Dynamic Key-Value Memory Networks With Rich Features for Knowledge Tracing

Xia Sun, Xu Zhao, Bo Li, Yuan Ma, Richard Sutcliffe, and Jun Feng

***Abstract*—Knowledge tracing is an important research topic in student modeling. The aim is to model a student's knowledge state by mining a large number of exercise records. The dynamic key-value memory network (DKVMN) proposed for processing knowledge tracing tasks is considered to be superior to other methods. However, through our research, we have noticed that the DKVMN model ignores both the students' behavior features collected by the intelligent tutoring system (ITS) and their learning abilities, which, together, can be used to help model a student's knowledge state. We believe that a student's learning ability always changes over time. Therefore, this article proposes a new exercise record representation method, which integrates the features of students' behavior with those of the learning ability, thereby improving the performance of knowledge tracing. Our experiments show that the proposed method can improve the prediction results of DKVMN.**

***Index Terms*—Dynamic key-value memory network (DKVMN), knowledge tracing, student clustering.**

I. Introduction

KNOWLEDGE tracing is a very important research topic in intelligent education [1]. By modeling student's learning behavior through their past exercise records, knowledge tracing can assess their mastery of knowledge skills. A student's current knowledge state can then be used to guide them to spend their time on developing skills where their knowledge is weak. In addition, it is more accurate to evaluate the student's knowledge state by using long-term exercise records than to use the results of one or more tests [2]–[4]. Knowledge tracing can be formalized as a supervised sequence prediction problem, as follows: given observations of interactions $x_0 \ldots x_t$ carried out by a student on a particular learning task, predict aspects of their next interaction x_{t+1} [5], [6].

For students, knowledge tracing can help them to develop a personalized learning path, so that they can quickly and effectively enter a new field and obtain timely, effective, and personalized guidance [7], [8]. For teachers, the knowledge-tracking algorithm can provide timely and accurate feedback concerning the students' mastery of knowledge skills, allowing them to understand the learning situation of students. Consequently, teachers have more data that can be used to adapt lesson plans and monitor course progress. As a result, students can be taught according to their aptitude [9], [10].

Although there are many existing knowledge tracing studies [11], most of these only model a student's knowledge state based on the number of exercises completed and the results achieved; they tend to ignore the behavioral features of the students, such as the attempt count and first action. Nevertheless, these behavior features can help the model to improve its predictive ability [12]. For example, when analyzing the behavioral feature "first action," a student may independently answer correctly or they may answer correctly only after being given further help. While the results of the exercise may be the same, the growth of their knowledge is different. In addition, a student's learning ability can change dynamically over time. For example, with an increase in the number of exercises and the consequent enhancement of knowledge state, a student's learning ability may be improved, a factor which is also not considered in the existing knowledge tracing models.

Therefore, in this work, we propose a multifunctional knowledge tracing model that divides the student's exercise records into segments, each containing the same number of exercises, calculates the learning ability of the students in each segment, and clusters the students according to their learning ability. Finally, we carry out cross-fusion of learning ability features and learning behavior features to improve the dynamic key-value memory network (DKVMN).

Our main contributions are summarized as follows.

1) We define a learning ability feature, which can dynamically reflect the changes of students' learning ability throughout the learning process. This feature can also distinguish between the learning ability of different students.
2) We use several student behaviors features in the learning process to capture more specific information about each student and, hence, improve the model's predictions. In this way, an enhanced exercise record representation is created, which effectively combines the student's learning behavior features with his or her learning ability features.
3) We improve the reading and writing process of the DKVMN, so that it can take into account both the

Manuscript received May 3, 2020; revised September 5, 2020 and November 30, 2020; accepted January 3, 2021. This work was supported in part by the National Natural Science Foundation Projects of China under Grant 61877050; in part by Northwest University Teaching Achievement Cultivation Project under Grant XM05190141; and in part by the Open Project Fund of Shaanxi Province Key Laboratory of Satellite and Terrestrial Network Tech of China. This article was recommended by Associate Editor F. Wu. *(Corresponding authors: Xia Sun; Richard Sutcliffe; Jun Feng.)*

The authors are with the Department of Computer Science, Northwest University, Xi'an 710127, China (e-mail: raindy@nwu.edu.cn; zxcsnwu@163.com; libo_ceasar@stumail.nwu.edu.cn; 1043331939@qq.com; rsutcl@nwu.edu.cn; fengjun@nwu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3051028.

