Master's Thesis of Science Education

Enhanced Item Response Theory: Integration of Response Consistency

응답정합성을 이용한 문항반응이론의 개선

August 2023

SEOUL NATIONAL UNIVERSITY
Department of Science Education (Physics Major)

YUNHWAN JANG

Enhanced Item Response Theory: Integration of Response Consistency

지도교수 조 정 효

이 논문을 교육학석사 학위논문으로 제출함
2023년 8월

서울대학교 대학원
과학교육과 물리전공
장 윤 환

장윤환의 석사 학위논문을 인준함
2023년 8월

위원장 유 준 희 (인)

부위원장 이 경 호 (인)

위 원 조 정 효 (인)

Enhanced Item Response Theory: Integration of Response Consistency

Examiner : Junghyo Jo

Submitting a master's thesis of
Science Education (Physics Major)

August 2023

Department of Physics Education
Seoul National University

Confirming the master's thesis written by
Yunhwan Jang

August 2023

Advisory Committee:

Professor June-Hee Yoo, Chair
Professor Gyoungho Lee, Vice Chair
Associate Professor Junghyo Jo, Examiner

Abstract

Item response theory (IRT) depicts the general tendency of interactions between items and examinees. IRT is applied in various areas, such as the item bank. In addition, diverse academic fields, such as psychology, adopt IRT as a methodology. Therefore, IRT holds both academic and practical significance.

IRT outperforms classical test theory (CTT) in terms of practicality and flexibility. However, due to the complex nature of an examinee's ability, existing models, especially unidimensional IRT (UIRT), excessively simplify the interaction between the examinees and items. This characteristic contributes to limitations in accuracy of diagnosing examinees' abilities and imputing missing data. Consequently, this limitation restricts the connection between evaluation and feedback.

To reinforce connectivity, a new IRT model is required to enhance its performance with respect to level diagnosis and imputation. To achieve this purpose, we have adopted interactions between two item pairs. Existing IRT models reflect these interactions indirectly, while the new IRT model does so directly. These interactions are conceptualized as response consistency.

In order to strengthen and verify the performance, methodologies relevant to machine learning were introduced. As a result, a more generalized level diagnosis of examinees has been accomplished. The

advanced diagnosis results served as the basis for further enhancing the imputation performance.

Response consistency is deemed to improve the performance of IRT by incorporating interactions between item pairs, which further segregate innocent responses from wild guessing. Meanwhile, it was confirmed that item categories sorted out by the response consistency coincided with item group classification in PISA 2018. This serendipitous finding is expected to open the window of opportunity for a data-driven approach in educational evaluation. In future studies, the interaction between two items is expected to be expanded into the interaction among multiple items for exploration towards the general response consistency.

Keyword : response consistency, multidimensional item response theory, item bank, imputation, machine learning, data-driven approach
Student ID : 2021-28401

Contents

List of Figures

Chapter 1. Introduction

1.1. Purpose of Research

According to Douglas Stone and Sheila Heen, "there are three types of feedback: evaluation, coaching, and appreciation." Evaluation simply rates points, whereas coaching provides information for further learning. In addition, appreciation offers sincere reactions from instructors. In other words, ultimately, feedback requires not only quantitative information but also qualitative information and emotional depth. Item Response Theory (IRT) may cover the varied aspect of feedback.

IRT depicts the general tendency of interactions between items and examinees. It is applicable in various areas such as achievement tests, the item bank, and computerized adaptive tests (CATs). Additionally, IRT is adopted in diverse academic fields such as psychology and medical science. Therefore, IRT holds both academic and practical significance.

Regarding IRT, it does not depend on the characteristics of examinees, unlike classical test theory (CTT). As a result, IRT is appraised as outperforming CTT in terms of practicality and

flexibility. Nevertheless, due to the complex nature of the interaction between items and examinees, existing models, especially unidimensional IRT (UIRT), excessively simplify this interaction. Only few IRT variables attempt to reenact the complexity of the interaction. As a result, these circumstances limit the performance of IRT, consequently restricting the connection between evaluation and feedback.

Before delving into a detailed discussion, there are two points to consider. First, the diversity of IRT variables for a more precise level of diagnosis is important. This point is expected to cover more aspects of the complex nature of the interaction. Second, the accuracy of imputation is significant for item banks as well. Item banks often encounter missing data due to nonresponse. The incompleteness of the item banks leads to the incompleteness of a customized test and further feedback. If unresponsive items are properly imputed, the quality of the customized test and feedback will be improved.

In this study, a new model called Ising Multidimensional Item Response Theory (IMIRT) is introduced. IMIRT incorporates a new exponential term derived from the Hamiltonian of the Ising model. The Hamiltonian of the Ising model is known for representing the interaction between adjacent two spins of a material. Similarly, the new exponential term in IMIRT reflects the interactions between two items of a test set. This introduced exponential term is expected to assist in more precise diagnosis of examinees' abilities. Furthermore, this term is expected to enhance the performance of imputation.

Regarding the verification process, multiple machine learning methods, such as gradient descent and train/test splitting, will be applied. Gradient descent is an algorithm used for exploring optimization through numerical analysis and is applicable to complex models. On the other hand, train/test splitting is a methodology used to verify the explanatory power of a model. Both methods are suitable for the verification process of complex data and models. Therefore, they are expected to accomplish the verification process of the new model, IMIRT.

1.2. Research Goals

In the process of the verification of IMIRT, two goals need to be accomplished.

[Goal 1] Is IMIRT model capable of superior performance in terms of the accurate imputation and the precise level diagnosis?

[Goal 2] What is the meaning of the parameters and variables in IMIRT model? In other words, what is the role of each parameter or variable in improving the performance of the IRT model?

Chapter 2. Theoretical Background

2.1. Item Response Theory

2.1.1 General Description, Assumptions and Types

IRT quantitatively evaluates the interaction between examinees and items. In comparison with CTT, IRT offers more flexibility in estimating item difficulty and item discrimination. In CTT, item difficulty and discrimination are estimated solely based on answer rates, while IRT takes into account the characteristics of both examinee groups and test items, along with answer rates, to calculate these two parameters. For example, if a group of examinees demonstrates a low level of achievement, IRT estimates the difficulty of items to be higher and the ability of examinees to be more generously assessed. In summary, the flexible nature of IRT ensures higher reliability in evaluation compared to CTT.

There are five basic assumptions in IRT. First, the location of examinees remains constant during the test. Second, the characteristics of test items remain static throughout the test. The first two assumptions exclude the possibility of interaction with the environment. Third, the response to one test item by an examinee

does not influence the response to other test items. This assumption is referred to as the assumption of independence. Fourth, the relationship between the ability level and the probability of answering correctly can be described as a continuous function. Fifth, as the probability of answering correctly increases, the ability level of the corresponding examinee monotonically increases. The final assumption represents the consistency of the model.

The number of variables and parameters determines the type of IRT model. If the location of ability is determined by a single indicator, the model is referred to as UIRT. If there are multiple indicators to determine the location of ability, the model is referred to as MIRT.

2.1.2 UIRT Models

Regarding the binary case, UIRT encompasses various models, including the Rasch model, the two-parameter logistic model (2PL model), and the three-parameter logistic model (3PL model).

First, the Rasch model, a one-parameter logistic model (1PL model), displays a probability distribution as follows:

$$P\left(Y_i^{\mu} = 1 \mid \beta_i, \theta^{\mu}\right) = \frac{e^{\theta^{\mu}-\beta_i}}{1+e^{\theta^{\mu}-\beta_i}}, \qquad (2.\ 1)$$

where β_i is the difficulty parameter of the ith item, and θ^{μ} shows the location of ability of the μth examinee. $Y_i^{\mu} = 1$ indicates that the μth examinee answered the ith item correctly. The 1PL model is fitted to the reference data with only one extrinsic parameter: β_i. In

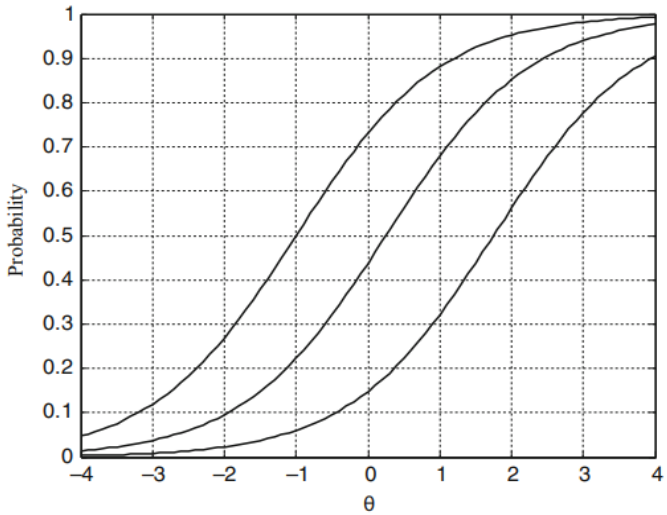


Figure 2-1. Three item characteristic curves (ICCs) of 1PL model. The left ICC curve represents a difficulty of -1, the middle ICC curve represents a difficulty of 0.2, and the right ICC curve represents a difficulty of 1.7. (Reckase 2009)

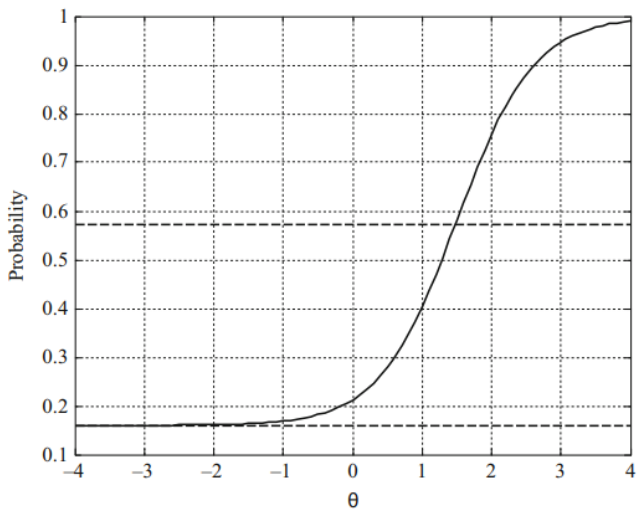


Figure 2-2. An item characteristic curve (ICC) of 3PL model. The asymptotic line, with a probability of 0.16, represents the likelihood of correctly answering the item through guessing. (Reckase 2009)

addition, the 1PL model exhibits high level of flexibility. However, the 1PL model lacks an important parameter for items, which is discrimination.

Second, the 3PL model has an expanded logistic form of the probability distribution as follows:

$$P(Y_i^{\mu} = 1|\alpha_i, \beta_i, \gamma_i, \theta^{\mu}) = (1 - \gamma_i)\frac{e^{\alpha_i(\theta^{\mu} - \beta_i)}}{1 + e^{\alpha_i(\theta^{\mu} - \beta_i)}} + \gamma_i\,, \qquad (2.\ 2)$$

where α_i is the discrimination parameter of the ith item, γ_i is the asymptotic parameter relevant to guessing.

The 3PL model includes two additional extrinsic parameters: α_i and γ_i. As a result, the 3PL model can exhibit a high level of explanatory power. However, the formula of the 3PL model is excessively complex for application.

Finally, 2PL model has a logistic form of probability distribution as follows:

$$P(Y_i^{\mu} = 1 \mid \alpha_i, \beta_i, \theta^{\mu}) = \frac{e^{\alpha_i(\theta^{\mu} - \beta_i)}}{1 + e^{\alpha_i(\theta^{\mu} - \beta_i)}}\,. \qquad (2.\ 3)$$

The 2PL model introduces an additional extrinsic parameter: α_i. When α_i increases, the item characteristic curve (ICC) exhibits a steep rise near the probability point of 0.5, as shown in **Figure 2−3**. Conversely, when α_i decreases, the ICC rises relatively gradually near the same point, as depicted in **Figure 2−3**. In summary, α_i is referred to as the discrimination parameter.

