



Dynamic Key-Value Gated Recurrent Network for Knowledge Tracing

Bo Xie, Lina Fu^(✉), Bo Jiang, and Long Chen

Zhejiang Gongshang University, Hangzhou 310018, China

Abstract. Knowledge tracing model is one of the important research fields to realize personalized intelligent education. The focus of the model is to trace students' knowledge mastery from the records of students' answering performance. However, the existing knowledge tracing models don't consider the problems of students' knowledge application ability and forgetting rules. Based on the external memory mechanism of Dynamic Key-Value Memory Networks (DKVMN), this paper proposes a knowledge tracing model based on a dynamic key-value gated recurrent network (DKVGRU), which designs a knowledge update network inspired the idea of Gated Recurrent Unit (GRU). DKVGRU calculates the proportion of concepts students apply and measures the degree of forgetting of learned concepts, which traces the knowledge state of each concept well. In this paper, the area under the receiver operating characteristic curve (AUC) of the prediction result is used as an evaluation indicator. The performance of DKVGRU is higher than DKVMN on four public datasets.

Keywords: Knowledge tracing · Deep learning · Data mining · Intelligent education system · Student evaluation

1 Introduction

Nowadays, online learning platform realizes the acquisition of high-quality learning resources without the constraints of time and space. Students can flexibly study on computers and mobile terminals, and can independently arrange study plans and tasks. Because of this, millions of students are learning a variety of courses through online learning platforms. However, there are many obstacles in online learning platform for the supervision of students and the provision of personalized learning guidance due to the large number of learners. In terms of providing personalized guidance, it is very important to evaluate students' knowledge state for online learning platform, which is also an important research topic in the field of intelligent education [1].

Knowledge tracing (KT) is a widely used model for predicting students' knowledge state in intelligent online learning platform [2]. KT can model the interaction process between students and exercises based on the students' past exercise records to trace students' knowledge state dynamically [3]. The goal of KT can be described as: given the interaction sequence of past exercises of a student $X = x_1, x_2, \dots, x_t$, KT acquires the knowledge state of the student, which is used to predict the probability of the correct

answer to the next exercise. The input $x_t = (q_t, a_t)$ contains the exercise q_t and the actual answer a_t [4].

Using KT model, online learning platforms not only customize learning materials for students based on the knowledge state of students, but also provide to students and teachers with feedback reports. Therefore, students reasonably allocate their study schedules to maximize their learning efficiency, and teachers can timely adjust appropriate teaching plans and schemes.

At present, the traditional KT model and the deep learning-based KT model are two kinds of models provided in the field of knowledge tracing. Among the traditional knowledge tracing models, the most typical one is Bayesian Knowledge Tracing (BKT) [5], which models each concept state separately. Therefore, BKT is limited to capture the correlations between different concepts, which ineffectively simulates the knowledge state transition between complex concepts. Researchers further applied deep learning to KT task and proposed Deep Knowledge Tracing (DKT) [6]. Compared with BKT, DKT uses a hidden state to sum up the knowledge state of all concepts. Considering correlations between multiple concepts, DKT delivers a better simulation in students' knowledge state. But DKT can't pinpoint which concepts a student has mastered like BKT. Consequently, DKT has its weakness in indicating the certain concept that students grasp or not. Combining the advantages of the BKT and DKT, DKVMN uses external memory to store the student's knowledge state [7], and its prediction performance is better than BKT and DKT.

However, existing KT models ignore two aspects in simulating the changes in students' knowledge states. Firstly, in the aspect of knowledge application, students apply different concepts according to their knowledge states for the same exercise. Secondly, according to the Ebbinghaus forgetting curve [8], the process of forgetting is not uniform. Students forget the knowledge they have just learned from the exercises very fast, but the knowledge they have learned before is slow. Existing models have limits in distinguishing the degree of forgetting the learned knowledge.

Based on the external memory mechanism of DKVMN, this paper designed a knowledge update network inspired by the idea of GRU's gating mechanism [9], and proposed a knowledge tracing model based on Dynamic Key-Value Gated Recurrent Network (DKVGRU). In the huge exercise data, DKVGRU uses the Key-Value matrix of DKVMN to explore the relationship between exercises and underlying concepts, while tracing the knowledge state of a certain concept. We provided two knowledge gates for simulating the change of students' knowledge states. The knowledge application gate calculates the proportion of knowledge concepts applied by students in solving exercises, and the knowledge forgetting gate measures the forgetting degree of the learned knowledge.