Digital Object Identifier 10.1109/TCYB.2021.3051028

current learning ability and the learning behavior of the student when predicting the results of the subsequent exercises.

II. Related Work

The Bayesian knowledge tracing (BKT) was first proposed by Corbett and Anderson and applied to the intelligent tutoring system (ITS) [13]. It is the most popular student learning modeling algorithm [14]. BKT tracks the changes in mastery of a student's knowledge skills. By maintaining a continuous assessment of the student's current probability of understanding each knowledge skill, and continuously updating that estimate based on the student's behavior, the student's mastery of the knowledge skill can be predicted [15]. Pardos and Heffernan [16] improved the reliability of students' predictions by personalizing the initial knowledge control parameters. Yudelson *et al.* proposed to construct multiple personalized BKT algorithms incrementally and to add specific parameters of the algorithm in batches. The final results showed that the combination of specific algorithm parameters and multiple models can improve the performance of the algorithm. In particular, optimizing students' learning rate parameters is better than optimizing prior knowledge parameters [17]. González-Brenes *et al.* identified a problem with the BKT algorithm: it assumes that students will not forget a knowledge skill after learning it. They addressed this by incorporating forgetting as a time-decay function in the student prediction model, thereby improving the prediction performance [18]. Nevertheless, the BKT model ignores the impact of the exercise order on learning, and needs to label the corresponding relationship between each exercise and related knowledge skills in advance, which greatly increases the task of manual labeling.

A recurrent neural network (RNN) is a time-series model, which has a high-dimensional representation of a continuous hidden state feature. An RNN is able to make predictions using previous information and has a good performance on sequence prediction problems [19], [20]. Piech *et al.* [21] applied RNNs to the knowledge tracing field and achieved good results with their deep knowledge tracing (DKT) model. DKT uses an RNN to model the student's knowledge state. When students practice through homework, it will try to use the information in the previous time step to better infer the student's future performance. Focusing on the problem of volatility in the prediction result of the DKT algorithm, Yeung and Yeung [22] proposed a method of adding three regularization terms to the loss function of the DKT algorithm to enhance the consistency of the algorithm's prediction and to improve its accuracy. Zhang *et al.* [23] improved the DKT model by incorporating more features at the exercise level and using an auto encoder to convert high-dimensional information into low-dimensional features. Minn *et al.* [24] proposed an improved DKT model based on student dynamic clustering, which achieves good results in knowledge tracing. In summary, we know that adding rich behavior features can improve the predictive performance of knowledge tracing models. However, while DKT models based on RNNs or LSTMs improve prediction accuracy compared with the BKT model, the training time is longer, the dimensions are increased, and the number of parameters is greater, thus restricting the wider application of the model.

DKVMNs are a variant of memory-augmented neural networks (MANNs), a type of model that adds storage modules and corresponding read-write mechanisms based on traditional neural networks [25]. DKVMN adds both a static matrix and a dynamic matrix as external memories, thus getting rid of the connection between the trainable parameters and the memory ability of the model; it also makes more efficient use of the trainable parameters; these changes make it easier to model long distance dependencies. So far, DKVMN performs the best in predicting student performance and is the best knowledge tracing model [26]. However, DKVMN only uses the exercise label and correctness label as inputs, and ignores such factors as students' behavior features, the changes in students' learning ability after practice, and the differences in learning ability between students. This can cause DKVMN to model a student's knowledge state inaccurately, which, in turn, affects its ability to predict the results of future exercises.

III. Method

Human learning is a process involving practice; we become proficient through constant practice. However, learning is also influenced by individual learning ability and individual learning behavior [27]. Therefore, we propose a DKT model based on the learning behavior features and learning ability of a student.

Our goal is to predict whether the student answered the current exercise correctly based on their past exercise records containing learning behavior and on their learning ability. Hence, we can evaluate a student's knowledge state through the results of long-term exercise sequences. This task can be formalized as a supervised sequence prediction problem, as follows: let $V = \{x_1, x_2, \ldots, x_n, g_{t1}, g_{t2}, g_{t3}\}$ denote the input space and $P = \{p_0, p_1\}$ denote the label space. Our model is to learn a function f: $V \rightarrow P$ from the training set $D = \{v, p\}$, which maps the next exercise x_{n+1}, which has not yet been done by the student, onto a proper label $p \in P$. p_1 means the exercise was answered correctly, while p_0 means the exercise was answered incorrectly. $x_i = \{q_i, b_i\}$, where q_i is the ith exercise tag and b_i denotes a student's learning behavior when (s)he answers the ith exercise. g_{t1}, g_{t2} and g_{t3} are the values of the average learning ability of three groups containing students whose ability is good, medium, or poor, respectively. Table I summarizes our notation.