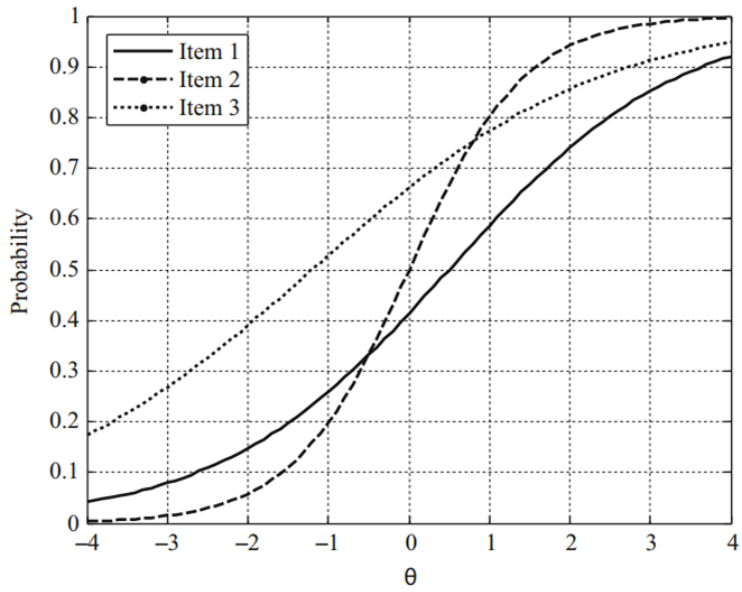


Figure 2-3. Three item characteristic curves (ICCs) of 2PL model. The ICC of item 1 shows middle-level discrimination and a difficulty of 0.5. The ICC of item 2 exhibits high-level discrimination and a difficulty of 0. The ICC of item 3 displays low-level discrimination and a difficulty of -1.2. (Reckase 2009)

2.1.3 MIRT Models

Regarding the binary case, MIRT includes two prominent models: the compensatory model and the partial-compensatory model. First, the compensatory model has a general form of the probability distribution as follows:

$$P\left(Y_i^{\mu}=1\middle|\boldsymbol{\alpha_i}, d_i, \boldsymbol{\theta^{\mu}}\right)=\frac{e^{\boldsymbol{\alpha_i}\cdot\boldsymbol{\theta^{\mu}}-d_i}}{1+e^{\boldsymbol{\alpha_i}\cdot\boldsymbol{\theta^{\mu}}-d_i}}, \qquad (2.\ 4)$$

where $\boldsymbol{\alpha_i}\cdot\boldsymbol{\theta^{\mu}}=\alpha_{i1}\theta_1^{\mu}+\alpha_{i2}\theta_2^{\mu}+\cdots=\sum_{v=1}\alpha_{iv}\theta_v^{\mu}$, v represents the number of ability variables and d_i is an intercept parameter. α_{i1}

represents discrimination parameter of the ith item corresponding the first variable of ability θ_1^{μ}.

In the compensatory model, the exponential term of the logistic model's probability distribution function contains a linear combination of multiple abilities. This allows the compensatory model to replenish a vacancy due to lack of a specific ability with other abilities. Furthermore, the intercept parameter d_i in MIRT differs from the difficulty parameter β_i in UIRT. While β_i interacts with a single ability, d_i needs to interact with a linear combination of multiple abilities.

Second, the partial-compensatory model has a probability distribution function in the following form:

$$P\left(Y_i^{\mu}=1\middle|\boldsymbol{\alpha_i}, d_i, \boldsymbol{\theta^{\mu}}\right)=\prod_{v}\frac{e^{\alpha_{iv}(\theta_v^{\mu}-\beta_{iv})}}{1+e^{\alpha_{iv}(\theta_v^{\mu}-\beta_{iv})}}. \qquad (2.5)$$

The formula of the partial-compensatory model consists of simple multiplications of a series of UIRT models. In the case of the partial-compensatory model, if an examinee experiences a significant loss in a specific ability, it becomes difficult to restore the damage with other abilities of high skill. Therefore, this model is referred to as a partial-compensatory model.

2.2. Ising Model

The Ising model is a theoretical model in statistical physics used primarily to describe sudden changes, such as phase transitions and the Curie temperature. In statistical physics, natural phenomena are studied using many-body systems through the use of Hamiltonian. In the case of the Ising model, when there is no external magnetic field, the Hamiltonian consists of the interaction between two neighboring spins. A detailed description of the Hamiltonian is provided below:

$$\hat{H}(\vec{s}_1, \vec{s}_2, \dots, \vec{s}_N) = -\sum_{i \neq j} J_{ij}\, \vec{s}_i \cdot \vec{s}_j \,. \qquad (2.6)$$

In the description of Hamiltonian, J_{ij} represents the interaction between two neighboring spins $\vec{s}_i$ and $\vec{s}_j$. If J_{ij} is positive $(J_{ij} > 0)$, the interaction is referred to as ferromagnetic. Conversely, if J_{ij} is negative $(J_{ij} < 0)$, the interaction is referred to as anti-ferromagnetic.

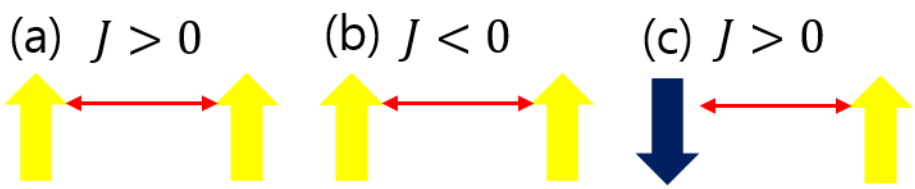


Figure 2-4 Spin configurations and spin interactions. (a) Two up-spins (yellow arrows) with a positive interaction ($J > 0$). (b) Two up-spins and a negative interaction ($J < 0$). (c) One up-spin and one down-spin (a dark blue arrow) with a positive interaction.

In the diagram (a) of **Figure 2-4**, interaction between two up-spins with a positive J decreases the Hamiltonian. On the other hand, for the diagram (b) of **Figure 2-4**, two up-spins with a negative J increase the Hamiltonian despite the same spin configuration of (a). Meanwhile, for the diagram (c) of **Figure 2-4**, the configuration of two inverse spins with a positive J increases the Hamiltonian. In summary, both the shape of spin configurations and the interaction J determine the change direction of the Hamiltonian in the Ising model.

Chapter 3. Research Procedure and Methods

3.1. Overview

The study followed a series of processes. First, we consider various models. The UIRT 2PL model was chosen as a control group. As an experimental group, IMIRT was selected. To establish IMIRT, the compensatory model of MIRT was selected as a framework, and the Hamiltonian of the Ising model was embedded into the exponential term of the compensatory MIRT. Second, the Program for International Student Assessment 2018 (PISA 2018) was selected as the reference data. Among the chosen data, only Computer-Based Test (CBT) items responded to by examinees from the Republic of Korea (ROK) were filtered for this study in order to maintain uniformity of the sample. Finally, we optimized the models by using gradient descent that is an optimization algorithm for finding a local extremum of differentiable objective functions. After completing the optimization, a verification process was conducted to determine the superiority of the new model.

3.2. Model Establishment Process

Among the UIRT models, the 2PL model is suitable for the control group. As mentioned in Chapter 2, the 1PL model is insufficient as it only includes the difficulty parameter. Additionally, the 3PL model is prone to overfitting, which can harm the predictive capability of the model.

For MIRT models, the compensatory model is adequate to serve as the framework for the new model. The compensatory model has the advantage of compatibility and simplicity. Unlike the partial-compensatory model, which experiences a steady decrease in the probability distribution as the model dimension increases, the compensatory model avoids this drawback. This characteristic facilitates the comparison of performance between the compensatory MIRT model and UIRT. Furthermore, the simplicity of the compensatory model allows for easy adoption of new parameters and variables.

Next, the Hamiltonian of the Ising model is converted into the new variable θ_2, establishing the new model ultimately. The conversion process consists of two steps. First, the Hamiltonian is normalized to form a pseudo-probability $\widehat{P}$ as shown below:

$$\widehat{P}^{\mu} = \frac{1}{2}\sum_{k\neq l}\frac{Q_{kl}\,Y'^{\mu}_{k}Y'^{\mu}_{l}}{\Sigma_{k'\neq l'}Q_{k'l'}} + \frac{1}{2}, \tag{3. 1}$$

where $\widehat{P}^{\mu}$ is a pseudo probability of the μth examinee, and k and l are index of items except missing data. In addition, if μth examinee

answers correctly the k'th item, then $Y'^{\mu}_{k} = 1$. If not, then $Y'^{\mu}_{k} = -1$. Then the pseudo probability ranges from 0 to 1. A pseudo probability is derived from the Hamiltonian of the Ising model to be inserted into the log odds, which requires a variable ranging from 0 to 1.

In the conversion process, the scale of Q is adjusted by dividing it with $\Sigma_{k' \neq l'} Q_{k'l'}$ in order to normalize the Hamiltonian of the Ising model. Since $Y'^{\mu}_{k} Y'^{\mu}_{l}$ ranges from -1 to 1, an additional step of scale adjustment is required. For this purpose, the normalized Hamiltonian is halved and 0.5 is added.

Second, the pseudo probability is transformed into log odds to complete the process as shown below:

$$\theta_2^{\mu} = \ln\left(\frac{\hat{P}^{\mu}}{1 - \hat{P}^{\mu}}\right). \quad (3.\ 2)$$

In this manner, the new variable θ_2 has been established, leading to the suggestion of the new model named IMIRT.

3.3. Data Selection and Preprocessing

3.3.1 Data Selection and Criteria

The PISA 2018 Student questionnaire data file in SPSS (TM) Data Files format was selected. As the data did not require additional human-targeted investigations and did not pose a risk of personal information leakage, it was evident that the data did not violate the Institutional Review Board (IRB).

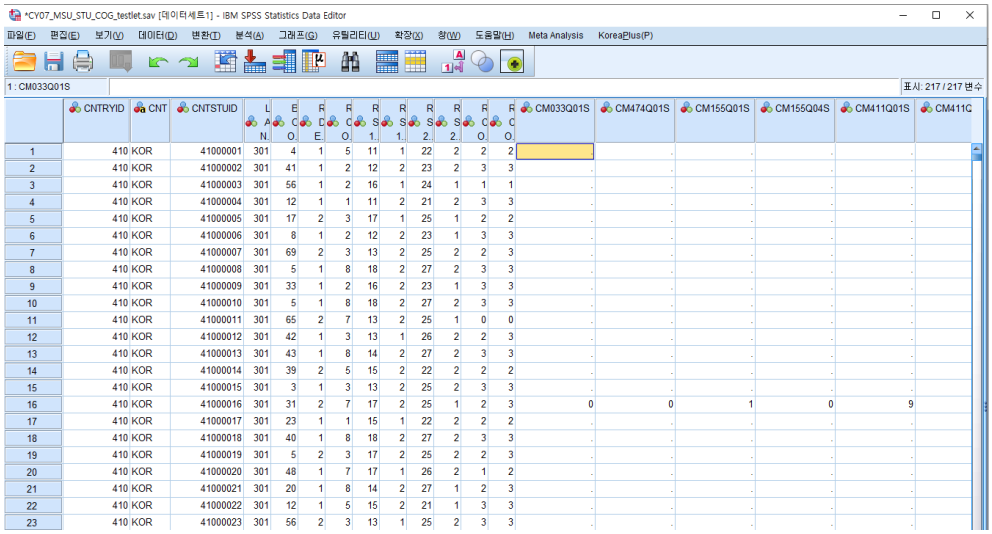


Figure 3-1. A part of data from PISA 2018 under preprocessing by IBM SPSS Statistics Data Editor software. The items responded to by ROK (Republic of Korea) students were exclusively sampled. Student data from other nations were excluded to maintain the uniformity of the sample. The entire dataset was anonymized from the beginning.

Among the data, only responses from ROK students were sampled exclusively to maintain sample uniformity. Additionally, items that had never been responded to by Korean students were removed. All the procedures thus far were conducted using IBM SPSS Statistics Data Editor version 26.