2 Related Work

There are two main types of KT models. One is the traditional KT model, the other is the KT model based on deep learning. In this chapter, we first introduce BKT, DKT and DKVMN. Besides, DKVGRU is inspired by the gating mechanism, and this chapter also introduces Recurrent Neural Network (RNN) [10] and its variants, which can capture long-term sequence data relations.

2.1 Bayesian Knowledge Tracing

BKT is the most commonly used among traditional KT models, which was introduced in the field of intelligent education by Corbett and Anderson and used to intelligent tutoring systems in 1995 [11]. BKT assumes that each concept is independent of each other and students have only two states for each concept: mastered or not mastered. As shown in Fig. 1, BKT uses Hidden Markov Model (HMM) to model a certain concept separately, and updates the state of a concept with the help of two learning parameters and two performance parameters. The original BKT assumes students do not forget knowledge in learning, which is obviously against students' regular learning pattern [12]. And researchers have proposed several aspects to optimize BKT from forgetting parameters [13], exercise difficulty [14], personalized parameters [15], emotions [16], etc.

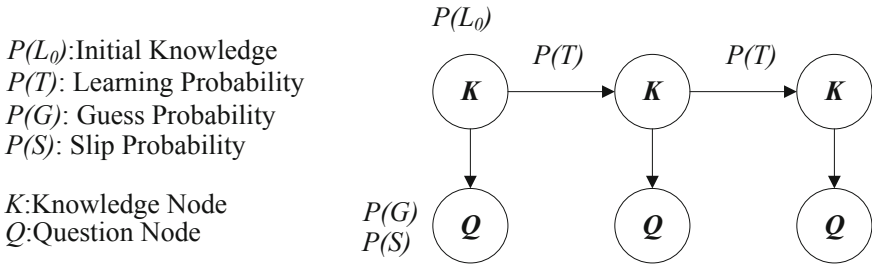


Fig. 1. The architecture of BKT

2.2 Deep Learning-Based Knowledge Tracing

In 2015, Piech et al. firstly applied deep learning to KT tasks and proposed DKT based on RNN and Long Short-Term Memory (LSTM) [17]. As illustrated in Fig. 2, DKT can represent the student's continuous knowledge state using a high-dimensional hidden state. And without the manual annotation, DKT can automatically discover the relationship between concepts from exercises. Using the forgetting gate of LSTM, DKT can simulate the knowledge forgetting that occurs in the learning process. Khajah et al. proved that the advantage of DKT lies in the ability to make good use of some statistical rules in data, which BKT cannot use [18]. Yeung et al. added a regularization term to the loss function to solve two problems of DKT: inaccuracy and instability [19]. Xiong et al. believe that DKT is a potential KT method if more features can be modeled, such as student abilities, learning time, and exercise difficulty [20]. And many variations were raised by adding dynamic student classification [21], side information [22] and other features [23] into DKT.

The Memory Augmented Neural Network (MANN) [24] uses an external memory module to store information, which has a stronger information storage capacity than using a high-dimensional hidden state. And MANN can rewrite local information through the external memory mechanism. Different from the general MANN which uses a simple

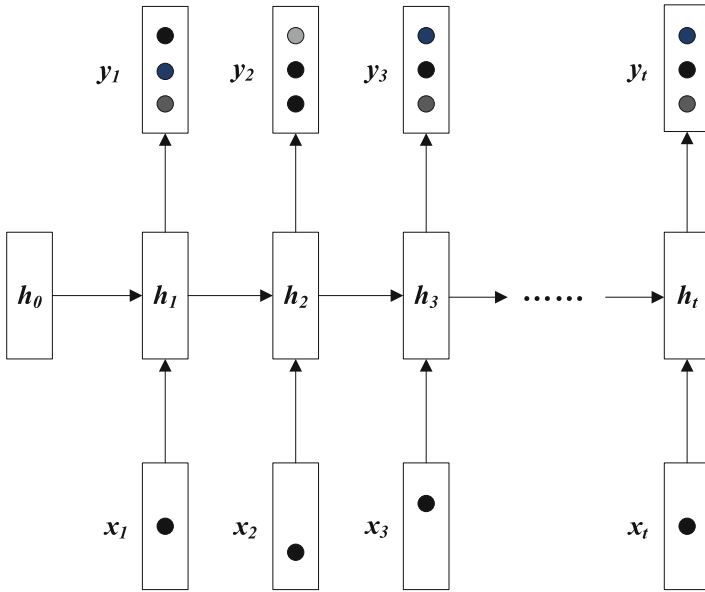


Fig. 2. The architecture of DKT based on RNN

memory matrix or two static memory matrices [25], DKVMN utilizes the key-value matrix to store all concepts and the knowledge state of each concept. The key matrix is used to calculate the correlation between exercises and concepts, and the value matrix is used to read and write the knowledge state of each concept. Ha et al. [26] optimized DKVMN from knowledge growth and regularization.

