WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 2, Number 1, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v2n1.39
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Application of Exercise Recommendation Model Based on Deep Reinforcement Learning

Juwei Dao, Li Hong*

College of Information, Yunnan Normal University, Kunming, China

ABSTRACT

In order to consolidate the learning achievements of students at a certain stage, Teachers often provide students with corresponding exercises before or after the beginning of this stage. Therefore, making the exercises appropriately challenging is one of the main goals of adaptive online learning systems. However, because each student's learning status is different, a student's learning situation in the same period of time will also be different. Therefore, it is also very challenging for students to choose appropriate exercises. In this paper, we propose a new method for problem recommendation. Firstly, we use GCKT model to model the user's answer sequence. Obtain the students' mastery of each concept. The learned user and problem representation are integrated into an extended framework to predict the likelihood of user mastery of the problem. Then use this as the basis for recommending exercises. On this basis, the deep reinforcement learning technology is used and the knowledge tracking model is used as a student simulator. The difference in the performance of the student simulator on all the exercises before and after solving the exercises provided by the exercise recommendation model is used as a reward, Make the model learn what kind of exercises can improve students' ability to the greatest extent, and recommend such exercises to students. Finally, the experiment is carried out in the actual use environment. The results show that the model has better performance than the current common recommendation models.

KEYWORDS

Deep Reinforcement Learning; Exercise Recommendation; Deep Learning

1. INTRODUCTION

Suppose there are U students, E exercises and K knowledge points. The answer sequence of the student $u(u \in U)$ is $[x_1, \ldots, x_t]$, where $x_t = \{q_t, r_t\}$ is a tuple, q_t is the problem solved by the student at time t, r_t represents an exercise of answering, $r_t = 1$ indicates that the result of the judgment is correct, $r_t = 0$ indicates that the result of the judgment is wrong.

Each exercise $q \in E$ is represented by a tuple $q = (k, d)$, where, $k \subseteq K$ represents the set of knowledge concepts examined by the exercise, and d represents the difficulty of the exercise. The difficulty of exercises depends on the error rate of students' answers, the error rate of submitting answers and the degree of discrimination.

In this paper, the process of problem recommendation is standardized as Markov Decision Process (MDP)[1]. For the four elements in the Markov decision process, the set of States S, the set of actions A, the reward function R, and the state transition function Γ, make the following definition.

(1) The set of States S. S represents the state space of the student. The state $s_t^u \in S$ is formed by the historical answer sequence of student u at time t and before, that is $s_t^u = x_{1:t}^u$.

(2) Action set A. A stands for the action space, which contains all the exercises in the system. The system exercise recommendation system takes action $a_t^u \in A$ in the state s_t^u, which is equivalent to recommending exercises $\hat{q}_{t+1}$ to the student u.

(3) The reward function R. R is the reward function, $r_t^u = R(s_t^u, a_t^u)$ represents the timely reward for the action a_t^u taken by the problem recommendation system in the state s_u^t. The size of the reward value is determined by the degree of growth of the student's learning ability status.

(4) State transition function Γ. When the exercise recommender system takes action a_t^u in the state s_t^u, the state transition function Γ can transition the current state from s_t^u to s_{t+1}^u, where $s_{t+1}^u = x_{1:t+1}^u$.

The goal of this paper is to find an optimal strategy $\pi: S \rightarrow A$ for recommending exercise sets to students. When the state s_t^u of the student is, the exercise recommendation system can select an exercise $\hat{q}_{t+1}$ from the exercise set E to recommend to the student u according to the strategy π, that the reward value accumulated by the whole recommendation process is maximized.

2. EXERCISE RECOMMENDATION MODEL

This paper proposes a GCER model for the problem recommendation task, and the overall architecture is shown in Figure 1. Firstly, the knowledge state of students is obtained by inputting the representation of exercises into the GCKT network. All exercises before and after the exercises provided by the exercise recommendation model are correctly solved by using the deep reinforcement learning technology and the knowledge tracking model. The average of the answer probability difference is used as the reward to train the recommendation model.

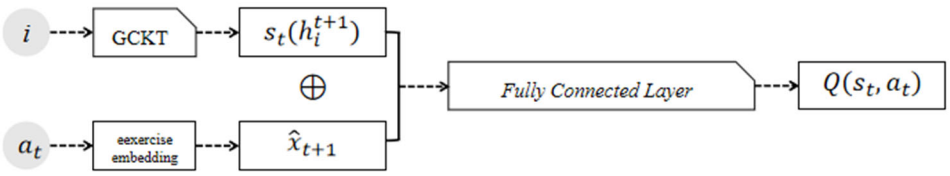

Figure 1. CGER Overall Architecture

CGER introduces the GCKT network to model the state of students at each moment through the historical answer sequence of students, and the specific steps are shown in formula (1):

$$h_{t+1} = \mathrm{GCKT}(x_t). \tag{1}$$

CGKT is a knowledge tracing model[2] inspired by Graph Convolutional Network (GCN)[3] and Graph-based Knowledge Tracing (GKT)[4] model.

