4-2019

Dynamic student classification on memory networks for knowledge tracing

Sein MINN
Polytechnic School of Montreal

Michel C. DESMARAIS
Polytechnic School of Montreal

Feida ZHU
Singapore Management University, fdzhu@smu.edu.sg

Jing XIAO
Ping An Technology (Shenzhen) Co Ltd

Jianzong WANG
Ping An Technology (Shenzhen) Co Ltd

Dynamic Student Classiffication on Memory Networks for Knowledge Tracing

Sein Minn[1(✉)], Michel C. Desmarais[1], Feida Zhu[2], Jing Xiao[3], and Jianzong Wang[3]

[1] Polytechnique Montreal, Montreal, Canada
{sein.minn,michel.desmarais}@polymtl.ca
[2] Singapore Management University, Singapore, Singapore
fdzhu@smu.edu.sg
[3] Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China
{xiaojing661,wangjianzong347}@pingan.com.cn

Abstract. Knowledge Tracing (KT) is the assessment of student's knowledge state and predicting whether that student may or may not answer the next problem correctly based on a number of previous practices and outcomes in their learning process. KT leverages machine learning and data mining techniques to provide better assessment, supportive learning feedback and adaptive instructions. In this paper, we propose a novel model called Dynamic Student Classification on Memory Networks (DSCMN) for knowledge tracing that enhances existing KT approaches by capturing temporal learning ability at each time interval in student's long-term learning process. Experimental results confirm that the proposed model is significantly better at predicting student performance than well known state-of-the-art KT modelling techniques.

Keywords: Massive open online courses · Knowledge tracing · Key-value memory networks · Student clustering · LSTMs

1 Introduction

Guiding human for solving problems efficiently and effectively is a recurring topic in educational research. Knowledge tracing (KT) gained credibility in this research community to provide appropriate and adaptive guidance in the learning process. KT aims to assess skills that are mastered or not, and use this information to tailor learning experience, whether in MOOCs, in a tutoring system or in web, results to name a few example for applications. For example, when a problem such as "$1 + 2 \times 3.5 =$?" is given to a student, she has to master the skills of *addition* and *multiplication* for solving that problem. The probability of getting a correct answer mainly depends on the mastery level of these two

This work was supported by NSERC Canada, Discovery grant program and Pinnacle lab for analytics at Singapore Management University.

skills behind that problem. Mastering a skill can be achieved by doing practices on that skill. The goal of knowledge tracing is to track the knowledge state of students based on observed outcomes on their previous practices [5]. This task is also known as student modelling. Research on KT can be traced back to the late 1970s and a wide array of Artificial Intelligence and Knowledge Representation techniques have been explored [3,14]. In environments where the student learns as she interacts with the system, which is specifically the case for learning environments such as MOOCs, modeling student skill mastery involves a temporal dimension. For instance, a sequence problems involving the same skills set may be failed at first, but succeeded later on because the student's skill mastery has increased. Yet, other factors can influence the success outcome, such as the two problem's difficulty level, forgetting, guessing and slipping, and an array of other factors that induce noise if they are not accounted for [11,12].

The dynamic nature of KT in learning environments leads to approaches that have the capacity to model temporal or sequential data. In this paper we propose a novel model for knowledge tracing, Dynamic Student Classified Memory Networks (DSCMN). The model can capture temporal learning ability in student's long-term memory and assess mastery of knowledge state simultaneously. Temporal learning ability refers to the rate of learning of specific skills. It can be tied to phenomena like wheel spinning, where a student fails to learn a skill even after numerous attempts [17]. It relies on an RNN architecture to improve performance prediction. The hypothesis we make is that learning ability can change in time and tracing this factor can help predict future performance.

The rest of this paper is organized as follow. Section 2 reviews the related work on the student modelling techniques for predicting student's performance from data. Section 3 presents the proposed DSCMN model. Section 4 mentioned experimental datasets used. Experimental results are described in Sect. 5 and finally Sect. 6 concludes this work and discusses future avenues of research.

