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	CAO YUCHEN	学籍番号	1931070
論 文 題 目	Deep-IRT incorporating Bayesian parameters estimation ベイズ母数推定を組み込んだ Deep-IRT		

要　　旨

テスト理論分野では，学習者のテスト(課題)への反応を基に，学習者の能力値を高精度に推定することが課題となっている。近年では，学習者の能力値を正しく推定するために，従来からテスト理論分野で用いられている項目反応理論（Item Response Theory: IRT）に深層学習手法を組み合わせた Deep-IRT が開発されている．既存研究では Deep-IRT は IRT より学習者の能力値を高精度に推定することが示されている。

しかし、Deep-IRT はデータ数が少ない場合に学習データに過学習してしまう問題がある。本論文では、少数データにおける過学習を避けるためにベイズ母数推定を組み込んだ Deep-IRT を提案する。提案手法ではニューラルネットワークにおける重みとバイアスパラメータを変分推定法を用いてベイズ推定することでパラメータの過学習を避けることができる。

評価実験では少数データにおいて提案手法が既存手法よりも学習者の能力値を正しく推定することを示した。さらに，提案手法は学習者の課題への反応を高精度に予測することを示した。

Deep-IRT incorporating Bayesian parameters estimation

電気通信大学大学院 情報理工学研究科

情報・ネットワーク工学専攻 情報数理工学プログラム

学籍番号　1931070

曹　宇塵　ソウ　ウチリ Yuchen Cao

主任指導教員 植野 真臣 教授

指導教員 川野 秀一 准教授

cyc19952019@gmail.com

Index

Abstract

Item Response Theory (IRT) [van der Linden et.al (2013)] is a test theory which enables to evaluate examinees who take different tests on the same scale. However, IRT assumes randomly sampling examinees' abilities from a statistical distribution. When actual examinees' abilities do not follow the distribution, the estimation accuracies of abilities tend to decrease significantly. To resolve this problem, Tsutsumi et.al (2021) proposed Deep-IRT which enables to estimate examinees' abilities without the assumption. However, the deep-learning-based methods tend to overfit the training data when the sample size is small. This study proposes a new Deep-IRT model, which incorporates Bayesian neural network into the final layer in the Deep-IRT model. Bayesian neural networks (BNN) is a method to improve the accuracies of estimates in deep learning, by mitigating the overfitting problem. To predict examinees' abilities, the proposed method employs the variational inference method for Bayesian inference. The proposed model is expected to have more accurate prediction than the Deep-IRT model does by mitigating the overfitting problem. Experiments show that the proposed model improves the prediction performances of the Deep-IRT model, while it provides interpretability for both students and items, because the proposed model mitigates the overfitting in learning the parameters.

Keywords ： Item Response Theory, Deep Learning, Bayesian neural networks

Chapter 1 Introduction

E-testing provides automatic assemblies of uniform test forms, for which each form comprises a different set of items but still has equivalent measurement accuracy [1–14].

Examinees’ test scores should be guaranteed to become equivalent, even if other examinees with the same ability take various tests. However, it isn't easy to develop perfectly consistent test forms, and the calibration process is fundamentally important when multiple test forms are used. Item Response Theory (IRT) [15] is a calibration method to solve this difficulty. Especially, IRT has been used widely along with the widespread use of computer-based testing. IRT offers the following benefits [16,17]: IRT can estimate examinee abilities by minimizing the effects of heterogeneous or aberrant items with low estimation accuracy. It can also assess the examinees’ responses to different items on the same scale. An individual examinee’s correct response probability to an item from the examinee’s past response histories can be predicted by IRT.

Evaluating examinee abilities on the same scale requires to estimate examinees' abilities on the same scale from different tests [18 –20]. For this purpose, IRT assumes that examinees' abilities are sampled from a normal distribution randomly. This assumption might sometimes be too strict for actual data [17]. Nevertheless, it requires much labor to design.

Previous studies proposed a test theory based on deep learning, Deep-IRT [13,14], which requires no assumption of a random sampling of examinee abilities from a statistical distribution. The Deep-IRT model represents an examinee's probability of answering an item correctly based on the examinee’s ability parameter and the item’s difficulty parameter. The main contributions of the Deep-IRT are as follows:

1. The Deep-IRT does not assume random sampling of examinees.

2. The Deep-IRT method provides more reliable and robust ability estimation for actual data than IRT does.

3. The Deep-IRT method predicts examinee responses to unknown items based on the examinee’s past response histories more accurately than IRT does.

However, Deep-IRT has the following problems.

1. The Deep-IRT model is a deep-learning-based model. Training data come from actual examinations, and they are usually too sparse for networks to clearly capture the features of examinees’ abilities. It tends to overfit easily to data.

2. The inputs of the Deep-IRT model are one-hot vectors, and the outputs of abilities and difficulties are calculated by back-propagation. Because the weights of ability are not assumed to have prior distributions, the estimation of the parameters tends to overfit to the training data.

On the other hand, for Knowledge Tracing [21 –24], Deep Knowledge Tracing (DKT) [25] has been proposed. DKT predicts the examinees’ performances. It can capture more complex representations of examinees’ knowledge components. However, DKT might also cause overfitting for small datasets. The current deep knowledge tracing with a simple RNN applies a back-propagation algorithm and batch gradient descent to adjust parameters, accessible to overfitting and prone to gradient disappearance and gradient explosion for long-dependent data. To solve these problems， Li, et. al, (2019) proposed

Bayesian Deep Knowledge Tracing (BDKT) [26], which corporates DKT and Bayesian neural networks.

Bayesian neural networks (BNN) [27,28] provides probabilistic interpretations of deep learning models by introducing model weights distributions. The model offers robustness for overfitting, uncertainty estimation, and ease of learning on small datasets. For the BDKT model, a Bayesian neural network is applied to examinees' behavior analysis and knowledge tracing. The results demonstrated that BNN improved the prediction performances of DKT.

This method can also be used in the field of E-testing. BNN provides posterior distributions of weights and biases given the training data. The Bayesian estimation avoids overfitting to data and then improves the parameter estimation accuracy.

This study proposes a new Deep-IRT model incorporating Bayesian parameters estimation. Unlike BDKT, which aims to trace examinees' knowledge states through time, the proposed model aims to predict examinees' abilities in E-testing.

This study implements the proposed model using Tensorflow. Two experiments compare the performances of the proposed model and the Deep-IRT model. Experiment 1 compares the estimation accuracies of the abilities of the two models from simulation datasets. Experiment 2 compares the prediction accuracies of unknown responses for the two models from actual datasets.

Chapter 2 Previous Study

2.1 Item Response Theory

Item Response Theory (IRT) is a standard framework to predict an examinee's probability of a correct answer to an item [15,29]. IRT is essentially a structured logistic regression to an examinee's probability of a correct answer to an item from the difference between the examinee’s ability and the item's difficulty. It is assumed that an examinee’s ability does not change during the examination.

This section introduces the two-parameter logistic model (2PLM), which is the most popular IRT model [15].

For the two-parameter logistic model, u_{ij} denotes the response of examinee i to item $j\ (1, \ldots, n)$ as

$$u_{ij} = \begin{cases} 1\ (examinee\ i\ answers\ correctly\ to\ item\ j) \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad (otherwise) \end{cases}.$$

$P_j(\theta_i)$ denotes the probability that an examinee i answers correctly to an item (question) j. $\theta_i \in (-\infty, \infty)$ represents i-th examinee's ability. This possibility is defined by the item response function by the difference of an examinee's ability level and an item's difficulty level,

Specifically, in the two-parameter logistic model, the following logistic function is used as an item response function:

$$P_j(\theta_i) = P\left(u_{ij} = 1 \mid \theta_i\right) = \frac{1}{1+\exp\left(-1.7a_j(\theta_i - b_j)\right)}\ , \tag{2.1}$$

where $a_j \in (0, \infty)$ is the j-th item’s discrimination parameter, and $b_j \in (-\infty, \infty)$ is the j-th item’s difficulty parameter. The examinees’ abilities are assumed to be sampled randomly from an examinees’ ability distribution.

