Deep-IRT with independent student and item networks

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

Ryo Kinoshita
University of
Electro-Communications
kinoshita@ai.lab.uec.ac.jp

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

ABSTRACT

Knowledge tracing (KT), the task of tracking the knowledge state of each student over time, has been assessed actively by artificial intelligence researchers. Recent reports have described that Deep-IRT, which combines Item Response Theory (IRT) with a deep learning model, provides superior performance. It can express the abilities of each student and the difficulty of each item such as IRT. However, its interpretability and applicability remain limited compared to those of IRT because the ability parameter depends on each item. Namely, the ability estimate for the same student and time might differ if the student attempts a different item. To overcome those difficulties, this study proposes a novel Deep-IRT model that models a student response to an item by two independent networks: a student network and an item network. Results of experiments demonstrate that the proposed method improves prediction accuracy and the interpretability of earlier KT methods

Keywords

Deep Learning, Item Response Theory, Knowledge Tracing

1. INTRODUCTION

Recently, along with the advancement of online education, Knowledge Tracing (KT) has attracted broad attention for helping students to learn effectively by presenting optimal problems and a teacher's support [5, 14, 16, 22, 23, 24, 37, 39, 43, 45, 46]. 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 prior learning history data. Furthermore, predicting a student's performance (correct or incorrect responses to an unknown item) accurately is important for adaptive learning. Many researchers have developed various methods to solve KT tasks. Methods for KT are divisible into probabilistic approaches and deep-learning approaches.

For example, Bayesian Knowledge Tracing (BKT), a traditional and well known probabilistic model for KT [1, 5, 8, 14, 16, 22, 23, 26, 45], employs a Hidden Markov Model to trace a process of student ability growth. It predicts the probability of a student responding to an item correctly. Item Response Theory (IRT) [3, 34, 35], which is used in the test theory area [10, 11, 12, 13, 28, 33, 36], has come to be used for KT [6, 40]. Actually, IRT predicts a student's correct answer probability to an item based on the student's latent ability parameter and item characteristic parameters.

Actually, a learning task is associated with multiple skills. Students must master the knowledge of multiple skills to solve a task. However, BKT and IRT have a restriction by which they express only uni-dimensional ability.

To overcome the limitations, Deep Knowledge Tracing (DKT) [24] was proposed as the first deep-learning-based method. DKT employs Long short - term memory (LSTM) [27] to predict a student's performance. LSTM relaxes the restrictions of skill separation and binary state assumptions. However, the hidden states include a summary of the past sequence of learning history data in LSTM. Therefore, DKT does not explicitly treat the student's ability of each skill.

To improve the DKT performance, various deep-learning-based methods have been proposed [2, 4, 17, 19, 29, 30, 31, 38, 42, 44]. Especially, the dynamic key-value memory network (DKVMN) was developed to exploit the relations among underlying skills and to trace the respective knowledge states [46]. To trace student ability, DKVMN uses a Memory-Augmented Neural Network and attention mechanisms. Furthermore, to improve the explanatory capabilities of the parameters, Deep-IRT was proposed by combining DKVMN with an IRT module [43]. In fact, Deep-IRT can estimate a student's ability and an item's difficulty just as standard IRT models can. However, the ability parameter of the Deep-IRT depends on each item characteristic because it implicitly assumes that items with the same skills are equivalent. The assumption does not hold when the item difficulties for the same skills differ greatly. Items for the same skills which are not equivalent hinder interpretation of a student's ability estimate.

Most recently, Gosh et al. (2020) proposed attentive knowledge tracing (AKT) [7], which incorporates a forgetting function of past data to attention mechanisms. Additionally, they indicated a problem by which earlier KT methods assumed that items with the same skills are equivalent. To re-

Emiko Tsutsumi, Ryo Kinoshita and Maomi Ueno "Deep-IRT with independent student and item networks". 2021. In: Proceedings of The 14th International Conference on Educational Data Mining (EDM21). International Educational Data Mining Society, 510-517. https://educationaldatamining.org/edm2021/
EDM '21 June 29 - July 02 2021, Paris, France

solve that difficulty, they employed both items and skills as inputs. The predictive accuracy of a student's performance was improved by AKT. However the interpretability of the parameters is limited because it cannot express a student's ability transition of each skill.

Earlier studies have tackled to develop deep-learning-based methods to give parameter interpretability similarly to IRT models, but those studies have not achieved it for student ability parameters, which are most important for student modeling. The problem is the difficulty of incorporating the ability parameters and item parameters independently into deep-learning-based methods so as not to degrade prediction accuracy. This study addresses that problem.