2 Knowledge Tracing

Successful learning environments such as the Cognitive tutors series and the ASSISTments platform rely on some form of KT [6]. In these systems, each problem is labeled with underlying skills required to correctly answer that problem. KT can be seen as the task of supervised sequential learning problem where the model is given student past interactions with the system that includes: skills $S = \{s_1, s_2, \ldots, s_t\}$ along with response outcomes $R = \{r_1, r_2, \ldots, r_t\}$. KT predict the probability of getting a correct answer to the next problem, which mainly depends on mastery of corresponding skill s associated with problems $P = \{p_1, p_2, \ldots, p_t\}$. So we can define the probability of getting correct answer as $p(r_t = 1|s_t, X)$ where $X = \{x_1, x_2, \ldots, x_{t-1}\}$ and $x_{t-1} = (s_{t-1}, r_{t-1})$ is a tuple containing response outcomes r to skill s at time $t-1$. Then, we review here four of the best known state-of-the-art KT modelling methods for estimating student's performance.

2.1 Bayesian Knowledge Tracing (BKT)

BKT is arguably the first model to relax the assumption on static knowledge states. Earlier approaches such as IRT would assume the student does not learn between answers, which is a reasonable assumption for testing, but not for learning environments. BKT was introduced for knowledge tracing within a learning environment [5]. In its original form, it also assumes a single skill is tested per item, but this assumption is relaxed in later work. The data are partitioned by skill and learning a model on each dataset leads to a specific model for each skill s. The standard BKT model is comprised of 4 parameters which are typically learned from the data while building a model for each skill. The model's inferred probability mainly depends on those parameters which are used to predict how a student masters a skill given that student's chronological sequence of incorrect and correct attempts to questions of that skill thus far [1]. To estimate the probability that a student knows the skill given his performance history, BKT needs to have four probabilities: $P(L_0)$, initial probability of mastery of skill L_0; $P(T)$, transition probability from a state of non mastery to mastery; and $P(S)$, slipping, the probability of a wrong answer in spite of mastery, and $P(G)$, guessing, the probability of a correct answer in spite of non mastery.

$$P(L_n|Correct) = \frac{P(L_{n-1})(1 - P(S))}{P(L_{n-1})(1 - P(S)) + (1 - P(L_{n-1}))P(G)} \tag{1}$$

$$P(L_n|Incorrect) = \frac{P(L_{n-1})P(S)}{P(L_{n-1})P(S) + (1 - P(L_{n-1}))(1 - P(G))} \tag{2}$$

$$P(L_n) = P(L_{n-1}|Outcome) + (1 - P(L_{n-1}|Outcome))P(T) \tag{3}$$

2.2 Deep Knowledge Tracing (DKT)

Similar to BKT, Deep Knowledge Tracing (DKT) [13] works on the skill sequence of attempts but the author leveraged the advantages of neural networks and break the restriction of skill separation and binary state assumption. It takes the previous history of attempts by students and transforms each attempt into one-hot encoded feature vector. Then, those features are fed into a neural network as input and pass information through the hidden layers of the network and onto the output layer. The output layer provides the predicted probability that the student would answer that particular problem correctly in the system.

DKT uses Long Short-Term Memory (LSTM) [8] to represent the latent knowledge space of students along with the number of practices dynamically. The increase in student's knowledge through an assignment can be inferred by utilizing the history of student's previous performance. DKT summarizes a student's knowledge state of all skills in one hidden state in hidden layer. A student's skill mastery state at certain time stamp is defined by the following equations:

$$h_t = \tanh(W_{hx}x_{t-1} + W_{hh}h_{t-1} + b_h), \tag{4}$$

$$p(s_t) \in y_t = \sigma(W_{yh}h_t + b_y), \tag{5}$$

In DKT, both tanh and the sigmoid function are applied element wise and parameterized by an input weight matrix W_{hx}, recurrent weight matrix W_{hh}, initial state h_0, and readout weight matrix W_{yh}. Biases for latent and readout units are represented by b_h and b_y.

2.3 Dynamic Key-Value Memory Network (DKVMN)

DKVMN was proposed an enhancement to DKT that utilizes a neural network module called external memory slots to encode the knowledge state of students and use as key and value components to encode the knowledge state of students [19]. Learning or forgetting of a particular skill are stored in those two components and controlled by read and write operations through additional attention mechanisms. Learning or forgetting of a particular skill is stored in those two components and controlled by read and write operations through additional attention mechanisms.