3.3.2 Procedure of Data Preprocessing

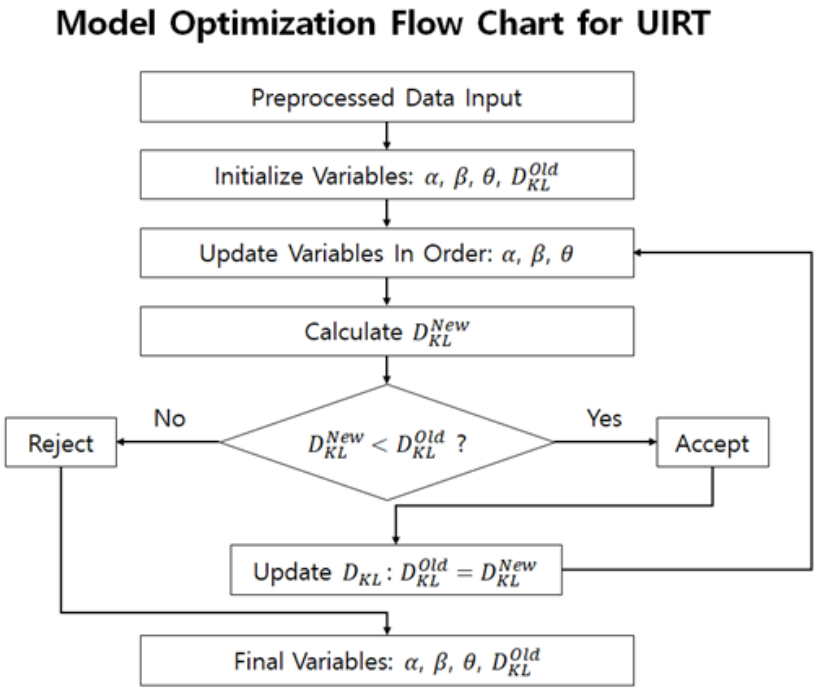
After the data selection, additional preprocessing was necessary to remove items with partial scores, which do not conform to the binary case. Additionally, data from examinees with no response were eliminated. As a result, the initial dataset of 52 items and 6650 examinees were refined to a dataset of 51 items and 2727 examinees.

The preprocessing was performed using Anaconda Jupyter Notebook version 3.6.

3.4. Model Optimization Algorithm

After inputting the preprocessed data, the model optimization algorithm for both UIRT and IMIRT follows four major steps in common (**Figure 3-2**): initialization, updating variables, judging of newD_{KL}'s acceptance, and refining final variables. First, in the initialization step, β and θ of UIRT and d and θ_1 for IMIRT were set in a special manner. The percentages of correct answers for examinees and for items were collected separately. Then the initial β and d were generated from the log odds of the correct answer rates for items, while the initial θ and θ_1 were from the log odds of the correct answer rates for examinees.

Additionally, for the optimization of the IMIRT model, extra steps were required involving reprocessing of reference data Y_i^{μ} and the variables Q and θ_2. First, in the reprocessing from Y_i^{μ} to $Y_i^{\mu\prime}$, set 1 for correct responses, -1 for incorrect responses, and 0 for non-responses. Then, the combination, namely the interaction, of two correct responses or two incorrect responses yields 1, whereas the interaction of one correct response and another incorrect response yields -1. Second, in the initialization of the symmetric hollow matrix Q, all the off-diagonal elements were set to 0.5 to avoid double counting. Then, the initial Q does not differ the weight of interactions,



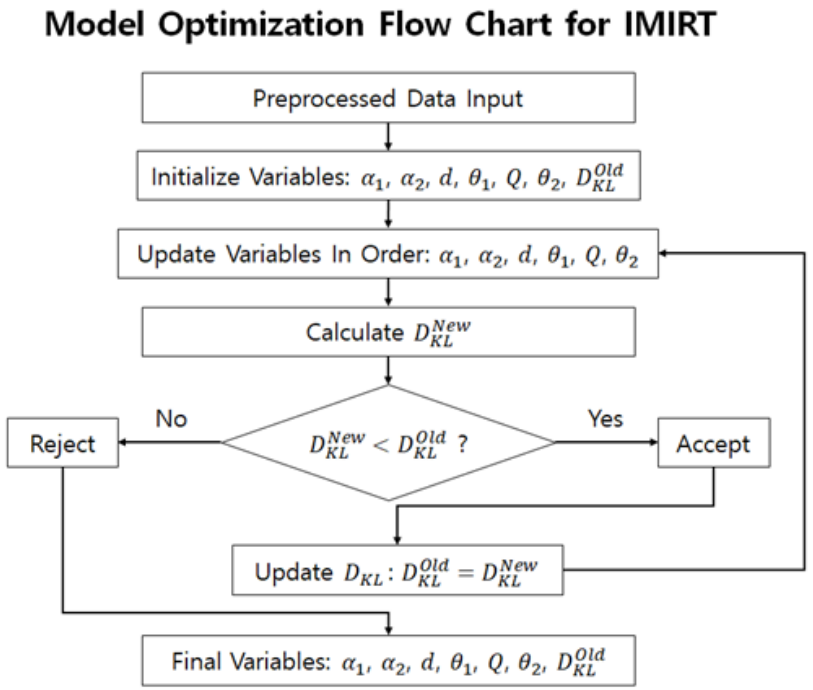


Figure 3-2. Model optimization flow charts for UIRT (Unidimensional Item Response Theory) and IMIRT (Ising Multidimensional Item Response Theory). Each flow chart consists of four major steps: initializing variables, updating variables, iteration, and finalizing variables. $\mathbf{D}_{KL}$ determines the continuation of the flow chart.

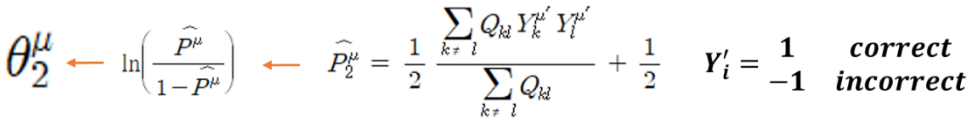


Figure 3-3. Conversion from Hamiltonian of Ising model into $\boldsymbol{\theta_2}$ of IMIRT (Ising Multidimensional Item Response Theory).

$Y_i^{\mu} Y_j^{\mu}$. Next, concerning the initialization of θ_2, the pseudo-probability, namely $\hat{P}^{\mu}$, is converted into θ_2 by a log-odds as equation (3.2). Before the conversion, the pseudo-probability is assembled with the weighted interaction, $\sum_{k \neq l} Q_{kl} Y_k^{\mu'} Y_l^{\mu'}$. Then, the weighted interaction undergoes normalization in order to set $0 \leq \hat{P}^{\mu} \leq 1$ as equation (3.1). The whole process of initializations is illustrated in Figure 3-3.

Second, the variables were updated using gradient descent, as shown below[①]:

$\alpha,\ \alpha_1,\ \alpha_1$: $$\alpha^{\mathrm{New}} = \alpha^{\mathrm{Old}} - \mathrm{A}\frac{\partial \mathrm{D_{KL}}}{\partial \alpha}, \quad (3.3)$$

β: $$\beta^{\mathrm{New}} = \beta^{\mathrm{Old}} - \mathrm{A}\frac{\partial \mathrm{D_{KL}}}{\partial \beta}, \quad (3.4)$$

d: $$\mathrm{d}^{\mathrm{New}} = \mathrm{d}^{\mathrm{Old}} - \mathrm{A}\frac{\partial \mathrm{D_{KL}}}{\partial \mathrm{d}}, \quad (3.5)$$

$\theta,\ \theta_1$: $$\theta^{\mathrm{New}} = \theta^{\mathrm{Old}} - \mathrm{A}\frac{\partial \mathrm{D_{KL}}}{\partial \theta}, \quad (3.6)$$

Q: $$\mathrm{Q}^{New} = Q^{Old} - A\frac{\partial D_{KL}}{\partial Q}. \quad (3.7)$$

Third, the iteration of the second process continued until the local

① Detailed calculation of variables by means of gradient descent is shown in Appendix A.

minimum was identified. If the newly calculated Kullback-Leibler divergence (D_{KL}^{New}) was larger than existing one(D_{KL}^{Old}), the last D_{KL}^{New} was rejected and the iteration stopped. Then the process proceeded to the next step.

Finally, the final variables were standardized to treat θ as a Z-score. The detailed formulas for standardization are as follows:

$1st\ \theta,\ \theta_1,\ \theta_2$: $$\theta^{std} = \frac{\theta - E[\theta]}{Std[\theta]}, \quad (3.\ 8)$$

2^{nd} set(UIRT): $$\alpha^{std}\left(\theta^{std} - \beta^{std}\right) = \alpha(\theta - \beta), \quad (3.\ 9)$$

2^{nd} set(IMIRT): $$\alpha_1^{std}\theta_1^{std} + \alpha_2^{std}\theta_2^{std} - d^{std} = \alpha_1\theta_1 + \alpha_2\theta_2 - d\,, \quad (3.\ 10)$$

$3^{rd}\ \alpha,\ \alpha_1,\ \alpha_2$: $$\alpha^{std} = Std[\theta]\,\alpha\,, \quad (3.\ 11)$$

$4^{th}\ \beta$: $$\beta^{std} = \frac{\beta - E[\theta]}{Std[\theta]}, \quad (3.\ 12)$$

$4^{th}\ d$: $$d^{std} = d - E[\theta_1]\,\alpha_1 - E[\theta_2]\,\alpha_2\,. \quad (3.\ 13)$$

3.5. Algorithm with Train/Test Splitting for Performance Verification

At this stage, two additional steps were introduced: sampling without replacement for the test set and calculating the Kullback-Leibler divergence of the train set and test set separately. Regarding the sampling, the number of items to which each examinee responded was taken into account. From the reference data, 758 examinees

responded to 18 items, 617 examinees to 16 items, 447 examinees to 17 items, 227 examinees to 15 items, and so on. The total number of combinations of responded items and corresponding examinees was 40,586. 3953 combinations, approximately 10% of the total combinations, were then sampled without replacement to generate the test set, while the remaining combinations formed the train set. The entire sampling process was initially conducted for UIRT, and the results of the sampling were subsequently shared with IMIRT.

Next, in terms of calculating D_{KL}, both the train set and the test set were utilized. Nonetheless, only the D_{KL} of the train set was considered to determine the continuation of training iteration. The D_{KL} of the test set would be collected to assess the explanatory power of the models.

Finally, in regard to the imputation performance comparison, the agreement ratio for each model was calculated. To perform the calculation, if the probability of an item being correct exceeded 0.5, the item was converted to a correct response value of 1. Additionally, a stricter agreement ratio was calculated for each model. Specifically, only when the difference between the probability and the reference data was within 0.3, the item was considered correct with a value of 1. The converted data and the reference data were collected and then compared.

3.6. Data Analysis Procedure

A series of collected data was exploited to verify the superiority of the suggested model, IMIRT. First, upon completing the optimization, the D_{KL} values of both the existing UIRT and the new IMIRT would be compared. Second, after the train/test splitting process, a comparison of the imputation performance and the train-test graph of both models would be conducted. Finally, the meaning of variables suggested with IMIRT would be investigated to infer the reasons why the new IMIRT model outperformed the existing UIRT, along with their implications in psychometric and evaluation theory.

