A Novel Approach for Knowledge State Representation and Prediction

Shreyansh Bhatt
Amazon.com
Seattle, USA
bhattshr@amazon.com

Jinjin Zhao
Amazon.com
Seattle, USA
jinjzhao@amazon.com

Candace Thille
Amazon.com
Seattle, USA
cthille@amazon.com

Dawn Zimmaro
Amazon.com
Austin, USA
dzimmaro@amazon.com

Neelesh Gattani
Amazon.com
Seattle, USA
neeleshg@amazon.com

ABSTRACT

Online learning systems with open navigation allow learners to select the next learning activity in order to achieve desired mastery. To help learners make an informed choice regarding the next learning activity, we propose to represent and communicate the learner's knowledge state as the average success rate in the course for each skill, rather than as the probability of correctly answering the next question. We first show that we can accurately estimate the proposed knowledge state. We then show that the proposed attention-based model to estimate the knowledge state requires fewer parameters, provides actionable information to the learners, and achieves equivalent or better accuracy compared to RNN (Recurrent Neural Network) based models.

Author Keywords

Knowledge tracing; Knowledge state; Attention based knowledge tracing.

CCS Concepts

•Applied computing → Learning management systems; Interactive learning environments; •Human-centered computing → *User models; User studies;* **•Computer systems organization → Neural networks;**

INTRODUCTION

Online learning systems supporting a "tell then test" provide a set of sequence of course content followed by an end of the course test. These systems let the learners decide at the end of the learning experience whether they have mastered a particular skill or whether they want to revisit the content from the beginning of the sequence. On the other hand, intelligent tutoring systems estimate the probability of answering the next question correctly and provide the next learning activity based on that estimate [5]. Open navigation based online learning systems provide learners with information about their current knowledge state and let the learners choose the next learning activity. Such a system provides skill directed practice where learners continually monitor their state for each skill after he/she they engage with a content block and attempt a question embedded in the content block. For such systems, it is essential to provide the learner with timely, frequent, and interpretable feedback for their current state in the course, combined with actionable information that can help them choose the next learning activity.

Intelligent tutoring systems use knowledge tracing to estimate a learner's knowledge state and provide the next learning activity based on the estimated knowledge state. In Deep Knowledge Tracing (DKT), piech et al showed that LSTM can predict the correctness of the next attempt. DKT can be significantly more accurate in predicting the correctness of the next attempt compared to BKT [1]. Several researchers have explored an extension to DKT to accurately predict the response to the next interaction. In DKT with side information [8], the authors proposed an extension of DKT and showed that it can improve the prediction accuracy. In Exercise Enhanced Knowledge Tracing [6], authors demonstrated that incorporating question text embedding can further improve the prediction accuracy. In DKT plus [10], authors showed that including the interaction reconstruction loss while predicting next interaction can improve the prediction accuracy. In another research, study authors explored training a separate deep knowledge tracing model for each skill and reported modest improvement for prediction accuracy a couple of datasets [2].

We propose to represent learner's knowledge state as the future average success rate for each skill than representing it as a probability to answer the next skill/exercise correctly. Predictive applications such as mastery prediction and next activity recommendation for a skill primarily leverage learner's ability to answer the next question. As a collection of questions can better represent a skill than one particular question, these ap-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
L@S '20, August 12–14, 2020, Virtual Event, USA.
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7951-9/20/08 ...$15.00.
http://dx.doi.org/10.1145/3386527.3406745

T	Q1	Q1	Q1	Q2	Q2	Q2	Q3	Q3	Q4
C	0	0	1	0	0	1	0	1	1
KS	0.5	0.57	0.5	0.6	0.75	0.67	1.0	1.0	-

Figure 1: Knowledge state representation for one skill. T: attempt to a question & C: success of the attempt. KS is the knowledge state updated after the attempt, E.g., KS after the first attempt is the ratio of successful attempts (4) to total attempts (8) in future.

plications can also benefit from knowledge state represented with multiple future attempts. In an open navigation based online learning system a knowledge state that represents multiple future attempts can help learners make an informed choice in selecting the learning activity. For example, consider a learner attempting questions associated with a skill s_1. We compute the average success rate (in future) after each attempt as knowledge state and communicates to the learner. Figure 1 shows how the average success rate varies after each attempt. A model is trained to predict the average success rate after every attempt based on the available learners' interaction data. Moreover, we train the model to simultaneously predict the average success rate for all the skills rather than the only the skill associated with the next interaction.