Unlike DKT, DKVMN performs reading and writing operations to perform local state transitions by avoiding global and unstructured state-to-state transformation in hidden layer. Knowledge state of a student is traced by reading and writing to the value memory slots using correlation weight computed from input skills and the key memory slots. It is comprised of three main steps:

Correlation: The correlation weight of input skill s_t is computed by utilizing the softmax activation of the inner product between k_t and key memory slot $M^k(i)$:

$$w_t = Softmax(k_t^T M^k(i)) \tag{6}$$

where k_t is the continuous embedding vector of s_t and $Softmax(z_i) = e^{z_i} / \sum_j e^{z_j}$ id differentiable. Correlation weight w_t is used in both reading and writing process in later.

Reading: The mastery m_t of s_t is retrieved by weighted sum of values in value memory slots by using w_t:

$$m_t = \sum_{i=1}^{N} (w_t(i) M_t^v(i)) \tag{7}$$

Prediction: The probability of answering the problem with underlying skill $p(s_t)$ is calculated by using mastery level m_t:

$$f_t = \tanh(W_1^T [m_t, k_t] + b_1) \tag{8}$$

$$p(s_t) = \sigma(W_2^T f_t + b_2) \tag{9}$$

Where $\tanh(z_i) = (e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i})$ and $\sigma(z_i) = 1/1 + e^{-z_i}$.

Writing: After the student answers the problem, the model will update the value memory according to response (r_t) of student. A joint embedding of $x_t = (s_t, r_t)$ is converted into embedding values v_t and written to the value memory with same correlation weight w_t used in read process. Erasing is performed before adding new information by using:

$$e_t = \sigma(E^T v_t + b_e), \tag{10}$$

$$\tilde{M}_t^v(i) = M_{t-1}^v(i)[1 - w_t(i)e_t], \tag{11}$$

where 1 is a row-vector of all 1-s. If both the weight at the location and the erase element are 1, the elements of a memory location are reset to zero. No changes are performed in the case of either erase signal or the weight is zero. After erasing previous memory, a_t is used to update each memory slots in value memory.

$$a_t = \tanh(D^T v_t + b_a)^T, \tag{12}$$

$$M_t^v(i) = M_{t-1}^v(i) + w_t(i)a_t, \tag{13}$$

where E and D are the transformation matrix with shape of $d_v \times d_v$. This erase-followed-by-add mechanism allows forgetting and strengthening knowledge states of student learning process [19] which is not able in other RNN based models.

2.4 Deep Knowledge Tracing with Dynamic Student Classification (DKT-DSC)

DKT-DSC was introduced to overcome the problem of short-term learning ability of student when applied to the KT task [10]. During the evaluation of student learning ability, DKT-DSC encodes student's past performance by using the following equation:

$$Correct(s_j)_{1:z} = \sum_{z=1}^{Z} \frac{(s_j = 1)}{|N_j|}, \tag{14}$$

$$Incorrect(s_j)_{1:z} = \sum_{z=1}^{Z} \frac{(s_j = 0)}{|N_j|}, \tag{15}$$

$$R(s_j)_{1:z} = Correct(s_j)_{1:z} - Incorrect(s_j)_{1:z}, \tag{16}$$

$$d_{1:z}^i = (R(s_1)_{1:z}, R(s_2)_{1:z}, \ldots, R(s_n)_{1:z}). \tag{17}$$

in which $Correct(s_j)_{1:z}$ represents the ratio of skill s_j being correctly answered and $Incorrect(s_j)_{1:z}$ for the ratio of incorrectly answered. $d_{1:z}^i$ is the vector of skills mastery for student i on n skills and for time interval 1 to z. $|N_j|$ is the total number of attempts that student i has done on each skill s_j. Evaluating temporal learning ability by assigning students into a group with similar ability c_z at each time interval z by using k-means clustering on encoded data $d_{1:z-1}^i$ [2,9,10] and then the model invokes an RNN to trace her knowledge according to her learning ability c_z at each time interval.