2.3 Recurrent Neural Network

For sequence data, researchers use RNN to obtain data relationships in general. However, RNN cannot effectively capture long-term sequence data relationships because of its structural defects. And Hochreiter et al. proposed LSTM to solve the problem of long-term in 1997, which used three gates to effectively deal with long-term and short-term dependence. And Cho et al. proposed GRU by optimizing the structure of LSTM in 2014, which not only guarantees model performance but also improves model training efficiency [27]. GRU uses two gates to determine which information needs to be memorized, forgotten, and output respectively, which effectively achieve long-term tracing of information. As shown in Fig. 3, the reset gate generates the weight to decide how much historical information is used according to the input information, and the update gate is used to generate the proportion of historical memory and current memory in new memories.

3 Model

DKVGRU can be divided into three parts: correlation weight, read process and write process, which are represented in Fig. 4. Correlation weight represents the weight of each

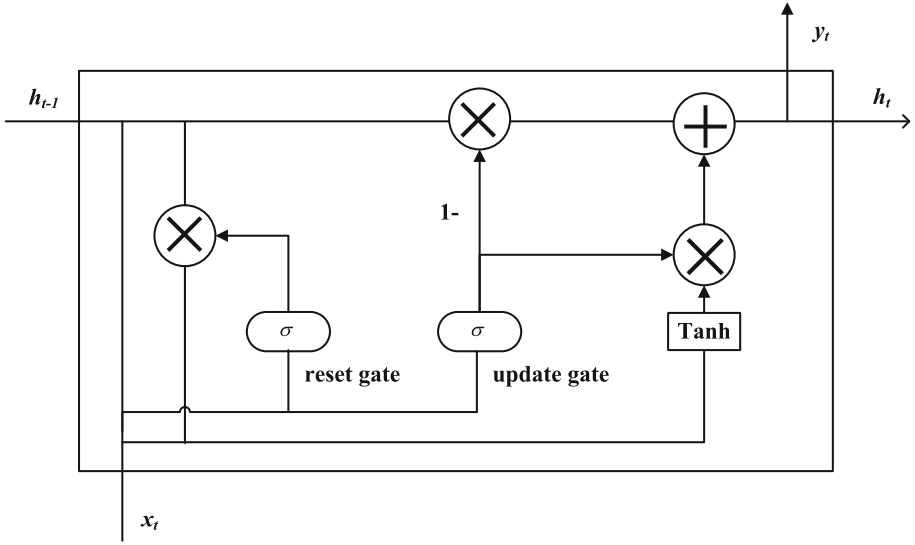


Fig. 3. The architecture of GRU

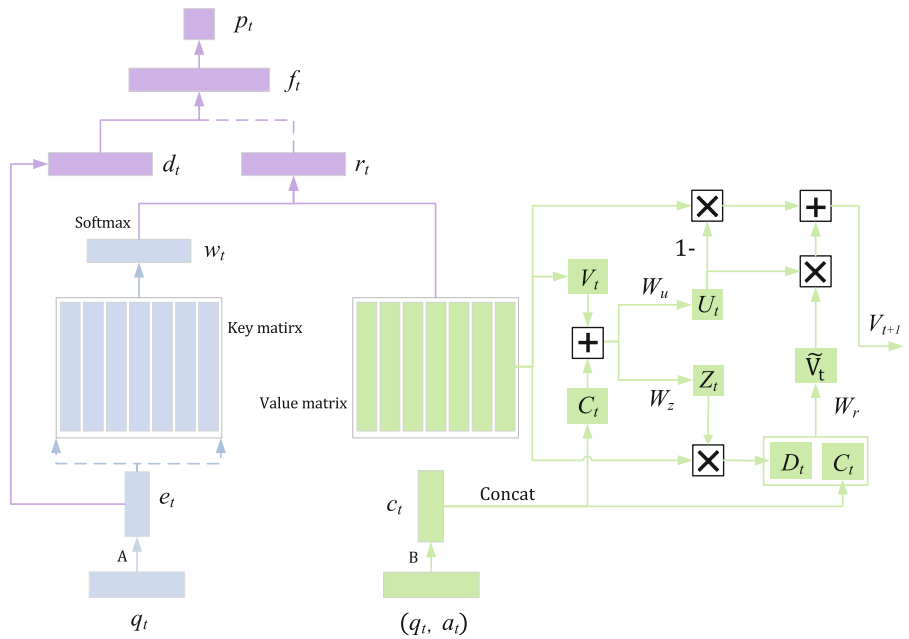


Fig. 4. In the framework of DKVGRU, the green part is write process we designed. The blue and purple parts are correlation weight and read process, which refer to DKVMN.

concept contained in the exercise. Read process can read the student’s current memory, which is used to predict students’ performance of a new exercise. And write process

is used to update the student's memory state after answering a exercise. Correlation weight and read process refer to DKVMN. The correlation weight, read process and write process are described in Sect. 3.2, 3.3, and 3.4.

3.1 Related Definitions

Given a student's past interaction sequence of exercises $X = x_1, x_2, \dots, x_{t-1}$, our task is to obtain the student's current knowledge state according to the student's interaction sequence and predict students' performance of the next exercise. The interaction tuple $x_t = (q_t, a_t)$ represents the student's answer to the exercise q_t , where a_t is 1 means the answer is correct and 0 means wrong.

Table 1. Symbols

Symbols	Explanation
X	The past exercise sequence of a student: x_1, x_2, \dots, x_{t-1}
E	Number of exercises
N	Number of concepts contained in the exercises
K	Key matrix, which is used to store all concepts
V	Value matrix, which is used to dynamically store the state of each concept
\tilde{V}	Knowledge growth matrix, which stores the knowledge growth of students after answering one exercise
