修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院　情報理工学研究科　情報・ネットワーク工学専攻　博士前期課程		
氏　　　名	BIAN NING	学籍番号	1931129
論 文 題 目	Deep-IRT incorporating hints in adaptive learning （アダプティブラーニングのためのヒントを組み込んだ Deep-IRT）		

要　　旨

近年、教育の現場では、機械学習を用いて学習履歴データから学習者の能力成長を把握し、個々の学習者に適切なヒントを提供するアダプティブラーニングが注目されている。学習者に最適なヒントを提供するためには、学習者が誤答した際に、各ヒントを提供した場合の正答確率を正確に推定する必要がある。最新の研究では、深層学習モデルと項目反応理論を組み合わせた Deep-IRT 手法が開発されており、学習履歴データから課題の難易度と学習者の多次元のスキルに対する能力変化を推定できるようになってきた。しかし、既存の Deep-IRT 手法では、課題に依存したヒントの提供しか想定しておらず、能力変化を考慮した最適ヒントは提案されていない。本論文では Deep-IRT モデルをアダプティブラーニングに適用できるようにするために、学習者が項目に正答するまでに必要とする最適なヒントを予測する新たなモデルを提案する。評価実験では実データを用いて学習者が課題に正答するために必要とするヒントを予測し、実際のデータと比較して提案手法の有効性を示す。

キーワード：アダプティブラーニング、Deep-IRT、ナレッジトレーシング、深層学習

Deep-IRT incorporating hints in adaptive learning

The University of Electro-Communications

1931129

BIAN NING

Advisors:

Prof. Maomi UENO

Assoc. Prof. Shuichi KAWANO

bianning@ai.lab.uec.ac.jp

CONTENTS

Table of Figures

ABSTRACT

Adaptive learning has been studied actively in the field of education and artificial intelligence for helping students by providing optimal teachers' supports (hints) when he/her answers incorrectly to a task. It is important to estimate a growth of the students' ability and predict the correct answer probability to a task with the supports(hints) based on learning history data. Recent researchers have proposed several Deep-IRT methods which combine the deep learning model with item response theory. The previous Deep-IRT methods estimate a student's ability changes under the multidimensional skills and predict the correct answer probability to a task. However, the previous Deep-IRT methods cannot handle items with hints. In this paper, in order to apply Deep-IRT to adaptive learning, we propose a new Deep-IRT method for selecting the optimal hints by adding a hint network. The experiments result shows that the proposed method improves the prediction accuracy of the students' performances with hints.

Keywords: adaptive learning, Deep-IRT, knowledge tracing, deep learning

Chapter 1: INTRODUCTION

Recently, adaptive learning has attracted attention in educational technology area. A typical adaptive learning system identifies the level of understanding and weaknesses of individual students, and then presents the optimal items and adaptive hints to a student.

Wood et al [2] established a scaffolding framework to facilitate student improvement, with teachers providing moderate support based on the students' abilities when they face to difficult tasks. Scaffolding dynamically assesses prior achievement and individual differences in learning, which estimates the student's level of competence and the need to estimate their ability during the learning process, and predicts their performance when the teacher has provided support. A good teacher can predict the performance of students with support and provide a conceptual framework for minimal support in problem-solving. However, conventional evaluation methods for evaluating the abilities, assessing, and supporting of students are based on the experience and intuition of a teacher. Therefore, it is difficult to adaptively provide the optimal amount of support for each student.

To solve this problem, Ueno and Matsuo [3], Ueno and Miyazawa [4], [5] developed an adaptive learning system that uses Item Response Theory(IRT) [6], [7], [8], [9] to present adaptive hints to achieve a specific probability of correct performance. The hints are presented to a student to maximize the amount of information on IRT before the correct answer is obtained. Additionally, ability is estimated sequentially after a task is presented, based on the student's response. Hints are chosen in order to give the student a probability of answering the task correctly close to 0.5. As a student's ability increases, support is automatically reduced and scaffolding is removed. The model results of Ueno and Miyazawa [5] indicate that scaffolding, which achieves a student success with a probability of 0.5, provides the best learning performance. Meanwhile, scaffolding systems that provide a probability of 0.5 automatically reduce the number of hints (the number of supports). However, the conventional IRT model tended to overfit the data until the number of learned tasks was sufficient. Thus, the adaptive hints presented to a student might be insufficient or they are presented more than necessary.

To solve this problem, Tsutsumi et al. [22] proposed a hidden Markov IRT model in which the ability values change over time during the learning process. To maximise the prediction of the performance after the presentation of hints, the proposed model provides a parameter that can forget the past training data when estimating the student's performance, and improves the prediction of the number of hints used by students to answer correctly compared to the IRT model. However, the conventional IRT approach cannot handle time-series data express a student's ability change.

In artificial intelligence area, various deep learning-based methods based on time series have been proposed to estimate a learner's correct answer probability for a task. As the first deep learning-based method, based on recurrent neural networks (RNNs), Deep Knowledge Tracking (DKT) [19] can model a student's knowledge status. However, it is difficult to track a student's mastery of each skill.

To improve the interpretability of deep learning-based methods, the Dynamic Key-Value Memory Network (DKVMN) [10] uses memory enhancement neural networks and attention mechanisms to track a student's ability. Although the DKVMN has the advantage of accurately predicting a student's performance, the interpretability of the parameters is still lacking.

Recent researchers have proposed several Deep-IRT [11] methods. Deep-IRT based on a time series can predict item difficulty parameters. The current Deep-IRT[22] responds to items through a student network and an item network. The Deep-IRT method estimates the changes in a student's ability under multidimensional skills and predicts the correct answer probability to the task. However, previous Deep-IRT methods cannot handle hints, which limits their applicability to learning support.

IDRT[12] as a form of Deep-IRT, can infer the abilities of students and the difficulties of items. The loss of accuracy in ability estimation is reduced even when students are not homogeneous and there are no common items between tests. To apply IDRT to adaptive learning, Cai[13] improved IDRT by adding a hint network. However, the IDRT model is not a time-series model and it is difficult to track a student's learning status over time. Therefore, it is difficult to present hints based on changes over time.