$$h_t = \tanh(W_{hx}[x_{t-1}, s_t, v_t] + W_{hh}h_{t-1} + b_h), \tag{18}$$

$$p(s_t^{c_z}) \in y_t = \sigma(W_{yh} h_t + b_y), \tag{19}$$

where v_t contains success and failure levels of skill s_t until time $t-1$ thus far.The probability of $p(s_t^{c_z}) \in y_t$ represents the probability of getting correctness of problem with associated skill s_t for the student with her temporal learning ability c_z in that time interval z while other models ignore the long-term learning ability in student learning process. DKT-DSC applies temporal value of student's learning ability at each time interval to improve the individualization in long-term knowledge tracing process.

3 Dynamic Student Classification on Memory Networks (DSCMN)

Despite a better accuracy to assess the mastery of skills than DKT, each of the above models has deficiencies for dealing with the KT task. In both DKT and DKVMN, temporal student's long-term learning ability is ignored. So the model cannot evaluate which level of learning ability the student achieved for a given time interval in a long term learning process. In DKT and DKT-DSC, LSTM uses single state vector to encode the temporal information of student knowledge state with corresponding learning ability in a single hidden layer.

To model learning ability, we propose a novel model called Dynamic Student Classification on Memory Networks (DSCMN) that builds upon the advantages of DKVMN and DKT-DSC. DSCMN predicts student performance based on both of evaluated temporal student's long-term ability and assessed mastery of skills simultaneously at each time interval.

Evaluating Temporal Student's Learning Ability: Learning is a process that involves practice: students become proficient through practice. Besides, learning is also affected by the individual's ability to learn, or to become proficient with more or less practice [10].

To detect the regularities and changes of temporal learning ability of a student over series of time intervals in long-term learning process, we need to encode student past performance for predicting her learning ability in the current time interval with DKT-DSC's Eq. 17. The encoded vector of student's past performance is updated after each time interval. The K-means algorithm [9] is used to evaluate the temporal long-term learning ability of students in both training and testing at each time interval z by measuring the Euclidean distance between centroids achieved after training the DKT-DSC process [10] and assigning a nearest cluster label c_z as the long-term learning ability of a student at time z. Evaluation is started after the first 20 attempts and updated after each 20 attempts have been made by a student. For first time interval, every student is assigned with initial learning ability 1 as described in Fig. 1.

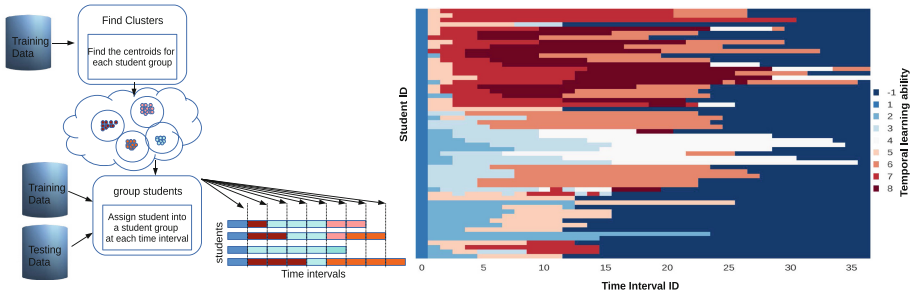

Fig. 1. Evaluation process of student's learning ability (Left) and Evolution of temporal learning ability in long-term learning process of random 56 students in ASSISTments 2009 dataset (Right)

Calculating Problem Difficulty: We measure problem difficulty as one of 10 levels [11,12]. Note that, in this study, the difficulty is associated with problems, not with skills themselves. The difficulty of a problem, $p_j \in D$, is determined as:

$$pd(p_j) = \begin{cases} \delta(p_j, pd), & \text{if } |N_j| \geq 4 \\ pd, & \text{else} \end{cases} \tag{20}$$

where:

$$\delta(p_j, pd) = \frac{\sum_i^{|N_j|} |\{p_{ij} == 0\}|}{|N_j|} \cdot pd \tag{21}$$

and where N_j is the set of students who attempted problem p_j, and p_{ij} is the outcome of the first attempt from student i, to problem p_j. An outcome of 0 is a failure. Constant pd is the problem difficulty (levels) that we wish to retain. It is described in function $\delta(p_j, pd)$ as shown in Eq. (20). Essentially, $\delta(p_j, pd)$ is a function that maps the average success rate of problem p_j onto (10) levels. For problems those do not have responses from at least 4 different students, problems with $|N_j| < 4$ in the dataset, we apply $pd_t = 5$ corresponding to 0.5 difficulty for those problems.