D	Knowledge application matrix, which stores the proportion of each concept used by students for a certain exercise
e	Exercise vector
c	Exercise interaction vector
w	Correlation weight
Z	Knowledge application gate, which calculates the proportion of concepts used in an exercise
U	Knowledge forgetting gate, which measures the forgetting degree of the learned knowledge
A	Exercise embedding matrix
B	Interaction embedding matrix
$;$	The operation of Concatenating two vectors or two matrices

As illustrated in Table 1, the definition of various symbols used in the model is described. The N represents the number of concepts, and the key matrix $K(N \times d_k)$ stores these concepts. Besides, the knowledge state of each concept is stored in the value matrix $V(N \times d_v)$.

3.2 Correlation Weight

Each exercise contains multiple concepts. The exercise q_t is firstly mapped into a vector $e \in R^{d_k}$ by an embedding matrix $A \in R^{d_k}$. The correlation weight $w_t \in R^N$ is computed by taking the softmax activation of the inner product between e_t and each k_i of the key matrix $K = (k_1, k_2, \dots, k_N)$:

$$w_t = \text{Softmax}(e_t \cdot K^T). \quad (1)$$

k_i is the key memory slot which is used to store the i^{th} concept. And w_t measures the correlation weight between this exercise and concepts.

3.3 Read Process

The probability of answering q_t correctly needs to consider two factors: the student's current knowledge state and exercise difficulty. Above all, w_t is multiplied by the each v_i of the value matrix $V = (v_1, v_2, \dots, v_N)$, which is to get the read content vector $r_t \in R^{d_v}$:

$$r_t = w_t \cdot V_t. \quad (2)$$

v_i is the value memory slot which is used to store the state of the i^{th} concept. And the read content r_t is regarded as the student's overall mastery of q_t .

Then considering that the difficulty of q_t , the exercise vector e_t passes through the fully connect layer and *Tanh* function to get the difficulty vector $d_t \in R^{d_k}$:

$$d_t = \text{Tanh}(e_t \cdot W_1 + b_1), \quad (3)$$

$$\text{Tanh}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}, \quad (4)$$

and W_i and b_i are the weight and bias of the full connect layer.

The summary vector f_t is obtained after concatenating the read content vector r_t and the difficulty vector d_t :

$$f_t = \text{Tanh}([r_t; d_t] \cdot W_2 + b_2). \quad (5)$$

Finally, the probability p_t is computed from the summary vector f_t :

$$p_t = \text{Sigmoid}(f_t \cdot W_3 + b_3), \quad (6)$$

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (7)$$

And *Sigmoid* function makes the probability p_t between 0 to 1.