Because it is difficult to estimate the parameters analytically, numerical calculation methods such as Markov Chain Monte Carlo methods (MCMC) are generally used to calculate the parameters. However, since the IRT model was initially designed to be used in educational testing environments, the model assumes that the examinees' ability does not change during the test.

IRT models assume randomly sampling examinees’ abilities from a statistical distribution. If actual examinees' abilities do not follow the distribution, the estimation accuracies of abilities tend to decrease significantly.

2.2 Deep-IRT

In order to solve the problem mentioned above, E. Tsutsumi, R. Kinoshita, and M. Ueno proposed Deep-IRT [13], which does not assume randomly sampling examinees' abilities from a statistical distribution. The Deep-IRT method is expected to estimate examinees' abilities more reliably and robustly than IRT. The Deep-IRT model combines two independent neural networks, an Examinee network and an Item network. Using the outputs of both networks, the probability of an examinee answering an item correctly is calculated. The structure of the Deep-IRT model is shown in Figure 1.

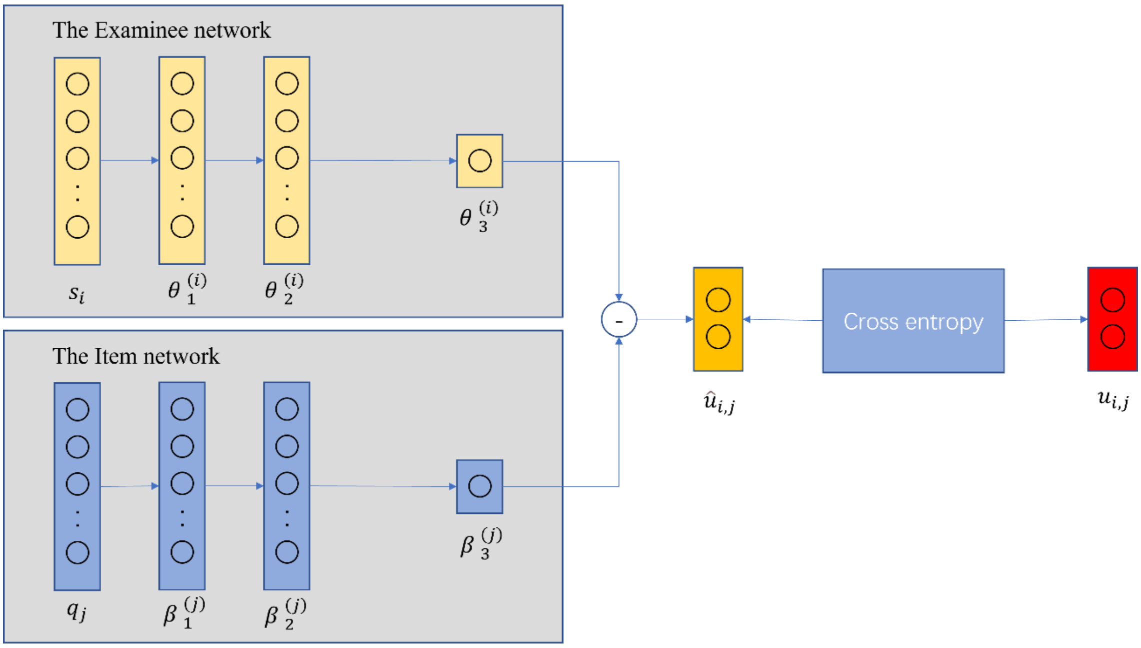


Figure 1 the structure of the Deep-IRT [23]

2.2.1 The Examinee network

To represent the *i-th* examinee in the examinee network, this study takes the one-hot-vector $s_i \in \{0,1\}^I$, where only the *i-th* element is one, and the other elements are 0, as input, and calculate the output for each layer as in equations (2.2), (2.3), and (2.4),

$$\theta_1^{(i)} = \tanh\left(\boldsymbol{W}^{(\theta_1)} s_i + \tau^{(\theta_1)}\right) \quad , \tag{2.2}$$

$$\theta_2^{(i)} = \tanh\left(\boldsymbol{W}^{(\theta_2)}\, \theta_1^{(i)} + \tau^{(\theta_2)}\right) \quad , \tag{2.3}$$

and
$$\theta_3^{(i)} = \boldsymbol{W}^{(\theta_3)}\, \theta_2^{(i)} + \tau^{(\theta_3)} \quad , \tag{2.4}$$

where tanh is the activation function and is calculated as

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad , \tag{2.5}$$

$\boldsymbol{W}^{(\theta_1)}$ and $\boldsymbol{W}^{(\theta_2)}$ represent the weight parameter matrices as

$$\boldsymbol{W}^{(\theta_1)} = \begin{pmatrix} w_{11}^{(\theta_1)} & \cdots & w_{1I}^{(\theta_1)} \\ \vdots & \ddots & \vdots \\ w_{|\theta_1|1}^{(\theta_1)} & \cdots & w_{|\theta_1|I}^{(\theta_1)} \end{pmatrix},$$

$$\boldsymbol{W}^{(\theta_2)} = \begin{pmatrix} w_{11}^{(\theta_2)} & \cdots & w_{1|\theta_1|}^{(\theta_2)} \\ \vdots & \ddots & \vdots \\ w_{|\theta_2|1}^{(\theta_2)} & \cdots & w_{|\theta_2||\theta_1|}^{(\theta_2)} \end{pmatrix},$$

$\boldsymbol{W}^{(\theta_3)}$ represents the weight parameter vector as

$$\boldsymbol{W}^{(\theta_3)T} = \begin{pmatrix} w_1^{(\theta_3)} \\ \vdots \\ w_{|\theta_2|}^{(\theta_3)} \end{pmatrix},$$

$\tau^{(\theta_1)}$ and $\tau^{(\theta_2)}$ represent the bias parameter vectors and $\tau^{(\theta_3)}$ is the bias parameter.

The weight parameters in $\boldsymbol{W}^{(\theta_1)}$, $\boldsymbol{W}^{(\theta_2)}$ and $\boldsymbol{W}^{(\theta_3)}$ are updated so as to maximize the model fitting to the data, and the output $\theta_3^{(i)}$ of the examinee network is regarded as the ability parameter of examinee i. In the Deep-IRT model, all weight parameters are updated when new response data are obtained, without an assumption of independence among examinee parameters.