Recent studies of deep learning have shown that redundancy of parameters for training data reduces generalization error, contrary to Occam's razor. The studies also clarify the reasons [9, 20, 21]. Based on state-of-the-art reports, this study proposes a novel Deep-IRT that models a student's response to an item by two independent redundant networks: a student network and an item network. The proposed method learns student parameters and item parameters independently to avoid impairing the predictive accuracy. A student network employs memory network architecture to reflect dynamic changes of student abilities as DKVMN does. Therefore, the ability parameters of the proposed method do not depend on each item characteristic. They have higher interpretability than those of Deep-IRT. Moreover, the proposed method employs both items and skills as inputs in a different mode of Gosh et al. (2020) [7]. Although Tsutsumi et al. previously proposed a Deep-IRT for test theory, it cannot be applied to KT because a student's ability is constant throughout a learning process [32].

2. RELATED WORK

2.1 Item response theory

There are many item response theory (IRT) models [3, 18, 34, 35, 41]. This subsection briefly introduces two-parameter logistic model (2PLM): an extremely popular IRT model. In 2PLM, the probability of a correct answer given to item j by student i with ability parameter $\theta_i \in (-\infty, \infty)$ is assumed as

$$P_j(\theta_i) = \frac{1}{1 + exp(-1.7a_j(\theta_i - b_j))}, \tag{1}$$

where $a_j \in (0, \infty)$ is the j-th item's discrimination parameter expressing the discriminatory power for student's abilities, and $b_j \in (-\infty, \infty)$ is the j-th item's difficulty parameter representing the degree of difficulty.

2.2 Dynamic key-value memory network

The salient feature of DKVMN is that it assumes N underlying skills and relations between the input (items). Underlying skills are stored in key memory $\boldsymbol{M}^k \in \mathbb{R}^{N \times d_k}$. However, value memory $\boldsymbol{M}_t^v \in \mathbb{R}^{N \times d_t}$ holds abilities of underlying skills at time t. Here, d_k and d_t are tuning parameters. To express the j-th item, the input of DKVMN is a one-hot vector $\boldsymbol{q}_j \in \{0, 1\}^J$, where J represents the number of items for which the j-th element is 1 and for which the other elements are zeroes. DKVMN predicts the performance of item j at time t as explained below.

First, DKVMN calculates the attention, which indicates how strongly an item j is related to each skill as

$$\boldsymbol{\beta}_1^{(j)} = \boldsymbol{W}^{(\beta_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(\beta_1)}, \tag{2}$$

$$w_{tl} = Softmax\left(\boldsymbol{M}_l^k \boldsymbol{\beta}_1^{(j)}\right), \tag{3}$$

where M_l^k represents a l th row vector and w_{tl} signifies the degree of strength of the relation between skill l and item j addressed by a student at time t. In addition, $\boldsymbol{W}^{(\cdot)}$ is the weight matrix and weight vector. $\boldsymbol{\tau}^{(\cdot)}$ is the bias vector and scalar. Next, student vector $\boldsymbol{\theta}_1^{(t)}$ is calculated using the weighted sum of value memory.

$$\boldsymbol{\theta}_1^{(t)} = \sum_{l=1}^{N} w_{tl} \left(\boldsymbol{M}_{tl}^v\right)^\top. \tag{4}$$

Finally, it concatenates $\boldsymbol{\theta}_1^{(t)}$ with $\boldsymbol{\beta}_1^{(j)}$ and predicts correct probability P_{tj} for an item j as

$$\boldsymbol{\theta}_2^{(t)} = \tanh\left(\boldsymbol{W}^{(\theta_2)}\left[\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\beta}_1^{(j)}\right] + \boldsymbol{\tau}^{(\theta_2)}\right), \tag{5}$$

$$P_{tj} = \sigma\left(\boldsymbol{W}^{(u)}\theta_2^{(t)} + \boldsymbol{\tau}^{(u)}\right), \tag{6}$$

where $\boldsymbol{M}_{tl}^v$ represents the l th row vector of $\boldsymbol{M}_t^v$, $[\cdot]$ is a concatenation of vectors, and $\sigma(\cdot)$ represents the sigmoid function. Reportedly, DKVMN has the capability of accurately predicting performance. However, unfortunately, a lack of the interpretability of the parameter remains.

2.3 Deep-IRT

Deep-IRT is implemented by combining DKVMN with an IRT module [43] to improve the DKVMN interpretability. Deep-IRT exploits both the strong prediction ability of DKVMN and the interpretable parameters of IRT. Deep-IRT adds a hidden layer to DKVMN to gain the applicable ability and item difficulty. Specifically, when a student attempts item j at time t, an ability $\theta_3^{(t,j)}$ and item difficulty $\beta_2^{(j)}$ are calculated as shown below.

$$\theta_3^{(t,j)} = \tanh\left(\boldsymbol{W}^{(\theta_3)}\boldsymbol{\theta}_2^{(t)} + \tau^{(\theta_3)}\right), \tag{7}$$

$$\beta_2^{(j)} = \tanh\left(\boldsymbol{W}^{(\beta_2)}\boldsymbol{\beta}_1^{(j)} + \tau^{(\beta_2)}\right), \tag{8}$$