3.1 Assessing Student's Mastery of Skill

To assess the mastery of skill according to temporal learning ability, we use read and write process into two key and value memory slots as like in DKVMN. DSCMN also assess the mastery of skills using the correlation weight computed from the input skill and the key memory. In DSCMN, instead of using embedding values, one-hot encoded inputs are directly fed into memory networks by using Eqs. (6) and (7). Mastery m_t of skill s_t is obtained from reading process before writing x_t to value memory. Then the model writes x_t into value memory by using Eqs. (10) and (12) after the student answered the problem at time t (Fig. 2).

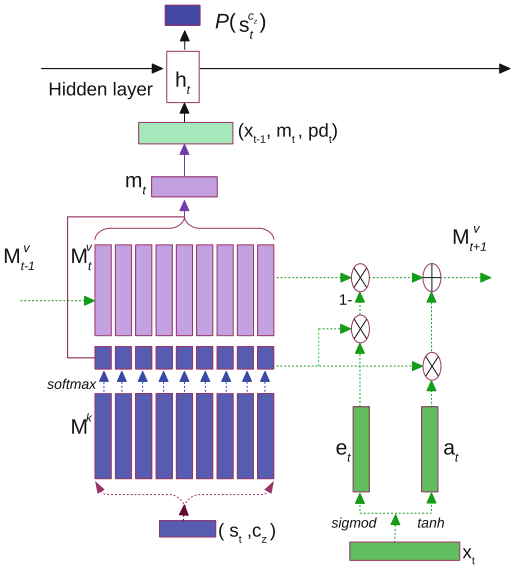

Fig. 2. Architecture of DSCMN

Prediction: The probability of answering the problem with underlying skill $p(s_t)$ of student in temporal learning ability c at time interval z is estimated by feeding previous response and mastery of skill in temporal learning ability of student into additional hidden layer and prediction is performed as follows:

$$h_t = \tanh(W_h[x_{t-1}, m_t, pd_t] + W_{hh}h_{t-1} + b_h), \tag{22}$$

$$p(s_t^{c_z}) \in y_t = \sigma(W_{yh}h_t + b_y), \tag{23}$$

Where c_z is the temporal learning ability of that student at time interval $t \in z$ and $[x_{t-1}, m_t, pd_t]$ encoded x_{t-1} previous response of skill s_{t-1} and mastery of skill s_t with skill id s_t and associated problem difficulty pd_t in temporal learning ability of student i at time interval z. DSCMN possess the ability to assess the mastery of skill based on temporal ong-term learning ability. Prediction is performed by using these factors and stored it in hidden state h_t.

Optimization: To improve the predictive performance of RNN based models, we trained with the cross-entropy loss l between p_t and actual response r_t for all RNN based models as follows:

$$l = \sum_t (r_t \log p_t + (1 - r_t) \log(1 - p_t)), \tag{24}$$

4 Datasets

In order to validate the proposed model, we tested it on four public datasets from two distinct tutoring scenarios in which students interact with a computer-

based learning system in the educational settings: (1) ASSISTments[1]: an online tutoring system that was first created in 2004 which engages middle and high-school students with scaffolded hints in their math problem. If students working on ASSISTments answer a problem correctly, they are given a new problem. If they answer it incorrectly, they are provided with a small tutoring session where they must answer a few questions that break the problem down into steps. Datasets are as follows: ASSISTments 2009–2010 (skill builder), ASSISTments 2012–2013, ASSISTments 2014–2015. (2) Cognitive Tutor. Algebra 2005–2006 [4][2]: is a development dataset released in KDD Cup 2010 competition from Carnegie Learning of PSLC DataShop. For all datasets, only first correct attempts to original problems are considered in our experiment. We remove data with missing values for skills and problems with duplicate records. To the best of our knowledge, these are the largest publicly available knowledge tracing datasets (Table 1).