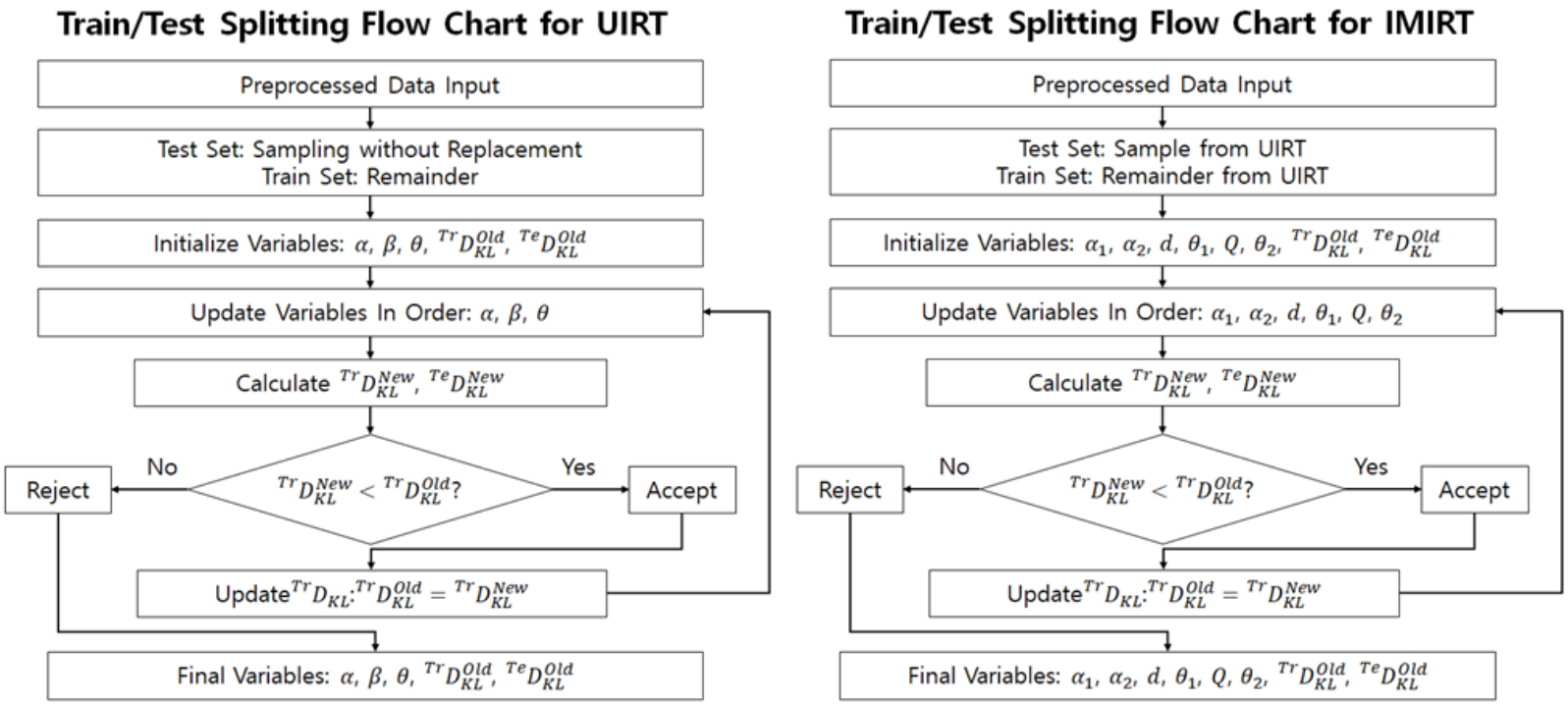


Figure 3-4. Train/test splitting flow charts for UIRT (Unidimensional Item Response Theory) and IMIRT (Ising Multidimensional Item Response Theory). In the flow chart, sampling for test set is inserted. Only ${}^{Tr}\boldsymbol{D}_{\boldsymbol{KL}}$ determines the continuation of the iteration.

Chapter 4. Result and Discussion

4.1. Improvement by IMIRT Model

4.1.1 Precise Level Diagnosis by IMIRT

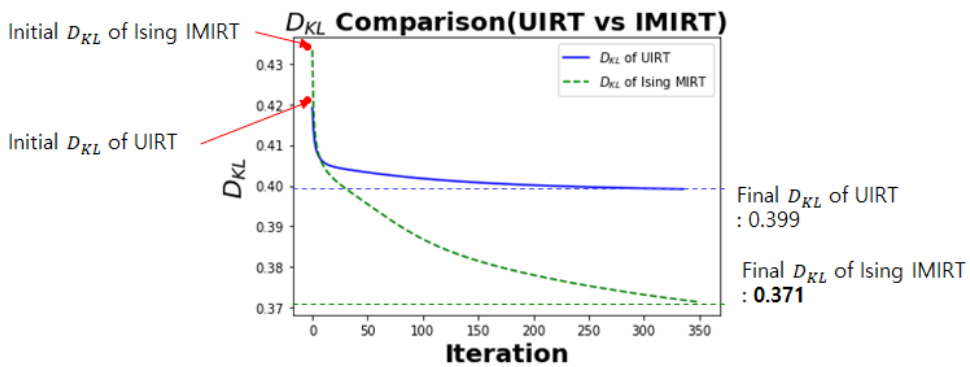


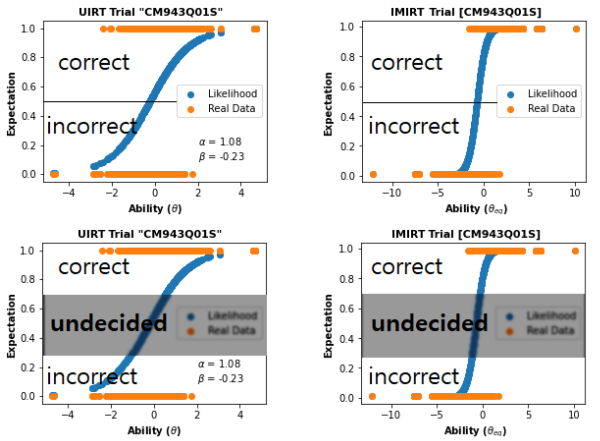
Figure 4-1. Quantitative comparison of model fitting between UIRT (Unidimensional Item Response Theory) and IMIRT (Ising Multidimensional Item Response Theory) with regard to $\mathbf{D_{KL}}$

Regarding the degree of model fitting, Kullback-Leibler divergence(D_{KL}) was selected as the criterion ②. After the optimization algorithm, the D_{KL} values of IMIRT and UIRT were

② As a model reaches the reference data closer, Kullback-Leibler divergence decreases.

compared in **Figure 4−1**. To explain in detail, the D_{KL} was calculated by averaging all the individual D_{KL}s of corresponding combinations consisting of an item and an examinee. At first, the UIRT model, with the initial D_{KL} value 0.419, appeared to be more efficient, compared to IMIRT model, with the initial D_{KL} value 0.433. However, as expected, the IMIRT model surpassed the UIRT model during the optimization process. As a result, the final D_{KL} of IMIRT reached 0.371, while the D_{KL} of UIRT 0.399. The quantitative result indicated the merit of IMIRT over UIRT in terms of model fitting.

4.1.2 Accuracy of Imputation by IMIRT



Accordance Ratios (%)

Bias \ Model	UIRT	IMIRT
0.5	74.62	76.65
0.3	55.22	60.36

where, $\theta_{eq} = \theta_1 + \frac{\alpha_2}{\alpha_1}\theta_2$

Figure 4-2. The illustration of imputation performance by means of accordance ratio (right table), and the criteria for accordance ratio (left graphs). The first criterion (upper graphs) sorted correct responses for an expectation value over 0.5. The second criterion (lower graphs) sorted correct responses for an expectation value over 0.7 and a bias less than 0.3. For the second criterion, expectations between 0.3 and 0.7 were considered undecided and excluded from the accordance ratio calculation.

The results shown in **Figure 4-2** once again indicate the superiority of IMIRT. The agreement ratio of IMIRT, 76.65%, outperformed that of UIRT, which was 74.62%. This represents an improvement of 2.03%.

Furthermore, for the stricter criterion with a bias of less than 0.3, the improvement was more significant. IMIRT achieved an agreement ratio of 60.36%, while UIRT of 55.22%. In this case, the improvement was enlarged to 5.14%. Given that the second criterion, a bias of less than 0.3, is stricter, the higher agreement ratio under the second criterion may reveal the higher quality of imputation accuracy. Then, IMIRT was suggested to own higher quality of imputation accuracy than UIRT.

Therefore, these results suggest that IMIRT not only improves the quantity of imputation but also the quality of imputation compared to UIRT.

4.1.3 Power of Explanation of IRT Improvement by IMIRT

Regarding the train sets of both models in **Figure 4-3**, D_{KL} gradually decreased as the iterations progressed. However, there was a difference in the trend of D_{KL} progression for the test sets. The UIRT train result indicated overfitting as the D_{KL} of the UIRT test set retrogressed against the D_{KL} of the UIRT train set. Meanwhile, the D_{KL} progression for the IMIRT test set gradually followed the trend of the D_{KL} of the IMIRT train set, with a slight rebound.

The fact that overfitting diminishes the power of explanation of a model suggests that the power of explanation of the UIRT model has

been compromised. Conversely, the absence of this phenomenon in the IMIRT model indicates its relatively superior power of explanation.

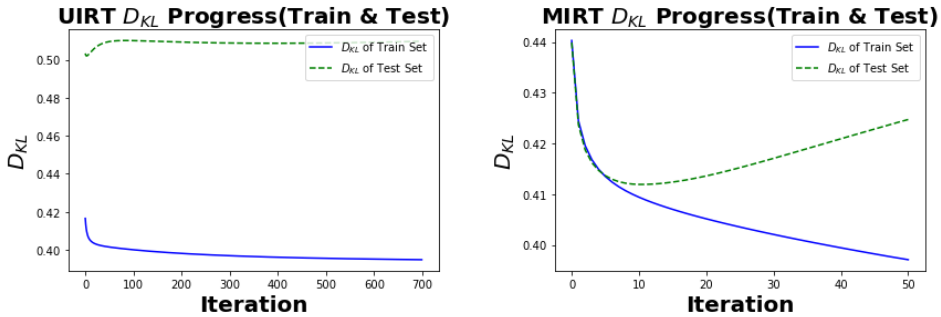


Figure 4-3. Progress of $\mathbf{D_{KL}}$ of train sets (blue line) and test sets (green dotted line) of UIRT (Unidimensional Item Response Theory) and IMIRT (Ising Multidimensional Item Response Theory). The $\mathbf{D_{KL}}$ of UIRT test set retrogressed then sidle along (left), whereas the $\mathbf{D_{KL}}$ of IMIRT softly landed first along with the train set and then rebounded slightly (right).

In summary, all the aspects of model fitting result, imputation performance and power of explanation reinforce the superiority of IMIRT.

4.2. Meaning of Parameters

4.2.1 Meaning of the New $\boldsymbol{\theta}$

Regarding the θ of IMIRT, there are two components: θ_1 and θ_2. To understand the meaning of this new θ, it is necessary to compare it with the θ of UIRT.

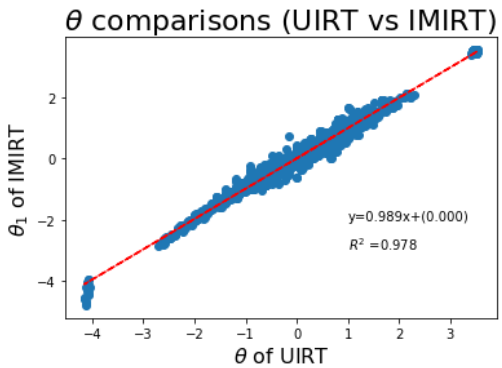


Figure 4-4. Correlation between **θ** of UIRT (Unidimensional Item Response Theory) and $\boldsymbol{\theta_1}$ of IMIRT (Ising Multidimensional Item Response Theory). The graph consists of the 2727 examinees (blue dots) and a trend line (red dotted line). The slope of the trend line is 0.989. The $\mathbf{R^2}$ value of the correlation is 0.978.

First, in regard to IMIRT θ_1, a significant correlation with UIRT θ was observed. As an example, the R^2 value of 0.978 indicates that θ_1 and θ are practically identical. Therefore, it can be concluded that IMIRT θ_1 is well-qualified to be regarded as the index of ability location as UIRT θ.

Second, regarding the parameter IMIRT θ_2, an intriguing pattern was observed in Figure 4-5 to a certain extent. For the initial case, namely before model fitting, the graph of θ_1 and θ_2 shows a parabolic pattern. Meanwhile, after model fitting, the graph of θ_1 and θ_2 exhibits a boomerang-shaped pattern. Fortunately, those patterns are reasonable as two graphs shows that low-level examinees tend

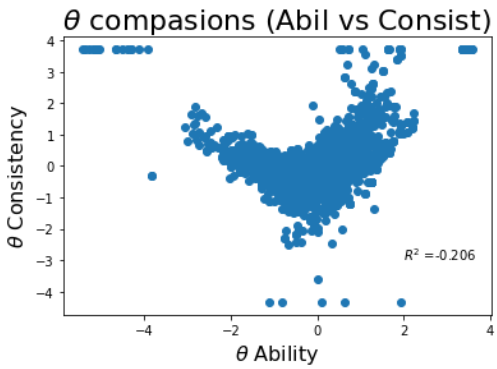


Figure 4-5. Distributions of examinees with respect to θ_1 (Ability) and θ_2 (Consistency) of IMIRT (Ising Multidimensional Item Response Theory) before model fitting (left) and after model fitting (right). The distribution of initial θs forms a parabolic shape (left), whereas the distribution of θs after model fitting appears as a dispersed boomerang shape (right).

to exhibit high-level consistency, similar to high-level examinees. The distinctive point is that θ_2 of post model fitting appears to scatter the examinees in the middle-level and upper-middle-level range from the initial pattern the most. Comparing the two graphs in Figure 4-5, it is certain to recognize the distinctive point.

As a result, it is possible to tentatively conclude that θ_2 has potential to differentiate the distribution of combinations (θ_1, θ_2) among similar abilities. Therefore, θ_2 can be denominated as the response consistency.