3.4 Write Process

The knowledge state of each concept are updated after the student answering the exercise q_t . The interaction tuple $x_t = (q_t, a_t)$ is turned into a number by $y_t = q_t + a_t * E$, which y_t represents the student's interactive information. And y_t is converted into an interaction vector $c_t \in R^{d_v}$ by an embedding matrix $B \in R^{E \times d_v}$. Considering that students apply knowledge to exercise according to their knowledge state, we adds the interaction vector c_t and each value memory slot v_i of the value matrix V_t , and pass the result through the fully connect layer and an activation function to obtain the knowledge application gate $Z_t \in R^{N \times d_v}$:

$$C_t = \text{Concat}(c_t, c_t, \dots, c_t), \quad (8)$$

$$Z_t = \text{Sigmoid}([V_t + C_t] \cdot W_z + b_z). \quad (9)$$

Z_t is used to calculate the proportion of concepts used in an exercise. The application knowledge state $D_t \in R^{N \times d_v}$ is obtained by using Z_t to weight the value matrix V_t :

$$D_t = Z_t * V_t. \quad (10)$$

Then, we concatenate the interaction vector c_t and each d_i of the value matrix $D_t = (d_1, d_2, \dots, d_N)$ to get the knowledge growth matrix $\tilde{V}_t \in R^{N \times d_v}$:

$$\tilde{V}_t = \text{Tanh}([D_t; C_t] \cdot W_r + b_r). \quad (11)$$

For the purpose of measuring student's forgetting degrees, We adds the interaction vector c_t and each value memory slot v_i of the value matrix V_t to obtain the knowledge forgetting gate $U_t \in R^{N \times d_v}$:

$$U_t = \text{Sigmoid}([V_t + C_t] \cdot W_u + b_u). \quad (12)$$

Each concept state of the value matrix V_t is updated by U_t . $(1 - U_t) * V_t$ represents the unforgettable part of the previous knowledge state, and $U_t * \tilde{V}_t$ represents the unforgettable part of the knowledge gained from this exercise. And V_{t+1} means the new student's knowledge state.

$$V_{t+1} = (1 - U_t) * V_t + U_t * \tilde{V}_t. \quad (13)$$

3.5 Optimization Process

The optimization goal of our model is that the predicted probability p_t is close to the student's answer a_t , that is to minimize the cross entropy loss L .

$$L = - \sum_t a_t \log(p_t) + (1 - a_t) \log(1 - p_t). \quad (14)$$

4 Experiments

4.1 Datasets

There are several datasets to test the performance of models in Table 2, including Statics2011, ASSISTments2009, ASSISTments2015 and ASSISTment Challenge. And these datasets come from real online learning systems.

- (1) Statics2011: This dataset has 1,223 exercise tags and 189,297 interaction records of 333 students, which comes from an engineering mechanics course of a university.

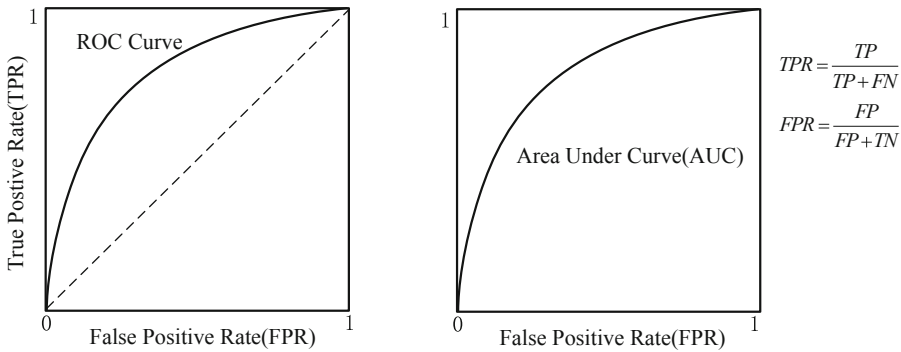
Table 2. Dataset statistics

Datasets	Students	Exercise tags	Records
Statics2011	333	1,223	189,297
ASSISTments2009	4,151	110	325,637
ASSISTments2015	19,840	100	683,801
ASSISTment Challenge	686	102	942,816

- (2) ASSISTments2009: This dataset contains 110 exercise tags and 325,637 interactions records for 4,151 students, which comes from the ASSISTment education platform in 2009.
- (3) ASSISTments2015: This dataset is collected from the ASSISTment education platform, which has 100 exercise tags and 683,801 interactions records of 19,840 students.
- (4) ASSISTment Challenge: This dataset was used in the ASSISTment competition in 2017, and it contains 102 exercise tags and 942,816 interaction records of 686 students.