The prediction is based on the difference between $\theta_3^{(t,j)}$ and $\beta_2^{(j)}$ such as IRT.

$$P_{tj} = \sigma\left(3.0 * \theta_3^{(t,j)} - \beta_2^{(j)}\right). \tag{9}$$

Here, ability $\theta_3^{(t,j)}$ is calculated using $\boldsymbol{w}_t$ in equation (6), which depends on the item to solve because it implicitly assumes that items with the same skills are equivalent. In other words, the ability estimate for the same student and time might differ if the student attempts a different item. Furthermore, in equation (7), Deep-IRT uses item vector $\beta_1^{(j)}$ to calculate $\theta_2^{(t)}$. An important difficulty is that a student's ability, which depends on each item, hinders the interpretability of the parameters. Although Tsutsumi et al. [32] also proposed a Deep-IRT as a test theory, the purpose is different from this study because it can not be available for KT as mentioned before.

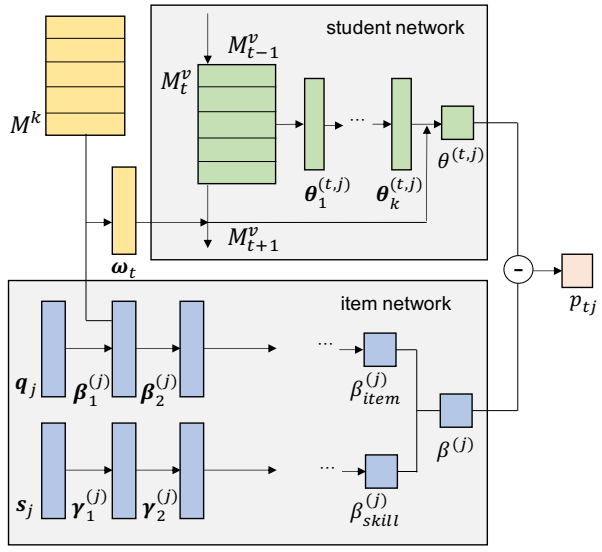

Figure 1: Network architecture for Deep-IRT with independent student and item networks. The yellow components represent the process of getting the attention weight. Also, the green components are associated with the student network and the process of updating the value memory. The blue components are associated with the item network.

3. DEEP-IRT WITH INDEPENDENT STUDENT AND ITEM NETWORKS

To resolve the difficulty described above, this study proposes a novel Deep-IRT method comprising two independent neural networks: the student network and Item deep network, as shown in Figure 1. The student network employs memory network architecture such as DKVMN to ascertain changes in student ability comprehensively. The item network includes inputs of two kinds: the item attempted by a student and the necessary skills to solve the item. Using outputs of both networks, the probability of a student answering an item correctly can be calculated.

The proposed method can estimate student parameters and item parameters independently such that prediction accuracy does not decline because the two independent networks are designed to be more redundant than with earlier methods , based on state-of-the-art reports [9, 20, 21]. The proposed method predicts P_{tj}, the probability of a correct answer assigned to item j at time t, using the item difficulties and the student abilities, as follows.

3.1 Item network

In the item network, two difficulty parameters of item j are estimated: the item characteristic difficulty parameter β^j_{item} and the skill difficulty β^j_{skill} to solve item j. The item characteristic difficulty parameter indicates the unique difficulties of the item, excepting the required skill difficulty. The proposed method expresses item difficulty as the sum of the two difficulty parameters of β^j_{item} and β^j_{skill}.

As with DKVMN, to express the j-th item, an input of the item network is a one-hot vector $\boldsymbol{q}_j \in \mathbb{R}^J$ as shown below.

$$q_{jm} = \begin{cases} 1 & (j = m) \\ 0 & (otherwise) \end{cases} \tag{10}$$

Here, J stands for the number of items. The item network comprises n layers. The item characteristic difficulty parameter of item j is calculated using a feed forward neural network as

$$\boldsymbol{\beta}^j_1 = \tanh\left(\boldsymbol{W}^{(q_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(q_1)}\right), \tag{11}$$

$$\boldsymbol{\beta}^j_k = \tanh\left(\boldsymbol{W}^{(\beta_k)}\boldsymbol{\beta}^j_{k-1} + \boldsymbol{\tau}^{(\beta_k)}\right), \tag{12}$$

$$\beta^j_{item} = \boldsymbol{W}^{(\beta_{item})}\boldsymbol{\beta}^j_n + \tau^{(\beta_{item})}, \tag{13}$$

where $k = \{2, ..., n\}$. The last layer $\beta^{(j)}_{item}$ represents the j-th item characteristic difficulty parameter.