The model architecture is shown in Fig. 1. The main stages are Xgboost, student clustering, feature splicing, and the DKVMN model. In the following sections, we will describe these stages in detail.

A. Xgboost

In order to carry out exercise record segmentation and learning ability calculation in follow-up stages, it is necessary to predict whether the student answered an exercise correctly. Xgboost performs this task.

Xgboost (extreme gradient boosting) was proposed by Chen and Guestrin [28]. It is an improved gradient lifting

TABLE I
MATHEMATICAL NOTATIONS

Name	Description
x_t	a exercise record containing exercise tag and learning behavior
g_{ti}	the learning ability of the student group
q_t	exercise tag
v_t	cross-feature which combines a student's learning behavior and learning ability values
M_t	value matrix
M_k	key matrix
w_t	correlation weight
e_t	erase vector
a_t	add vector
r_t	knowledge state
p_t	the exercise was answered correctly or not.
f_t	learning function
k	the number of segmentation
$C(x)_{1:k}$	the correct rate of students in the exercise in a partition of length k
$I(x)_{1:k}$	the error rate of students in the exercise in a partition of length k
$R(x)_{1:k}$	the difference between the accuracy rate and the error rate
$\|N_t\|$	the total number of student's exercises in every segment
ac_t	the times of a student attempts to do the exercise
fr_t	whether the student asks for help or not
M	the vector size at each memory location
N	the number of memory locations

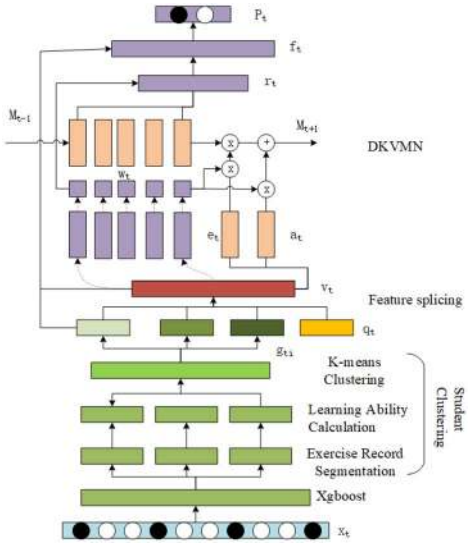

Fig. 1. Architecture of the proposed model.

learning algorithm, which is a method in boosting. The principle is to use the idea of the iterative operation to convert a large number of weak classifiers into strong classifiers in order to achieve accurate classification results. Xgboost generates a weak learner at each step and accumulates it into the total model. The Xgboost algorithm has the characteristics of fast running speed, good accuracy, and generalization ability, making it suitable for our model.

Xgboost takes the student's exercise number, attempt number, and prompt request as input. The output is a prediction of whether the student answered the exercise correctly. These behavior features allow us to better analyze the student's learning process in order to accurately shape the student's knowledge state and hence further improve the accuracy of knowledge tracing.

B. Student Clustering

In this stage, students are clustered into three groups based on whether their ability is good, medium, or poor. Distinguishing learning abilities can help us model each student's knowledge state more accurately. A student's learning abilities are evaluated based on their previous learning and exercise-solving abilities. Students' exercise records are also segmented, each segment containing the same number of exercises. For each student in each segment, we use their probability of answering correctly as a measure of their learning ability. Then, *k*-means clustering is used to divide students into three groups with different learning abilities. The segmentation reflects the changes in student's learning abilities as they do the exercises. We now describe these steps in more detail.

1) Exercise Record Segmentation: We divide each student's sequence of exercises into multiple segments according to a certain length. This has two functions: first, the student's learning ability is a dynamic process of change, so it needs to be re-evaluated in each segment. Second, dividing the student learning sequence into multiple segments can reduce computational burden and memory space allocation for learning in long sequences. In our experiments, we chose a segment that comprised five exercises.