The principle underlying the segregation by θ_2 is proposed as a qualitative explanation with the aid of the diagrams shown in Figure 4-6 and Figure 4-7. Based on the chart presented above, when there

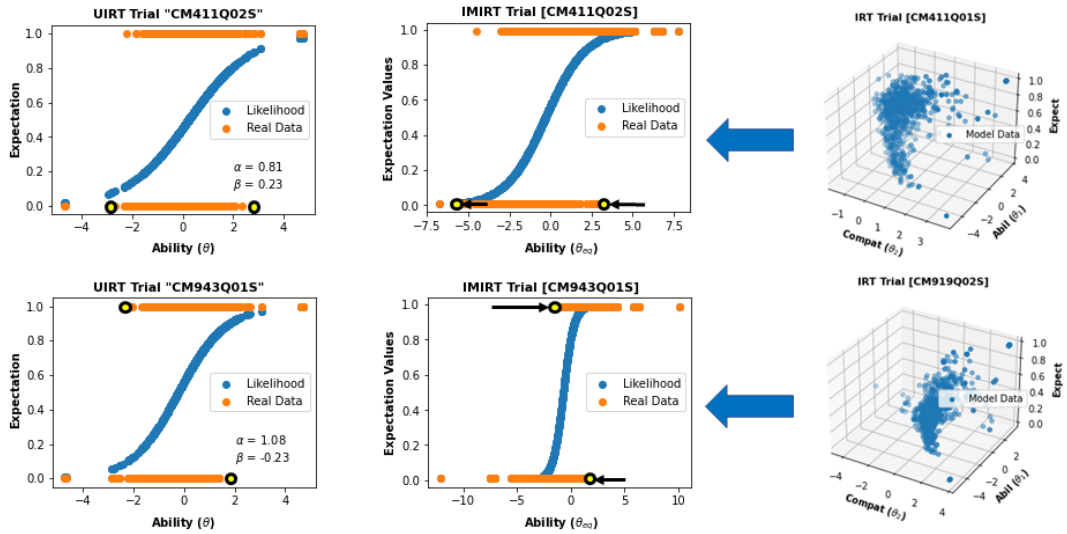


Figure 4-6. 2D the graphs (left) for the transition of position of reference data (yellow dots) and the data from corresponding expectation values (blue dots) and 3D graphs of IMIRT (Ising Multidimensional Item Response Theory) before transition. In the middle 2D graph, the positions of reference data of examinees who missed the item are shown as indicated by the red arrows. For the lower case (item code: CM919Q02S), the positions of reference data of examinees who missed the item progressed. On the right side, the transition from the 3D graph to the 2D graph is depicted, as indicated by the blue arrows. The transition follows the relation: $\boldsymbol{\theta_{eq}} = \boldsymbol{\theta_1} + \frac{\alpha_2}{\alpha_1}\boldsymbol{\theta_2}$

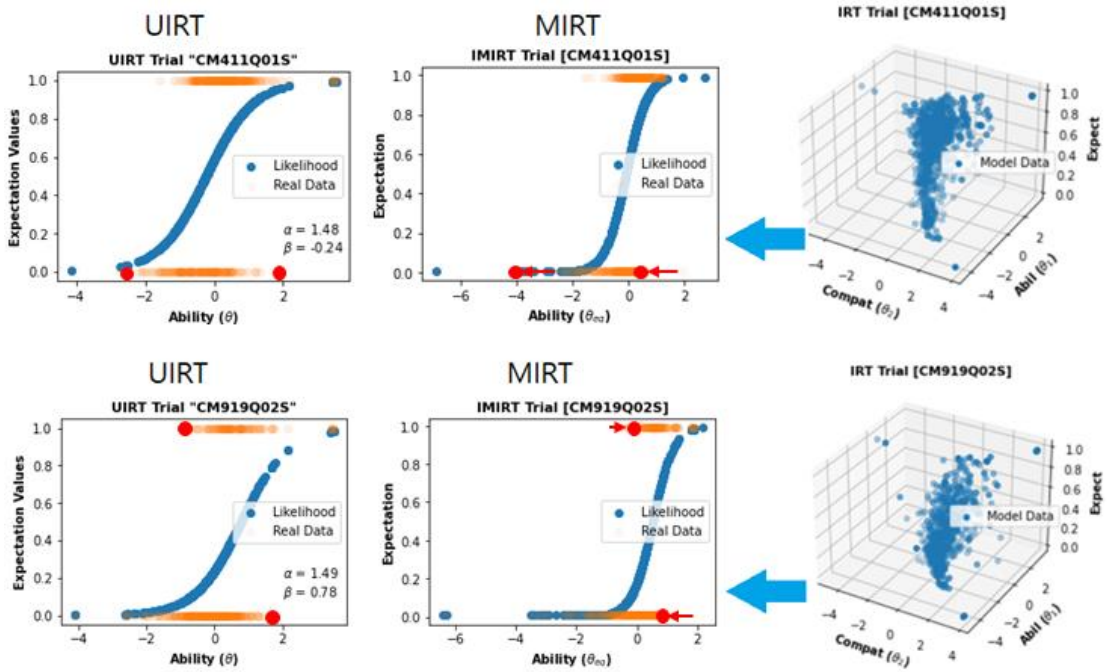


Figure 4-7. 2D the graphs (left) for the transition of position of reference data (red dots) and the data from corresponding expectation values (blurred blue dots) and 3D graphs of IMIRT (Ising Multidimensional Item Response Theory) before transition. The position of reference data of examinees who missed or correct items unexpectedly was adjusted as red arrows indicated. The clearance of orange dots represents the density of reference data distribution.

is a scenario of one correct response to one item and one incorrect response to another item, there is a deduction of points by '−1'. This trend suggests that θ_2 imposes a penalty for answer inconsistency.

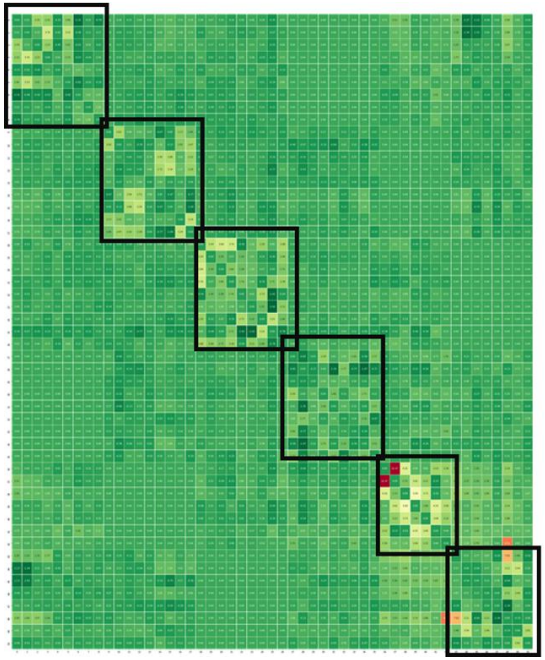
Furthermore, it is expected that θ_2 would aid in distinguishing sincere responses from wild guessing. Since wild guessing often leads to inconsistent answers, namely low consistency, θ_2 is anticipated to highlight this characteristic.

Before conducting the actual analysis of **Figure 4−6** and **Figure 4−7**, the concept of the converted ability, θ_{eq}, was introduced. θ_{eq} is defined as below:

$$\theta_{eq} = \theta_1 + \frac{\alpha_2}{\alpha_1}\theta_2 . \qquad (4.14)$$

After introducing of θ_{eq}, it is indeed possible to conduct a qualitative analysis through a direct comparison with UIRT. The yellow dots and black arrows in **Figure 4−6**, and red dots and red arrows in **Figure 4−7** illustrate cases where θ_2 becomes relevant. In both example items, θ_2 is assumed to suppress incorrect responses and push them towards the left. Additionally, for the item "CM919Q02S", it was observed that correct responses tend to shift towards the right.

In summary, IMIRT θ_1, one of the new variables, is virtually identical to the existing variable UIRT θ. Whereas the response consistency, θ_2, serves various roles: segregating ties and discerning wild guessing among answers.



Item Groups	Item Code PISA	Item Type	Sorted Groups	Item Number	Collected Items
M1	CM033Q01S	single select	M1	0	CM033Q01S
	CM474Q01S	single select		1	CM474Q01S
	DM155Q02C	constructive(H)			
	CM155Q01S	multiple select		2	CM155Q01S
	DM155Q03C	constructive(H)			
	CM155Q04S	multiple select		3	CM155Q04S
	CM411Q01S	constructive(C)		4	CM411Q01S
	CM411Q02S	single select		5	CM411Q02S
	CM803Q01S	constructive(C)		6	CM803Q01S
	CM442Q02S	multiple select		7	CM442Q02S
	DM462Q01C	constructive(H)			
	CM034Q01S	constructive(C)		8	CM034Q01S
M2	CM305Q01S	single select	M2	9	CM305Q01S
	CM496Q01S	multiple select		10	CM496Q01S
	CM496Q02S	constructive(C)		11	CM496Q02S
	CM423Q01S	single select		12	CM423Q01S
	CM192Q01S	multiple select		13	CM192Q01S
	DM406Q01C	constructive(H)			
	DM406Q02C	constructive(H)			
	CM603Q01S	multiple select		14	CM603Q01S
	CM571Q01S	single select		15	CM571Q01S
	CM564Q01S	single select		16	CM564Q01S
	CM564Q02S	single select		17	CM564Q02S
M3	CM447Q01S	single select	M3	18	CM447Q01S
	CM273Q01S	multiple select		19	CM273Q01S
	CM408Q01S	multiple select		20	CM408Q01S
	CM420Q01S	multiple select		21	CM420Q01S
	CM446Q01S	constructive(C)		22	CM446Q01S
	DM446Q02C	constructive(H)			

Figure 4-8. Diagram of the distribution of **Q** depicted by 51 X 51 matrix (left) and the contrast table of PISA 2018 reference data and data-driven block tendency (right). The scale of **Q** ranges from -1.62 to 12.47. The six black squares indicate the block tendency of the interaction among items. The item groups categorized by blocks are identical to the item groups categorized in the reference research report.

4.2.3 Meaning of the Parameter Q

Interaction term: $$\sum_{k\neq l} Q_{kl}\, Y'^{\mu}_{k} Y'^{\mu}_{l}\,. \qquad (4.\ 15)$$

Considering the effect of the new interaction term originated from the Hamiltonian of the Ising model, it is significant to analyze the identity of Q, a weight parameter.

In **Figure 4-8**, a series of block tendencies is observed, forming six minor off-diagonal square matrices. This block tendency implies that items may interact exclusively with adjacent items within the same block.

In reality, according to PISA 2018 research report, mathematics proficiency is categorized into 6 levels: M1, M2, M3, M4, M5, M6A[③]. It has been observed that the range of each proficiency level aligns closely with the block tendency identified. However, there is one exception, $Q_{42,48}$, which deviates from the overall block tendency. Despite this exception, Q can still be used to track the items that each examinee personally responded to, with only minor discrepancies.

Meanwhile, it should be noted that the Hamiltonian of the Ising model and the interaction term in the IMIRT model **(4.4)** are not strictly identical. In the context of the Ising model, the spins of a material flip due to interactions with adjacent spins. However, in the context of the IMIRT model, the responses of the reference data

[③] Some parts of the categorization are shown in the **Figure 4-8**. The whole categorization and the check list are enumerated in **Appendix C**.

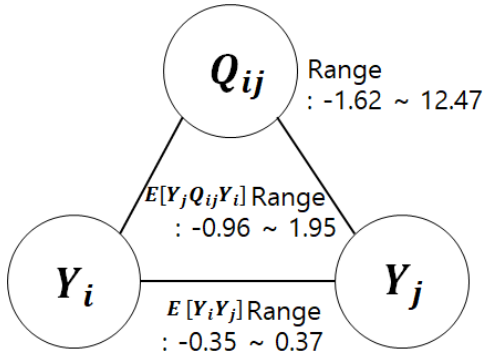


Figure 4-9. Correlation triangle scheme between item responses (Y_i, Y_j) and Q. $E[Y_iY_j]$ ranges -0.35~0.37, $E[Y_jQ_{ij}Y_i]$ -0.96~1.95, Q -1.62~12.47

never flip by interactions with adjacent responses.