4.2 Evaluation Method

In the field of knowledge tracing, we usually use AUC as the evaluation criteria for model classification. The advantage of AUC is that even if the sample is unbalanced, it can still give a more credible evaluation result [28]. This paper also uses AUC as the evaluation of the model. And the higher the value of AUC, the better the classification result. As shown in Fig. 5, ROC curve is drawn according to TPR and FPR, and AUC is obtained from the area under ROC curve.



		Actual Label	
		Positive	Negative
Predictive Label	Positive	TP	FP
	Negative	FN	TN

Fig. 5. Description of the AUC calculation

4.3 Implementation Details

In this paper, the training set and test set of each dataset was randomly assigned, 70% of which is the training set and the remaining 30% is the test set. The five-fold cross-validation method was used on the training set, and 20% of the training set was divided into the validation set. We used early stopping and selected hyperparameters of model on the validation set. And the performance of the model was evaluated on the test set.

Gaussian distribution was used to initialize the parameters randomly. Stochastic gradient descent method was adopted as the optimization method for training. And batch size was set to 50 on all datasets. The maximum number of training times of the model was set to 100 epochs. The epoch with the best AUC value on the validation set was selected for testing. And the average value of AUC on the test set was used as the model evaluation result.

Using different initial learning rates, we compared the performance of DKVMN and DKVGRU models when the sequence length was 200. Then, we set sequence lengths of 100, 150, and 200 to compare the performance of DKVMN and DKVGRU.

4.4 Result Analysis

On the four datasets, the experiment used the initial learning rate of 0.02, 0.04, 0.06, 0.08, and 0.1 to measure the AUC scores of DKVMN and DKVGRU. And AUC of 0.5 represents the score that can be obtained by random guessing. The higher the AUC score, the better the prediction effect of the model. As shown in Table 3, there are the test AUC score of DKVMN and DKVGRU of all datasets. It can be clearly seen that DKVGRU performs better than DKVMN on all datasets.

Table 3. The test AUC scores of DKVMN and DKVGRU with different initial learning rates on all datasets

Dataset	The initial learning rate	Test AUC	
		DKVMN	DKVGRU
Statics2011	0.02	0.814900	0.816630
	0.04	0.817625	0.831916
	0.06	0.818041	0.834688
	0.08	0.818591	0.834850
	0.1	0.819070	0.834766
ASSISTments2009	0.02	0.801624	0.806886
	0.04	0.803469	0.808587
	0.06	0.804472	0.808334
	0.08	0.802909	0.806580
	0.1	0.804652	0.808877
ASSISTments2015	0.02	0.726578	0.728503
	0.04	0.725727	0.728791
	0.06	0.724857	0.729177
	0.08	0.724883	0.728526
	0.1	0.724837	0.728371
ASSISTment Challenge	0.02	0.662156	0.676517
	0.04	0.664859	0.684211
	0.06	0.667367	0.687498
	0.08	0.669780	0.689189
	0.1	0.671637	0.689144

For Statics2011 dataset, the average AUC of DKVMN is 81.76%, while the average AUC of DKVGRU is 83.06%, which indicates a 1.29% higher than DKVMN. On the ASSISTments2009 dataset, DKVMN produces the average test AUC value of 80.34%, which shows a 0.44% difference compared with 80.70% for DKVGRU. For ASSISTments2015 dataset, the average AUC of DKVGRU is 72.87% and DKVMN is 72.54%. On the ASSISTment Challenge dataset, DKVGRU achieves the average AUC of 68.53%, which improves 1.82% as DKVMN in 66.72%. Therefore, DKVGRU has a better performance than DKVMN on all four datasets. For both models, the paper observes that a larger initial learning rate might lead to a better AUC score from the aforementioned experiments.

Then, we set the initial learning rate of 0.1 and sequence lengths of 100, 150, and 200 to evaluate these two models. And the experimental results indicate that DKVGRU performs better than DKVMN at different sequence lengths in Table 4.

