Accuracy-aware Deep Knowledge Tracing with Knowledge State Vector Loss

Qiushi PAN & Taro TEZUKA
University of Tsukuba, Japan
qiugits@klis.tsukuba.ac.jp, tezuka@slis.tsukuba.ac.jp

Abstract: In major e-learning platforms such as intelligent tutoring systems (ITSs) and massive open online courses (MOOCs), the students are often recommended what course materials to take based on their past interactions. Knowledge Tracing (KT) is the task of modeling students' academic abilities. Given a sequence of student's learning history, it predicts how well they will perform in the next interaction. Deep Knowledge Tracing (DKT) uses a recurrent neural network (RNN) to capture the underlying structure of the student's understanding. In this paper, we point out the *accuracy rate problem* that the model won't reproduce the accuracy ratio. This is a limitation of the existing loss function in DKT that it only learns the probability of correctly answering a problem in the next interaction. We introduced the *Knowledge State Vector loss*, which captures the accuracy rate of all knowledge concepts, to measure and train the model.

Keywords: Deep Knowledge Tracing, Machine Learning, Educational Data Mining

1. Background

On e-learning platforms such as intelligent tutoring systems and massive open online courses, students and teachers can benefit from a more personalized educational system. Knowledge Tracing is a task to model the dynamics of a student’s acquisition of knowledge concepts (KC). KC is a general term that represents a unit of knowledge, and can also be expressed as skill, ability, etc. Given a sequence of a student's learning history, it predicts how they will perform in the next interaction. Bayesian Knowledge Tracing (BKT) (Corbett et al., 1994) tackles this task with an interpretable model that requires domain specialists to tune its parameters. Deep Knowledge Tracing (DKT) (Piech et al., 2015) uses a recurrent neural network (RNN) (Zaremba et al., 2014). It achieves higher accuracy without a domain specialist, but its parameters remain difficult to be explained. To optimize for the next interaction result, the loss function of a DKT model is $L = \sum_{t=1}^{T} \kappa\left(y_t^T \delta_m(q_{t+1}), a_{t+1}\right)$ where t is a time step in the learning history, y_t is an m-dimensional vector representing the prediction for each KC’s probability being correctly answered. m is the total number of KCs. q_{t+1} is a scalar index representing the label of the KC studied at time $t+1$, and δ_m converts the label to an m-dimensional one-hot vector. a_{t+1} is the ground truth answer result that indicates whether the student answered correctly (1) or not (0). The loss κ is binary cross-entropy. Input x_t is given as a one-hot representation such that $x_t = \delta_{2m}(q_{t+1} + a_t m)$. Based on the method of compressed sensing, we embedded $x_t \in \{0, 1\}^{2m}$ into a ceil($\log 2m$) length vector and passed to the RNN cells that recursively outputs y_t. Among a number of problems reported in DKT, the reconstruction problem is where the model does not reproduce the last study log in the predictions, and the wavy transition problem is that the predicted performance fluctuates rapidly over time (Yeung et al., 2018).

2. Method

We point out the accuracy rate problem, the inability to trace the accuracy performance of students properly. Although the model with the existing loss function L models the probability of answering the immediately following KC q_{t+1}^{α} correctly, it does not model the correct answer rate. This comes from the limitation of the existing loss function L that in order to make an estimation, it requires knowing what question the student will choose to solve in the next step. This means, in turn, the model can not predict well about a KC q^{β} if it does not appear at $t+1$. Also, the probability of answering correctly at $t+1$ is not ideal as the assessment of academic ability since it can easily fluctuate over time. We consider the accuracy rate is more suitable as a representation of students' academic skills.

To solve this problem, we propose a novel loss function that optimizes the accuracy rate as a regularization problem. It is reasonable that in the first place, the loss function is invented to utilize the next answered KC result because it is the only information we can directly get from the dataset. In this paper, by calculating the average accuracy rate for all KCs, we can access target data of KCs not limited to what is actually answered in the next time step.

Our new loss function is based on the assumption that at each time point, a student has a probability distribution on whether he or she can solve each problem correctly. We call this a knowledge state vector (KSV). The KSV loss is considered more suitable as an optimization target because it considers all information in the given input when optimizing to the next time step.

Let T be a sequence length. $\delta_m(q_{t+1})$ represents an m-dimensional one-hot representation of the skill involved in solving a problem at timestamp t. The sum of $\delta_m(q_t)$ over time is a vector of the frequency of the skills that appear in the sequence. Since $a_s\delta_m(q_t)$ is the product of a scalar and vector, their sum is the frequency of skills $q_1, ..., q_t$ of correctly solved problems. Let m be the number of KCs, 2 be the set $\{0, 1\}$, and be the set of integers. The simplest KSV loss is