Table 1. Overview of datasets

Dataset	Number of				Description
	Skills	Problems	Students	Records	
Cognitive Tutor	437	15663	574	808,775	KDD Cup 2010 [4]
ASSISTments	123	13002	4,163	278,607	2009–2010 [15]
	198	41918	28,834	2,506,769	2012–2013 [7]
	100	NA	19,840	683,801	2014–2015 [18]

5 Experimental Study

In this experiment, we assume every 20 attempts made by a student is a time interval. The total number of temporal values for student's learning ability used in our experiment is 8 (7 clusters and 1 for initial ability before evaluation in initial time interval for all students) for DKT-DSC and DSCMN. Five fold cross-validations are used to make predictions on all datasets. Each fold involves randomly splitting each dataset into 80% training students and 20% test students of the each datasets. For the input of DKVMN, initial values in both key and value memory are learned in training process. For other models, one hot encoding is applied. Initial values in value memory represents the initial knowledge state as prior difficulty for each skill and is fixed in the testing process.

We implement the all models with Tensorflow and DKT, DKT-DSC and DSCMN share same structure of fully-connected hidden nodes for LSTM hidden layer with the size of 200 for DKT, 200 for DKT-DSC and output size of memory

[1] https://sites.google.com/site/assistmentsdata/.
[2] https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

networks for DSCMN. For speeding up the training process, mini-batch stochastic gradient descent is used to minimize the loss function. The batch size for our implementation is 32, corresponding 32 to split sequences from each student. We train the model with a learning rate 0.01 and dropout is also applied to avoid over-fitting [16]. We set the number of epochs to 100. All models are trained and tested on the same sets of training and testing students.

For BKT, we use the Expectation Maximization (EM) algorithm and limit the number of iterations to 200. We learn models for each skill and make predictions separately. The results for each skill are averaged.

Table 2. AUC result for all tested datasets. Note that the results of DKT-DSC are slightly different than [10] after fixing bugs in the original code.

Datasets	Model				
	BKT	DKT	DKVMN	DKT-DSC	DSCMN
Cognitive Tutor	64.2 ± 1.0	78.4 ± 0.6	78.0 ± 0.0	79.2 ± 0.5	$\mathbf{86.0 \pm 0.5}$
ASSISTments09	65.1 ± 1.0	72.1 ± 0.5	71.0 ± 0.5	73.5 ± 0.6	$\mathbf{81.2 \pm 0.4}$
ASSISTments12	62.3 ± 0.0	71.3 ± 0.0	70.7 ± 0.1	72.1 ± 0.1	$\mathbf{78.5 \pm 0.1}$
ASSISTments14	61.1 ± 1.0	70.7 ± 0.4	70.0 ± 0.1	$\mathbf{71.6 \pm 0.2}$	71.0 ± 0.01

In Table 2, DSCMN performs significantly better than state-of-the-art models in three datasets. On the Cognitive Tutor dataset, compared with the standard DKT which has an maximum test AUC of 78.4, 79.2 in DKT-DSC and only 78.0 in DKVMN. The DSCMN model can achieve AUC = 86.0, with a notable gain of 10% over the original DKT and DKVMN, and 8% over DKT-DSC. For the ASSISTments09 dataset, DSCMN also achieves about a 10% gain with AUC = 81.2, above DKT-DSC = 78.5, and well above the original DKT, with AUC = 71.3, and DKVMN with AUC = 70.7. On the ASSISTments12 dataset, DSCMN only achieved AUC = 0.71. In the latest ASSISTments14 dataset (which contains more students and less data compared to other three datasets and lacks problem information) DSCNM has AUC slightly lower than DKT-DSC.