In this study proposes to add a hint network to Deep-IRT, to predict the least hints a student needs to answer to an item correctly, and the optimal hint for each item. The experimental results indicate that the proposed method improves the prediction accuracy of student performance with hints.

This paper is structured as follows: Chapter 2 describes the adaptive hints system. Chapter 3 introduces the deep-learning methods with high accuracy related to the proposed model. Chapter 4 introduces Deep-IRT to predict the optimal hints. In Chapter 5, we describe the experimental methods and experimental results. Finally, Chapter 6 concludes the study.

Chapter 2: ADAPTIVE HINTS SYSTEM

IRT-Based Adaptive Hints

To promote students' development, it is important for teachers to provide scaffolding when students face higher-order tasks, by providing them with moderate support according to their abilities. Scaffolding is a step-by-step process that supports students to solve a difficult tasks. Effective scaffolding requires accurate prediction of a student's current ability and the student's performance after a hint is provided. The capability evaluation for effective scaffolding is called dynamic assessment, and the accuracy of the dynamic assessment is crucial for effective scaffolding.

Ueno and Miyazawa [4], [5] used Item Response Theory to predict a student's performance on several hints and then selected the best hint. A sample programming task is shown in Figure 1. In this system, students first study the basics of programming. Subsequently, the student solved programming tasks. In the tasks, step-by-step hints such as explanations about programming syntax and code meaning were given when students answer incorrectly, and the hints were given more specifically until the student completed the task. Ueno and Miyazawa [4], [5] used a dynamic assessment system to improve the performance of students who answered incorrectly with hints.

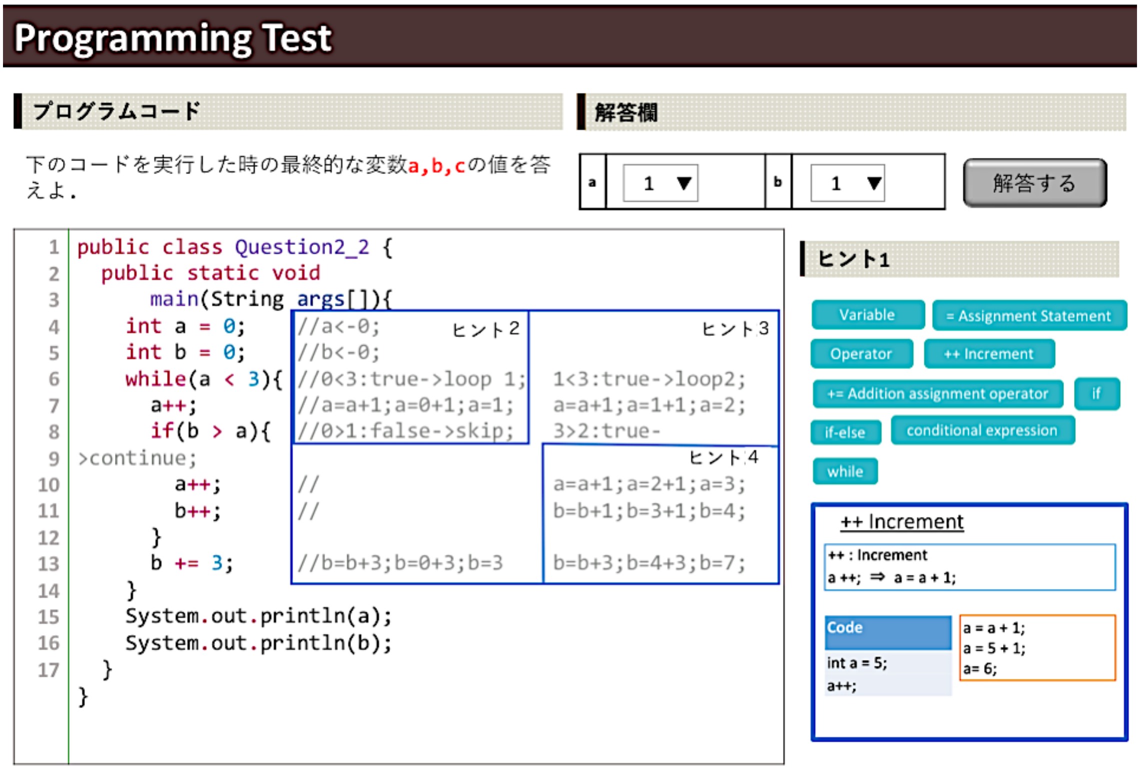

Figure 1: Sample step-by-step hints [4]

In the system, Ueno and Miyazawa employed a series of graded hints $\{k\}, (k = 1, \cdots, K-1)$ for task i. Initially, task i was presented to student j without hints. If the student answered task i incorrectly, hint $k = 1$ was given, and hint $k = K-1$ was given for each further incorrect answer. If the student answered correctly or if they still answered incorrectly after the last hint, feedback was given and the next task $i+1$ was presented. This procedure was repeated until the number of task i was reached. Let the number of students be J, the number of the tasks be I and the number of hints be K. The dynamic assessment data is given by (1) and (2), and x_{jiK} indicates the response data when student j answered incorrectly with hint $K-1$.

$$x_{jik} = \begin{cases} 1: student\ j\ answers\ corectly\ to\ task\ i\ when\ hint\ k\ is\ presented \\ 0: else\ other \end{cases}, \quad (1)$$

$$X = \{x_{jik}\}, (j = 1, \ldots, J, i = 1, \ldots, I, k = 0, \ldots, K)\ . \quad (2)$$

After giving the hints for each step, we must accurately estimate the value of the student's ability and predict the student's performance. The prediction results were used to decide whether to continue providing learning support.