On the other hand, both the Ising model and the IMIRT model allow for the alteration of the interaction parameter, such as Q in the IMIRT model. When the magnitude of Q is changed, it also affects the impact of interactions between adjacent items. For example, if Q is positive, analogous to a ferromagnetic interaction, it strengthens the effect of interactions. Conversely, if Q is negative, analogous to an anti-ferromagnetic interaction, it reverses the effect of interactions. Consequently, the scale of the interaction between two adjacent items, expressed as $E[Y_iY_j]$, is amplified to $E[Y_jQ_{ij}Y_i]$ by Q as illustrated in Figure 4-9.

Searching for the identity of Q, a hint can be suggested from the Riemannian geometry. In the Riemannian geometry, which is applied in General Relativity, it is possible for the Riemannian metric to distort vectors of Euclidean space. Similarly, it is feasible for Q to

distort the connection between two responses Y.

In summary, several aspects of the complex nature of Q have been discovered. First, Q represents the interaction between two adjacent items, analogous to the interaction in the Ising model. Second, the block tendency of Q can be excavated by data-driven approach. Finally, Q plays a role in distorting the correlation between two items. Then, Q has the potential to both intensify and diminish the correlation between two interacting items.

Chapter 5. Conclusion

To summarize, this study aimed to investigate whether the IMIRT model, which applies the Hamiltonian of the Ising model, can enhance the performance of IRT. In particular, the introduction of interaction among item responses implied the potential of merit. Specifically, the IMIRT model outperformed the existing UIRT model in terms of model fitting, imputation, and explanatory power. Additionally, this study examined the significance of the newly suggested variables and parameters, namely θ_1, θ_2 and Q, to understand the underlying reasons for the performance improvement of the IMIRT model. The findings suggest that θ_2 is proposed to represent the response consistency of examinees. That θ_2 segregated innocent responses from wild guessing is assumed to contribute to the advance. Finally, Q is identified as a factor that distorts the correlation between two items and it exhibited a block tendency with minor exceptions.

Afterwards, we propose two follow-up subjects for further exploration of general consistency factors. First, it is possible to expand the category of interactions among item responses. In this study, we only introduced the interactions between two item

responses for the IMIRT model. Then, the influence by interactions among three or more item responses is required to be explored. Second, we will explore general consistency factors on the scope of data-driven approaches as well as model-driven approaches. The block tendency of Q had confirmed the potential of data-driven approaches. Then, application of data-driven approaches is expected to contribute to discover new aspects of general consistency factors. Thus, these subjects are worth of exploring in future studies.

Appendix A. Detailed Derivations of Formula

A.1. Basic Information of Kullback-Leibler Divergence

Kullback-Leibler divergence ($D_{KL}(Y||P)$), also known as relative entropy, quantifies the disparity between the probability distribution of the model (P) and the reference probability distribution (Y). In the binary case, Kullback-Leibler divergence, serving as an objective function, is defined as follows:

$$D_{KL}(Y||P) \equiv Y\ln\frac{Y}{P} + (1-Y)\ln\frac{(1-Y)}{(1-p)}. \qquad \text{(A. 16)}$$

Kullback-Leibler divergence is always non-negative. This property is also called Gibb's inequality:

$$D_{KL}(Y||P) \geq 0\,. \qquad \text{(A. 17)}$$

Kullback-Leibler divergence equals zero if and only if $Y = P$, indicating that Y is identical to P. The inequality implies that minimizing Kullback-Leibler divergence allows the model to approach the real data more closely.

A.2. Probability Distribution and Variables

The probability distribution of the Multi-dimensional item response model takes the form of a sigmoid function as shown below:

$$P\left(Y_i^{\mu}=1\middle|\boldsymbol{\alpha_i}, d_i, \boldsymbol{\theta}^{\mu}\right)=\left[1+\exp\left(-\boldsymbol{\alpha_i}\cdot\boldsymbol{\theta}^{\mu}+d_i\right)\right]^{-1}. \quad \text{(A. 18)}$$

where $\boldsymbol{\alpha_i}\cdot\boldsymbol{\theta}^{\mu}=\alpha_{i,1}\theta_1^{\mu}+\alpha_{i,2}\theta_2^{\mu}$. And '$Y_i^{\mu}=1$' means that the μth examinee corrected the ith item. Meanwhile, θ_2 has the two step route for assembly as below:

1st step
$$\widehat{P}^{\mu}=\frac{1}{2}\sum_{k\neq l}\frac{Q_{kl}\,Y'^{\mu}_{k}Y'^{\mu}_{l}}{\Sigma_{k'\neq l'}Q_{k'l'}}+\frac{1}{2}, \quad \text{(A. 19)}$$

2nd step
$$\theta_2=\ln\left(\frac{\widehat{P}^{\mu}}{1-\widehat{P}^{\mu}}\right). \quad \text{(A. 20)}$$

where $\widehat{P}^{\mu}$ is a pseudo probability with $0\leq\widehat{P}^{\mu}\leq 1$, and k' and l' are index of items without missing data. In addition, if μth examinee corrects the kth item, then $Y'^{\mu}_{k}=1$. If not, then $Y'^{\mu}_{k}=-1$.

A.3. Detailed Procedures of Calculations for Model Optimization

To minimize the objective function, appropriate variables such as $\boldsymbol{\alpha}$, d, $\boldsymbol{\theta}$, are required for P to fit Y. By calculating the argument minimum of the objective function, it will be possible to determine the variables as follows:

$$\underset{\boldsymbol{\alpha},\mathrm{d},\boldsymbol{\theta}}{\operatorname{argmin}} D_{KL}(Y||P)\,. \tag{A. 21}$$

However, finding the argument minimum of the objective function analytically is convoluted. Therefore, it is plausible to suggest a numerical method such as Gradient Descent. Using Gradient Descent, the optimized variables of $\boldsymbol{\alpha}$, d, $\boldsymbol{\theta}$ are explored step by step.

First, in order to search the optimized α, the derivative should be calculated as follows:

$$\alpha^{new} = \alpha^{old} - A\frac{\partial D_{KL}}{\partial \alpha}, \tag{A. 22}$$

where A is the learning rate.

To perform the calculation, complex calculations of the partial derivative term should be conducted as follows:

$$\frac{\partial D_{KL}}{\partial \alpha} = \frac{\partial D_{KL}}{\partial P}\frac{\partial P}{\partial \alpha}, \tag{A. 23}$$

where
$$\frac{\partial D_{KL}}{\partial P} = \frac{P - Y}{P(1-P)}, \tag{A. 24}$$

and
$$\frac{\partial P}{\partial \alpha} = \theta P(1-P)\,. \tag{A. 25}$$

Then, the partial derivatives of the objective function of the whole data with respect to both α_1 and α_2 are given as follows:

$$\frac{\partial D_{KL}}{\partial \alpha_{i,1}} = \sum\nolimits_{\mu} \theta_1 (P_i^{\mu} - Y_i^{\mu})\,, \tag{A. 26}$$

$$\frac{\partial D_{KL}}{\partial \alpha_{i,2}} = \sum\nolimits_{\mu} \theta_2 (P_i^{\mu} - Y_i^{\mu})\,, \tag{A. 27}$$

where $\alpha_{i,1}$ and $\alpha_{i,2}$ are the α_1 and the α_2 of the ith item respectively,

θ_1^{μ} and θ_2^{μ} are the θ_1 and the θ_2 of the μth examinee respectively.

Second, in order to search for the optimized d, the derivative should be calculated as follows:

$$\mathrm{d}^{\mathrm{new}} = \mathrm{d}^{\mathrm{old}} - \mathrm{A}\frac{\partial \mathrm{D_{KL}}}{\partial \mathrm{d}}, \tag{A. 28}$$

where A is the learning rate.

Then, the partial derivatives of the objective function of the whole data with respect to d is given as follows:

$$\frac{\partial \mathrm{D_{KL}}}{\partial \mathrm{d}} = \frac{\partial \mathrm{D_{KL}}}{\partial \mathrm{P}}\frac{\partial \mathrm{P}}{\partial \mathrm{d}}, \tag{A. 29}$$

where,
$$\frac{\partial \mathrm{D_{KL}}}{\partial \mathrm{P}} = \frac{\mathrm{P}-\mathrm{Y}}{\mathrm{P}(1-\mathrm{P})}, \tag{A. 30}$$

and
$$\frac{\partial \mathrm{P}}{\partial \mathrm{d}} = -\mathrm{P}(1-\mathrm{P})\,. \tag{A. 31}$$

Then, the partial derivative of the objective function of the whole data by d is given as follows:

$$\frac{\partial \mathrm{D_{KL}}}{\partial \mathrm{d}} = \sum\nolimits_{\mu} -(\mathrm{P}_{\mathrm{i}}^{\mu} - \mathrm{Y}_{\mathrm{i}}^{\mu})\,. \tag{A. 32}$$

Third, in order to search for the optimized θ_1, the derivative should be calculated as follows:

$$\theta_1^{\mathrm{new}} = \theta_1^{\mathrm{old}} - \mathrm{A}\frac{\partial \mathrm{D_{KL}}}{\partial \theta_1}, \tag{A. 33}$$

where, A is the learning rate.

Then, the partial derivatives of the objective function of the whole data with respect to θ_1 is given as follows:

$$\frac{\partial D_{KL}}{\partial \theta_1} = \frac{\partial D_{KL}}{\partial P}\frac{\partial P}{\partial \theta_1}, \quad \text{(A. 34)}$$

where,
$$\frac{\partial D_{KL}}{\partial P} = \frac{P - Y}{P(1-P)}, \quad \text{(A. 35)}$$

and
$$\frac{\partial P}{\partial \theta_1} = \alpha P(1-P). \quad \text{(A. 36)}$$

Then, the partial derivatives of the objective function of the whole data by θ_1 is given as follows:

$$\frac{\partial D_{KL}}{\partial \theta_1^{\mu}} = \sum\nolimits_i \alpha_{i,1}(P_i^{\mu} - Y_i^{\mu}), \quad \text{(A. 37)}$$

where θ_1^{μ} is the θ_1 of the μth examinee, $\alpha_{i,1}$ and $\alpha_{i,2}$ are the α_1 and the α_2 of the ith item respectively.

Finally, in order to search for the optimized θ_2, the derivative of Q should be conducted as follows:

$$Q^{new} = Q^{old} - A\frac{\partial D_{KL}}{\partial Q}, \quad \text{(A. 38)}$$

where A is the learning rate.

To perform the calculation, complex calculations of the partial derivative term need to be conducted as shown below:

$$\frac{\partial D_{KL}}{\partial Q} = \frac{\partial D_{KL}}{\partial P}\frac{\partial P}{\partial \theta}\frac{\partial \theta}{\partial \widehat{P}}\frac{\partial \widehat{P}}{\partial Q}, \quad \text{(A. 39)}$$

where
$$\frac{\partial D_{KL}}{\partial P} = \frac{P - Y}{P(1-P)}, \quad \text{(A. 40)}$$

$$\frac{\partial P}{\partial \theta} = \alpha_2 P(1-P), \quad \text{(A. 41)}$$

$$\frac{\partial \theta}{\partial \widehat{P}} = \frac{1}{\widehat{P}(1-\widehat{P})}, \tag{A. 42}$$

And $$\frac{\partial \widehat{P}}{\partial Q} = \frac{Y'^{\mu}_{k} Y'^{\mu}_{l} - 2\widehat{P} + 1}{2\,\Sigma_{k' \neq l'} Q_{k'l'}}. \tag{A. 43}$$

Then, the partial derivative of the objective function with respect to Q for the entire dataset is given as follows:

$$\frac{\partial D_{KL}}{\partial Q} = \sum_{i,\mu} \frac{\alpha_{i,2}(P_i^{\mu} - Y_i^{\mu})}{\widehat{P}^{\mu}(1-\widehat{P}^{\mu})} \left(\frac{Y'^{\mu}_{k} Y'^{\mu}_{l} - 2\widehat{P}^{\mu} + 1}{2\,\Sigma_{k' \neq l'} Q_{k'l'}} \right), \tag{A. 44}$$

where $\widehat{P}^{\mu}$ is a pseudo probability with $0 \leq \widehat{P}^{\mu} \leq 1$, and k and l are index of items without missing data. In addition, if μth examinee corrects the kth item, then $Y'^{\mu}_{k} = 1$. If not, then $Y'^{\mu}_{k} = -1$. $\alpha_{i,2}$ is the α_2 of the ith item.

θ_2 can be updated with the newly learned Q using equation (A. 5).