Similarly, to compute the difficulty of skills, the proposed method uses the input of necessary skills $\boldsymbol{s}_j \in \mathbb{R}^S$ as presented below.

$$s_{jm} = \begin{cases} 1 & (item\ j\ requires\ skill\ m) \\ 0 & (otherwise) \end{cases} \tag{14}$$

Here, S represents the number of skills:

$$\boldsymbol{\gamma}^j_1 = \tanh\left(W^{(\gamma_1)}\boldsymbol{s}_j + \tau^{(\gamma_1)}\right), \tag{15}$$

$$\boldsymbol{\gamma}^j_k = \tanh\left(W^{(\gamma_k)}\boldsymbol{\gamma}^j_{k-1} + \tau^{(\gamma_k)}\right), \tag{16}$$

$$\beta^j_{skill} = \boldsymbol{W}^{(\beta_{skill})}\boldsymbol{\gamma}^j_n + \tau^{(\beta_{skill})}, \tag{17}$$

where $k = \{2, ..., n\}$. The last layer β^j_{skill} denotes the difficulty parameter of the required skills to solve the j-th item.

3.2 Student network

In the student network, the proposed method calculates $\boldsymbol{\theta}^t_1$ based on the past response history as

$$\boldsymbol{\theta}^{(t,j)}_1 = \sum_{l=1}^{N} \boldsymbol{M}^v_{t,l}, \tag{18}$$

where $\boldsymbol{M}^v_t$ is a memory matrix holding a students' latent knowledge state, which are estimated similarly to DKVMN. Next, an interpretable student's ability vector θ^t_n is estimated as follows. Therein, n represents a number of hidden layers decided depending on the prediction accuracy of actual data.

$$\boldsymbol{\theta}^{(t,j)}_k = \tanh\left(W^{(\theta_k)}\boldsymbol{\theta}^{(t,j)}_{k-1} + \tau^{(\theta_k)}\right), \tag{19}$$

$$\theta^{(t,j)} = \boldsymbol{w}^\top_t \boldsymbol{\theta}^{(t,j)}_k, \tag{20}$$

where $k = \{2, ..., n\}$. As a difference between the proposed method and Deep-IRT, the proposed method does not multiply the attention in equation (18). In addition, $\boldsymbol{\theta}^{(t,j)}_k$ is not calculated using features of items such as equations (5) and (7). Therefore, the ability parameter vector $\theta^{(t,j)}$ does not depend on each item. Namely, it is independent from the difficulty parameter. The value of which denotes the ability for the corresponding latent skill because it is independent of any item. Therefore, $\boldsymbol{\theta}^{(t,j)}_k$ can be interpreted as a measurement model such as a multidimensional IRT [25].

3.3 Prediction of student response to an item

The proposed method predicts a student's response probability to an item using the difference between a student's ability $\theta^{(t,j)}$ to solve item j at time t and the sum of two difficulty parameters β^j_{item} and β^j_{skill}.

$$P_{tj} = \sigma\left(3.0 * \theta^{(t,j)} - (\beta^j_{item} + \beta^j_{skill})\right). \quad (21)$$

After the procedure, the value memory is updated using $\boldsymbol{c}_j$ based on the input $\boldsymbol{q}_j$ and actual performance such as DKVMN [46].

The loss function of the proposed method employs cross-entropy, which reflects classification errors. The cross-entropy of the predicted responses P_{tj} and the true responses u_{tj} is calculated as

$$\ell(u_t, P_{tj}) = -\sum_t \left(u_{tj} \log P_{tj} + (1 - u_{tj}) \log(1 - P_{tj})\right), \quad (22)$$

where u_{tj} is the true response to item j at time t. The student's response u_{tj} is recorded as 1 when the student answers the item correctly and 0 otherwise. All parameters are learned simultaneously using a well known optimization algorithm: adaptive moment estimation [15].

4. PREDICTIVE ACCURACY

4.1 Datasets

We conduct experiments to compare the performance of our approach against existing solutions. This section presents comparison of the prediction accuracies for student performance of the proposed method with those of earlier methods (DKT, DKVMN, Deep-IRT, AKT) using four benchmark datasets as ASSISTments2009[1], ASSISTments2015[2], Statics2011 [3], KDDcup[4]. ASSISTments2009 and KDDcup have item and skill tags, although most methods explained in the relevant literature adopt only the skill tag as an input. However, methods with skill inputs rely on the assumption that items with the same skill are equivalent [7]. That assumption does not hold when an item's difficulties in the same skill differ greatly. Therefore, as inputs to AKT and the proposed method, we employ not only skills but also items.ASSISTments2015 has only the skill tag. Therefore, we employ only the skill tag as an input.

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), the average length of items which students addressed (Learning length), and the rate of items in which the number of student addressed is less than 10 (Sparsity). For all the datasets, we excepted students who addressed fewer than five items. Additionally, we set 200 items as the upper limit of the input length according to an earlier study [43]. When the input length of items becomes greater than 200, we use the first 200 response data for all methods.