In the IRT model, the probability of student j answering task i correctly with hint $\{k\}, (k = 1, \cdots, K-1)$ is given as

$$P_{ijk} = P^*_{ijk-1} - P^*_{ijk}, and \tag{3}$$

$$P^*_{ijk} = \frac{1}{1 + \exp\left(-a_i(\theta_j - b_{ik})\right)}, \tag{4}$$

where $P^*_{ij0} = 1$ and $P^*_{ijK} = 0$, where a_i is a parameter representing the discriminatory ability of task i, b_{ik} is a hint parameter representing the difficulty of hint k when it appears in task i, and θ_j is a parameter representing the ability value of student j.

The experimental results indicate high accuracy in predicting the probability of the student's correct answer when given a hint. This method improves the reliability of dynamic assessment. However, conventional IRT methods cannot handle time-series data to express a student's ability change.

Chapter 3: DEEP-LEARNING APPROACH

1. Deep Knowledge Tracing

To overcome these difficulties, a deep learning approach that can process time-series data is necessary. As the first deep learning-based method, basing on recurrent neural networks (RNNs), Deep Knowledge Tracking (DKT) [19] can model the knowledge status of students.

In DKT, single-point encoding is first used to convert the interaction (q_t, a_t) into a fixed-length input vector $\boldsymbol{x}_t$. Thereafter, DKT passes $\boldsymbol{x}_t$ to the hidden layer, and uses a long and short-term memory (LSTM) [20] unit to calculate the hidden state $\boldsymbol{h}_t$. To compute the output vector $\boldsymbol{y}_t$ that shows the correct probability of answering each KC, the potential knowledge state to the output layer has been extended. For instance, a student has a series of question-and-answer interactions of length T, and the DKT model maps the input $(x_1, x_2, \cdots, x_T)$ to the output $(y_1, y_2, \cdots, y_T)$ accordingly.

However, DKT summarizes the knowledge status of all students' skills in a hidden state. The disadvantages of the model are as follows : (1) it makes it difficult to track the students' mastery of a certain concept and (2) it is difficult to identify the concepts that students are familiar or unfamiliar with.

2. Dynamic Key-Value Memory Network

To improve the interpretability of deep learning-based methods, a Dynamic Key-Value Memory Network (DKVMN) [10] that uses key-value pairs instead of a single matrix of storage structure. Memory-Augmented Neural Network and attention mechanisms are exploited to trace student abilities in multi-dimensions.

To express the j-th item, one-hot vector $\boldsymbol{q}_j \in \{0,1\}^J$, where J is the number of items for which the j-th element is 1 and the other elements are 0 has been inputted.

First, the attention w_{ji} that indicates how strongly an item j is related to each skill according to equation (6).

$$\boldsymbol{\beta}_1^{(j)} = \boldsymbol{W}^{(\beta_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(\beta_1)}, \text{ and} \tag{5}$$

$$w_{ji} = Softmax\left(\boldsymbol{M}_i^k \boldsymbol{\beta}_1^{(j)}\right), \tag{6}$$

where $Softmax(z_i) = e^{z_i} / \sum_j e^{z_i}$ and is differentiable, and $\boldsymbol{W}^{(\beta_1)}$ is the weight matrix, and $\boldsymbol{\tau}^{(\beta_1)}$ is the bias parameter.

The student vector $\boldsymbol{\theta}_1^{(t)}$ is then regarded as a summary of the students' proficiency in the exercise. $\boldsymbol{\theta}_1^{(t)}$ and input $\boldsymbol{\beta}_1^{(j)}$ were concatenated to obtain the summary vector $\boldsymbol{\theta}_2^{(t)}$, which contained the proficiency level of the student and the difficulty of the previous exercise.

$$\boldsymbol{\theta}_1^{(t)} = \sum_{i=1}^{N} w_{ti} (M_{ti}^v)^T, \text{ and} \tag{7}$$

$$\boldsymbol{\theta}_2^{(t)} = tanh\left(\boldsymbol{W}^{(\theta_2)}\left[\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\beta}_1^{(j)}\right] + \boldsymbol{\tau}^{(\theta_2)}\right), \tag{8}$$

where $\tanh(z_i) = (e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i})$.

Finally, $\boldsymbol{\theta}_2^{(t)}$ was used to predict student performance.

$$p_{tj} = \sigma\left(\boldsymbol{W}^{(y)}\theta_2^{(t)} + \boldsymbol{\tau}^{(y)}\right), \tag{9}$$

where $\sigma(z_i) = 1/(1 + e^{-z_i})$, and p_{tj} represented the probability of answering $\boldsymbol{q}_j$ correctly.

$\boldsymbol{W}^{(\theta_2)}$ is the weight matrix, and $\boldsymbol{W}^{(y)}$ is the weight vector, $\boldsymbol{\tau}^{(\theta_2)}$ is the bias vector, and $\boldsymbol{\tau}^{(y)}$ is the scalar. [10]

Although the DKVMN can accurately predict performance, the interpretability of the parameters is still lacking.

3. Deep-IRT with independent student and item networks

To improve the interpretation ability of the parameters, Deep-IRT[11] was proposed by combining the DKVMN and IRT. Deep-IRT completes parameter updates based on time series, and can estimate student ability and predict difficulty, similar to IRT. For the current Deep-IRT which improves on the original Deep-IRT, Tsutsumi et al. [21] used two independent redundant networks (student network and item network) to model student responses to the project without reducing the accuracy of prediction. The student network uses a memory network architecture to reflect the dynamic changes in student abilities, similar to the DKVMN. The proposed method independently learns the characteristics of the items and students.

4. IDRT-Based Adaptive Hints

According to current research, although the Deep-IRT cannot predict optimal hints, IDRT (Item Deep Response Theory) [12] as a form of Deep-IRT that enables inferences to be made about the ability of the students and the difficulty of the items, can predict the hints needed for the students to solve a task.

The three main benefits of IDRT are as follows: (1) It can estimate ability with a high degree of accuracy although there are no common items between tests. (2) It can estimate ability with a high degree of accuracy although it cannot be assumed that the test subjects are random samples. (3) It can estimate ability with a high degree of accuracy although the group of items is not homogeneous. To apply IDRT to adaptive learning, Cai [13] proposed a new IDRT model that adds a network of independent hints and predicts the hints needed for the students to solve a task.