Appendix B. Detailed Algorithms for Sampling, Variable $\boldsymbol{\theta_2}$ Fitting of Ising MIRT embodied by Python

B.1. Sampling without Replacement to Generate Train Set and Test Set

```
def simple_random(num_residues, num_division):       # Number
Distribution in Random

    result = []
    count = 0

    for i in range(num_division):
        if count < num_residues:
            result.append(1)
        else:
            result.append(0)
        count += 1

    random.shuffle(result)
    result_np = np.array(result)

    return result_np        # return is yielded in numpy form

def random_colrow_extractor(df_bf_gagong, df_pray_gagong, rate_sam):
# df_pray_gagong is of pandas, list_cols is of list.

    cols_num_samp = []       # the number of samples for each item
    coord_list = []
    ind_n = 0

    df_decay_train = df_bf_gagong.drop(['NS'], axis=1)
    df_decay = df_pray_gagong.drop(['NS'], axis=1)
    list_cols = basket_column.copy()
```

```
    row_min = df_decay.shape[0]
    col_min = df_decay.shape[1]

    num_sam = math.trunc(tot_num_ref * rate_sam)     # tot_num_ref is
universal variable.

    # To distribute samples for each item
    how_quotient = num_sam // col_min
    how_residue = num_sam % col_min

    num_dist_col = simple_random(how_residue, col_min) + how_quotient
    num_dist_rsh = num_dist_col.reshape(1,col_min)
    num_dist_col_pd = pd.DataFrame(num_dist_rsh)
    num_dist_col_pd.columns = list_cols[:51]

    # To distribute samples for each examinee
    how_quotient_mu = num_sam // row_min
    how_residue_mu = num_sam % row_min

    num_dist_row = simple_random(how_residue_mu, row_min) +
how_quotient_mu
    num_dist_rshr = num_dist_row.reshape(row_min,1)
    num_dist_row_pd = pd.DataFrame(num_dist_rshr,
index=df_decay.index.tolist())

    # data for test set
    data_collect = []
    coord_col = []
    coord_row = []
    row_col_val = []

    # result for test set
    basket_trial_np = np.zeros((rows,columns))
    basket_trial_nan = np.where(basket_trial_np == np.nan,
basket_trial_np, np.nan)
    basket_test = pd.DataFrame(basket_trial_nan)
    basket_test.columns = list_cols[:51]

    # shuffle examinee's index
    shf_index = df_decay.index.tolist().copy()
    random.shuffle(shf_index)

    for mu in shf_index:

        col_decay = list_cols[:51].copy()

        for j in list_cols[:51]:
            if np.isnan(df_decay.loc[mu][j]):
                col_decay.remove(j)
            elif num_dist_col_pd.loc[0][j] == 0:
                col_decay.remove(j)
```

```
        col_decay_len = len(col_decay)
        num_col_pick = num_dist_row_pd.loc[mu][0]
        picked = simple_random(num_col_pick, col_decay_len)
        picked_np = np.array(picked)
        loc_picked = np.where(picked_np == 1)[0]

        for nm in loc_picked:
            col_picked = col_decay[nm]
            coord_col.append(col_picked)
            coord_row.append(mu)
            row_col_val.append(df_decay.loc[mu][col_picked])
            num_dist_col_pd.loc[0][col_picked] -= 1
            df_decay_train.loc[mu][col_picked] = np.nan

            basket_test.loc[mu][col_picked] =
df_decay.loc[mu][col_picked]

    data_collect.append(coord_row)
    data_collect.append(coord_col)
    data_collect.append(row_col_val)
    data_collect_np = np.array(data_collect)

    return df_decay_train, basket_test, data_collect_np      # processed
train set, test set and the set of coordinates of test set

# sampling responses to test set

basket_ini = pd.concat([num_dfdf, p_solves], axis=1)   # nametagging of
num_dfdf

num_dfdf_stunt = num_dfdf.copy()    # num_dfdf's understudent
num_dfdf_stunt.columns = fil4.columns.to_list()

basket_column = fil4.columns.to_list()
basket_column.append('NS')     # NS stands for 'N'umber of the 'S'olved
problems

basket_ini.columns = basket_column

gagong_univ1 = basket_ini.copy()
#gagong_univ21 = gagong_univ1[gagong_univ1['NS'] >= 3]
#gagong_univ31 = gagong_univ21.notnull().sum()

less_2 = []

for i in range(rows):
    if basket_ini['NS'][i] <= 15:
        less_2.append(i)

print(less_2)
```

```
basket_sel = basket_ini.copy()
basket_sel.drop(less_2, axis=0, inplace=True)

tot_num_ref = int(gagong_univ1.sum()[-1])

train_gagongs = []
test_gagongs = []
num_iter = 10

for i in range(num_iter):

    num_df_gagong, test_set_gagong, test_set_coord =
random_colrow_extractor(basket_ini, basket_sel, 0.1)
    # 'Gumeong' mean 'a hole' in Korean.

    train_gagongs.append(num_df_gagong)
    test_gagongs.append(test_set_gagong)
```

B.2. List of Functions for Updating $\boldsymbol{\theta_2}$ Only

```
# Both samjin_data and Q_let are of numpy.    'samjin' means 'trinary'
in Korean.
def Shell_gagong(samjin_data, Q_let):

    num_gagong = samjin_data.copy()
    rows_let = num_gagong.shape[0]
    columns_let = num_gagong.shape[1]

    shell_list = []

    for i in range(rows_let):
        garo_pre = num_gagong[i, :]  # response vector(Y) of 1D. 'garo'
means 'horizon' in Korean.
        garo_T = np.reshape(garo_pre, (columns_let, 1))  # vertical form
        sero = garo_T.copy()    # 'sero' means 'the vertical' in Korean.
        garo = np.transpose(garo_T)
        shell_rough = sero * garo  # 2D matrix of all the combination of
Y_i Y_j (symmetric)

        carrier = Q_let * shell_rough    # 2D matrix with intensity Q
        np.fill_diagonal(carrier, 0)     # off-diagonal
        shell_list.append(carrier)

    shell_result = np.array(shell_list)  # The result is yielded in 3-
Rank Tensor

    return shell_result
```

```
# The function to generate ingredient for pseudo-probability from Ising
Hamiltonian
# Gagong_data is of pandas and Q_let is of numpy.
def answer_covari_bfsum(gagong_data, Q_let):

    num_gagong_bf = gagong_data.to_numpy()
    rows_let = num_gagong_bf.shape[0]
    columns_let = num_gagong_bf.shape[1]
    Yij_shell_let = Yij_shell.copy()

    Q_np = Q_let.copy()

    # Conversion of (1,0) binary data into (1,-1) binary data (NaN is
transformed to zero)
    # Refinement for avoiding 'divided by zero' error
    num_gagonged_bf = np.where(num_gagong_bf == 0.01, -0.99,
num_gagong_bf)
    num_gag_pd = pd.DataFrame(num_gagonged_bf)
    num_gag_fna = num_gag_pd.fillna(0)
    num_gagonged_np = num_gag_fna.to_numpy()

    p_bfsum = Shell_gagong(num_gagonged_np, Q_np)       # the numberator
before sum of the formula above

    #--------simple sum up ------- Normalization down -----------------
    # generation of denominator of the formula above

    denomin = []
    for i in range(rows):
        bf_Qsam = Yij_shell[i] * Q_let
        af_Qsam = bf_Qsam.sum()
        denomin.append(af_Qsam)

    P2_carrier = p_bfsum.copy()          # 3-Rank Tensor

    # Ingredient of pseudo-probability
    for i in range(rows_let):
        if denomin[i] == 0:
            P2_carrier[i] = 0 * P2_carrier[i] # Get rid of the
information of examinees who solved only one item.
        else:
            P2_carrier[i] = P2_carrier[i] / denomin[i]

    return P2_carrier


# Generation of pseudo-probability is accomplished in the end.
# Gagong_data is of pandas and Q_let is of numpy.
def answer_covari_afsum(gagong_data, Q_let):

    # collection of the whole ingredient
```

```
    p_bfsum = answer_covari_bfsum(gagong_data, Q_let)
    gagong_np = gagong_data.to_numpy()
    rows_let = gagong_np.shape[0]
    columns_let = gagong_np.shape[1]

    covari_ini = p_bfsum.sum(axis=2)
    covari_mid = covari_ini.sum(axis=1)
    covari_carry = np.reshape(covari_mid, (rows_let, 1))  # keep the
vertical shape
    # pseudo-probability of range between 0 and 1

    mid_result = (49/98.01) * (covari_carry) + 0.5  # refinement
avoding 'divided by zero' error

    # refinement avoding 'divided by zero' error
    scarub = np.where(mid_result > 0.99, 0.99, mid_result)
    scourge = np.where(scarub < 0.01, 0.01, scarub)
    P2_result = scourge

    return P2_result       # pseudo-probability of numpy form

# The function to calculate the derivative of KLD by Q
# Gagong_data is of pandas the others are of numpy.
def Q_deriv(alp1, alp2, d_let, tht1, tht2, Q_let, gagong_data):

    num_gagong_bf = gagong_data.to_numpy()
    rows_let = num_gagong_bf.shape[0]
    columns_let = num_gagong_bf.shape[1]
    Yij_shell_let = Yij_shell.copy()

    Q_np = Q_let.copy()
    Q_nuul = Q_halves.copy() # The initialized Q matrix of Universality

    # Conversion of (1,0) binary data into (1,-1) binary data (NaN is
transformed to zero)
    # Refinement for avoiding 'divided by zero' error
    num_gagonged_bf = np.where(num_gagong_bf == 0.01, -0.99,
num_gagong_bf)
    num_gag_pd = pd.DataFrame(num_gagonged_bf)
    num_gag_fna = num_gag_pd.fillna(0)
    num_gagonged_np = num_gag_fna.to_numpy()

    p_bfsum_nossi = Shell_gagong(num_gagonged_np, Q_nuul)
    p_bfsum = Shell_gagong(num_gagonged_np, Q_np) # Before calculation

#---------------------------division line----------------------------#
    # generation of denominator of the formula above
    denomin = []
    for i in range(rows):
        bf_Qsam = Yij_shell[i] * Q_let
        af_Qsam = bf_Qsam.sum()
```

```
        denomin.append(af_Qsam)

    P2_carrier1 = p_bfsum_nossi.copy()
    P2_carrier20 = p_bfsum.copy()          # 3-Rank Tensor

    # The 1st term of the numerator
    for i in range(rows_let):
        if denomin[i] == 0:
            P2_carrier1[i] = 0 * P2_carrier1[i]
        else:
            P2_carrier1[i] = P2_carrier1[i] / denomin[i]

    # The 2nd term of the numerator
    for i in range(rows_let):
        if denomin[i] == 0:
            P2_carrier20[i] = 0 * P2_carrier20[i]
        else:
            P2_carrier20[i] = P2_carrier20[i] / (denomin[i] *
denomin[i])

    covari2_ini = P2_carrier20.sum(axis=2)
    covari2_mid = covari2_ini.sum(axis=1)
    P22_part = np.reshape(covari2_mid, (rows_let, 1))   # keep the
vertical shape

    P2_list = []
    for i in range(rows_let):
        carrier = Yij_shell_let[i] * P22_part[i]
        P2_list.append(carrier)

    P2_carrier2 = np.array(P2_list)

    return P2_carrier1, P2_carrier2 # former: the 1st term, latter: the
2nd term of the numerator


# The function to sum all the ingredient of the formula above in the
end
# Gagong_data is of pandas the others are of numpy.
def Q_learn(alp1, alp2, d_let, tht1, tht2, Q_let, gagong_data):

    Q_np_test = Q_let.copy()               # Matrix to be learned
    gagonged_data = gagong_data.to_numpy()
    rows_let = gagonged_data.shape[0]
    columns_let = gagonged_data.shape[1]

    # the chain of the derivative: 3-Rank Tensor form
    P2_mu = answer_covari_afsum(gagong_data, Q_np_test)
    Normed_Y = (49/98.01) * (Q_deriv(alp1, alp2, d_let, tht1, tht2,
Q_let, gagong_data)[0] - Q_deriv(alp1, alp2, d_let, tht1, tht2, Q_let,
gagong_data)[1])
```

```
#----------------------------division line--------------------------#
    # common part
    com_pt = preprocess_diff(alp1, alp2, d_let, tht1, tht2,
gagong_data)

    # calculation start
    common_unit_np = com_pt * alp1     # 2-dimensional Matrix

    common_unit_T = np.transpose(common_unit_np)  # mu for axis=1; in
order to link mu with 3-Rank Tensor
    decoy_1st = pd.DataFrame(common_unit_T)
    decoy_2nd = decoy_1st.fillna(0)
    common_unit = decoy_2nd.to_numpy()

#-----------------Now, it's time to build a 4-Rank tensor ------------#

    P_hat_list = []    # Initialize the list to store a 4-Rank Tensor
    P_hat_3D = []      # Initialize the list to store a 3-Rank Tensor
    carrier_2D = []

    for i in range(columns_let):
        for j in range(columns_let):
            for mu in range(rows_let):
                carrier = common_unit[:, mu] * Normed_Y[mu, i, j] /
(P2_mu[mu, 0] * (1 - P2_mu[mu, 0]))
                carrier_2D.append(carrier)
            P_hat_3D.append(carrier_2D) # combination of mu and k
components is added.
            carrier_2D = []              # Reset the 2D matrix
        P_hat_list.append(P_hat_3D)     # complete the ith component
        P_hat_3D = []                    # Reset the 3-Rank Tensor

    P_hat_np = np.array(P_hat_list)     # complete the 4-Rank Tensor

    #Then, sum it up in terms of k and mu axes.
    # KLD Gradient Discent
    Q_pre = P_hat_np.sum(axis=3)        # sum it up in mu axis
    Q_presum = Q_pre.sum(axis=2)        # sum it up in k axis

    # Final Gradient Descendent: update
    Q_med = Q_np_test - A * Q_presum
    np.fill_diagonal(Q_med, 0)
    Q_result = Q_med/(2 * Q_med.mean()) # Normalization: the average of
all the component should be 0.5.

    return Q_result # The result is yielded in 2D matrix of numpy form.

# the function to update theta_2
# Only the gagong_data is given in pandas.
# Theta_2 is updated via the imaged process above.
```

```
def set_theta_Q(gagong_data, Q_let):

    rate_result = answer_covari_afsum(gagong_data, Q_let)
    theta_result = np.log((rate_result)/(1 - rate_result))

    return theta_result       # The result is yielded in numpy form.
```