2) Learning Ability Calculation: Because the student's sequence of exercises is a time series, changes in learning ability can be deduced from changes in exercise results. Therefore, we express the learning ability by calculating the proportion of a student's exercises which are answered correctly—the correct rate. We define the student's learning ability value as the difference between the correct rate and the error rate for the student's exercises in the sequence, as follows:

$$C(x)_{1:k} = \sum_{t=1}^{k} \frac{p_t == 1}{|N_t|} \tag{1}$$

$$I(x)_{1:k} = \sum_{t=1}^{k} \frac{p_t == 0}{|N_t|} \tag{2}$$

$$R(x)_{1:k} = C(x)_{1:k} - I(x)_{1:k} \tag{3}$$

where p_t is a binary value indicating whether the student has answered the exercise correctly, $C(x)_{1:k}$ and $I(x)_{1:k}$ are the correct rate and error rate in a partition of length k, $|N_t|$ is the total number of student's exercises in this segment, and $R(x)_{1:k}$ is the difference between the accuracy rate and the error rate in this division, that is, the student's learning ability.

3) K-Means Clustering: Through *k*-means clustering, students are assigned to three groups with similar abilities based on the learning ability values reflected in each practice segment. By defining three centroids, each one clusters a given dataset. The algorithm flow is as follows.

1) Randomly select three points as centroids.
2) Assign each object to the group with the closest centroid.
3) After all the objects have been specified, recalculate the positions of the three centroids.

TABLE II
STUDENT GROUP LEARNING ABILITY

Segment Length	Excellent	Medium	Low
5	0.8217	0.7597	0.6312
10	0.8260	0.7602	0.6214
20	0.8036	0.7560	0.6102
30	0.8219	0.7607	0.6177

4) Repeat steps 2 and 3 until the center of mass no longer moves.

After the clustering is completed, the average learning ability of each group can be calculated. Finally, each student in a group is assigned the ability value calculated for that group. Table II shows the average learning ability values for students in different groups and for various segment sizes.

C. Feature Splicing

Crossover is a method of encoding two or more features into one feature in order to represent the concurrent performance of these features [29]. We propose the cross-feature v_t that combines a student's learning ability values as follows:

$$v_t(q_t, p_t, g_t) = q_t + \frac{1}{ac_t * p_t + fr_t * \max(fr_t) * p_t} + g_{ti} \quad (4)$$

where q_t is the exercise number, p_t is 1 if the student correctly answered an exercise (otherwise 0), g_t is the learning ability of the student, and g_{ti} indicates the learning ability of the student group. fr_t is whether the student asks for help or not, and ac_t is the number of student attempts. As a new feature, v_t is input to the DKVMN to understand the similarities of exercises and to track the knowledge that students have.

D. DKVMN Model

The DKVMN model includes three processes: 1) a weight calculation; 2) a read mechanism; and 3) a write mechanism. The weight calculation determines the relationship between the input fusion feature and the knowledge skill. The read mechanism is used to predict the student's exercise result, and the write mechanism dynamically updates the student's knowledge status. The overall structure of the model is shown at the top of Fig. 1, above. The lower purple part of the figure is the weight calculation, that is, finding the relationship between the exercises and the knowledge skills. The upper purple part is the reading process, which predicts a student's future results, and the light brown is the writing process, which updates their knowledge status according to their record.

In the figure, Mt is an $N \times M$ matrix, where N is the number of memory locations and M is the vector size at each location. At each timestamp, the input is the fusion feature v_t, and the output is the probability of correctly answering the exercise. The storage matrix Mt is then updated with the tuple of the fusion feature and the result of the exercise. v_t is an embedding feature that contains the student's exercise number, the behavior of the exercise, and the value of the student's learning ability (see Section III-C). Add vector at to update each memory slot. at is a row vector. The value memory is updated at each time by it. Specifically, the correlation weight between the input and the knowledge skill is first calculated by SoftMax activation.