In the hint network, h_{ijk} represents the number of hints used by the student i when item j is correctly answered. If the answer is correct using hints from 0 to k, one-hot vector $\boldsymbol{h}_{ijk}$ with only element $h_{ij0}\dots h_{ijk}$ are 1, and other elements are 0 has been inputted. The layer-by-layer outputs are calculated according to equations (10), (11), and (12).

$$\boldsymbol{\delta}_1^{(k)} = \tanh\left(\boldsymbol{W}^{(\alpha_1)}\boldsymbol{h}_{ijk} + \boldsymbol{\tau}^{(\alpha_1)}\right), and \tag{10}$$

$$\boldsymbol{\delta}_2^{(k)} = \tanh\left(\boldsymbol{W}^{(\alpha_2)}\boldsymbol{\delta}_{\mathbf{1}}^{(\boldsymbol{k})} + \boldsymbol{\tau}^{(\alpha_2)}\right), and \tag{11}$$

$$\delta_3^{(k)} = \mathrm{relu}\left(\boldsymbol{W}^{(\alpha_3)}\boldsymbol{\delta}_{\mathbf{2}}^{(\boldsymbol{k})} + \tau^{(\alpha_3)}\right), \tag{12}$$

where $\boldsymbol{W}^{(\alpha_1)}$,$\boldsymbol{W}^{(\alpha_2)}, \boldsymbol{W}^{(\alpha_3)}$ are the weight parameter matrices, $\boldsymbol{\tau}^{(\alpha_1)}, \boldsymbol{\tau}^{(\alpha_2)}$ are the bias parameter vectors. The weight parameters $\boldsymbol{W}_{\alpha 1}, \boldsymbol{W}_{\alpha 2}, W_{\alpha 3}$, and $\mathbf{W}_{\mathrm{y}}$ were updated to maximize the prediction of the obtained response data. The output of the hint network $\delta_3^{(k)}$ is considered as the difficulty parameter of the hint k for item j of student i.

We used the difference between the student's ability parameter, item difficulty, and hint difficulty parameters to model the student's response to an item when given a hint.

Specifically, the prediction probability $\boldsymbol{p}^{(i,j,k)}$ is determined according to the following equation when student i answers item j correctly as hint k has been given

$$\boldsymbol{p}^{(i,j,k)} = (\boldsymbol{W}^{(y)})^T \left(\theta_3^{(i)} - \beta_3^{(j)} + \delta_3^{(k)}\right) + \boldsymbol{b}^{(y)}. \tag{13}$$

In addition, student's response to item j at hint k is predicted by

$$\hat{y}_{i,j}^{(c)} = softmax\left(\boldsymbol{p}^{(i,j,k)}\right) = \frac{\exp\left(p_c^{(i,j,k)}\right)}{\sum_c \exp\left(p_c^{(i,j,k)}\right)}. \tag{14}$$

However, the IDRT model cannot handle time series data, which makes it difficult to track a student's learning status over time.

Chapter 4: THE PROPOSED MODEL

Tsutsumi et al. [21] improved on the original Deep-IRT, by using two independent redundant networks (student network and item network) to model student responses to the project without reducing the accuracy of prediction. The Deep-IRT with independent student and item networks has the following benefits:

(1) Because it considers time series changes, it can represent the change in ability in the learning process.

(2) It is the most accurate method for predicting responses to unknown items based on past response history.

(3) The estimation of students' abilities is independent of item characteristics, and it is possible to express multidimensional competences in terms of multiple skills.

This chapter describes the Deep-IRT model and the proposed model.

1. Components of the Deep-IRT

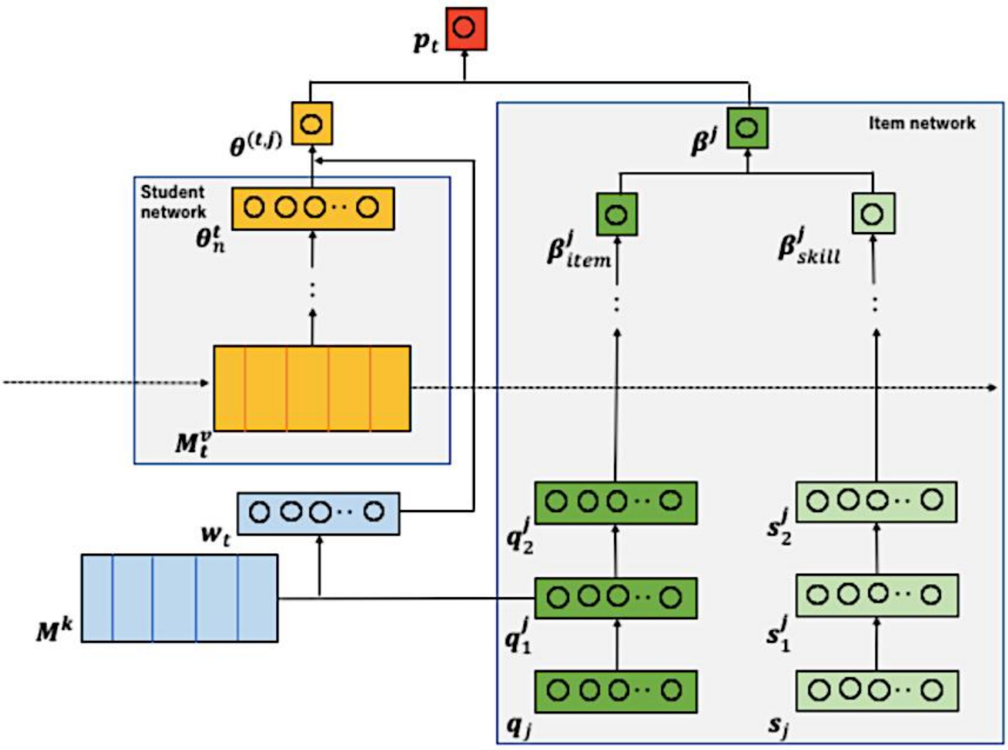

Figure 2: Deep-IRT with independent student and item networks [21]

In the item network, q_{jm} represents the input of the j-th item, as defined in equations (15). If $j = m$, the elements q_{jm} of the one-hot vector $\boldsymbol{q}_j \in R^J$ are 1, and the other elements are 0.

$$q_{jm} = \begin{cases} 1 & (j = m) \\ 0 & (otherwise) \end{cases}, \tag{15}$$

where J is the number of items.