In order to provide a ranked list of activities to revisit to achieve desired mastery, we propose an attention-based model that predicts the knowledge state (average success rate for each skill). The attention-based model weighs each attempt of a learner to best predict the average success rate. The resulting attention weights provide the knowledge state explanation to a learner in the form of the attempts that were weighted to compute the knowledge state. If the average success rate is lower than expected, the learner can choose to focus on the unsuccessful attempts that were weighted more by the model in computing the knowledge state.

To evaluate, we first compared the accuracy of estimating the knowledge state with the accuracy of predicting the response of the next attempt. We used five datasets to evaluate the accuracy of knowledge state prediction and used DKT (RNN based knowledge tracing) as a baseline to predict the outcome of the next attempt. Although the proposed knowledge state is estimated based on multiple future attempts and for all skills as compared to estimating it based on the next attempt for the next skill, the proposed knowledge state can be estimated more accurately than the knowledge state based on the next attempt. We found that the proposed attention-based model achieves better AUC and Accuracy to predict the proposed knowledge state than the RNN based model. We also demonstrate the actionable explanation achieved by the proposed attention-based model.

Main contributions of the work are as follows.

- A novel approach for knowledge state representation.
- A simple one layer attention model to accurately predict the proposed knowledge state.
- A framework that provides actionable information to help achieve desired mastery to a learner.

APPROACH

In this section, we describe the proposed knowledge state representation and attention-based model to predict the knowledge state.

Knowledge State Representation

For a learner, we want to estimate the average success rate in each skill based on whether "similar" learners (available in the training data) in the same state achieved the success. Here, the model determines "similar" learners based on the learner's interaction with the online learning system. The average success rate after an attempt t for skill k can be represented as p_t^k (equation 1). p_t^k is computed by taking the ratio of the number of successful future attempts to the skill ($success_i^k$, i from t+1 to m) compared to the total number of future attempts to the skill($total_i^k$, i from t+1 to n).

$$p_t^k = \sum_{i=t+1}^{n} success_i^k / \sum_{i=t+1}^{n} total_i^k \tag{1}$$

We compute a knowledge state (as an n dimensional vector) at an attempt t as $[KS_1, KS_2, ..., KS_n]$ where n is the number of skills and each KS_k indicates p_t^k for k^{th} skill. Next, we describe the model to compute the knowledge state vector.

Attention-based Knowledge State Estimation

We propose to use an attention-based model to predict the proposed knowledge state. The attention-based model considers all the past attempts of a learner instead of considering the latent representation computed based on the previous attempt by RNN. This helps predict the average success rate over multiple future attempts as well as predicting average success rate for all the skills. Moreover, the resulting attention weights from the model, provide the most important attempts in predicting the current knowledge state. Hence, it can help identify most important activities that a learner can revisit to achieve desired knowledge state (mastery).

The model consists of the attention layer followed by a fully connected layer. For any learner L and his/her attempt set $I = \{i_1, i_2, \ldots, i_m\}$, each attempt is represented as a one-hot encoded vector indicating the success of an attempt. Specifically, if there are n skills in a course then the input vector i_t is $2n$ dimensional vector where each skill is represented by two dimensions (corresponding to correct or incorrect attempt to that skill).

We use the scaled dot-product self-attention $Att(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{n}}\right)$[7]. Here, Q, K, and V are query, key, and value matrices and n is the total number of dimensions. Applying attention on the input sequence I is to compute a weighted sum for each i_t based on $i_1, i_2, \ldots, i_t$, where t is the current timestamp. Instead of applying a fixed weight vector, in attention, the weight vector is also learned based on the relation between the query and key.

The computed weighted representation is then applied to a shared fully connected linear layer to compute the knowledge state of a learner at time t. $KS_t = sigmoid(i_tW + ba)$. Here, KS_t is the knowledge state at time t. W is a $2n$ by n weight

Dataset	Next_Pred		KS_Pred	
	AUC	ACC	AUC	ACC
A_2009_corrected	0.75	0.7	0.81	0.8
A_2015	0.7	0.71	0.78	0.8
A_challenge	0.69	0.67	0.78	0.76
Course1	0.74	0.78	0.76	0.8
Course2	0.68	0.7	0.82	0.81

Table 1: The proposed knowledge state is predicted for all the skills. The proposed knowledge state can be predicted more accurately than predicting the next attempt.

matrix and *ba* is bias. Attention weights applied to compute the resulting vectors are different for different sequences [7]. The fully connected layer converts the attention layer output ($2n$) to n dimensional vector (representing n skills). The final knowledge state is computed by taking the sigmoid of the n dimensional output vector.