In the reading process, the student's mastery of the exercise is obtained by computing the weighted sum of all the memory slots in the knowledge state matrix M_t. Compared with the original DKVMN, our model not only considers the student's knowledge status but also takes into account the learning ability of the student group. We apply feature splicing to the degree of knowledge of a student and their learning ability to form a composite feature (Section III-C), and then obtain a summary vector through a fully connected layer to indicate the degree of knowledge of the student and their learning ability. Finally, we output the probability f_t of a student answering a question correctly through a fully connected layer with sigmoid activation

$$f_t = \mathrm{Tanh}\big(w_1\big[r_t, v_t, g_{ti}\big] + b_1\big). \quad (5)$$

In the writing process, when the student answers the exercise, the model updates the student's knowledge state matrix according to the correctness of the student's answer. Through the joint vector v_t, which is defined in 4, the student's knowledge growth after completing this exercise is obtained, and this is written into the knowledge state matrix. Before adding new information, the previous content is erased by using the erasure vector. Then, we update the student's knowledge status by updating the vector. This erasing and updating mechanism simulates the student's forgetting in the learning process.

TABLE III
DATASET DESCRIPTION

Name	Number of students	Number of records	Number of concepts	time
ASSISTment2009	4,151	325,637	124	2009-2010

TABLE IV
DATASET FEATURE NAMES

Name	Description
user_id	Student ID
problem_id	Exercise number
correct	Exercise results: 1 = correct, 0 = wrong
attempt_count	Number of student attempts at an exercise
first_response	Time spent by a student on an exercise
first_action	Whether the student requested help during the first exercise

IV. EXPERIMENTS

A. Datasets

To validate the effectiveness of our proposed method, we used the ASSISTment2009 public dataset for the experiments [17]. This dataset is a record of student's answers to mathematics exercises collected from the ASSISTments online tutorial platform and is a standard dataset in the field of knowledge tracing. Table III provides summary information for the dataset and Table IV shows the main feature names.

B. Experiment Setup

In our experiments, five-fold cross-validation is used. Each fold involves randomly splitting the dataset into 80% training data and 20% test data at the student level. So, both training and test subsets within a fold contain response records from different students. In order to measure the performance of our algorithm and compare it with existing approaches, we adopt area under curve (AUC) which is a standard measure in the knowledge tracing field. The input exercise data are presented to neural networks using "one-hot" input vectors. For the DKVMN, the learning rate and the number of iterations are set to 0.05 and 50. We learn the initial value of both the key matrix and the value matrix in the training process. For the Xgboost and k-means modules, the parameters used are those provided as defaults in the Python toolbox.

C. Experimental Results

We carried out four experiments. Experiment 1 compares DKVMN-LA with the knowledge tracking algorithms in the existing literature, including the three classic algorithms: 1) BKT; 2) DKT; and 3) DKVMN, and the improved algorithms DKT-F and DKT-DSC based on the DKT algorithm; DKT-F adds behavioral features to DKT, while DKT-DSC dynamically groups students. In Table V, DKVMN-LA performs significantly better than state-of-the-art models. Compared with the standard DKVMN that has an AUC of 81.6%, our DKVMN-LA model achieves an AUC of 91.9%, which represents a gain of 12%. In addition, our model gives superior predictions to the latest improved DKT and DKVMN, as we show in the next experiment.

Experiment 2 compares DKVMN-LA with the DKT-LA algorithm, which is a knowledge tracking algorithm proposed by combining the method of this article with the DKT algorithm. The results are in Table VI. We found that in DKT-LA, there are many improvements in the prediction performance compared to the original DKT model. However, the model is still not as good as the model in this article. This is because the improved dynamic key-value neural network can better track the student's knowledge status.

In Experiment 3, the performance of DKVMN-LA with different segment sizes is investigated. The results are in Table VII. In this experiment, segments contain 5, 10, 20, and 30 exercises. DKVMN-LA performs the best when each segment contains five exercises. Generally, we found that when the number of exercises in each segment is less, the prediction results of the model are more accurate. This is because the smaller the segment, the more it reflects the continuous change of a student's learning ability with exercises. So, we can prove that the learning ability features obtained by segmentation really help improve the performance of the model.

Experiment 4 tries five different combinations of features: 1) attempt count and first action; 2) attempt count, attempt time, first action, and learning ability; 3) attempt count, attempt time, and learning ability; 4) attempt time, first action, and learning ability; and 5) attempt count, first action, and learning ability. As shown in Table VIII, the algorithm's predictive performance is best when using only the three features of combination 5), that is, attempt count, first action, and learning ability. We think this is because there is a clear linear relationship between the attempt time and the attempt count. We have experimentally proved that the method of this article only needs to collect the two behavioral characteristics of the attempt count and first action to achieve the best predictive performance, which helps us improve the efficiency of collecting student record data. In addition, when using combination 5), the prediction performance of the algorithm is 2.2% higher than combination 1). It can be seen that the learning ability feature has an important influence on the modeling of a student's knowledge status.