The layer-by-layer outputs which are considered as the characteristic item difficulty parameters of item j, are calculated according to equations (16), (17), and (18).

$$\boldsymbol{q}_1^j = tanh\left(\boldsymbol{W}^{(q_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(q_1)}\right), \text{ and} \tag{16}$$

$$\boldsymbol{q}_k^j = tanh\left(\boldsymbol{W}^{(q_k)}\boldsymbol{q}_{k-1}^j + \boldsymbol{\tau}^{(q_k)}\right), \text{ and} \tag{17}$$

$$\beta_{item}^j = \boldsymbol{W}^{(q_n)}\boldsymbol{q}_n^j + \tau^{(q_n)}, \tag{18}$$

where $\boldsymbol{W}^{(q_1)}, \boldsymbol{W}^{(q_k)}, \boldsymbol{W}^{(q_n)}$ are the weight matrices; $\boldsymbol{\tau}^{(q_1)}, \boldsymbol{\tau}^{(q_k)}$ are the bias vectors.

The output of the item network β_{item}^j is considered as the characteristic difficulty parameter of item j.

s_{jm} represents the input of the j-th item requiring skill m which is defined in equations (19). If item j requires skill m, the elements s_{jm} of the one-hot vector $\boldsymbol{s}_j \in R^S$ are 1, and the other elements are 0.

$$s_{jm} = \begin{cases} 1 & (item\ j\ requires\ skill\ m) \\ 0 & (otherwise) \end{cases}, \tag{19}$$

where S is the number of skills.

The layer-by-layer outputs which are considered as the difficulty parameters of the required skills to solve item j are calculated according to equations (20), (21), and (22).

$$\boldsymbol{s}_1^j = tanh\left(\boldsymbol{W}^{(s_1)}\boldsymbol{s}_j\right), \text{ and} \tag{20}$$

$$\boldsymbol{s}_k^j = tanh\left(\boldsymbol{W}^{(s_k)}\boldsymbol{s}_{k-1}^j + \tau^{(s_k)}\right), \text{ and} \tag{21}$$

$$\beta_{skill}^j = \boldsymbol{W}^{(s_n)}\boldsymbol{s}_n^j + \tau^{(s_n)}, \tag{22}$$

where $\boldsymbol{W}^{(s_1)}, \boldsymbol{W}^{(s_k)}, \boldsymbol{W}^{(s_n)}$ are the weight matrices, $\tau^{(s_k)}, \tau^{(s_n)}$ are the bias vectors.

The last layer β_{skill}^j is considered as the difficulty parameter of the required skills to solve item j.
Item difficulty β^j is calculated from the sum of the two difficulty parameters β_{item}^j and β_{skill}^j as

$$\beta^j = \beta_{item}^j + \beta_{skill}^j\ . \tag{23}$$

In the student network, θ_1^t which is calculated based on the past response history and independently from item j according to equation (24).

$$\boldsymbol{\theta}_1^t = \sum_{i=1}^{N} \boldsymbol{M}_{t,i}^v, \tag{24}$$

where M_t^v is memory matrix that stores and updates students' understanding of each knowledge state.

θ_n^t which is the student's ability vector, is estimated according to equation (25). n is the number of hidden layers and it is determined by the prediction accuracy of the actual data.

$$\boldsymbol{\theta}_k^t = \tanh\left(W^{(\theta_k)}\boldsymbol{\theta}_{k-1}^t + \tau^{(\theta_k)}\right)\ (k = 2, \cdots, n), \tag{25}$$

where $\boldsymbol{W}^{(\theta_k)}$ is the weight matrix and $\boldsymbol{\tau}^{(\theta_k)}$ is the bias vector.

The last layer $\boldsymbol{\theta}_n^t$ is the parameter vector of a student's ability, which is not independent of each item.

To estimate the correct response probability prediction, the proposed model sums up the student's ability vector. An attention vector $\boldsymbol{w}_j$ is calculated according to equation (26).

$$\boldsymbol{w}_j = Softmax\left(\boldsymbol{M}^k \boldsymbol{q}_1^{(j)}\right), \text{ and} \tag{26}$$

$$\theta^{(t,j)} = \boldsymbol{w}_j^T \boldsymbol{\theta}_n^t, \tag{27}$$

where $\boldsymbol{M}^k$ is a key memory matrix that holds the strength of the relationship between each item and skill. $\theta^{(t,j)}$ is the ability of the student to correctly answer item j at time t. Specifically, the predicted probability p_{tj} is determined according to equation (28) when the student answers item j correctly at time t.

$$p_{tj} = \sigma\left(\theta^{(t,j)} - \beta^j\right). \tag{28}$$

2. Components of the Proposal Model

The previous Deep-IRT method estimates the changes in a student's ability under multidimensional skills and predicts the correct answer probability for the task. However, the methods do not support items with hints, which limits their applicability to learning support. Therefore, we propose a novel Deep-IRT method for predicting optimal hints. To apply Deep-IRT to adaptive learning, we propose a new Deep-IRT model that adds an independent hint network and predicts the hints for the students to solve a task.

The proposed model has the following benefits:

(1) It can track the learning status of students based on time series; therefore, it is possible to present hints over time.

(2) It is possible to estimate the ability with a high degree of accuracy to predict the response to unknown tasks based on the past response history.