In order for CGER to provide problem sets that will maximize the growth of student programming ability. In this paper, reinforcement learning technology and knowledge tracing model are used as student simulator, which simulates students in the state s_t. The reward letter is the average of the difference in the probability of correctly solving all the exercises before and after the exercise $\hat{q}_{t+1}$ provided by the recommended model $R(s_t, a_t)$ for solving the exercises. Numerical value, reward function, as formula (2):

$$R(s_t, a_t) = \frac{\sum_{qm \in E}[p(l_m=1|s_{t+1}) - p(l_m=1|s_t)]}{|E|}. \quad (2)$$

Among them, $p(l_m = 1|s_{t+1})$and $p(l_m = 1|s_t)$ are the probability that the student simulator can solve the problem correctly at the time $t+1$ and t. The greater the difference between the probability of correct solution of the student simulator at the time $t+1$ and t, the growth degree of relevant knowledge concepts will be greater. At the same time, the exercise recommendation model can also get a larger reward value. So that the model can learn what kind of exercises can maximize the probability of students to solve the exercises correctly. Such exercises are recommended to students to achieve the most significant purpose of increasing students' corresponding abilities.

In order to find the best strategy π to recommend problem sets to students. The exercise recommendation model maximizes the expectation of the cumulative recommendation reward value $Q(s_t, a_t)$ obtained by taking action a_t based on any state s_t, which satisfies formula (3):

$$Q(s_t, a_t) = E_\pi\{R(s_t, a_t) + \gamma \max[Q(s_{t+1}, a_{t+1})]\}. \quad (3)$$

Where $\gamma \in (0,1)$ is the penalty factor, $\max_{a_{t+1}}[Q(s_{t+1}, a_{t+1})]$ represents the maximum value of Q obtained after taking all possible actions in the state s_{t+1}.

Because there are a large number of exercises in the exercise recommendation system and new exercises are often added, Calculating and storing the Q values of all state-action pairs consumes a lot of resources. In order to solve this problem, inspired by Deep Q-Network [5], this paper adopts a deep reinforcement learning scheme. Use the parameter θ of the CGER model fits Q, as shown in Equation (4):

$$Q(s_t, a_t) \approx Q(s_t, a_t; \theta). \quad (4)$$

The model parameters are updated by minimizing a loss function as follows:

$$L(\theta) = \sum_u \sum_{t=1}^{T}[\frac{1}{2}(y - Q(s_t, a_t; \theta))^2]. \quad (5)$$

$$y = R(s_t, a_t) + \gamma \max_{a_{t+1}}[Q(s_{t+1}, a_{t+1}; \theta')]. \quad (6)$$

Where y is the target at the time t; θ' is the relevant parameters of the previous target network.

The partial derivatives of $L(\theta)$ as (7):

$$\nabla L(\theta) = \sum_u \sum_{t=1}^{T}\{[Q(s_t, a_t; \theta) - y]\nabla Q(s_t, a_t; \theta)\}. \quad (7)$$

When the student's status s_t is, this paper takes the value $Q(s_t, a_t; \theta)$ as the recommended degree of CGER for exercise $\hat{q}_{t+1}$. The candidate exercises are sorted from high to low according to the degree of recommendation, thus generating a recommendation list. The calculation method of $Q(st, at; \theta)$ is shown in formula (8):

$$Q(s_t, a_t; \theta) = \text{sigmoid}[W_q^T(h_t' \oplus \hat{x}_{t+1}) + b_q]. \quad (8)$$

Where, $h_t' \in \mathbb{R}^{d_h}$ is the representation vector of the student's state s_t; $\hat{x}_{t+1}$is the representation vector of the exercise $\hat{q}_{t+1}$; $W_q \in \mathbb{R}^{d_h+d_x}$is the trainable parameter vector; $b_q \in \mathbb{R}$ is a trainable bias term parameter.

3. EXERCISE RECOMMENDATION ASSESSMENT

3.1. Data set

In this paper, the public data set ASSIS2009 [6] and the self-made data set DH-jhs2023 are selected as the experimental data. Both data set include problem information and user submission record, Wherein each piece of data in the exercise information represents information of one exercise,The fields include: exercise ID, title, exercise description, knowledge point label (may be missing); Each piece of data in the user submission record contains the following fields: submission ID, user ID, exercise ID, system decision result (true/false), procedure Program, programming language type, run time, and commit timestamp. The basic data sets of ASSIS2009 and DH-jhs2023 are listed in Table 1.80% of the students were randomly selected from each data set to train the recommendation model and the student simulator.10% of the students were used to validate the recommended model and the student simulator, and 10% were used to test the recommended model.