2.2.2 The Item network

To represent the j-th item in the item network, this study employs the one-hot vector $q_j \in \{0,1\}^J$, in which only the j-th element is one and the other parts are 0, as the input, and calculate the output for each layer as in Equations (2.6), (2.7), and (2.8),

$$\beta_1^{(j)} = \tanh(\boldsymbol{W}^{(\beta_1)} q_j + \tau^{(\beta_1)}) \quad , \tag{2.6}$$

$$\beta_2^{(j)} = \tanh\left(\boldsymbol{W}^{(\beta_2)} \beta_1^{(j)} + \tau^{(\beta_2)}\right) \quad , \tag{2.7}$$

and
$$\beta_3^{(j)} = \boldsymbol{W}^{(\beta_3)} \beta_2^{(j)} + \tau^{(\beta_3)} \quad , \tag{2.8}$$

where $\boldsymbol{W}^{(\beta_1)}$, $\boldsymbol{W}^{(\beta_2)}$ are the weight parameter matrices shown as

$$\boldsymbol{W}^{(\beta_1)} = \begin{pmatrix} w_{11}^{(\beta_1)} & \cdots & w_{1J}^{(\beta_1)} \\ \vdots & \ddots & \vdots \\ w_{|\beta_1|1}^{(\beta_1)} & \cdots & w_{|\beta_1|J}^{(\beta_1)} \end{pmatrix},$$

$$\boldsymbol{W}^{(\beta_2)} = \begin{pmatrix} w_{11}^{(\beta_2)} & \cdots & w_{1|\beta_1|}^{(\beta_2)} \\ \vdots & \ddots & \vdots \\ w_{|\beta_2|1}^{(\beta_2)} & \cdots & w_{|\beta_2||\beta_1|}^{(\beta_2)} \end{pmatrix},$$

$\boldsymbol{W}^{(\beta_3)}$ is the weight parameter vector shown as

$$\boldsymbol{W}^{(\beta_3)T} = \begin{pmatrix} w_1^{(\beta_3)} \\ \vdots \\ w_{|\beta_2|}^{(\beta_3)} \end{pmatrix},$$

$\tau^{(\beta_1)}$ and $\tau^{(\beta_2)}$ are the bias parameter vectors, and $\tau^{(\beta_3)}$ is the bias parameter. The weight parameters in $\boldsymbol{W}^{(\beta_1)}$, $\boldsymbol{W}^{(\beta_2)}$ and $\boldsymbol{W}^{(\beta_3)}$ are updated so as to fit the obtained response data, and the output of the item network is calculated by $\beta_3^{(j)}$ alone via the weight parameters. The output of the network $\beta_3^{(j)}$ in the Deep-IRT model is interpreted as the difficulty parameter of item *j*. In the Deep-IRT model, all the item difficulty parameters are dependent one another.

2.2.3 The output of the Deep-IRT model

The difference between an examinee’s ability parameter and an item's difficulty latent variable parameter predicts the examinee's correct response probability to the item. Specifically, the response of examinee *i* to item *j*, the hidden layer $h^{(i,j)} = (h_0^{(i,j)},\ h_1^{(i,j)})$, is represented by

$$h^{(i,j)} = \left(\boldsymbol{W}^{(y)}\right)^T \left(\theta_3^{(i)} - \beta_3^{(j)}\right) + \tau^{(y)} \qquad , \qquad (2.9)$$

where $\boldsymbol{W}^{(y)}$ is the weight parameter vector, and the $\tau^{(y)}$ is the bias parameter vector. The weight parameters in $\boldsymbol{W}^{(y)}$ are updated so as to maximize the model fitting to the data.

The correct response probability of examinee i to item j is obtained by

$$\hat{u}_{i,j} = softmax\left(h^{(i,j)}\right) = \frac{\exp\left(h_1^{(i,j)}\right)}{\exp\left(h_0^{(i,j)}\right)+\exp\left(h_1^{(i,j)}\right)} \quad . \tag{2.10}$$

Deep-IRT uses a deep learning method to estimate the relationship between an examinees' ability and all the other examinees' abilities so as to maximize the model fitting to the data. The unique feature of this method is to estimate an examinee's ability by adjusting the other examinees' ability estimates.

In general, deep-learning-based models learn their parameters using the back-propagation algorithm by minimizing a loss function. Because the Deep-IRT model is a classification model, the cross-entropy is often used as a loss function. It is possible to calculate the predicted responses and the actual responses as

$$\text{cross entropy} = -u_{ij}\log\hat{u}_{ij} - (1-u_{ij})\log(1-\hat{u}_{ij}) \quad , \tag{2.11}$$

where u_{ij} are the actual responses and the $\hat{u}_{ij}$ are the predicted responses.

However, the Deep-IRT model is deep-learning-based. Overfitting is an unavoidable problem in standard deep learning models.

To mitigate the overfitting problem of the Deep-IRT, the next chapter will propose a new model incorporating Bayesian estimation of an examinee's ability and an item difficulty into Deep-IRT.

Chapter 3 The proposed model

3.1 Bayesian neural networks

Bayesian neural networks (BNNs) apply Bayesian learning to deep learning. It realizes the combination of probabilistic programming and deep learning, bringing massive innovation to deep learning [26,27].

For BNNs, we can use the following Bayesian methods,

$$\text{Input data:} \quad D = \{x, y\} \ ,$$

$$\text{Prior:} \quad p(w) \ ,$$

$$\text{Posterior:} \quad p(w|D) = \frac{p(w)p(D|w)}{p(D)} \ ,$$

$$\text{and} \quad \text{Prediction:} \quad p(\hat{y}|D) \ = \int p(\hat{y}|w)p(w|D)dw, \tag{3.1}$$

where D is the dataset, and it is made up of x and y. x is the input of the neural network. y is the label data. w is the weights vector or matrix in neural networks. $p(w)$ is the prior over the weights vector or matrix. This study estimates the posterior $p(w\,|\,D)$, which assumes the posterior distribution of the weights vector or matrix influenced by the dataset. $\hat{y}$ is the prediction or the output of the neural network. It is possible to calculate the gradient descent by comparing y and $\hat{y}$ and update parameters.

It is difficult to analytically calculate the posterior $p(w\,|\,D)$. Therefore, it needs to assume a variational posterior $Q(w; \xi)$ to approximate $p(w \mid D)$. This study employs the variational inference method. $Q(w; \xi)$ is the variational posterior parameterized by parameters ξ.

Peterson (1987) [30] and Hinton & Van Camp (1993) [31] firstly applied variational inference to neural networks. Variational reasoning uses optimization instead of Bayesian modeling marginalization. Namely, a derivative is used instead of an integral calculation. In contrast to the optimization methods often used in deep learning, in this case, distributions of the weights are estimated instead of estimating the points. This method retains advantages of Bayesian modeling (such as the balance between a complex model and a model that can explain the data satisfactorily). It leads to a probabilistic model that captures the uncertainty of the model.

The approximated distribution is estimated to be as close as possible to the posterior distribution obtained from the original model. Therefore, this study minimizes the Kullback-Leibler (KL) divergence [27] to measure the distance between the variational posterior $Q(w; \xi)$ and the Posterior $p(w \mid D)$ as

$$KL\big(Q(w; \xi)||p(w|D)\big) = \int Q(w; \xi) \log \frac{Q(w;\xi)}{p(w|D)} dw \quad . \tag{3.2}$$

Because the KL divergence is still difficult to be calculated, it is transformed as follows

$$\int Q(w; \xi) \log \frac{Q(w; \xi)}{p(w|D)} dw = -\int Q(w; \xi) \log \frac{p(w|D)}{Q(w; \xi)} dw$$

$$= -\left(\int Q(w; \xi) \log \frac{p(D,w)}{Q(w;\xi)} dw - \int Q(w; \xi) \log p(D) dw\right)$$

$$= -\int \mathrm{Q}(w;\xi)\log\frac{p(D,w)}{\mathrm{Q}(w;\xi)}dw \; + \; \int \mathrm{Q}(w;\xi)\log p(D)dw \quad , \tag{3.3}$$

where

$$\int \mathrm{Q}(w;\xi)dw = 1 \qquad . \tag{3.4}$$

As a result, we obtain

$$KL\big(\mathrm{Q}(w;\xi)||p(w|D)\big) = -\int \mathrm{Q}(w;\xi)\log\frac{p(D,w)}{\mathrm{Q}(w;\xi)}dw \; + \; log\, p(D) \; . \tag{3.5}$$

The first term of the right side of (3.5) is called as $-ELBO(\xi)$,

$$\int \mathrm{Q}(w;\xi)\log\frac{p(D,w)}{\mathrm{Q}(w;\xi)}dw \; = \; \int \mathrm{Q}(w;\;\xi)\log p(D|w)dw \; - \; \int \mathrm{Q}(w;\;\xi)\log\frac{\mathrm{Q}(W;\;\xi)}{p(w)}dw \quad , \tag{3.6}$$

$$ELBO(\xi) = \; \int \mathrm{Q}(w;\;\xi)\log p(D|w)dw \; - \; \int \mathrm{Q}(w;\;\xi)\log\frac{\mathrm{Q}(W;\;\xi)}{p(w)}dw \quad . \tag{3.7}$$

The Kullback-Leibler divergence between the $\mathrm{Q}(w;\xi)$ and $p(w|D)$ can be represented by

$$KL\big(\mathrm{Q}(w;\xi)||p(w|D)\big) = -ELBO(\xi) \; + \; log\, p(D) \quad . \tag{3.8}$$

Because the $log\, p(D)$ is constant for the variational posterior, to minimize the KL, the $ELBO(\xi)$ (equation 3.7) has to be maximized.