TABLE V
EXPERIMENT 1: COMPARISON OF DKVMN-LA WITH EXISTING ALGORITHMS

Model	AUC(%)
BKT [25]	63.0
DKT [25]	80.5
DKVMN [25]	81.6
DKT-F [23]	86.7
DKT-DSC [24]	91.0
DKVMN-LA	**91.9**

TABLE VI
EXPERIMENT 2: DKVMN-LA AND DKT-LA COMPARED

Model	AUC(%)
DKT-LA	90.5
DKVMN-LA	**91.9**

TABLE VII
EXPERIMENT 3: PERFORMANCE OF DKVMN-LA WITH DIFFERENT SEGMENT

Segment Length	AUC(%)
5	**91.9**
10	91.8
20	91.7
30	91.3

TABLE VIII
EXPERIMENT 4: PERFORMANCE OF DKVMN-LA WITH DIFFERENT FEATURE SPLICINGS

Feature Splicing	AUC(%)
(1) Attempt count, first action	89.7
(2) Attempt count, Attempt time, First action, Learning ability	91.6
(3) Attempt count, Attempt time, Learning ability	90.9
(4) Attempt time, First action, Learning ability	91.2
(5) Attempt count, First action, Learning ability	**91.9**

D. Visualizations

In order to visually demonstrate the learning ability and interpretability of our approach, we conducted some visualization experiments. Usually, each exercise is associated with a single knowledge concept. We randomly choose 30 distinct exercises from the ASSISTment2009 public dataset. These 30 exercises were drawn from five concepts: 1) congruence;

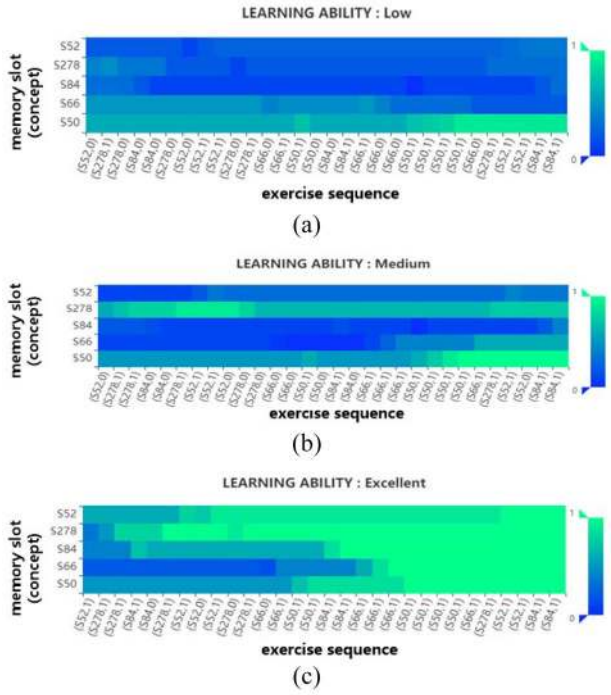

Fig. 2. Knowledge state of different learning ability. (a) Knowledge state of Low learning ability. (b) Knowledge state of medium learning ability. (c) Knowledge state of excellent learning ability.

2) addition and subtraction positive decimals; 3) prime number; 4) write linear equation from graph; and 5) ordering fractions. Three examples of students' changing knowledge states on these five concepts after they have answered 30 exercises are shown in Fig. 2. Fig. 2(a)–(c) shows the changing knowledge state of students whose learning ability is low, medium, and excellent, respectively. Knowledge concepts are on the left-hand side of the heatmap in Fig. 2. The five knowledge concepts are labeled S52, S278, S84, S66, and S50. At the bottom of the heatmap, exercise sequences are listed. (S52, 0) means a student incorrectly answered the exercise that is associated with knowledge concept S52; while, (S278, 1) means the student correctly answered the exercise associated with knowledge concept S278. Blue in the figures indicates that the student did not master the knowledge concepts at all. Conversely, green indicates that the student completely mastered the knowledge concepts. The change from blue to green reflects the change in a student's mastery of the underlying knowledge concepts. Each time the student answers an exercise, the concept state of the discovered concept will increase or decrease. Thus, every horizontal color bar in Fig. 2 shows the student group's changing knowledge state relative to the underlying concepts.