(3) It is possible to predict the optimal hint for each item based on the past response history.

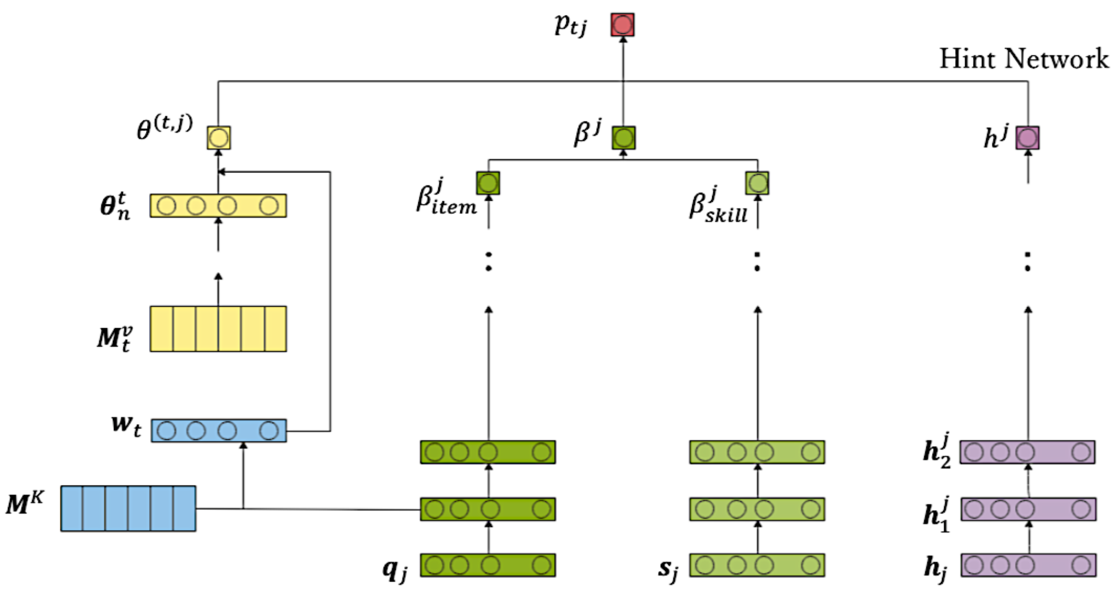

Figure 3: Network architecture of the Deep-IRT hint network

3. Hint Network

In the hint network, h_{jh} represents the input of the j-th item which hint h has been given, as defined in equations (29). If item j presents hint h, the elements h_{jh} of the one-hot vector $\boldsymbol{h}_j \in R^H$ are 1, and the other elements are 0.

$$h_{jh} = \begin{cases} 1 & (item\ j\ presents\ hint\ h) \\ 0 & (otherwise) \end{cases}, \tag{29}$$

where H is the number of hints.

The layer-by-layer outputs are calculated according to equations (30), (31), and (32).

$$\boldsymbol{h}_1^j = tanh\left(\boldsymbol{W}^{(h_1)}\boldsymbol{h}_j + \boldsymbol{\tau}^{(h_1)}\right), and \tag{30}$$

$$\boldsymbol{h}_k^j = tanh\left(\boldsymbol{W}^{(h_k)}\boldsymbol{h}_{k-1}^j + \boldsymbol{\tau}^{(h_k)}\right), and \tag{31}$$

$$h^j = relu\left(\boldsymbol{W}^{(h_n)}\boldsymbol{h}_k^j + \tau^{(h_n)}\right), \tag{32}$$

where $\boldsymbol{W}^{(h_1)}$,$\boldsymbol{W}^{(h_k)}, \boldsymbol{W}^{(h_n)}$ are the weight parameter matrices; $\boldsymbol{\tau}^{(h_1)}, \boldsymbol{\tau}^{(h_2)}$ are the bias parameter vectors. Weight parameters $\boldsymbol{W}_{h1}, \boldsymbol{W}_{h2}, W_{h3}$ are updated to maximize the prediction of the obtained response data. The output of the hint network h^j is considered as the parameter of hint for item j. Specifically, the predicted probability p_{tj} is determined according to equation (33), which predicts a student's response probability to an item using the difference between the student's ability to solve item j at time t $\theta^{(t,j)}$, item difficulty β^j and the hint parameter h^j.

$$p_{tj} = (\boldsymbol{W}^{(y)})^T\left(\theta^{(t,j)} - \beta^j + h^j\right) + \boldsymbol{b}^{(y)}. \tag{33}$$

We use a backpropagation algorithm for deep learning models to learn their parameters by minimizing the loss function. The loss function of the proposed model employs cross-entropy, which reflects the classification errors. The predicted responses p_{tj} and the true responses u_t are calculated by

$$\ell\left(u_t, p_{tj}\right) = -\sum_t\left(u_t \log p_{tj} + (1-u_t)\log\left(1-p_{tj}\right)\right). \tag{34}$$

Chapter 5: EXPERIMENTS

1. Actual data

In this study, we use the existing IDRT method and the proposed model to estimate the hint usage status and ability of students from the learning data and to predict their responses to the tasks. To estimate the parameters, this study used the system developed by Ueno [14], [15], [16], [17] to collect the response data 75 university students, who were beginners in programming, for 18 programming learning tasks. The students learned the grammar of "four arithmetic operations on variables", "conditional branching while loops", "for loops", "arrays", and "functions and method calls", and answered the corresponding tasks after learning each domain. However, there are four questions each for "four arithmetic operations on variables", "conditional branching while loop" and "for loop", and three questions for "array" and "function/method call".

Table 1 lists the number of hints for each task and the accuracy without hints for each task.