Table 1. Dataset Statistics

Dataset name	Number of exercises	Number of users	Number of committed records
ASSIS2009	6772	1585	39075
DH-jhs2023	150	414	12420

3.2. Experimental details

In this paper, Py Torch is used to implement the proposed method, and the accuracy of GCKT model on ASSIS2009 and DH-jhs2023 validation sets is 73.2% and 76.7%, respectively. The accuracy rate is high, which shows that the GCKT model can accurately predict the results of students' answers. Some important hyper parameters are set uniformly as follows:

Table 2. Experimental hyperparameter settings

Parameter	Set up	Description
KT	GCKT	Student Simulator
emb-dim	16	Dimensions of Knowledge Concept Embedding
S-dim	128	Knowledge points represent vector dimensions
epochs	150	The limit of training iteration
batch-size	128	Number of samples per batch
optimizer	Adam	Optimizer selection
γ	0.9	Penalty factor

Recall @ K (Recall Rate of Top-K items) and MRR @ K (Mean Reciprocal Rank of Top-K items) are used in this paper As an evaluation index to analyze the performance of the exercise

recommendation model on the next interactive exercise prediction task, Examine whether the model can meet the learning needs of students.

Recall @ K is defined in detail as follows:

$$\text{Recall@K} = \frac{1}{|U|}\sum_{u\in U}\frac{|R_u \cap T_u|}{R_u}. \tag{9}$$

Among them, $|U|$ is the number of students; R_u indicates the problem set in the test set that is relevant to the student u; T_u represents the set of the top K problem sets recommended to the student.

The detailed definition of MRR @ K is as follows:

$$\text{MRR@K} = \frac{1}{|U|}\sum_{u\in U}\frac{1}{\text{rank}_u}. \tag{10}$$

Where rank_u is the position of the first correctly recommended exercise in the first K exercises of the student's recommended list. In this paper, K is set to 10, and 100 negative cases are randomly selected for each actual output. And rank the actual output along with the negative cases.

3.3. Experimental results

In order to evaluate the performance of the CGER model, this paper selects the methods based on knowledge tracing, DKT and DKVMN, as the benchmark methods.

Table 3. Comparison of experimental results

Method	Recall@10		MRR@10	
	ASSIS2009	DH-jhs2023	ASSIS2009	DH-jhs2023
DKT	0.203	0.172	0.161	0.109
DKVMN	0.206	0.186	0.168	0.120
CGER	0.357	0.331	0.313	0.295

According to the experimental results listed in the table, because CGER uses the code implementation of exercises to mine the correlation between exercises, The purpose of recommendation is to improve students' programming ability to the greatest extent. Therefore, its performance on problem recommendation tasks is better than that of the benchmark method.

From different data sets, the CGER model achieves better performance on the denser data set ASSIS2009.At the same time, the performance on sparse data sets has also achieved better results. Compared with the traditional knowledge tracking model, This model uses reinforcement learning to consider long-term rewards. On dense data sets, various methods usually perform better. Because the data set contains rich user interaction information. However, on the sparse data set, There is a lack of sufficient interaction information to describe user preferences, Therefore, the method based on traditional time series analysis is difficult to solve this problem. The method based on reinforcement learning can improve the performance of the model by calculating the maximum reward.

4. SUMMARY

To a large extent, the concept of knowledge can help learners quickly find the focus of learning, according to the different learning status of each student. Choosing the right exercises is also quite

challenging for students. Traditional time series recommendation methods rely heavily on the similarity between students and exercises. This makes the recommended exercises not suitable for students' knowledge level. In this paper, a personalized exercise recommendation algorithm based on reinforcement learning and students' cognitive state is adopted. The method takes a history answer submission sequence of a student as an input, Firstly, a knowledge level vector representing the current hidden learning state of a student is obtained and combined with a personal knowledge level vector, Find an optimal strategy for recommending problem sets to students, The exercise recommendation system can select an exercise from the exercise set to recommend to the student according to the strategy, that the reward value accumulated by the whole recommendation process is maximized. Finally, the effectiveness and rationality of the method proposed in this paper are proved by experiments.

REFERENCES

[1] Kaelbling L P , Littman M L , Cassandra A R .Acting Optimally in Partially Observable Stochastic Domains[J].Artificial Intelligence, 1994, 101(1-2): 99-134.DOI: 10.1016/ s0004- 3702(98) 00023-x.

[2] Albert, T., CorbettJohn, R., & Anderson. (1994). Knowledge tracing: modeling the acquisition of procedural knowledge. User Modeling & User Adapted Interaction.DOI:10.1007/BF01099821.

[3] Kipf, T. N. , & Welling, M. . (2016). Semi-supervised classification with graph convolutional networks.DOI:10.48550/arXiv.1609.02907.

[4] Nakagawa, H. , Iwasawa, Y. , & Matsuo, Y. . (2019). Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. IEEE/WIC/ACM International Conference on Web Intelligence. ACM.DOI:10.1145/3350546.3352513.

[5] MNIH V,KAVUKCUOGLU K,SILVERD,etal.Playing atari with deep reinforcement learning[J].arXiv:1312.5602,2013.

[6] Feng, M. , Heffernan, N. , & Koedinger, K. . (2009). Addressing the assessment challenge with an online system that tutors as it assesses. User Modeling and User-Adapted Interaction, 19(3), 243-266.DOI:10.1007/s11257-009-9063-7.