DeepIRT with a Hypernetwork to Optimize the Degree of Forgetting of Past Data

Emiko Tsutsumi  
University of Electro-Communications  
tsutsumi@ai.lab.uec.ac.jp

Yiming Guo  
University of Electro-Communications  
guo_yzmzng@ai.lab.uec.ac.jp

Maomi Ueno  
University of Electro-Communications  
ueno@ai.lab.uec.ac.jp

ABSTRACT

Knowledge Tracing (KT), the task of tracing students’ knowledge state, has attracted attention in the field of artificial intelligence. Recently, many researchers have proposed KT methods using deep learning to predict student performance on unknown tasks based on learning history data. Especially, the latest DeepIRT reportedly has high predictive accuracy and parameter interpretability. Nevertheless, some room remains for improvement of its prediction accuracy because it does not optimize the degree of forgetting of past data. Specifically, although its forgetting parameters are optimized solely using current input data, it should use both current input and past data to optimize them. Therefore, for better parameter estimation to improve accuracy, this study proposes a new DeepIRT that optimizes the degree of forgetting of past data. The proposed method has a hypernetwork to balance both the current and the past data in memory, which stores a student’s knowledge states. Results of experiments demonstrate that the proposed method improves the prediction accuracy compared to earlier KT methods.

Keywords

Deep Learning, Hypernetwork, Item Response Theory, Knowledge Tracing

1. INTRODUCTION

Recently, with the development of online education [24, 25, 26], Knowledge Tracing (KT) has attracted broad attention for helping students to learn effectively by presenting optimal problems and a teacher’s support. [3, 7, 10, 15, 16, 17, 28, 29, 33, 34, 35]. Important tasks of KT are tracing the student’s evolving knowledge state and discovering concepts that the student has not mastered based on the student’s past learning history data. Furthermore, accurate prediction of a student’s performance (correct or incorrect response to an unknown item) is important for adaptive learning. Although KT methods have been proposed as probabilistic approaches [3, 7, 28, 29, 34] and deep-learning-based approaches [17, 28, 29, 33, 35], the latter have been studied more actively in recent years because they reportedly have high prediction accuracies.

Various deep-learning-based approaches have been proposed to improve the prediction accuracy of a student’s performance[1, 20, 21, 32]. Most recently, Ghosh et al. (2020) proposed attentive knowledge tracing (AKT) [5], which incorporates a forgetting function of past data to attention mechanisms: the Transformer method [27]. In addition, AKT optimizes the parameters to weight the data necessary for student performance prediction from past learning data. Therefore, AKT has the best performance for predicting a student’s responses among earlier KT methods. However, the interpretability of the parameters is limited because it cannot express a student’s ability transition of each skill [5, 14, 22].

On the other hand, to express a student’s knowledge state transition for deep-learning-based approaches, Zhang proposed the dynamic key-value memory network (DKVMN) [35]. DKVMN traces the knowledge state transition using a Memory-Augmented Neural Network and attention mechanisms. It can estimate the relations between underlying skills and items addressed by students. In addition, DKVMN has a memory updating component to allow forgetting and updating of the latent variable memory, which stores the students’ knowledge states in the learning process [35]. For interpretability of the parameters, the memory updating component in DKVMN is more effective than the forgetting function of AKT because it updates the current latent variable memory, which stores the students’ skills and abilities, using only the immediately preceding values.

To improve the interpretability of the parameters of DKVMN, DeepIRT was proposed by combining DKVMN with an Item Response Theory (IRT) [2, 11, 30] module [33]. It includes the students’ ability parameters and the items’ difficulty parameters. However, it was insufficient to improve the interpretability because a student’s ability of DeepIRT depends on each item characteristic. To resolve this shortcoming, Tsutsumi et al. proposed DeepIRT methods with independent redundant student and item networks [22, 23]. They can learn the student’s ability and item difficulty independently to avoid impairing the predictive accuracy. For DeepIRT [23], a student’s ability is constant throughout a learning process because it is structured for test theory.
Therefore, it can not be applied to KT. To apply DeepIRT to KT, DeepIRT [22] was proposed using architecture of DKVMN. In DeepIRT [22], a student network employs memory network architecture to reflect dynamic changes of student abilities as DKVMN does. Because the student’s ability parameters of the DeepIRT [22] are independent of each item characteristic, it has higher interpretability than the earlier method has [33]. Furthermore, the DeepIRT [22] can express a student’s ability transition for each skill by estimating relations among the multidimensional skills. Consequently, the DeepIRT provides high interpretability without impairing the predictive accuracy.