The calculation of derivatives is usually much more accessible than integration, making many approximations easier to handle. By adding $ELBO(\xi)$ in loss function, this study uses the re-parameterization trick for backpropagation [32] to update the variational parameters.

3.2 The proposed model

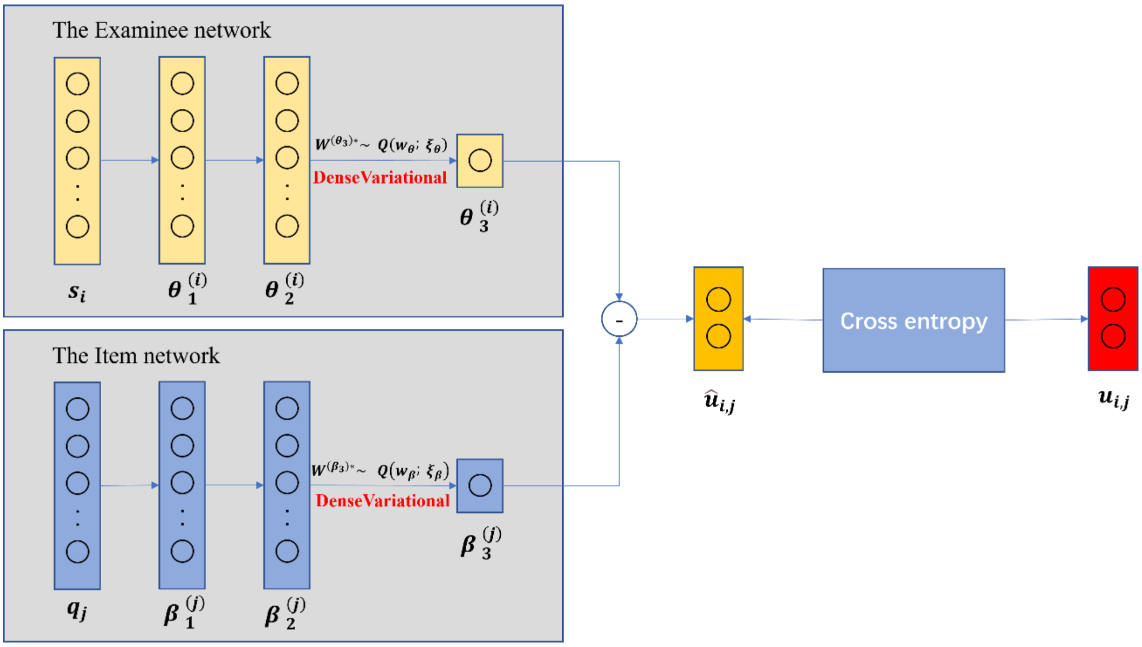


Figure 2 the structure of the proposed model

The Deep-IRT model has two networks to estimate examinees' abilities and the items' difficulties. Specifically, this study incorporates Bayesian neural network [27] into the final layers of the examinee network and the item network as

$$\theta_3^{(i)} = \boldsymbol{W}^{(\theta_3)} \theta_2^{(i)} + \tau^{(\theta_3)},$$

and

$$\beta_3^{(j)} = \boldsymbol{W}^{(\beta_3)} \beta_2^{(j)} + \tau^{(\beta_3)}, \tag{3.9}$$

where $\boldsymbol{W}^{(\theta_3)}$ and $\boldsymbol{W}^{(\beta_3)}$ follow the posteriors $p\left(\boldsymbol{W}^{(\theta_3)} \middle| D\right)$ and $p\left(\boldsymbol{W}^{(\beta_3)} \middle| D\right)$, respectively. The structure of the proposed model is shown in Figure 2.

3.2.1 learning parameters

The priors $p\left(\boldsymbol{W}^{(\theta_3)}\right)$ and $p\left(\boldsymbol{W}^{(\beta_3)}\right)$ are defined as a standard multivariate normal distribution.

It is difficult to analytically calculate the posterior $p\left(\boldsymbol{W}^{(\theta_3)}\middle|D\right)$ and $p\left(\boldsymbol{W}^{(\beta_3)}\middle|D\right)$, this study introduces the variational posteriors $\mathrm{Q}\left(\boldsymbol{W}^{(\boldsymbol{\theta}_3)};\ \xi_\theta\right)$ and $Q\left(\boldsymbol{W}^{(\beta_3)};\ \xi_\beta\right)$ of the variational Bayes [27] to approximate them as

$$\mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\ \xi_\theta\right) = N(\mu_\theta, \Sigma_\theta) \quad ,$$

$$\text{and} \qquad \mathrm{Q}\left(\boldsymbol{W}^{(\beta_3)};\ \xi_\beta\right) = N(\mu_\beta, \Sigma_\beta) \quad , \tag{3.10}$$

where $N(\mu_\theta, \Sigma_\theta)$ and $N(\mu_\beta, \Sigma_\beta)$ are the multivariate normal distributions parameterized by the variational parameters $\xi_\theta = \{\mu_\theta, \Sigma_\theta\}$ and $\xi_\beta = \{\mu_\beta, \Sigma_\beta\}$ respectively. μ_θ and μ_β are the mean vectors. Σ_θ and Σ_β are the covariance matrices. The Σ_θ and Σ_β are diagonal matrices as

$$\Sigma_\theta = \begin{pmatrix} \sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{\theta\theta} \end{pmatrix},$$

$$\text{and} \qquad \Sigma_\beta = \begin{pmatrix} \sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{\beta\beta} \end{pmatrix}.$$

The variational Bayes uses the following Kullback-Liebler (KL) divergence to measure the distance between the variational posterior $\mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\ \xi_\theta\right)$ and the Posterior $p\left(\boldsymbol{W}^{(\theta_3)}\middle|D\right)$ as

$$KL\left(\mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\ \xi_\theta\right)||p\left(\boldsymbol{W}^{(\theta_3)}\middle|D\right)\right) = \int \mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\ \xi_\theta\right)\log\frac{\mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\xi_\theta\right)}{p\left(\boldsymbol{W}^{(\theta_3)}\middle|D\right)}d\boldsymbol{W}^{(\theta_3)}, \quad (3.11)$$

where $KL\left(\mathrm{Q}\left(\boldsymbol{W}^{(\boldsymbol{\theta_3})};\ \xi_\theta\right)||p\left(\boldsymbol{W}^{(\theta_3)}\middle|D\right)\right)$ is obtained as

$$KL\left(\mathrm{Q}\left(\boldsymbol{W}^{(\boldsymbol{\theta_3})};\ \xi_\theta\right)||p\left(\boldsymbol{W}^{(\theta_3)}\middle|D\right)\right) = -ELBO(\xi_\theta)\ +\ log\,p(D), \quad (3.12)$$

and

$$ELBO(\xi_\theta) =$$
$$\int \mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\ \xi_\theta\right)\log p\left(D\middle|\boldsymbol{W}^{(\theta_3)}\right)d\boldsymbol{W}^{(\theta_3)} - \int \mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\ \xi_\theta\right)\log\frac{\mathrm{Q}\left(\boldsymbol{W}^{(\theta_3)};\xi_\theta\right)}{p\left(\boldsymbol{W}^{(\theta_3)}\right)}d\boldsymbol{W}^{(\theta_3)}. \quad (3.13)$$

This study estimated the variational parameters $\xi_\beta = \{\ \mu_\beta\ ,\Sigma_\beta\}$ in the item network with the same process.