[1]https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data

[2]https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builder-data

[3]https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507

[4]https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

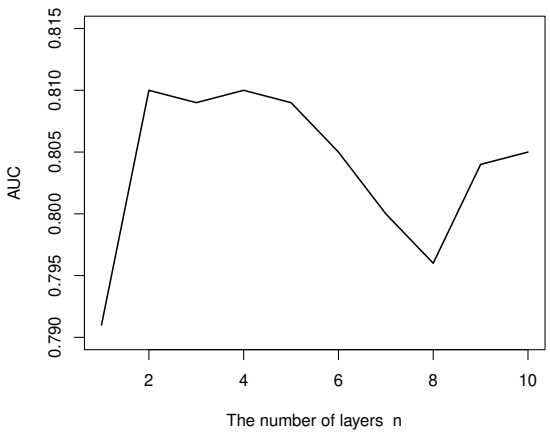

Figure 2: AUC and the Number of Layers

4.2 Hyperparameter selection and evaluation

We used ten-fold cross-validation to evaluate the prediction accuracies of the methods. The item parameters and the hyperparameters are learned using 70% of datasets. Given the estimated hyperparameters, a student's ability can be estimated at each time using the remaining 30% of each dataset. For all methods, the hidden layer size and memory dimension are chosen from $\{10, 20, 50, 100, 200\}$ using cross-validation. In addition, for the earlier methods, we used the hyperparameters reported from earlier studies [7, 43].

To ascertain the number of layers n for the proposed method, we conducted some experiments to gain experience using ASSISTments2009 while changing the layer number. The results are presented in Figure 2. As shown in the figure, AUC score reaches its highest level when $n = 2$ and $n = 4$. Based on this result, we employ $n = 2$ for the following experiments because the computation time of the proposal increases exponentially as the number of layers increases.

If the predicted correct answer probability for the next item is 0.5 or more, then the student's response to the next item is predicted as correct. Otherwise, the student's response is predicted as incorrect. For this study, we leverage three metrics for prediction accuracy: Accuracy (Acc) score, AUC score, and F1 score. The first, Acc, represents the concordance rate between the student predictive performance and the true performance. The second, AUC, represents the predictive accuracy of the correct answer probabilities. F1 indicates the average of the F1 score of incorrect answer prediction and the F1 score of correct answer prediction.

4.3 Results

The respective values of Acc, AUC, and F1 for those benchmark datasets are shown in Table 2. Results show that the proposed method with item and skill inputs provides the best performance for the metrics: averages of Acc and F1. Especially noteworthy is that the proposed method outperforms AKT, which is the most advanced method. Furthermore, the proposed method with item and skill inputs provides better performance than that with skill or item inputs. These results indicate that parameter estimation, not only with skill but also with item, improves the predictive accuracy.

Table 1: Summary of Benchmark Datasets

Dataset	No. students	No. skills	No. Items	Rate Correct	Learning Length	Sparsity
ASSIST2009	4,151	111	26,684	68.0%	70.8	55.2%
ASSIST2015	19,840	100	N/A	73.2%	34.2	12.6%
Statics2011	333	156	1,223	77.7%	180.9	2.6%
KDDcup	820	43	476	78.3%	11.9	57.8%

Table 2: Predictive Accuracy of Student Performance with Benchmark Datasets

		DKT	DKVMN	Deep-IRT	AKT	AKT (item&skill)	Proposed	Proposed (item&skill)
ASSIST2009	Acc	0.759	0.763	**0.768**	0.692	0.755	**0.768**	0.765
	AUC	0.781	0.807	0.806	0.717	0.811	**0.818**	0.810
	F1	0.697	0.714	0.718	0.639	**0.726**	0.725	0.722
ASSIST2015	Acc	0.754	0.749	0.747	**0.757**	N/A	0.752	N/A
	AUC	0.730	0.732	0.727	**0.760**	N/A	0.751	N/A
	F1	0.433	0.541	0.540	**0.616**	N/A	0.543	N/A
Statics2011	Acc	0.769	0.805	0.817	0.809	0.818	0.819	**0.822**
	AUC	0.666	0.819	0.822	0.821	**0.827**	0.821	0.821
	F1	0.483	0.679	0.681	**0.690**	0.677	0.679	**0.690**
KDDcup	Acc	0.784	0.773	0.792	0.774	0.780	0.786	**0.802**
	AUC	0.538	0.594	0.588	0.606	**0.610**	0.588	**0.610**
	F1	0.439	0.439	0.455	0.441	0.449	0.469	**0.478**
Average	Acc	0.767	0.773	0.781	0.758	0.784	0.781	**0.796**
	AUC	0.679	0.738	0.736	0.726	**0.749**	0.745	0.747
	F1	0.513	0.593	0.599	0.597	0.617	0.604	**0.630**

However, AKT with item and skill inputs shows the best average values of AUC. Actually, AKT with item and skill inputs also provides higher performance than that achieved with skill or item inputs, as shown in [7]. Gosh et al. (2020) reported that AKT is more effective for large datasets. Therefore, AKT provides the best performance for all the metrics of ASSISTments2015, which has an extremely large number of students.