Table 3. RMSE result for all tested datasets

Datasets	Model				
	BKT	DKT	DKVMN	DKT-DSC	DSCMN
Cognitive Tutor	0.44 ± 0.00	0.38 ± 0.01	0.38 ± 0.00	0.37 ± 0.03	$\mathbf{0.35 \pm 0.00}$
ASSISTments09	0.47 ± 0.01	0.45 ± 0.00	0.45 ± 0.01	0.43 ± 0.00	$\mathbf{0.40 \pm 0.00}$
ASSISTments12	0.51 ± 0.00	0.43 ± 0.00	0.43 ± 0.00	0.43 ± 0.00	$\mathbf{0.40 \pm 0.00}$
ASSISTments14	0.51 ± 0.00	0.42 ± 0.00	0.42 ± 0.00	0.42 ± 0.00	0.42 ± 0.00

In Table 3, when we compare the models in term of RMSE, BKT is lowest at 0.46 for ASSISTments09, 0.51 for ASSISTments12 and ASSISTments14, and 0.44 for Cognitive Tutor. RMSE results in all dataset is lowest for DSCMN, with 0.40, while all other models are no over 0.43 (except DKT in the Cognitive Tutor dataset and DSCMN in ASSISTments14). According to these results, DSCMN shows better performance than DKT-DSC and significantly better than other models in Cognitive Tutor, ASSISTments09, ASSISTments12 but a little lower than DKT-DSC in ASSISTments14.

6 Conclusion and Future Work

In this paper, we propose a new model, DSCMN, which can predict the student performance by gathering information from skills, problems and student: mastery level of skills of student on various problems at each time step, along with student learning ability at each time interval.

Experiments with four datasets show that the proposed model performs better in predictive performance than state-of-the-art KT models. Dynamic evaluation of student's temporal learning ability at each time interval plays a critical role and helps DSCMN capture more variance in the data, leading to more accurate predictions.

In our future work, we plan to adapt this model to problems associated with multiple skills and apply it in the recommendation of related problems.

References

1. d Baker, R.S.J., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_44
2. Ball, G., Hall Dj, I.: A novel method of data analysis and pattern classification. Isodata, a novel method of data analysis and pattern classification. Technical report 5ri, project 5533 (1965)
3. Brown, J.S., Burton, R.R.: Diagnostic models for procedural bugs in basic mathematical skills. Cogn. Sci. **2**(2), 155–192 (1978)
4. Corbett, A.: Cognitive computer tutors: solving the two-sigma problem. User Model. **2001**, 137–147 (2001)
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-Adapt. Interact. **4**(4), 253–278 (1994)
6. Desmarais, M.C., Baker, R.S.: A review of recent advances in learner and skill modeling in intelligent learning environments. User Model. User-Adapt. Interact. **22**(1–2), 9–38 (2012)
7. Feng, M., Heffernan, N., Koedinger, K.: Addressing the assessment challenge with an online system that tutors as it assesses. User Model. User-Adapt. Interact. **19**(3), 243–266 (2009)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

9. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, vol. 1, pp. 281–297 (1967)
10. Minn, S., Yu, Y., Desmarais, M.C., Zhu, F., Vie, J.J.: Deep knowledge tracing and dynamic student classification for knowledge tracing. In: IEEE International Conference on Data Mining (2018)
11. Minn, S., Zhu, F., Desmarais, M.C.: Improving knowledge tracing model by integrating problem difficulty. In: IEEE International Conference on Data Mining, Ph.D. Forum (2018)
12. Pardos, Z.A., Heffernan, N.T.: 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
13. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, pp. 505–513 (2015)
14. Polson, M.C., Richardson, J.J.: Foundations of Intelligent Tutoring Systems. Psychology Press, London (2013)
15. Razzaq, L., et al.: The assistment project: blending assessment and assisting. In: Proceedings of the 12th Annual Conference on Artificial Intelligence in Education, pp. 555–562 (2005)
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
17. Wan, H., Beck, J.B.: Considering the influence of prerequisite performance on wheel spinning. In: International Educational Data Mining Society (2015)
18. Xiong, X., Zhao, S., Van Inwegen, E., Beck, J.: Going deeper with deep knowledge tracing. In: EDM, pp. 545–550 (2016)
19. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774. International World Wide Web Conferences Steering Committee (2017)