However, room for improvement of prediction accuracy of the DeepIRT remains [22] because it does not optimize the degree of forgetting the past data. Specifically, in DKVMN and DeepIRT methods, the forgetting parameters which control the degree of forgetting the past data are optimized from only the current input data: the student’s latest response to an item. As a result, it might degrade the prediction accuracy of the DeepIRT because the value memory insufficiently reflects the past learning history data. Namely, it might be difficult to reflect the past data accurately in a long learning process. It should use not only the current input data but also past data to optimize the forgetting parameters.

In this study, we propose the new DeepIRT with a hypernetwork to optimize the forgetting parameters. The hypernetwork [4, 6, 8, 9, 12, 13, 19, 31] balances both current and the past data in the latent variable memory, which stores a student’s knowledge state data. Before the model updates the latent variable memory, it optimizes not only the weights of the forgetting parameters but also the past latent variable memory. Experiments were conducted to compare the performances of the proposed method and those of the earlier KT methods. The results demonstrate that the proposed method improves the prediction accuracy of the DeepIRT [22]. They also indicate the proposed method as effective, especially for tasks with a long-term learning process.

2. DKVMN AND DEEP-IRT METHODS

DKVMN and DeepIRT methods [22, 33, 35] have the same memory updating component to update and forget the students’ knowledge states in the learning process [35]. The value memory \( M^t \), which traces the process of student ability growth, is updated in this memory updating component. They use \( c_j \) based on input \( q_j \), which reflects a latest student’s response data \( u_{tj} \) to item \( j \) at time \( t \).

\[
c_j = \begin{cases} [0, q_j] & u_{tj} = 1 \\ [q_j, 0] & u_{tj} = 0. \end{cases} \quad (1)
\]

Here, \( 0 \) is a zero vector consisting of \( J \) zero values. They updated the value memory \( M^t \) as

\[
v_t = W^v c_j + \tau^v, \quad (2)
\]

\[
e_t = \sigma(W^e v_t + \tau^e), \quad (3)
\]

\[
a_t = \tanh(W^a v_t + \tau^a), \quad (4)
\]

and

\[
\tilde{M}^t_{t+1} = M^t_{tJ} \otimes (1 - w_{tt} e_t) + w_{tt} a_t^\top, \quad (6)
\]

where \( W^v, W^e \) and \( W^a \) are the weight matrices, and \( \tau^v, \tau^e \) and \( \tau^a \) are the bias vectors. Furthermore, \( w_{tt} \) signifies the degree of strength of the relations between the underlying skill \( I \) and skill tags addressed by a student at time \( t \). It is noteworthy that \( e_t \) and \( a_t \) are forgetting parameters, which adjust the degrees of forgetting the past data and reflecting the current input data. \( e_t \) influences how much the value memory forgets (remembers) the past ability. Additionally, \( a_t \) controls how much the value memory reflects the current input data.

For the interpretability of the parameters, this memory updating component is more effective than the forgetting function of AKT because it updates the current latent variable memory which stores the student’s skills and abilities using only the immediately preceding values. However, the forgetting parameters are optimized only from current input data. It should use not only the current input data but also past data to optimize them. Additionally, the weights are fixed values and are not optimized for each time point. As a result, DKVMN and DeepIRT might degrade the prediction accuracies because of value memory \( M^t_{tJ} \) which only insufficiently reflects past learning history data. Especially, it might be difficult to reflect past data accurately in a long learning process.

3. PROPOSED METHOD

The preceding section described that the forgetting parameters of DeepIRT are not optimized using both current input data and past data. However, when using both current input data and past data, it is difficult to optimize the weight parameters directly because the number of parameters increases dynamically.

Recent studies in the field of Natural Language Processing (NLP) proposed the extension components to LSTM [18] in the form of mutual gating of the current input data and the previous output hidden variables [6]. These extension components are called hypernetworks. A hypernetwork supports the main recurrent neural network by optimizing the non-shared weights for each time point in the hidden layers [6]. In standard LSTM [18], the hidden variables change with time, but the weights used to update them are fixed values and are not optimized for each time point. To resolve this difficulty, various hypernetworks have been proposed to optimize the non-shared weights in the LSTM at each time point. [4, 6, 8, 9, 12, 13, 31]. Their results demonstrate that LSTM with a hypernetwork works better than the standard LSTM [18].