Furthermore, surprisingly, the averages of ACC, AUC, and F1 obtained using the proposed method with skill input are better than Deep-IRT, although the proposed method separates student and item networks. This result implies that redundant deep student and item networks function effectively for performance prediction. These results are explainable from reports of state-of-the-art methods [9, 20, 21].

The performance results obtained using DKVMN are almost identical to those obtained using Deep-IRT because they have almost identical network structures. Results show that DKT provides the worst performance among the methods studied here.

5. PARAMETER INTERPRETABILITY

5.1 Interpretability of difficulty parameters

To evaluate the interpretability of the difficulty parameters of the proposed method, we compare the parameters of IRT with those of Deep-IRT using a simulation data. The dataset includes 2000 students' responses to 50 items and it is generated from 2PLM as shown in equation (1). The priors of the parameters have $\boldsymbol{\theta} \sim N(0,1), \boldsymbol{a} \sim LN(0,1), \boldsymbol{b} \sim N(0,1)$. We estimated the parameters of the proposal and Deep-IRT using the dataset. Table 3 shows the Pearson correlation between the true parameters of the true models and the estimated parameters, respectively, of the proposed method and Deep-IRT. Additionally, we show the prediction accuracies of the proposed method and Deep-IRT for the dataset. The proposal provides higher correlations with true parameters than Deep-IRT does, whereas the proposed method has higher accuracy than Deep-IRT has. The results demonstrate that the two independent networks of the proposed method function effectively for the interpretability of the estimated parameters and for the prediction accuracies.

Table 3: Pearson correlation

parameter	Deep-IRT	Proposed
difficulty	0.611	0.886
accuracy	0.694	0.695

5.2 Student ability transitions

This section shows student ability transitions using the proposed method. Visualizing the ability transition for each skill is helpful for both students and teachers because they can discover student strengths and weaknesses and can improve the learning method to fill in the learning gaps. Yeung [43] demonstrated a student ability transition for each skill using Deep-IRT. However, their results included some counter-intuitive ability estimates. For example, even when the student answered incorrectly, the corresponding student ability estimate increased. Moreover, Deep-IRT cannot identify a relation among multidimensional skills. There are cases in which a student's ability for low-level skills decreases even when the student responds correctly to items for high-level skills. These unstable behaviors of Deep-IRT might engender serious difficulties, which will consequently confuse students and teachers, as a student model.

Figure 3 depicts a student's ability transitions of the proposal for the ASSIST2009 dataset. The vertical axis shows

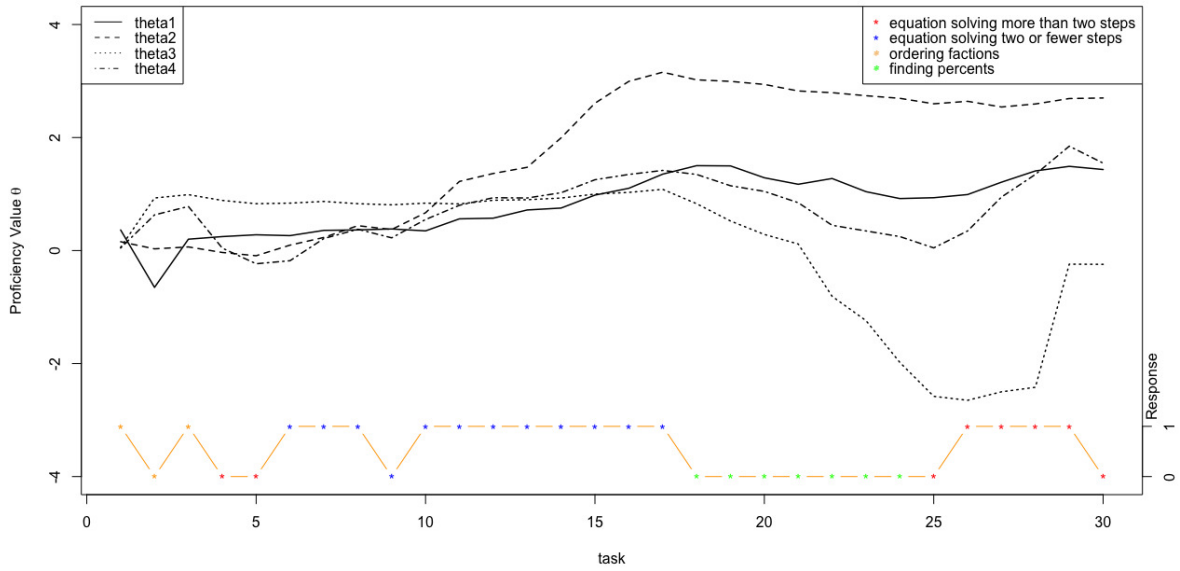

Figure 3: An example of a student ability transition from the ASSIST2009 dataset. The skill tags are classified respectively as equation solving two or fewer steps (blue), ordering fractions (orange), finding percents (green), and equation solving more than two steps (red). The student responses to items are shown at the bottom of the graph.

the student ability on the left side, with the student's response to an item on the right side. The horizontal axis shows the item number. The student's response is 1 when the student answers the item correctly; it is 0 otherwise. The student attempted skills of "equation solving more than two steps" (shown in red), "equation solving two or few steps" (shown in blue), "ordering factions" (shown in orange), and "finding percents" (shown in green). Figure 3 can be interpreted as explained below.