Table 1. Number of hints and Accuracy without hints for each task

	Task 1	Task 2	Task 3	Task 4	Task 5
Number of hints	8	8	8	9	10
Accuracy(%)	60%	66.7%	65.3%	46.7%	50.6%
	Task 6	Task 7	Task 8	Task 9	Task 10
Number of hints	11	9	8	13	12
Accuracy(%)	46.7%	54.7%	54.7%	50.6%	53.7%
	Task 11	Task 12	Task 13	Task 14	Task 15
Number of hints	12	13	17	10	11
Accuracy(%)	48%	80%	81.3%	86.7%	49.3%
	Task 16	Task 17	Task 18		
Number of hints	6	9	8		
Accuracy(%)	94.7%	80%	49.3%		

2. Experimental method

The experiments are conducted in the same experimental environment as in the existing IDRT method. We select a 10-fold cross-validation method with 90% of the experimental data as the training set and 10% of the experimental data as the prediction set under the same experimental conditions similar to those of the existing IDRT method.

To predict the optimum hints the students will need in the next task, we calculate the consistency rate with the hints that the students actually used to get the correct answer.

Using the actual hints x_{ij} used by student i in answering task j and the predict hints $\hat{x}_{ij}$, we calculate the consistency rate c_j for each task j as

$$c_j = \frac{1}{I}\sum_{i=1}^{I} \psi(\hat{x}_{ij}, x_{ij}), \tag{35}$$

where $\psi(\hat{x}_{ij}, x_{ij})$ is a function which takes the value 1 when x_{ij} and $\hat{x}_{ij}$ are identical, otherwise, takes the value 0.

The percentage of consistency is averaged over all tasks and the prediction accuracy of the proposed model c is given by the equation(36).

$$c = \frac{1}{M-1}\sum_{m=2}^{M} c_m \, . \tag{36}$$

The predict hints $\hat{h}_{ij}$ for task j is compared with the actual hints h_{ij}, and the extra-prediction and missing-prediction rates are calculated for all students using equations (37) and (38).

$$e_j = \frac{1}{I}\sum_{i=1}^{I} \nabla(\hat{h}_{ij}, h_{ij}) \, , and \tag{37}$$

$$m_j = \frac{1}{I}\sum_{i=1}^{I} \Delta\left(\hat{h}_{ij}, h_{ij}\right), \tag{38}$$

where $\nabla\left(\hat{h}_{ij}, h_{ij}\right)$ is a function which takes the value 1 when h_{ij} than $\hat{h}_{ij}$, otherwise, takes the value 0.

where $\Delta\left(\hat{h}_{ij}, h_{ij}\right)$ is a function which takes the value 1 when h_{ij} than $\hat{h}_{ij}$, otherwise, takes the value 0.

3. Comparison of the accuracies in predicting responses to an unknown task

We analyze the response prediction accuracy c for the proposed model, the existing method IDRT model and the existing method IRT model on the training data. The experimental results are shown in Table 2.

Table 2. Prediction rates for each task

	Task1	Task2	Task3	Task4	Task5	Task6	Task7
proposed model	75%	37.5%	75%	37.5%	75%	50%	50%
IDRT model	50%	62.5%	62.5%	37.5%	75%	62.5%	87.5%
IRT model	---	34.2%	60.3%	46.3%	45.4%	64.8%	52.8%
	Task8	Task9	Task10	Task11	Task12	Task13	Task14
proposed model	50%	62.5%	87.5%	100%	100%	100%	62.5%
IDRT model	50%	75%	62.5%	62.5%	100%	50%	100%
IRT model	52.7%	46.7%	56%	40.2%	67%	83%	87%
	Task15	Task16	Task17	Task18	Average		
proposed model	75%	100%	87.5%	75%	**72.9%**		
IDRT model	75%	100%	87.5%	50%	69.4%		
IRT model	49.7%	85.3%	90.7%	69%	60.3%		

The accuracy (Accuracy) represents the prediction accuracy c of the response obtained in the previous section, refer to equations (35) and (36). We also show the percentage of correct responses of all students for each training data. Table 2 shows that the proposed method predicts the task more accurately than the existing IDRT method and the existing IRT method. In other words, the proposed method is able to predict students' performance more accurately than the existing IDRT method and the existing IRT method.

4. Error analysis

In order to analyze the error in the hints predicted by the proposed method and the existing IDRT method, we used the equations (37) and (38) described in the previous section to calculate the error in the hints predicted by each method. The results are shown in Table 3.

Table 3. Extra-prediction rate and missing-prediction rate for the used hints

		Task1	Task2	Task3	Task4	Task5
proposed model	extra	12.5%	12.5%	25.0%	12.5%	12.5%
	missing	12.5%	50.0%	0.0%	50.0%	12.5%
IDRT model	extra	50.0%	37.5%	37.5%	62.5%	25.0%
	missing	0.0%	0.0%	0.0%	0.0%	0.0%
IRT model	extra	---	29.3%	15.1%	8.7%	13.7%
	missing	---	30.1%	25.1%	45.9%	41.3%
		Task6	Task7	Task8	Task9	Task10
proposed model	extra	12.5%	25.0%	25.0%	25.0%	12.5%
	missing	25.0%	25.0%	25.0%	12.5%	0.0%
IDRT model	extra	37.5%	12.5%	50.0%	12.5%	37.5%
	missing	0.0%	0.0%	0.0%	12.5%	0.0%
IRT model	extra	14.1%	8.3%	18.3%	15.5%	7.5%
	missing	19.7%	39.3%	29.2%	37.6%	37.9%
		Task11	Task12	Task13	Task14	Task15
proposed model	extra	0.0%	0.0%	0.0%	25.0%	25.0%
	missing	0.0%	0.0%	0.0%	12.5%	12.5%
IDRT model	extra	37.5%	0.0%	50.0%	0.0%	25.0%
	missing	0.0%	0.0%	0.0%	0.0%	0.0%
IRT model	extra	21.9%	20.7%	3.3%	4.0%	3.5%
	missing	37.9%	11.9%	13.7%	8.4%	46.8%
		Task16	Task17	Task18	Average	
proposed model	extra	0.0%	12.5%	25.0%	14.6%	
	missing	12.5%	0.0%	0.0%	13.9%	
IDRT model	extra	0.0%	12.5%	50.0%	27.1%	
	missing	0.0%	0.0%	0.0%	0.7%	
IRT model	extra	12.9%	2.7%	0.4%	11.7%	
	missing	2.1%	6.7%	30.5%	27.3%	

The existing IDRT method always over-predicts the hints compared to proposed method. The existing IDRT method underestimates the students, thus the students to overlearn. Besides, the existing IRT method always under-predicts the hints compared to proposed method. The existing IRT method overestimates the students.