Melis et al. earlier proposed the "Mogrifier component" which is a kind of hypernetwork for LSTM in the field of NLP [12]. Mogrifier also scales the weights and the hidden variables using not only the current inputs but also the output of the hidden variable at the previous point in time. They reported that the LSTM with Mogrifier component outperforms the other methods for a long input data length. Inspired by those studies, this study proposes a new hypernetwork that optimizes the degree of forgetting of past data in the DeepIRT [22] to improve prediction accuracy with the parameter interpretability. We incorporate the proposed hypernetwork in the memory updating component, which updates the latent variable \( M^t_{tJ} \), to avoid greatly increasing
number of parameters. Before the model updates the latent variable \( M^r_t \), the proposed hypernetwork optimizes not only the weights of the forgetting parameters but also the past latent variable \( M^r_t \). The proposed hypernetwork estimates the optimal forgetting parameters by balancing both the current input data and the past latent variable. In addition, the Mogrifier component [12] used constant values as the tuning parameters in the hypernetwork. For this study, we optimize the tuning parameters to adjust the hypernetwork for each dataset. No report of the relevant literature has described a study of the use of the hypernetworks for KT methods. Figure 1 presents the architecture of the proposed method. The right side of Figure 1 presents the hypernetworks and the memory updating component. The left side of Figure 1 shows the independent student and item networks.

### 3.1 Hypernetwork

To optimize the forgetting parameters at time \( t \), the proposed hypernetwork balances the current input data and the past value memory \( M^r_{t+1} \) to store sufficient information of the learning history data before calculating the latent variables \( M^r_{t+1} \). The proposed hypernetwork structure is located at the beginning of the Memory Updating Component on the right side of Figure 1 (shown in red).

Figure 2 shows the structure of the proposed hypernetwork. The inputs of the hypernetwork are the past value memory \( M^r_t \) and current input data \( (s_j, u_j) = s_j + u_j * S \) when a student responds to item \( j \) of skill \( s_j \). Therein, \( S \in \{1, 2, ..., 2S\} \) represents the number of skills. The embedding vector of \( (s_j, u_j) \) denoted as \( v_t \in \mathbb{R}^{d_v} \). Because of the repeating multiplications as shown in Figure 2, this hypernetwork balances current data \( v_t \) and past value memory \( M^r_t \). For the proposed methods, we optimize the number of rounds \( r \) for each learning dataset.

### 3.2 Memory Updating Component

Next, we estimate the forgetting parameters \( e_t \) and \( a_t \) using the optimized \( v_t \) and \( M^r_t \) in the hypernetwork. These forgetting parameters are important to update the latest value memory \( M^r_{t+1} \) optimally. The earlier memory updating component of DKVMN and DeepIRT methods calculates the forgetting parameters from \( v_t \) with only current input information in equation (3), (4). By contrast, we calculate them using the optimized current input data \( v_t^* \) and the past latent value \( M^r_t \). Therefore, the forgetting parameters \( e_t \) and \( a_t \) are also be estimated as optimizing the degree of forgetting of past data and as reflecting the current input data. Furthermore, the proposed method can capture the student knowledge state changes accurately because the latent knowledge state \( M^r \) has sufficient information of the past learning history data.

### 4. EXPERIMENTATION

#### 4.1 Datasets and Experiment Setting

This section presents comparisons of the prediction accuracies of the proposed method with those of earlier methods (Tsutsumi et al. and AKT) [5, 22]. We use the standard benchmark datasets ASSISTments2009 and ASSISTments2017 collected from an online tutoring system. Table 1 presents the number of students (No. Students), the number of skills (No. Skills), the number of items (No. Items), the rate of correct responses (Rate Correct), and the average number of rounds (No. Rounds) for each dataset.
Table 1: Summary of datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. students</th>
<th>No. skills</th>
<th>No. Items</th>
<th>Rate Correct</th>
<th>Learning length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSISTments2009</td>
<td>4151</td>
<td>111</td>
<td>26684</td>
<td>63.6%</td>
<td>52.1</td>
</tr>
<tr>
<td>ASSISTments2017</td>
<td>1709</td>
<td>102</td>
<td>3162</td>
<td>39.0%</td>
<td>551.0</td>
</tr>
</tbody>
</table>