The whole network is then trained to predict the n dimensional knowledge state. The loss function is as follows,

$$L = \sum_{b} \sum_{k=1}^{n} \sum_{t=1}^{T} MSE(p_t^k, a_t^k) \quad (2)$$

Here, b is the batch size which indicates the number of learners. For each learner, the loss is computed based on the actual knowledge state computed for each skill (a_t^k) using equation 1 and the predicted knowledge state (p_t^k). As the (prediction term) knowledge state is continuous, we use Mean Squared Error (MSE) to predict the knowledge state.

EXPERIMENTS AND RESULTS

In this section, we describe datasets, experiment setup, and results.

We used three publicly available datasets A_2009_corrected [9] (4217 students, 124 skills), A_2015 (19971 students, 100 skills), A_Challenge [3] (1709 students, 102 skills). We used two datasets from the open navigation system course 1 (509 students, 28 skills) and course 2 (315 students, 21 skills).

We used Accuracy (ACC) and Area Under Curve (AUC) measures for evaluating the proposed knowledge state prediction. We considered a 60-20-20 train, validation, and test split. Hyper-parameters were determined using validation data and evaluation was performed on the test data. We refer the knowledge state representing the average success rate for each skill as the knowledge state (KS_pred) and the knowledge state representing the success of the next attempt as next attempt correctness prediction (Next_Pred).

AUC and ACC for the next attempt correctness prediction are calculated by comparing the learner's actual response at the attempt $t+1$ with the model's predicted response after t. For the proposed knowledge state, we compare p_t^k and with average success rate computed based on actual learner's responses 1. To compute accuracy, we threshold the actual average success rate with 0.5. Note that the knowledge state consists of prediction for all the skills instead of the skill associated with the next response.

We use the DKT [4] to predict the next attempt's correctness prediction. It uses LSTM to predict the response for the next attempt. Table 1 shows the results of the comparison for predicting the proposed knowledge state (KS_Pred) with the next attempt correctness prediction (Next_Pred).

For all the datasets, we found that the proposed knowledge state can be predicted more accurately than predicting correctness of the next attempt. Hence, we can accurately estimate the proposed knowledge state of a learner. Note that the proposed knowledge state estimation includes predicting the average success rate for all the skills. In fact, this has a positive effect on prediction accuracy as simultaneously predicting all skills and all the future attempts (as indicated in the loss function equation 2) serves as regularization and helps with the prediction waviness as identified by DKT+ [10]. Table 2 shows the proposed knowledge state prediction and the next attempt's correctness prediction for three randomly selected learners. The x-axis indicates the attempts while the y-axis indicates the actual and predicted value. As the next question can be wrong or correct (0 or 1), the actual knowledge state takes 0 or 1 value. The proposed knowledge state is a continuous measure (across the number of attempts) and hence it has a value between 0-1. The orange lines indicate actual knowledge state value while the blue line indicates the predicted knowledge state. Table 2 also indicates that less variation in knowledge state helps the model to better predict the knowledge state – blue line (predicted knowledge state) follows the orange line (actual knowledge state) for the proposed knowledge state prediction.

To evaluate the attention-based model for the proposed knowledge state prediction, we used a modified version of DKT (LSTM_ALL) that uses LSTM to predict the proposed knowledge state based on the loss function equation 2. Table 3 shows the number of parameters associated with the proposed attention-based model and LSTM based model (DKT). LSTM based model has more parameters than the proposed model (AV_ATT) as the number of parameters for the proposed model depends on the input dimensions while the number of parameters for DKT depends on input dimensions and hidden units.

Table 4 shows the comparison results for the LSTM based model and the proposed attention-based model (Av_Att) for the knowledge state estimation. The proposed model achieves equivalent or better AUC and ACC for all the datasets.

We compute the weight vector from the attention-based knowledge. The weighted vector indicates the relative importance of all the past attempts in achieving the current knowledge state. Each attempt is associated with a particular question or learning activity. Based on the mapping and relative importance of attempts, we create a ranked learning activity revision list for the learner. The learner can then choose to revisit the activities to if the average success rate does fall below the learner's expectation. Hence, after each attempt, we provide the learner with the knowledge state indicating the average success rate for each skill (output of the attention model) and a revised activity list.