Because the proposed model is also a classification model, it can still use the cross-entropy as a loss function (equation 2.11). This study calculates $ELBO(\xi_\theta)$ and $ELBO\left(\xi_\beta\right)$ to update the variational parameters ξ_θ and ξ_β by minimizing the following loss function.

$$L = \text{cross entropy} - ELBO(\xi_\theta) - ELBO(\xi_\beta) \ , \tag{3.14}$$

where $ELBO(\xi_\theta)$ and $ELBO(\xi_\beta)$ (equation 3.7) are calculated using the DenseVariational layer. The variational parameters ξ_θ and ξ_β are trained by the re-parameterization trick for backpropagation so as to maximize the $ELBO(\xi_\theta)$ and $ELBO(\xi_\beta)$.

This study uses the DenseVariational layer to achieve the re-parameterization trick for backpropagation to update parameters. The DenseVariational layer is an API (Application Programming Interface) from the TensorFlow Probability library [33].

The variational inference learns the distributions by maximizing the $ELBO(\xi)$ (equation 3.7), and two terms of the $ELBO(\xi)$ are computed in the DenseVariational layer separately.

For each epoch of training, we employ the batch processing. Each input dataset is divided into several small batches. $\int \mathrm{Q}(w;\xi)\log p(D|w)dw$ is calculated by approximating it with a single random sample from $\mathrm{Q}(w;\ \xi)$ on each small batch, because the sampling is repeated for each batch. Thus, by simply drawing a random set of weights from $\mathrm{Q}(w;\ \xi)$ and then computing the loss function, the first term of ELBO is approximated automatically. On the other hand, $\int \mathrm{Q}(w;\ \xi)\log\frac{\mathrm{Q}(w;\ \xi)}{p(w)}dw$ is computed analytically and then added to the layer as a regularization loss.

3.2.2 Estimation of an examinee's ability and an item's difficulty

By incorporating the Bayesian neural networks into the Deep-IRT model, the weights vector in the examinee network (equation 2.4) follows

$$\boldsymbol{W}^{(\theta_3)} \sim N(\mu_\theta, \Sigma_\theta) \quad . \tag{3.15}$$

Each weight parameter $(w_1^{(\theta_3)} \cdots w_{|\theta_2|}^{(\theta_3)})$ in the vector is sampled from the variational posterior distribution. The variational parameter μ_θ and Σ_θ are trained by the DenseVariational layer.

The weights vector in the item network (equation 2.8) also follows

$$\boldsymbol{W}^{(\beta_3)} \sim N(\mu_\beta, \Sigma_\beta) \quad . \tag{3.16}$$

Each weight parameter $(w_1^{(\beta_3)} \cdots w_{|\beta_2|}^{(\beta_3)})$ in the vector is sampled from the variational posterior distribution. The variational parameter μ_β and Σ_β are trained by the DenseVariational layer.

The estimation of an examinee's ability θ in the examinee network is obtained as follows

$$\theta_l|D \sim p(\theta|D) = \int p(\theta|w_\theta)p(w_\theta|D)dw,$$

and

$$\bar{\theta}|D = \frac{1}{n}\sum_{l=1}^{n}\theta_l|D, \tag{3.17}$$

where n is the sample size from the variational posterior of θ.

The estimation of an item's difficulty β in the item network is obtained as follows

$$\beta_l|D \sim p(\beta|D) = \int p(\beta|w_\beta)p(w_\beta|D)dw,$$

and

$$\bar{\beta}|D = \frac{1}{n}\sum_{l=1}^{n}\beta_l|D, \tag{3.18}$$

where n is the sample size from the variational posterior of β.

3.2.3 Prediction of an examinee's response $\hat{u}_{ij}$ to an item

The prediction probability p_{ij} of a correct answer to the item j by the examinee i is calculated by

$$p_{ij}|D = \frac{1}{1+\exp\left[-(\theta_{ij}|D-\beta_{ij}|D)\right]} \quad . \tag{3.19}$$

This study predicts the examinee’s unknown response as $\hat{u}_{ij} = 1$ if $p_{ij} \geq 0.5$, otherwise, $\hat{u}_{ij} = 0$.

The proposed model learns its parameters using the back-propagation algorithm by minimizing a loss function. It is calculated from the predicted responses $\hat{u}_{ij}$ and the true responses u_{ij} by adding them in the loss function (equation 3.14).

Chapter 4 Experiments

This chapter demonstrates two experiments to evaluate the performances of the proposed model. To compare the Deep-IRT model and the proposed model, this study implements the models using the GPU, Radeon R9 M390.

4.1 Datasets

4.1.1 Simulation datasets

To demonstrate the effectiveness of the proposed model when examinees' abilities are not randomly sampled, this subsection compares the estimation accuracies with changing examinee assignments for different tests. This study generates simulation experiments' data as Tsutsumi et.al (2021) [13,14] did.

This experiment generates 10 test datasets that have no common examinees. In addition, the k*-th* test $(k = 1, \ldots, 10)$ has common items only among the *(*$k-$*1)-th* test and the *(k + 1)- th* test. The actual parameters were generated randomly:

$$\theta \sim N(0,1),\ log\, a \sim N(0,1),\ b \sim N(1, 0.4)\ . \tag{4.1}$$

Here, the simulation data were generated based on 2PLM in the following two ways. The first way is that examinees are assigned randomly to each test from Equation (4.1). The other way is that examinees are given systematically to each test as described below.

1. Examinees are sampled randomly from Equation (4.1).

2. The examinees are sorted in order of their ascending ability. Furthermore, the examinees are divided equally into groups of 10 examinees in charge of their respective abilities.

3. The *k-th* examinee group is assigned to the *k-th* test.

The simulation data is set on Items in {10, 50, 100} and examinees in {100, 500, 1000}.

4.1.2 The actual datasets

This study used the same actual datasets used in Tsutsumi et.al (2021) [13,14]. These datasets are originally from [35] ~ [39].

The summary of actual datasets is shown in Table 1.

Table 1 Summary of actual datasets

Dataset	**Examinees**	**Items**
Benesse Japanese test	314	60
Discrete mathematics	77	125
programming1	148	7
programming2	75	18

4.2 Experiments 1: Estimation of the estimated abilities

This experiment compares the estimation accuracies of the estimated abilities of Deep-IRT with those of the proposed model using simulation datasets.

The training epochs are 300, and we divide each dataset into 10 batches.

This experiment employs the root mean square error (RMSE) and the correlation coefficients between the estimated abilities and the true values.

Because the proposed model samples several times to estimate the examinee's ability, we can calculate the standard error of the estimated abilities θ. The standard error can be estimated by

$$SE = \sqrt{\frac{1}{n}\sum_{l=1}^{n}(\theta_l|D - \bar{\theta}|D)^2}, \tag{4.2}$$

where n is the sample size from the posterior of θ. This experiment calculates the standard errors when sample size = {5,10,20,30,50}.