1. Theta 1 decreases when the student responds to item 2 "ordering factions" (orange) incorrectly and it increases when the student responds to item 3 correctly. Therefore, theta 1 indicates the ability of "ordering factions".
2. Items 6–17 correspond to the skill of "equation solving two or few steps"(blue). Theta 2 indicates the ability of "equation solving two or few steps" because theta 2 greatly increases while the student answers correctly.
3. For the skill of "finding percents" (green), the student answers all items incorrectly. Theta 3 indicates the ability of "finding percents" (green) because it greatly decreases in items 18–24.
4. Items 4, 5, and 25–30 correspond to the skill of "equation solving more than two steps" (red). Theta 4 decreases when the student answers to item 4 and 5 incorrectly, and increases when the student answers to items 26–29 correctly. Therefore, theta 4 represents the ability of "equation solving more than two steps" (red).

Figure 3 shows that the proposed method estimates the ability of each skill to reflect the student responses. Additionally, it estimates relations among the skills. Therefore, when a student responds to an item correctly/incorrectly, not only does the corresponding skill ability increase/decrease; those for other skills increase/decrease as well. Consequently, the results demonstrate that the proposed method improves both the interpretability and the prediction accuracies of Deep-IRT.

6. CONCLUSIONS

This study proposed a novel Deep-IRT that models a student's response to an item by two independent redundant networks: a student network and an item network. Because two independent redundant neural networks are used, the parameters of the proposed method can be highly interpreted with keeping hight prediction accuracy. Moreover, the proposed method employs both items and skills as inputs. Experiments demonstrated that the proposed method with item and skill inputs provided the best performance for the metrics: averages of Acc and F1. deep-learning-based methods. The result also showed AKT with item and skill inputs provided the best average values of AUC. Especially, AKT provided the best performances for large datasets as Gosh et al. (2020) reported [7]. In addition, results of experiments show that the parameters of the proposed method are more interpretable than those of Deep-IRT. This study employed slightly redundant deep networks compared to earlier methods. As future work, we intend to use the proposed method to investigate the performances of more redundant and deeper networks. In addition, we will try to optimize a forgetting function for past data to maximize the prediction accuracy for large data sets.

7. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers 19H05663 and 19K21751.

8. REFERENCES

[1] D. Agarwal, R. Baker, and A. Muraleedharan. Dynamic knowledge tracing through data driven recency weights. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, pages 725–729, 2020.

[2] F. Ai, Y. Chen, Y. Guo, Y. Zhao, Z. Wang, G. Fu, and G. Wang. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In *EDM*, 2019.

[3] F. Baker and S. Kim. *Item Response Theory: Parameter Estimation Techniques, Second Edition*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis, 2004.

[4] P. Chen, Y. Lu, V. Zheng, and Y. Pian. Prerequisite-driven deep knowledge tracing. In *In IEEE International Conference on Data Mining, ICDM 2018*, pages 39–48, 2018.

[5] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.*, 4(4):253–278, Dec 1995.

[6] C. Ekanadham and Y. Karklin. T-skirt: Online estimation of student proficiency in an adaptive learning system. *CoRR*, abs/1702.04282, 2017.

[7] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[8] S. Gowda, J. Rowe, R. Baker, M. Chi, and K. Koedinger. Improving models of slipping, guessing, and moment-by-moment learning with estimates of skill difficulty. In *EDM 2011 – Proceedings of the Fourth International Conference on Educational Data Mining*, pages 199–208, 01 2011.

[9] H. He, G. Huang, and Y. Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In *Advances in Neural Information Processing Systems 32*, pages 2553–2564. Curran Associates, Inc., 2019.

[10] T. Ishii, P. Songmuang, and M. Ueno. Maximum clique algorithm for uniform test forms assembly. In *The 16th International Conference on Artificial Intelligence in Education*, volume 7926, pages 451–462, 07 2013.

[11] T. Ishii, P. Songmuang, and M. Ueno. Maximum clique algorithm and its approximation for uniform test form assembly. *IEEE Transactions on Learning Technologies*, 7:83–95, 01 2014.

[12] T. Ishii and M. Ueno. Clique algorithm to minimize item exposure for uniform test forms assembly. In *International Conference on Artificial Intelligence in Education*, pages 638–641, 06 2015.