Comparing the three figures, it is obvious that students with low learning ability improve their knowledge status slowly: green appears later as they answer questions. In contrast, the students with excellent learning ability improve their knowledge status quickly, so green appears earlier. For example, after answering 30 exercises, the low-ability students only mastered concept S50, while the high-ability ones mastered all the concepts. We also found that the change of knowledge state differs between the three student groups when answering the same exercise. In Fig. 2(a), after low-ability students correctly answer the exercise associated with S50 five times, they almost understand S50. In Fig. 2(c), however, high-ability students correctly answer the S50 concept only three times before they totally master it. We can conclude from these results that our approach is successful at modeling student learning ability.

V. Conclusion

In this article, we have proposed DKVMN-LA, a knowledge-tracking algorithm that combines learning capabilities and behavioral features. DKVMN-LA solves two problems with existing algorithms. First, they ignore the differences in learning ability between students, and second, they assume that students will not change their learning ability through answering questions. The DKVMN-LA algorithm segments students' long-term exercise records, defines and calculates students' learning abilities in each segment, and dynamically clusters students into three different groups. We also proposed a new exercise record representation method and combined the learning ability features to improve the DKVMN. Our experiments showed that the proposed model is significantly better than other models in predicting performance. The original DKVMN model ignored the student behavior data collected by the online learning platform and assumed that all students have the same learning ability, regardless of the difference between each student's ability and learning rate. In contrast, our model improves DKVMN by introducing behavior features derived from student's exercises and introducing student's learning ability values, determined by how well they learn.

References

[1] S. Trivedi, Z. A. Pardos, and N. T. Heffernan, "Clustering students to generate an ensemble to improve standard test score predictions," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2011, pp. 377–384.

[2] H. Mei, M. Bansal, and M. Walter, "Coherent dialogue with attention-based language models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 3252–3258.

[3] M. K. Hashem and B. J. Oommen, "Using learning automata to model the 'learning process' of the teacher in a tutorial-like system," in *Proc. IEEE 22nd Int. Symp. Comput. Inf. Sci.*, 2007, pp. 1–6.

[4] T. M. Nguyen and Q. J. Wu, "Bounded asymmetrical student's-T mixture model," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 857–869, Jun. 2014.

[5] X. Wan, F. Anma, T. Ninomiya, and T. Okamoto, "Collaboenote: A hybrid recommender system for group learning support," *Int. J. Comput. Sci. Netw. Security*, vol. 8, no. 9, pp. 166–171, 2008.

[6] C. V. Le, Z. A. Pardos, S. D. Meyer, and R. Thorp, "Communication at scale in a MOOC using predictive engagement analytics," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2018, pp. 239–252.

[7] F. Wang, B. Chen, C. Lin, J. Zhang, and X. Meng, "Adaptive neural network finite-time output feedback control of quantized nonlinear systems," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1839–1848, Jun. 2018.

[8] J.-J. Vie. (2018). *Deep Factorization Machines for Knowledge Tracing*. [Online]. Available: arXiv:1805.00356.

[9] J.-J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 750–757.

[10] Y. Huang, "Deeper knowledge tracing by modeling skill application context for better personalized learning," in *Proc. Conf. User Model. Adapt. Pers.*, 2016, pp. 325–328.

[11] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5329–5336.