Table 2 Estimation accuracies and standard errors of the estimated abilities

Model	**Deep-IRT**	**Sample size-5**		**Sample size-10**		**Sample size-20**		**Sample size-30**		**Sample size-50**	
Dataset:	**RMSE**	**RMSE**	**SE**	**RMSE**	**SE**	**RMSE**	**SE**	**RMSE**	**SE**	**RMSE**	**SE**
{100.10}	0.8200	0.4557	0.0436	**0.4552**	0.0474	0.4560	0.0490	0.4561	0.0492	0.4553	0.0514
{500.10}	0.7572	0.5816	0.0409	0.5813	0.0449	**0.5812**	0.0468	0.5816	0.0478	0.5816	0.0481
{1000.10}	0.8801	0.6250	0.0484	0.6249	0.0516	**0.6246**	0.0537	0.6253	0.0547	0.6253	0.0554
{100.50}	0.55	0.3208	0.0507	**0.3199**	0.0562	0.3201	0..0599	0.3205	0.0607	0.3201	0.0618
{500.50}	0.451	**0.3433**	0.0603	0.3437	0.0662	0.3443	0.0691	0.3445	0.0696	0.3441	0.0706
{1000.50}	0.500	0.3362	0.0719	0.3361	0.0805	**0.3356**	0.0840	0.3360	0.0849	0.3360	0.0856
{100.100}	0.783	0.4805	0.0345	0.4809	0.0371	0.4805	0.0389	0.4803	0.0392	**0.4800**	0.0394
{500.100}	0.633	0.4330	0.0708	0.4329	0.0779	**0.4326**	0.0806	0.4328	0.0825	0.4333	0.0830
{1000.100}	0.614	0.4268	0.0797	0.4269	0.0874	0.4267	0.0911	**0.4265**	0.0927	0.4266	0.0935

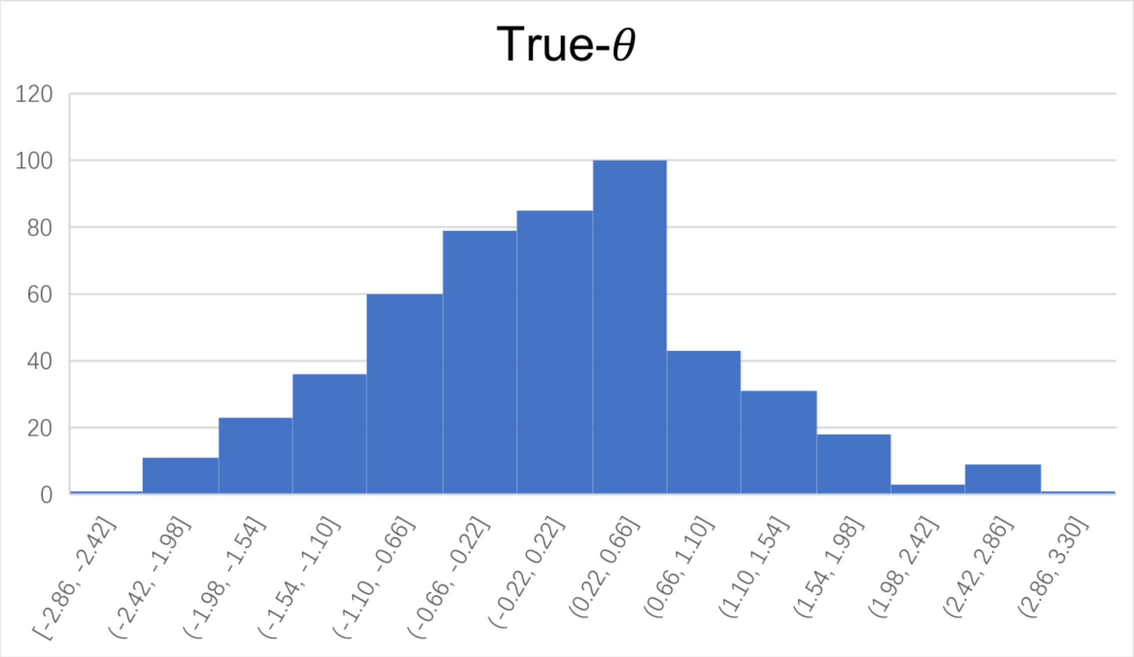
Table 2 shows the RMSEs of the proposed model tend to be less than those of the Deep-IRT model. Table 2 also demonstrates the estimated standard errors slightly decrease as the sample size increases. In this study, (sample size = 5) is used as an example to compare the performances of the correlation coefficients.

Table 3 Estimation accuracies of the estimated abilities (Sample size = 5)

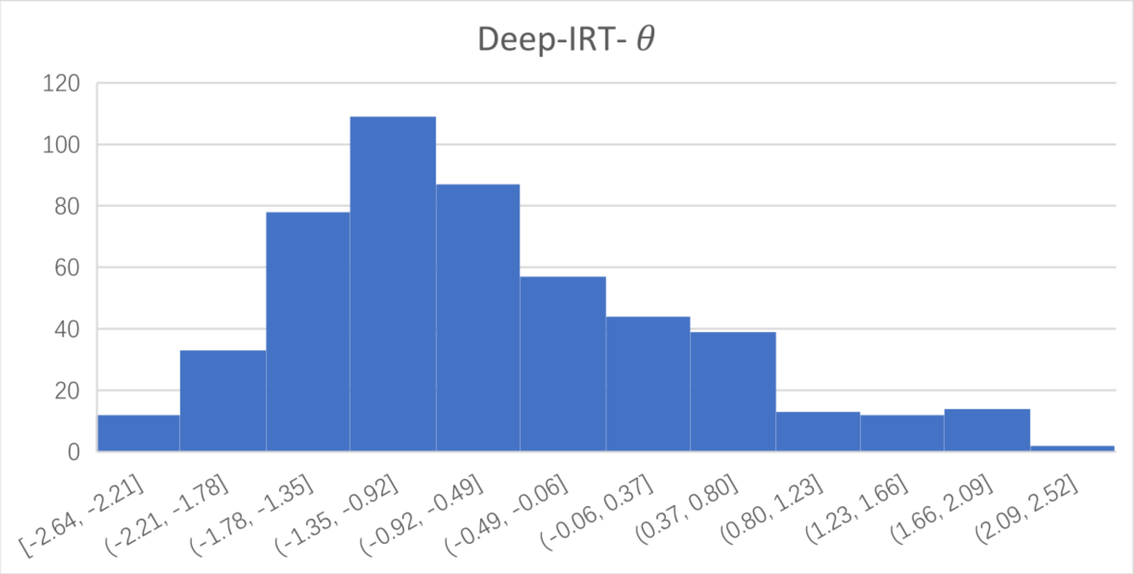
Items	Examinees	Theta θ	RMSE	Pearson	Kendall	Spearman
10	100	Deep-IRT	0.82	0.691	0.520	0.691
		Proposed	**0.45**	**0.735**	**0.553**	**0.735**
	500	Deep-IRT	0.757	0.766	0.571	0.767
		Proposed	**0.581**	**0.778**	**0.579**	**0.775**
	1000	Deep-IRT	0.880	0.595	0.420	0.591
		Proposed	**0.625**	**0.620**	**0.463**	**0.645**
50	100	Deep-IRT	0.55	**0.928**	0.775	0.927
		Proposed	**0.32**	0.926	**0.784**	**0.936**
	500	Deep-IRT	0.451	**0.924**	0.787	0.939
		Proposed	**0.343**	0.921	**0.791**	**0.942**
	1000	Deep-IRT	0.500	0.925	0.788	0.940
		Proposed	**0.336**	**0.932**	**0.796**	**0.944**
100	100	Deep-IRT	0.783	0.943	0.796	0.940
		Proposed	**0.480**	**0.952**	**0.802**	**0.942**
	500	Deep-IRT	0.633	**0.936**	0.803	0.949
		Proposed	**0.433**	0.930	**0.809**	**0.951**
	1000	Deep-IRT	0.614	0.936	0.825	0.958
		Proposed	**0.427**	**0.942**	**0.827**	0.958

Table 3 shows the proposed model tends to have higher the correlation coefficients than the Deep-IRT model does. Namely, the abilities predicted by the proposed model is closer to the true abilities than those by the Deep-IRT.