According to Fig. 6, the AUC results of DKVGRU and DKVMN become better with the increase of sequence length except for Statics2011 dataset. The findings support that the setting of the sequence length of the exercises has a positive correlation with the models performance, which means a longer sequence length results in a better prediction performance for the model. That is, the model can more accurately trace students' knowledge state by utilizing more exercise records.

Table 4. The test AUC scores of DKVMN and DKVGRU with different sequence length on all datasets

Dataset	Sequence length	Test AUC	
		DKVMN	DKVGRU
Statics2011	100	0.833927	0.849142
	150	0.827971	0.842618
	200	0.819070	0.834766
ASSISTments2009	100	0.799399	0.804384
	150	0.800944	0.804276
	200	0.804652	0.808877
ASSISTments2015	100	0.723672	0.727784
	150	0.723034	0.727913
	200	0.724837	0.728371
ASSISTment Challenge	100	0.658292	0.672841
	150	0.664720	0.682050
	200	0.671637	0.689144

On the Statics2011 dataset, the reason why the AUC results have a negative correlation with the sequence length is that exercise tag is the largest among the four datasets, which included 1,223 exercise tags. The more exercise labels in the sequence, the more complex relationships between exercises and concepts need to be considered by the model. Nonetheless, the AUC score of DKVGRU on the Statics2011 dataset is higher, which means DKVGRU can simulate students' knowledge state better than DKVMN.

In summary, DKVGRU performs better than DKVMN with different learning rates and sequence lengths, which shows that the gating mechanism of DKVGRU effectively simulates the changes of students' knowledge state.

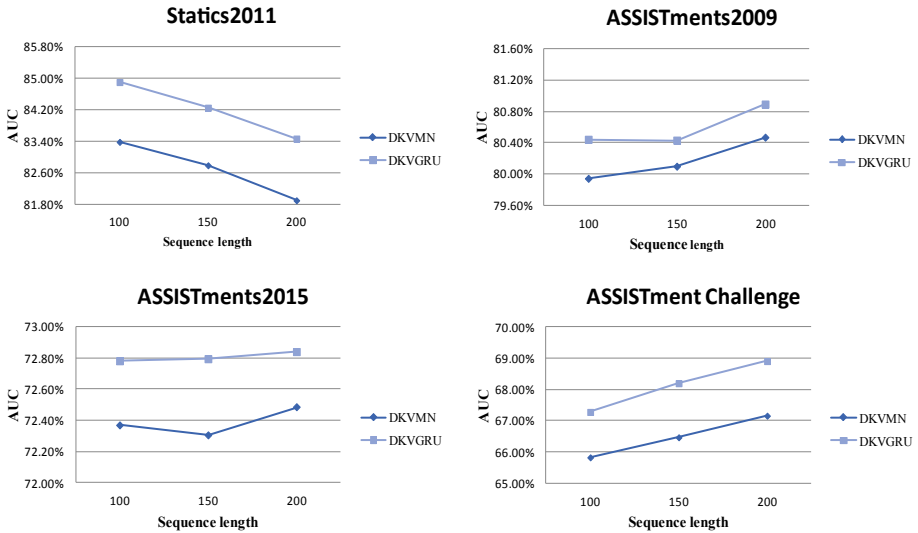


Fig. 6. The test AUC scores of DKVMN and DKVGRU with different sequence length on all datasets

5 Conclusions and Prospects

For the existing shortcomings of knowledge tracing, such as ignoring students apply different concepts to the same exercise and failing to consider the forgetting process of concepts they have learned, we propose a knowledge tracing model DKVGRU, which is based on the dynamic Key-Value matrix and gating mechanism. DKVGRU updates students' knowledge state by the gating mechanism. The experimental data comes from four public datasets. And the experiments demonstrate that DKVGRU performs better than DKVMN.

In addition to the students' exercise records, online learning platforms also record various learning activities of students, such as watching videos, viewing exercise explanations and other learning actions. For future work, we will consider these features in KT tasks. And using these data, we also can classify students according to students' learning attitude and habits, which simulates students' knowledge state reasonably.