Therefore, in contrast to the IDRT model and the IRT model, the proposed model is more effective for adaptive learning than the previous methods.

Chapter 6: CONCLUSIONS

Adaptive learning has been studied actively in the field of artificial intelligence to help students by providing optimal hints when they answer a task incorrectly.

In this study, we proposed a novel Deep-IRT method for predicting the optimal hints by adding a hint network to the Deep-IRT. Specifically, we extended the idea of [11], by applying Deep-IRT to adaptive learning. Therefore, we propose a novel Deep-IRT method for predicting optimal hints by adding a hint network. We established that the proposed method improved the prediction accuracy of the students' performance with hints.

In the evaluation experiment, we compared the prediction accuracy of the existing method and the proposed method, and we established that the proposed method improved the prediction accuracy of the existing method. From actual data experiments, it is found that the proposed method has the following advantages.

1. The proposed method can handle time series data; therefore, it can present the optimum hints to a student according to his/her ability change.

2. The proposed method provided the most accurate prediction of the optimum hints.

With the current rapid development in educational engineering and adaptive learning, applying the proposed model to adaptive learning system is an important future task.

ACKNOWLEDGEMENT

It is a great honor to be able to study at the University of Electro-Communications as a master. During these three years, I have learned not only knowledge, but also a rigorous attitude towards research and a resilient spirit when facing difficulties.

I am very grateful to Prof. Ueno for all the encouragement and guidance which he has given me over the years. When I made mistakes, I was still given the opportunity to correct them.

I would also like to thank Prof. Kawano for his guidance and advice on writing my thesis. I would like to thank Ms. Tsutsumi for her advice on research content and thesis writing. I would like to thank Mr. Guo for his help in programming and clarifying ideas.

I would like to thank my mother and father. They gave me encouragement and love during my toughest time. They still believed in me when I was having a hard time and worked with me through it.

Last but not least, I would also like to thank myself for not giving up in my last year. The future is not always a straightforward one, and I can always be the light that shines on me.

REFERENCES

[1] https://www.digital-knowledge.co.jp/product/edu-ai/adaptive/

[2] D. Wood, “Scaffolding contingent tutoring and computer-supported learning”, Int. J. Artificial Intelligence in Education, pp.280–292, 2001.

[3] M. Ueno and J. Matsuo, “A scaffolding system for presenting adaptive hints with IRT”, IEICE(D),vol.J98-D, no.1, pp.17–29, 2015. (in Japanese)

[4] M. Ueno and Y. Miyazawa, “Probability based scaffolding system with fading”, Artificial Intelligence in Education - 17th International Conference, AIED 2015, pp.492–503, 2015.

[5] M. Ueno and Y. Miyazawa, “IRT- Based Adaptive Hints to Scaffold Learning in Programming”, IEEE Transactions on Learning Technologies, IEEE computer Society, Vol. 11, Issue 4, pp.415- 428, 2018.

[6] F. M. Lord, “Applications of Item Response Theory to Practical Testing Problems”, Mahwah, NJ, USA: Lawrence Erlbaum Associates, 1980.

[7] F. B. Baker and S. Kim, “Item Response Theory: Parameter Estimation Techniques”, 2nd ed. New York, NY, USA: Marcel Dekker, 2004.

[8] M. Ueno, “E-Learning in the knowledge society”, Baifukan, 2007. (in Japanese)

[9] M. Ueno and K. Nagaoka, “E-Testing”, Baifukan, 2009. (in Japanese)

[10] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, “Dynamic key-value memory networks for knowledge tracing”, *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW ’17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, pp. 765–774, 2017.

[11] C. Yeung, “Deep-irt: Make deep learning based knowledge tracing explainable using item response theory”, *Proceedings of the 12th* [36] *International Conference on Educational Data Mining*, EDM, 2019.

[12] E. Tsutsumi, R. Kinoshita, and M. Ueno, “Deep Item Response Theory as a Novel Test Theory Based on Deep Learning”, electronics, Vol.10, Issue.9, no.1020, 2021.

[13] R. Cai, “Deep learning hint prediction model for adaptive learning”, Information Science and Technology, University of Electro-Communications, 2021. (in Japanese)

[14] A. Tanizawa and Y. Honda, “E-Testing for the Information Technology Engineer Exam”, JART 12th, Vol. 33, pp. 54–57, 2014. (in Japanese)

[15] M. Ueno, “Animated agent to maintain learner’s attention in e-learning”, Proceedings of E-Learn: World Conference on E-Learning in Corporate,

Government, Healthcare, and Higher Education 2004, eds. by J. Nall and R. Robson, Association for the Advancement of Computing in Education (AACE), Washington, pp. 194–201, 2004.

[16] M. Ueno, “Data mining and text mining technologies for collaborative learning in an ilms "samurai"”, ICALT’ 04 Proceedings of the IEEE International Conference on Advanced Learning Technologies, pp. 1052–1053, 2004.

[17] M. Ueno, “Intelligent with an agent that learns from log data”, Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005, ed. by G. Richards, pp. 3169–3176, 2005.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, arXiv: 1412. 6980, 2014.

[19] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing”, Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. pp. 505–513, 2015.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] E. Tsutsumi, R. Kinoshita, and M. Ueno, “Deep-IRT with independent student and item networks”, International Conference on Educational Data Mining, EDM, 2021.

[22] E. Tsutsumi, M. Uto and M. Ueno, “Hidden Markov IRT model for dynamic assessment”, IEICE(D), vol.J102-D, no.2, pp.79–92, 2019. (in Japanese)