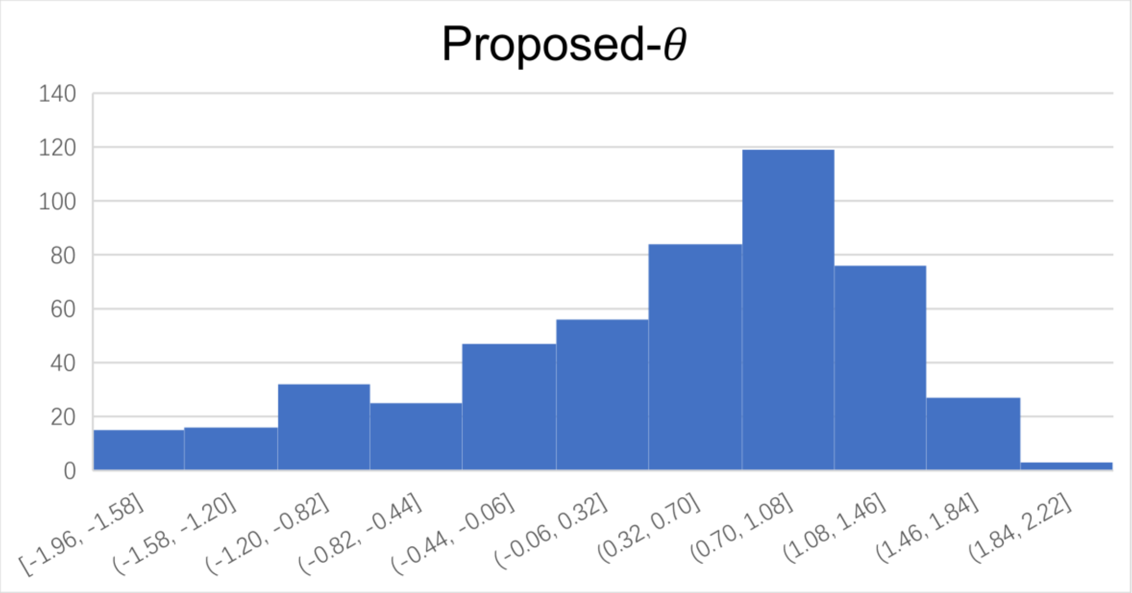
The histogram of estimated abilities for items {100} and examinee {500} is depicted as follows.



Histogram 1 The true θ



Histogram 2 The θ predicted by Deep-IRT



Histogram 3 The θ predicted by the proposed model

The peak value of the true θ in [0.22, 0.66], and the proposed model's distribution is closer than that of the Deep-IRT model. It shows that Bayesian neural networks make the model more explainable.

4.3 Experiment 2: Prediction accuracies

This experiment compares the prediction accuracies of predicted examinees' responses of the proposed model and the previous Deep-IRT model. This experiment used some simulation datasets and actual datasets.

The training epochs are 300, and we divide each dataset into 10 batches.

For all the datasets, 20% of the sequences are held out as a test set, and the remaining 80% are used as a training set. Furthermore, five-fold cross-validation is applied to the training set. This study compares the performances of the models using accuracy (Acc), the AUC scores and the F1 scores.

Table 4 Prediction accuracies in simulation datasets

Items	Examinees	Metrics(%)	Acc	AUC	F1
10	100	Deep-IRT	67.0	68.3	63.0
		Proposed	**74.0**	**77.9**	**72.5**
	500	Deep-IRT	74.0	76.3	67.6
		Proposed	**74.8**	**78.2**	**72.2**
	1000	Deep-IRT	76.5	76.2	67.5
		Proposed	**76.6**	76.2	**74.1**
50	100	Deep-IRT	70.5	75.4	70.0
		Proposed	**72.5**	**76.4**	**72.2**
	500	Deep-IRT	70.5	75.7	70.3
		Proposed	**71.2**	**76.3**	**71.1**
	1000	Deep-IRT	73.6	78.3	73.4
		Proposed	**74.3**	**79.4**	**74.2**
100	100	Deep-IRT	71.1	74.8	71.3
		Proposed	**71.8**	**75.1**	**72.1**
	500	Deep-IRT	71.8	76.3	71.3
		Proposed	**72.7**	**76.3**	**72.5**
	1000	Deep-IRT	73.5	78.3	73.4
		Proposed	73.5	**78.7**	**73.6**

Table 4 shows that the proposed model performs better than the Deep-IRT model especially for small datasets because the Deep-IRT model tends to over-fit for small datasets, but the proposed model mitigates it. The proposed model has a better F1 score because it mitigated the overfitting in deep learning.

Table 5 Prediction accuracies in actual datasets

Dataset	Metrics(%)	Acc	AUC	F1
Benese	Deep-IRT	73.8	74.8	73.5
	Proposed	**74.6**	**75.8**	**74.4**
Discrete mathematics	Deep-IRT	71.2	78.5	71.5
	Proposed	**72.2**	**78.9**	**73.0**
programming1	Deep-IRT	66.4	73.4	63.0
	Proposed	**71.0**	**76.0**	**64.4**
programming2	Deep-IRT	73.4	78.3	71.5
	Proposed	**74.5**	**79.1**	**73.4**

Table 5 shows the results: the average of metrics of the proposed model is significantly higher than that of the Deep-IRT model. The proposed model can predict examinees' responses to unknown items more accurately than the Deep-IRT model can, especially in programming1 and programming2 which are relatively small datasets.

The results show that Bayesian neural network helps the Deep-IRT model to predict more accurately an examinee's responses to unknown items.

Chapter 5 Conclusions

This study proposed a new Deep-IRT model based on Bayesian neural networks, which used the Bayesian approach to model examinees' reactions to an item. Due to the effectiveness of Bayesian neural networks, the parameters of the proposed model can be highly explained. The proposed model mitigated the overfitting problem of the previous Deep-IRT because it showed higher F1 scores than the previous model did. The results also demonstrated the proposed model accurately estimated the examinees’ abilities. In addition, the proposed model improved the accuracy of response probability prediction of the previous model. As a future work, we will incorporate Bayesian neural network into all the layers of the Deep-IRT model. Furthermore, as another future work, we will apply the proposed model for Computer Adaptive Testing (CAT) [40,41] to improve the examinee’s ability estimation accuracy.

Acknowledgements

It was my pleasure to have the opportunity as a student to study in UEC (The University of Electro-Communications). The two-year course was a precious experience in my life. I learned a lot from the courses and the experiments. I believe this experience is helpful very much for my future career. Many thanks to all the people I met here.

Specially thanks to my thesis advisor Prof. Ueno. Without his help, I could not finish my thesis. The door to Prof. Ueno office was always open whenever I ran into a trouble spot or had a question about my research or writing. He also provided indispensable assistance and gave me many crucial and helpful suggestions and comments. With the guidance and assistance, I learned many experimental techniques and experience and can finish my research. I would like to express my deep gratitude to him.

I also want to thank to Miss. Tsutsumi at The University of Electro-Communications. Her work provided huge support for my experiments. I am grateful very much to her for her support.