References

1. Liu, H., Zhang, T., Wu, P., Yu, G.: A review of knowledge tracking. *Comput. Sci. Eng.* (5), 1–15 (2019)
2. Pardos, Z., Heffernan, N.: Modeling individualization in a bayesian networks implementation of knowledge tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) *UMAP 2010. LNCS*, vol. 6075, pp. 255–266. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13470-8_24
3. Xu, M., Wu, W., Zhou, X., Pu, Y.: Research on multi-knowledge point knowledge tracing model and visualization. *J. Audio-visual Educ. Res.* **39**(10), 55–61 (2018)

4. Yeung, C.K., Lin, Z., Yang, K., et al.: Incorporating features learned by an enhanced deep knowledge tracing model for STEM/Non-STEM job prediction. *J. CoRR* (2018)
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User Adap. Interact.* **4**(4), 253–278 (1994)
6. Piech, C., et al.: Deep knowledge tracing. *Comput. Sci.* **3**(3), 19–23 (2015)
7. Zhang, J., Shi, X., King, L., Yeung, D.: Dynamic key-value memory networks for knowledge tracing. In: *The Web Conference*, pp. 765–774. ACM (2017)
8. Deng, Y., et al.: Application of Ebbinghaus forgetting curve theory in teaching medical imaging diagnosis. *Chin. J. Med. Educ.* **33**(4), 555–557 (2013)
9. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Comput. Sci.* 1724–1734 (2014)
10. Elman, J.L.: Finding structure in time. *Cogn. Sci.* **14**(2), 179–211 (1990)
11. Li, F., Ye, Y., Li, X., Shi, D.: Application of knowledge tracking model in education: a summary of related research from 2008 to 2017. *Distance Educ. China* (007), 86–91 (2019)
12. Wang, Z., Zhang, M.: MOOC student assessment based on Bayesian knowledge tracing model. *China Sciencepaper* **10**(02), 241–246 (2015)
13. Ye, Y., Li, F., Liu, Q.: The influence of forgetting and data volume into the knowledge tracking model on prediction accuracy. *China Acad. J. Electron. Publishing House* (008), 20–26 (2019)
14. Pardos, Z., Heffernan, N.: KT-IDEM: introducing item difficulty to the knowledge tracing model. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) *UMAP 2011. LNCS*, vol. 6787, pp. 243–254. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22362-4_21
15. Yudelson, M., Koedinger, K., Gordon, G.: Individualized Bayesian knowledge tracing models. In: Chad Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) *AIED 2013. LNCS (LNAI)*, vol. 7926, pp. 171–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_18
16. Spaulding, S., Breazeal, C.: Affect and inference in Bayesian knowledge tracing with a robot tutor. In: *The Tenth Annual ACM/IEEE International Conference*, pp. 219–220. ACM (2015)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
18. Khajah, M., Lindsey, R.L., Mozer, M.C.: How deep is knowledge tracing? In: *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 94–101 (2016)
19. Yeung, C.K., Yeung, D.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: *5th Annual ACM Conference on Learning at Scale*, pp. 1–10 (2018)
20. Xiong, X., Zhao, S., Van Inwegen, E.G., Beck, J.E.: Going deeper with deep knowledge tracing. In: *9th International Conference on Educational Data Mining*, North Carolina, pp. 545–550 (2016)
21. Minn, S., Yu, Y., Desmarais, M.C., Zhu, F., Vie, J.J.: Deep knowledge tracing and dynamic student classification for knowledge tracing. In: *2018 IEEE International Conference on Data Mining*, pp. 1182–1187. IEEE (2018)
22. Wang, Z., Feng, X., Tang, J., Huang, G., Liu, Z.: Deep knowledge tracing with side information. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) *AIED 2019. LNCS (LNAI)*, vol. 11626, pp. 303–308. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23207-8_56
23. Zhang, L., Xiong, X.L., Zhao, S.Y., Botelho, A.F., Heffernan, N.T.: Incorporating rich features into deep knowledge tracing. In: *ACM Conference on Learning* (2017)
24. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D.: Meta-learning with memory-augmented neural networks. In: *33rd International Conference on Machine Learning*, pp. 1842–1850. IMLS, New York City (2016)

25. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents (2016)
26. Ha, H., Hwang, U., Hong, Y., Yoon, S.: Memory-augmented neural networks for knowledge tracing from the perspective of learning and forgetting. In: 33rd Innovative Applications of Artificial Intelligence Conference Honolulu, Hawaii, USA (2018)
27. Yang, L., Wu, Y., Wang, J., Liu, Y.: Research on recurrent neural network. *J. Comput. Appl.* **38**(S2), 6–11+31 (2018)
28. Wang, Y., Chen, S.: A survey of evaluation and design for AUC based classifier. *Pattern Recogn. Artif. Intell.* **24**(001), 64–71 (2011)