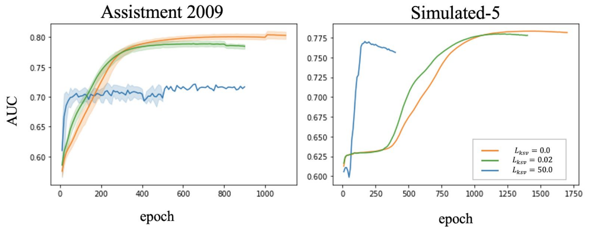
$$L_{ksv,1} = \sum_{t=1}^{T} \kappa\left(y_t \circ \sum_{s=2}^{t+1} \delta_m(q_s),\ \sum_{s=2}^{t+1} a_s\delta_m(q_s)\right) \tag{1}$$

where $\delta_m(q_s) \in 2^m$, $\sum_{s=2}^{t+1} \delta_m(q_s) \in Z^m$, $a_s\delta_m(q_s) \in 2^m$, $\sum_{s=2}^{t+1} a_s\delta_m(q_s) \in Z^m$. $\circ$ is the Hadamard product. The original loss function takes skill frequency and accuracy rate as arguments. The loss function κ is a mean squared error. By calculating the loss, we can optimize the model such that its prediction will be closer to the accuracy rate. To model the process of forgetting, we introduce an alternative model that has a parameter β that puts exponential weights to questions that were answered recently.

$$L_{ksv,\beta} = \sum_{t=1}^{T} \kappa\left(y_t \circ \sum_{s=2}^{t+1} \beta^s\delta_m(q_s),\ \sum_{s=2}^{t+1} a_s\beta^s\delta_m(q_s)\right) \tag{2}$$

In the experiments, we used $\beta = 1.2$. Equation 1 is an example of Equation 2, where β is set to 1. Finally, the updated loss function is formulated as $L' = L + \lambda_{ksv}L_{ksv,\beta}$ where L is the existing loss. λ_{ksv} is a regularization parameter for KSV loss L_{ksv}.

3. Results



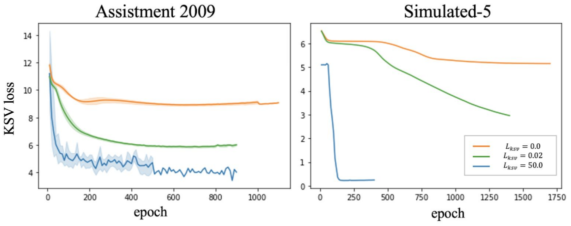


Figure 1. Learning curves of the proposed method for different datasets. Left is about the AUC, and right is about the KSV loss. The horizontal axis represents the iteration epoch number.

Table 1. *Comparison of DKT results.*

Assistments 2009

λ_{ksv}	AUC	KSV loss	w_1	w_2	AUC(C)
0.00	0.8010	9.1607	0.0887	0.0275	0.8656
0.02	0.7893	5.9381	0.0616	0.0153	0.8597
50.00	0.7132	**4.8588**	**0.0389**	**0.0090**	0.8531

Simulated-5

λ_{ksv}	AUC	KSV loss	w_1	w_2	AUC(C)
0.00	0.7832	5.1550	0.0365	0.0032	0.5898
0.02	0.7796	2.9585	**0.0328**	**0.0027**	0.5843
50.00	0.7701	**0.2384**	0.0933	0.0536	0.5827

We used the ASSISTments "Skill Builder" Dataset 2009–2010 (Feng et al., 2009). Each problem in this dataset is manually tagged with a single required skill among 124 knowledge concepts. We also used Simulated-5 (Piech et al., 2015), a simulation of 4,000 virtual students solving 50 problems. For comparison, we reconstructed the DKT as well as the waviness indicators (w1 and w2) and current time step AUC (C) (Yeung et al., 2018). The baseline DKT is set up on the basis of previous research. We used a hidden dimension size of 200 and a batch size of 128. The learning rate is set to 0.05. We experimented for $\lambda_{ksv} \in \{0, 0.02, 50\}$. Since the AUC is the result of categorical optimization, our model's decrease in AUC in Table 1 is expected. Figure 1 shows the learning curve of the KSV loss. Training with the existing classification loss L decreased KSV loss for some degree. However, when trained with the KSV loss, it reduces the loss dramatically. Figure 1 also shows the learning curve occasionally fluctuates. The weight added might be the cause: when the latest time step is weighted heavily, the target value changes more, making the loss fluctuate. Without adding them to the loss function and only training the model with our KSV loss, w1 and w2 dropped significantly on real datasets. For the Simulated-5 dataset, they increased possibly by overfitting to the accuracy rate. Since the Simulated-5 is a simulation dataset, it is different from real datasets. When the prediction overfits the accuracy rate, its changes between time steps become larger, causing waviness indicators to increase. We also examined AUC for the current time step in the column AUC (C). We noticed training the models with KSV loss has even less negative impacts on AUC (C).

In this paper, we addressed a new problem in DKT, namely the accuracy rate problem. To quantize and solve this problem, we proposed a new learning loss: KSV loss. The results show that our loss function is useful to measure how well the model fits the accuracy rate. Training with the KSV loss also reduced the waviness loss compared with the baseline. In the real world application, there are various ways of utilizing the model's predictions. For example, a teacher may want to use the prediction results to assess students' academic abilities, as is suggested in many research papers. With a more intuitive output of accuracy rate, our method is more suitable to be shown to students or teachers and opens up wider opportunities for KT models.

References

Corbett, A. T., Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-adapted Interaction, Vol. 4, No. 4, pp. 253–278.

Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. The Journal of User Modeling and User-Adapted Interaction, 19, 243-266.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. arXiv.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep Knowledge Tracing. Advances in Neural Information Processing Systems, pp. 505–513.

Yeung, C.-K., & Yeung, D.-Y. (2018). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. Proc. of the 5th ACM Conf. on Learning at Scale, p. 5.