I also want to say thank to Prof. Kawano. His comments of my thesis is very useful and they helped me a lot.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

References

1. Songmuang, P.; Ueno, M. Bees Algorithm for Construction of Multiple Test Forms in E-Testing. IEEE Trans. Learn. Technol. 2011, 4, 209–221.

2. Ishii, T.; Songmuang, P.; Ueno, M. Maximum Clique Algorithm for Uniform Test Forms Assembly. In Proceedings of the 16th International Conference on Artificial Intelligence in Education, Memphis, TN, USA, 9–13 July 2013; Volume 7926, pp. 451–462._46.

3. Ishii, T.; Songmuang, P.; Ueno, M. Maximum Clique Algorithm and Its Approximation for Uniform Test Form Assembly. IEEE Trans. Learn. Technol. 2014, 7, 83–95.

4. Ishii, T.; Ueno, M. Clique Algorithm to Minimize Item Exposure for Uniform Test Forms Assembly. In Proceedings of the International Conference on Artificial Intelligence in Education, Madrid, Spain, 22–26 June 2015; pp. 638–641.

5. Ishii, T.; Ueno, M. Algorithm for Uniform Test Assembly Using a Maximum Clique Problem and Integer Programming. In Proceedings of the Artificial Intelligence in Education, Wuhan, China, 28 June–1 July 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 102–112.]

6. Lin, Y.; Jiang, Y.S.; Gong, Y.J.; Zhan, Z.H.; Zhang, J. A Discrete Multiobjective Particle Swarm Optimizer for Automated Assembly of Parallel Cognitive Diagnosis Tests. IEEE Trans. Cybern. 2018, 1–14.

7. Vie, J.J.; Popineau, F.; Bruillard, E.; Bourda, Y. Automated Test Assembly for Handling Learner Cold-Start in Large-Scale Assessments. Int. J. Artif. Intell. Educ. 2018, 28.

8. RodrÃguez-Cuadrado, J.; Delgado-GÃ³mez, D.; Laria, J.; Rodriguez-Cuadrado, S. Merged Tree-CAT: A fast method for building precise Computerized Adaptive Tests based on Decision Trees. Expert Syst. Appl. 2019, 143, 113066.

9. Linden, W.; Jiang, B. A Shadow-Test Approach to Adaptive Item Calibration. Psychometrika 2020, 85.

10. Ren, H.; Choi, S.; Linden, W. Bayesian adaptive testing with polytomous items. Behaviormetrika 2020, 47.

11. Ueno M, AI based e-testing as a common yardstick for measuring human abilities.In: The 18th International Joint Conference on Computer Science and Software ngineering, IEEE computer society, pp 1–6, (2021)

12. Ueno,M., Fuchimoto,K., and Tsutumi,E., "e-testing from artificial intelligence aproach," Behaviormetrika, vol.48, no.2,(2021).

13. Tsutsumi, Emiko, Ryo Kinoshita, and Maomi Ueno. "Deep Item Response Theory as a Novel Test Theory Based on Deep Learning." Electronics 10.9 (2021): 1020.

14. Tsutsumi, E.; Kinoshita, R.; Ueno, M. "Deep-IRT with independent student and item networks." Proceedings of the 14thInternational Conference on Educational Data Mining, EDM, Paris, France, 29 June–2 July 2021.

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

16. Lord, F., and Novick, M. 1968. Statistical Theories of Mental Test Scores. Addison-Wesley.

17. van der Linden, W., and Barrett, M. D. 2016. Linking item response model parameters. Psychometrika 81(3):650–673.

18. Lord, F. 1980. Applications of item response theory to practical testing problems. L. Erlbaum Associates Hillsdale, N.J.

19. W.J. van der Linden. 2016a. Handbook of Item Response Theory, Volume Three: Applications. Chapman and Hall/CRC Statistics in the Social and Behavioral Sciences. Chapman and Hall/CRC.

20. Joo, S.-H.; Lee, P.; and Stark, S. 2017. Evaluating anchoritem designs for concurrent calibration with the ggum. Applied Psychological Measurement 41(2):83–96.

21. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., Advances in Neural Information Processing Systems

28, 505–513. Curran Associates, Inc.

22. Le, C. V.; Pardos, Z. A.; Meyer, S. D.; and Thorp, R. 2018. Communication at scale in a MOOC using predictive engagement analytics. In Artificial Intelligence in Education - 19th International Conference, AIED 2018, London, UK, June 27-30, 2018, Proceedings, Part I, 239–252.

23. Vie, J., and Kashima, H. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019., 750–757.

24. J. Zhang, X. Shi, I. King, and D.-Y. Yeung, “Dynamic key-value memory networks for knowledge tracing,” in 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, 2017, pp. 765–774.

25. Wilson, K. H.; Karklin, Y.; Han, B.; and Ekanadham, C. 2016. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. volume 1, 539–544.

26. Donghua, Li, et al. "Deep Knowledge Tracing Based on Bayesian Neural Network." International Conference on Intelligent and Interactive Systems and Applications. Springer, Cham, 2019.

27. Gal Y. Uncertainty in deep learning[J]. University of Cambridge, 2016, 1(3).

28. Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In International Conference on Machine Learning, pp. 1613–1622, 2015.

29. Georg Rash. Probabilistic models for some intelligence and attainment tests. Copenhagen: Danish Institute for Educational Research, 1960.

30. Carsten Peterson. A mean field theory learning algorithm for neural networks.

Complex systems, 1: 995–1019, 1987.

31. Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In Conference on Computational Learning Theory, pp. 5–13, 1993.

32. Hernández-Lobato, J.M., Adams, R.: Probabilistic back-propagation for scalable learning of bayesian neural networks. In: International Conference on Machine Learning, pp. 1861– 1869 (2015)

33. https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DenseVariational

34. Uysal, I.; Kilmen, S. Comparison of Item Response Theory Test Equating Methods for Mixed Format Tests. Int. Online J. Educ. Sci. 2016, 8, 1–11. [CrossRef]

35. M. Ueno, Animated agent to maintain learner's attention in e-learning. In Proceedings of the E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004; Nall, J., Robson, R., Eds.; Association for the Advancement of Computing in Education (AACE): Washington, DC, USA, 2004; pp. 194–201.

36. M. Ueno, Data Mining and Text Mining Technologies for Collaborative Learning in an ILMS "Samurai". In Proceedings of the ICALT '04 Proceedings of the IEEE International Conference on Advanced Learning Technologies, 2004; pp. 1052–1053.

37. M. Ueno, Intelligent LMS with an agent that learns from log data. In Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005; Richards, G., Ed.; Association for the Advancement of Computing in Education (AACE): 2005; pp. 3169–3176

38. M. Ueno, and Y. Miyazawa, Probability Based Scaffolding System with Fading. In Proceedings of the Artificial Intelligence in Education, 17th International Conference, AIED, pp. 237–246._49, 2015

39. M. Ueno and Y. Miyazawa, "IRT-Based Adaptive Hints to Scaffold Learning in Programming," in IEEE Transactions on Learning Technologies, vol. 11, no. 4, pp. 415-428, 2018, doi: 10.1109/TLT.2017.2741960.

40. M. Ueno, P. Songmuang, Computerized Adaptive Testing Based on Decision Tree. In

Proceedings of the Advanced Learning Technologies (ICALT), 2010 IEEE Tenth International Conference, Sousse, Tunisia, 5–7 July 2010; pp. 191–193.

41. M. Ueno, Adaptive testing based on Bayesian decision theory. In Proceedings of the International Conference on Artificial Intelligence in Education, Memphis, TN, USA, 9–13 July 2013; pp. 712–716.