[12] Y. Gong, J. E. Beck, and N. T. Heffernan, "How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis," *Int. J. Artif. Intell. Educ.*, vol. 21, nos. 1–2, pp. 27–46, 2011.
[13] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User Adapt. Interact.*, vol. 4, no. 4, pp. 253–278, 1994.
[14] B. Li, Y. Liu, and X. Wang, "Gradient harmonized single-stage detector," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8577–8584.
[15] Y. Wang and N. T. Heffernan, "Leveraging first response time into the knowledge tracing model," in *Proc. Int. Educ. Data Mining. Soc.*, 2012, pp. 176–179.
[16] Z. A. Pardos and N. T. Heffernan, "Modeling individualization in a Bayesian networks implementation of knowledge tracing," in *Proc. Int. Conf. User Model. Adapt. Pers.*, 2010, pp. 255–266.
[17] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized Bayesian knowledge tracing models," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2013, pp. 171–180.
[18] J. González-Brenes, Y. Huang, and P. Brusilovsky, "General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge," in *Proc. 7th Int. Conf. Educ. Data Mining.*, 2014, pp. 84–91.
[19] D.-R. Liu, S.-P. Chuang, and H.-y. Lee, "Attention-based memory selection recurrent network for language modeling," 2016. [Online]. Available: arXiv:1611.08656.
[20] J. Weston *et al.*, "Towards ai-complete question answering: A set of prerequisite toy tasks," 2015. [Online]. Available: arXiv:1502.05698.
[21] C. Piech *et al.*, "Deep knowledge tracing," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 505–513.
[22] C.-K. Yeung and D.-Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proc. 5th Annu. ACM Conf. Learn. Scale*, 2018, pp. 1–10.
[23] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan, "Incorporating rich features into deep knowledge tracing," in *Proc. 4th ACM Conf. Learn. Scale*, 2017, pp. 169–172.
[24] S. Minn, Y. Yu, M. C. Desmarais, F. Zhu, and J.-J. Vie, "Deep knowledge tracing and dynamic student classification for knowledge tracing," in *Proc. IEEE Int. Conf. Data Mining. (ICDM)*, 2018, pp. 1182–1187.
[25] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 765–774.
[26] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.
[27] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, 1997.
[28] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining.*, 2016, pp. 785–794.
[29] H. Yang and L. P. Cheung, "Implicit heterogeneous features embedding in deep knowledge tracing," *Cogn. Comput.*, vol. 10, no. 1, pp. 3–14, 2018.

Xia Sun received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2006.

She is a Professor with the School of Information Science and Technology, Northwest University, Xi'an. She has coauthored 40 articles and is an editor or co-editor of four books. Her current research interests include natural language processing and intelligent education.

Prof. Sun has served as a Reviewer for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Chinese Journal of Electronics*.

Xu Zhao received the B.S. and M.S. degrees in computer science from Northwest University, Xi'an, China, in 2017 and 2020, respectively.

In addition to student classification, he has applied deep learning to student performance prediction. His research is in the areas of knowledge tracing and deep learning.

Bo Li received the B.S. degree in computer science from Northwest University, Xi'an, China, in 2018.

His research is in the areas of exercise prediction. In addition, he combines matrix factorization with the cognitive diagnosis method to predict exercise individually.

Yuan Ma received the B.S. degree in computer science from Northwest University, Xi'an, China, in 2019, where she is currently pursuing the M.S. degree.

Her research combines neural networks with natural language processing. Recent works have developed new neural-network algorithms and frameworks for sentiment classification.

Richard Sutcliffe received the Ph.D. degree from the University of Essex, Colchester, U.K., in 1989.

He is an Associate Professor with Northwest University, Xi'an, China. Recent projects have included persuasive conversational agents, public sector message classification, analysis of classical music texts, and personality and translation ability. He has coauthored 101 articles and is a co-editor of three books and ten conference proceedings. His research interests are in the areas of natural language processing, information retrieval, and music information retrieval.

Dr. Sutcliffe has reviewed for *Artificial Intelligence Review*, *Computational Linguistics*, *Computers and the Humanities*, *Information Processing and Management*, *Information Retrieval Journal*, *Journal of Natural Language Engineering*, and *Journal Traitement Automatique des Langues*. He has reviewed for conferences including ACL, CIKM, COLING, IJCNLP, LREC, NAACL-HLT, and SIGIR.

Jun Feng received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2006.

She is a Professor with the School of Information Science and Technology, Northwest University, Xi'an, China. She has coauthored 132 articles and is a co-editor of three books. Recent projects have included medical image analysis with deep learning, and intelligent education based on AI and brain–human interaction. Her research areas include pattern recognition and machine learning, especially in the fields of medical imaging analysis and intelligent education.

Prof. Feng has reviewed for many journals, including IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE ACCESS, JIVP, MTAP, JDIM, CJC, JCAD, OPE, and INFPHY. She has reviewed for conferences, including IEEE-VR, MICCAI, SIGCSE, IWCSE, and CompEd. She is a member of ACM.