Table 2: Prediction accuracies of students’ performances

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metrics</th>
<th>Tsutsumi et al.</th>
<th>AKT</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>80.70 +/- 0.56</td>
<td>82.20 +/- 0.25</td>
<td>81.57 +/- 0.39</td>
</tr>
<tr>
<td>ASSISTments2009</td>
<td>Acc</td>
<td>76.13 +/- 0.58</td>
<td>77.30 +/- 0.55</td>
<td>76.85 +/- 0.56</td>
</tr>
<tr>
<td></td>
<td>Loss</td>
<td>0.54 +/- 0.10</td>
<td>0.49 +/- 0.10</td>
<td>0.53 +/- 0.13</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>74.15 +/- 0.27</td>
<td>74.54 +/- 0.21</td>
<td>76.85 +/- 0.39</td>
</tr>
<tr>
<td>ASSISTments2017</td>
<td>Acc</td>
<td>68.73 +/- 0.11</td>
<td>69.83 +/- 0.15</td>
<td>71.08 +/- 0.50</td>
</tr>
<tr>
<td></td>
<td>Loss</td>
<td>0.57 +/- 0.06</td>
<td>0.58 +/- 0.06</td>
<td>0.55 +/- 0.06</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>77.42</td>
<td>78.37</td>
<td>79.21</td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>72.43</td>
<td>73.56</td>
<td>74.00</td>
</tr>
<tr>
<td></td>
<td>Loss</td>
<td>0.56</td>
<td>0.54</td>
<td>0.54</td>
</tr>
</tbody>
</table>

We used five-fold cross-validation to evaluate the prediction accuracies of the methods. The item parameters and hyper-parameters are trained by 70% of each dataset. Given the estimated parameters, the students’ abilities are estimated at each time using the remaining 30% of each dataset according to an earlier study [22]. We employ Adam optimization with a learning rate of 0.003 and batch-size 32. Additionally, 200 items was set as the upper limit of the input length according to the earlier studies [22, 33, 35]. For this study, we leverage three metrics for prediction accuracy: Accuracy (Acc) score, AUC score, and Loss score.

4.2 Prediction Accuracy

The respective values of Acc, AUC, and Loss for ASSISTments2009 and ASSISTments2017 datasets [5, 22] are presented in Table 2. We compared the performances of the proposed method with those of DeepIRT [22] and AKT for each dataset with item and skill tag inputs according to [5]. Additionally, this report describes the standard deviations across five test folds.

Results indicate that the proposed method, which optimizes the forgetting parameters, provides the best average scores for all metrics. Especially, the proposed method outperforms the Tsutsumi et al. [22] and AKT for ASSISTments2017. ASSISTments2017 has a long learning length. By contrast, the proposed method tends to have lower prediction accuracies for ASSISTments2009 with a shorter learning length than AKT has. Results suggest that the proposed hypernetwork functions effectively, especially for datasets with long learning lengths.

5. CONCLUSIONS

Recently, to express a student’s the knowledge state transition for deep-learning-based approaches, DKVMN and DeepIRT methods have been proposed. Tsutsumi et al. (2021) proposed a DeepIRT with independent redundant student and item networks [22]. It can learn the student’s ability and item difficulty independently to avoid impairing the predictive accuracy. Furthermore, the DeepIRT [22] can express a student’s ability transition for each skill by estimating relations among the multidimensional skills. the DeepIRT [22] has a memory updating component to allow forgetting and updating of the latent variable memory, which stores the students’ knowledge states in the learning process. However, the forgetting parameters which control the degree of forgetting the past data are optimized from only the current input data. It might degrade the prediction accuracy of the DeepIRT because the value memory insufficiently reflects the past learning history data. It should use not only the current input data but also past data to optimize the forgetting parameters.

This study proposed a new DeepIRT with a hypernetwork that optimizes the degree of forgetting of the past data for parameter estimation to improve prediction accuracy with the parameter interpretability. In the proposed method, the hypernetwork balances the current input data and the past value memory to store sufficient information of the learning history data before calculating the latent variables. Specifically, it scales not only the weights of the forgetting parameters but also the hidden variables using the current inputs and the output of the hidden variable at the previous point in time.

Experiments conducted with the benchmark datasets demonstrated that the proposed method improves the prediction accuracies of the earlier KT methods. Especially, results showed that the proposed method is effective for tasks with a long-term learning process. As future work, we will evaluate the interpretability of the ability parameters of the proposed method by comparing the parameter estimates with those of the earlier DeepIRTs [22, 33]. Furthermore, we will clarify the mechanism of how the proposed hypernetwork functions to increase the predictive accuracy.

6. ACKNOWLEDGMENTS

This research was supported by JSPS KAKENHI Grant Numbers JP19H05663, JP19K21751 and JP22J15279.
7. REFERENCES