[13] T. Ishii and M. Ueno. Algorithm for uniform test assembly using a maximum clique problem and integer programming. In *Artificial Intelligence in Education. Springer International Publishing*, pages 102–112, 06 2017.

[14] M. Khajah, Y. Huang, J. Gonzalez-Brenes, M. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. *Personalization Approaches in Learning Environments*, 1181:5–17, 2014.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the ICLR*, 2014.

[16] J. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the Fifth International Conference on Educational Data Mining*, pages 118–125, 01 2012.

[17] X. Liangbei and D. Mark. Dynamic knowledge embedding and tracing. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, pages 524–530, 2020.

[18] F. Lord and M. Novick. *Statistical Theories of Mental Test Scores*. Addison-Wesley, 1968.

[19] Y. Lu, D. Wang, Q. Meng, and P. Chen. Towards interpretable deep learning models for knowledge tracing. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, pages 185–190, 2020.

[20] A. Morcos, H. Yu, M. Paganini, and Y. Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems 32*, pages 4932–4942. Curran Associates, Inc., 2019.

[21] V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems 32*, pages 11615–11626. Curran Associates, Inc., 2019.

[22] Z. Pardos and N. Heffernan. T.: Modeling individualization in a bayesian networks implementation of knowledge tracing. In *In Proceedings of the 18th International Conference on User Modeling, Adaption, and Personalization*, pages 255–266, 06 2010.

[23] Z. Pardos and N. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *Proceedings of 19th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011)*, pages 243–254, 01 2011.

[24] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 505–513. Curran Associates, Inc., 2015.

[25] M. Reckase. Multidimensional item response theory models, springer. 2009.

[26] J. Reye. Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education*, 14:63–96, 2004.

[27] H. Sepp and S. Jurgen. Long short-term memory. *Neural Computation*, 14:1735–1780, 1997.

[28] P. Songmuang and M. Ueno. Bees algorithm for construction of multiple test forms in e-testing. *IEEE Transactions on Learning Technologies*, 4:209–221, 07 2011.

[29] Y. Su, Q. Liu, Q. Liu, Z. Huang, Y. Yin, E. Chen, C. H. Q. Ding, S. Wei, and G. Hu. Exercise-enhanced sequential modeling for student performance prediction. In *AAAI*, pages 2435–2443, 2018.

[30] X. Sun, X. Zhao, Y. Ma, X. Yuan, F. He, and J. Feng. Multi-behavior features based knowledge tracking

using decision tree improved dkvmn. In *Proceedings of the ACM Turing Celebration Conference – China*, New York, NY, USA, 2019. Association for Computing Machinery.

[31] H. Tong, Y. Zhou, and Z. Wang. Exercise hierarchical feature enhanced knowledge tracing. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, pages 324–328, 2020.

[32] E. Tsutsumi, R. Kinoshita, and M. Ueno. Deep item response theory as a novel test theory based on deep learning. *Electronics*, 10(9), 2021.

[33] M. Ueno. Adaptive testing based on bayesian decision theory. *International Conference on Artificial Intelligence in Education*, pages 712–716, 2013.

[34] M. Ueno and Y. Miyazawa. Probability based scaffolding system with fading. *Artificial Intelligence in Education – 17th International Conference, AIED*, pages 237–246, 2015.

[35] M. Ueno and Y. Miyazawa. Irt-based adaptive hints to scaffold learning in programming. *IEEE Transactions on Learning Technologies*, 11(4):415–428, Oct 2018.

[36] M. Ueno and S. Pokpong. Computerized adaptive testing based on decision tree. In *Advanced Learning Technologies (ICALT), 2010 IEEE Tenth International Conference*, pages 191–193, 2010.

[37] X. Wang, J. Berger, and D. Burdick. Bayesian analysis of dynamic item response models in educational testing. *The Annals of Applied Statistics*, 7(1):126–153, 2013.

[38] Z. Wang, X. Feng, J. Tang, G. Huang, and Z. Liu. Deep knowledge tracing with side information. In *The 20th International Conference on Artificial Intelligence in Education (AIED)*, pages 303–308, 2019.

[39] R. Weng and D. Coad. Real-time bayesian parameter estimation for item response models. *Bayesian Analysis*, 13, 12 2016.

[40] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. In *9th International Conference on Educational Data Mining*, volume 1, pages 539–544, 06 2016.

[41] W.J. van der Linden. *Handbook of Item Response Theory, Volume Two: Statistical Tools*. Chapman and Hall/ CRC Statistics in the Social and Behavioral Sciences. Chapman and Hall/ CRC, 2016.

[42] X. Xiong, S. Zhao, V. Inwegen, E. G., and J. E. Beck. Going deeper with deep knowledge tracing. In *Proceedings of International Conference on Education Data Mining*, 2016.

[43] C. Yeung. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM*, 2019.

[44] C. K. Yeung and D.-Y. Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth ACM Conference on Learning @ Scale*, pages 1–10, 2018.

[45] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[46] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory network for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 765–774, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.