CONCLUSIONS

In this research, we proposed a novel knowledge state representation to help learners make a better decision in choosing the

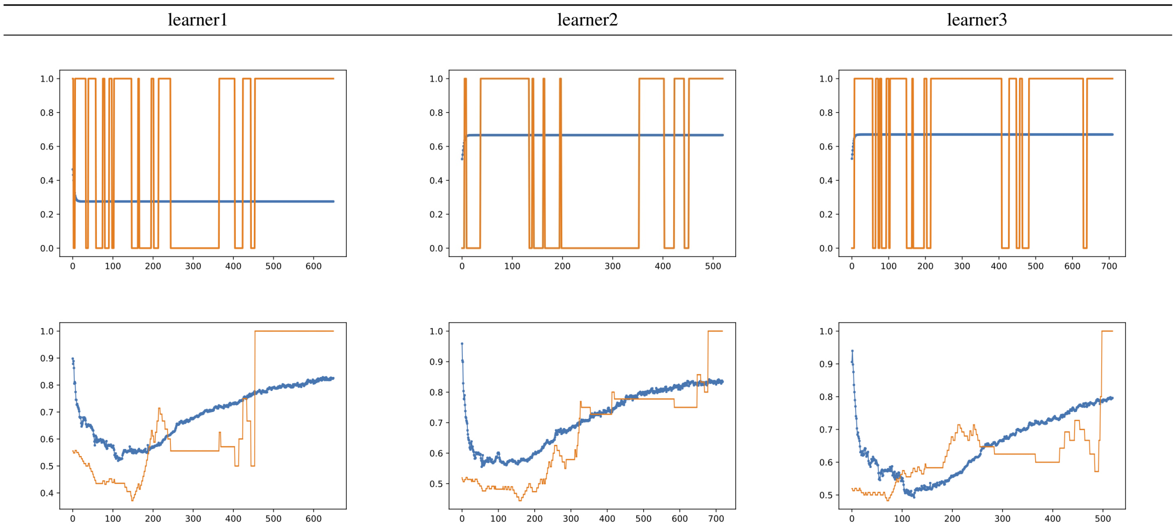

Table 2: Knowledge state prediction (bottom) and next response prediction (top) for three learners. x axis: number of attempts, y axis: knowledge state. Yellow indicates the actual knowledge state and blue indicates the predicted knowledge state. Prediction for the proposed knowledge state follows the actual knowledge state.

Dataset	LSTM+Next	AV_Att
A_2009_corrected	460,000	405,000
A_2015	340,000	180,000
A_challenge	342,002	183,618
Course 1	206,368	12,544
Course 2	104,482	7938

Table 3: Number of parameters required. The proposed model requires fewer parameters hence less prone to overfit.

Dataset	LSTM		AV_Att	
	AUC	ACC	AUC	ACC
A_2009_corrected	0.78	0.78	0.81	0.8
A_2015	0.78	0.8	0.78	0.81
A_challenge	0.76	0.76	0.78	0.76
Course1	0.74	0.79	0.76	0.8
Course2	0.8	0.81	0.81	0.81

Table 4: LSTM and attention based model comparison for predicting the proposed knowledge state. Attention based model achieves equal or better accuracy for all the datasets.

next learning activity. We also proposed the attention-based model to predict this new knowledge state. We showed that the proposed knowledge state provides more information to the learner regarding that current state compared to providing the probability to predict the next attempt. Even though it provides more information, it can be predicted more accurately than predicting the response to the next attempt. The proposed attention-based model provides actionable information to learners and yet does not compromise on the prediction accuracy. Future research shall include exploring the role of the proposed knowledge state in downstream applications such as mastery prediction and next activity recommendation as these applications currently utilize the next attempt prediction based knowledge state. More research is required in representing multiple future attempts than averaging as averaging does not capture the distribution of attempts. Application specific aggregation or future attempt representation warrants a different underlying learning model.

REFERENCES

[1] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[2] Shirly Montero, Akshit Arora, Sean Kelly, Brent Milne, and Michael Mozer. 2018. Does Deep Knowledge Tracing Model Interactions among Skills?. *International Educational Data Mining Society* (2018).

[3] Thanaporn Patikorn, Neil T Heffernan, and Ryan S Baker. 2018. ASSISTments Longitudinal Data Mining Competition 2017: A Preface. In *Proceedings of the Workshop on Scientific Findings from the ASSISTments Longitudinal Data Competition, International Conference on Educational Data Mining*.

[4] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[5] Martha C Polson and J Jeffrey Richardson. 2013. *Foundations of intelligent tutoring systems*. Psychology Press.

[6] Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris Ding, Si Wei, and Guoping Hu. 2018. Exercise-enhanced sequential modeling for student performance prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[8] Zhiwei Wang, Xiaoqin Feng, Jiliang Tang, Gale Yan Huang, and Zitao Liu. 2019. Deep Knowledge Tracing with Side Information. In *International Conference on Artificial Intelligence in Education*. Springer, 303–308.

[9] Xiaolu Xiong, Siyuan Zhao, Eric G Van Inwegen, and Joseph E Beck. 2016. Going deeper with deep knowledge tracing. *International Educational Data Mining Society* (2016).

[10] Chun-Kit Yeung and Dit-Yan Yeung. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM, 5.