B.3. Iteration Process for $\mathbf{D}_{KL}$ Calculation of Train and Test Set

```
albetheQKLD = []
num_iter = 0
#train_trial = []
#train_trial.append(train_gagongs[0])

#for gagong_carrier in train_trial:
for gagong_carrier in train_gagongs:
    carrier_shell = []

    num_dfdf = gagong_carrier.copy()
    p_df = num_dfdf.copy()
    num_np = num_dfdf.to_numpy()

    # theta_1 initialization
    row_pre = p_df.mean(axis=1)
    row_prob_1 = row_pre.to_numpy()
    row_prob = np.reshape(row_prob_1, (rows,1))

    theta_1 = np.log(row_prob/(1-row_prob))

    # d initialization
    col_pre = p_df.mean(axis=0)
    col_prob_1 = col_pre.to_numpy()
    col_prob = np.array([col_prob_1])
    d0 = np.log(col_prob/(1-col_prob))
    d = np.mean(d0) - d0

    # alpha_1 and alpha_2 initialization
    alpha = np.ones((1,columns))

    A = 0.005            # learning rate

    # transformation of (1,0) binary responses into (1,-1)
binary responses
    num_exp1 = num_np.copy()
```

```
    num_exp2 = np.where(num_exp1 == 0.01, -0.99, num_exp1) #
transformation

    num_exp_df = pd.DataFrame(num_exp2)
    num_exp_af = num_exp_df.fillna(0)       # get rid of NaN
    num_exp_np = num_exp_af.to_numpy()

    # Q initialization
    Q_np_ini = np.ones((columns, columns))
    np.fill_diagonal(Q_np_ini, 0)
    Q_halves = Q_np_ini / 2

    # theta_2 initialization
    shell_list = []

    for i in range(rows):
        garo_pre = num_exp_np[i, :]
        garo_T = np.reshape(garo_pre, (columns, 1))   # vertial
vector form
        sero = garo_T.copy()
        garo = np.transpose(garo_T)
        carrier = sero * garo
        np.fill_diagonal(carrier, 0) # off-diagonal

        shell_list.append(carrier)

    shell_ini = np.array(shell_list) # initial combination of
Y_iY_j

    # the reference to indicate the location of solved items
    Y_solved0 = num_np.copy()
    Y_solved1 = np.where(Y_solved0 == 0.01, 1, Y_solved0)
    Y_solved2 = np.where(Y_solved1 == 0.99, 1, Y_solved1)
    Y_pd = pd.DataFrame(Y_solved2)
    Y_fna = Y_pd.fillna(0)         # set NaN as zero
    Y_solved = Y_fna.to_numpy()

    Yij_solved = []

    for i in range(rows):
        garo_pre = Y_solved[i, :]
        garo_T = np.reshape(garo_pre, (columns, 1))
        sero = garo_T.copy()
        garo = np.transpose(garo_T)
        carrier = sero * garo
        np.fill_diagonal(carrier, 0)

        Yij_solved.append(carrier)

    Yij_shell = np.array(Yij_solved)
```

```
    denominator = []
    for i in range(rows):
        bf_Qsum = Yij_shell[i] * Q_halves
        af_Qsum = bf_Qsum.sum()
        denominator.append(af_Qsum)  # generation of the
denominator

    P_carrier = []     # basket for initial pseudo-probability
    for i in range(rows):
        garo_pre = num_exp_np[i, :]
        garo = np.reshape(garo_pre, (1, columns))
        sero_T = np.copy(garo)
        sero = np.transpose(sero_T)

        vectorman1 = sero * garo
        vectorman11 = Q_halves * vectorman1
        vectorman111 = vectorman11.sum(axis=1)
        vectorman2 = vectorman111.sum(axis=0)

        if denominator[i] == 0:
            P_mu = 0
        else:
            P_mu = vectorman2 / denominator[i]

        P_carrier.append(P_mu)

    P_norm = np.array(P_carrier)

    theta_pre = (49/98.01) * (P_norm) + 0.5    # final form of
pseudo-probability initialization

    # final initialization of theta_2
    theta1_bfT = np.log(theta_pre / (1 - theta_pre))
    theta_2 = np.reshape(theta1_bfT, (rows,1))

    # initialization of the probability distribution of the
model
    exp1 = alpha * theta_1
    exp2 = alpha * theta_2
    ex_prob = np.exp(exp1 + exp2 - d)/(1+np.exp(exp1 + exp2 -
d))

    ex_prob_real = ex_prob.copy()

    for n in range(ex_prob.shape[0]): # reflect the distribution
of NaN
        for m in range(ex_prob.shape[1]):
            if np.isnan(num_np[n][m]):
                ex_prob_real[n][m] = np.nan

    # KLD of each response
```

```
    KLD_indiv = num_np * np.log(num_np / ex_prob_real) + (1 -
num_np) * np.log((1 - num_np) / (1 - ex_prob_real))

    # get rid of missing values
    KLD_indiv_df = pd.DataFrame(KLD_indiv)
    KLD_NaNga_df = KLD_indiv_df.fillna(0)
    KLD_NaNga_np = KLD_NaNga_df.to_numpy()

    # KLD initialization
    KLD_RowSum = np.sum(KLD_NaNga_np, axis=1)
    KLD_TotalSum_np = np.sum(KLD_RowSum, axis=0)

    # Model Optimization Start
    alpha1_mod, alpha2_mod, d_mod, theta1_mod, theta2_mod,
Q_mod, KLDs_mod, KLDs_test_mod = opt_model(alpha, d, theta_1,
theta_2, Q_halves, p_df, test_gagongs[num_iter], 20)

    # save for further analysis
    carrier_shell.append(alpha1_mod)       # 0
    carrier_shell.append(alpha2_mod)       # 1
    carrier_shell.append(d_mod)            # 2
    carrier_shell.append(theta1_mod)       # 3
    carrier_shell.append(theta2_mod)       # 4
    carrier_shell.append(Q_mod)            # 5
    carrier_shell.append(KLDs_mod)         # 6
    carrier_shell.append(KLDs_test_mod)    # 7

    albetheQKLD.append(carrier_shell)
    num_iter += 1
```

Appendix C. Contrast Table of Item Codes with PISA 2018

Item Groups	Item Code PISA	Item Type	Sorted Groups	Item Number	Collected Items
M1	CM033Q01S	single select	M1	0	CM033Q01S
	CM474Q01S	single select		1	CM474Q01S
	DM155Q02C	constructive(H)			
	CM155Q01S	multiple select		2	CM155Q01S
	DM155Q03C	constructive(H)			
	CM155Q04S	multiple select		3	CM155Q04S
	CM411Q01S	constructive(C)		4	CM411Q01S
	CM411Q02S	single select		5	CM411Q02S
	CM803Q01S	constructive(C)		6	CM803Q01S
	CM442Q02S	multiple select		7	CM442Q02S
	DM462Q01C	constructive(H)			
	CM034Q01S	constructive(C)		8	CM034Q01S
M2	CM305Q01S	single select	M2	9	CM305Q01S
	CM496Q01S	multiple select		10	CM496Q01S
	CM496Q02S	constructive(C)		11	CM496Q02S
	CM423Q01S	single select		12	CM423Q01S
	CM192Q01S	multiple select		13	CM192Q01S
	DM406Q01C	constructive(H)			
	DM406Q02C	constructive(H)			
	CM603Q01S	multiple select		14	CM603Q01S
	CM571Q01S	single select		15	CM571Q01S
	CM564Q01S	single select		16	CM564Q01S
	CM564Q02S	single select		17	CM564Q02S
M3	CM447Q01S	single select	M3	18	CM447Q01S
	CM273Q01S	multiple select		19	CM273Q01S
	CM408Q01S	multiple select		20	CM408Q01S
	CM420Q01S	multiple select		21	CM420Q01S
	CM446Q01S	constructive(C)		22	CM446Q01S
	DM446Q02C	constructive(H)			
				23	CM559Q01S
				24	CM828Q03S
				25	CM464Q01S
	CM800Q01S	single select		26	CM800Q01S

M4	CM982Q01S	constructive(C)	M4	27	CM982Q01S
	CM982Q02S	constructive(C)		28	CM982Q02S
	CM982Q03S	multiple select		29	CM982Q03S
	CM982Q04S	single select		30	CM982Q04S
	CM992Q01S	constructive(C)		31	CM992Q01S
	CM992Q02S	constructive(C)		32	CM992Q02S
	DM992Q03C	constructive(H)			
	CM915Q01S	single select		33	CM915Q01S
	CM915Q02S	constructive(C)		34	CM915Q02S
	CM906Q01S	single select		35	CM906Q01S
	DM906Q02C	constructive(H)			
	DM00KQ02C	constructive(H)			
M5	CM909Q01S	constructive(C)	M5	36	CM909Q01S
	CM909Q02S	single select		37	CM909Q02S
	CM909Q03S	constructive(C)		38	CM909Q03S
	CM949Q01S	multiple select		39	CM949Q01S
	CM949Q02S	multiple select		40	CM949Q02S
	DM949Q01C	constructive(H)			
	CM00GQ01S	constructive(C)		41	CM00GQ01S
	DM955Q01C	constructive(H)			
	DM955Q02C	constructive(H)			
	CM955Q03S	constructive(C)			
	DM998Q02C	constructive(H)			
	CM998Q04S	multiple select		42	CM998Q04S
M6A	CM905Q01S	multiple select	M6A	43	CM905Q01S
	DM905Q02C	constructive(H)			
	CM919Q01S	constructive(C)		44	CM919Q01S
	CM919Q02S	constructive(C)		45	CM919Q02S
	CM954Q01S	constructive(C)		46	CM954Q01S
	DM954Q02C	constructive(H)			
	CM954Q04S	constructive(C)		47	CM954Q04S
	CM943Q01S	single select		48	CM943Q01S
	CM943Q02S	constructive(C)		49	CM943Q02S
	DM953Q02C	constructive(H)			
	CM953Q03S	constructive(C)		50	CM953Q03S
	DM953Q04C	constructive(H)			

Bibliography

Douglas Stone, Sheila Heen. Thanks for the Feedback: The Science and Art of Receiving Feedback Well Douglas Stone, Sheila Heen (0150).

성태제. (2009). 교육평가의 기초. 서울: 학지사.

Millman, J., & Arter, J. A. (1984). Issues in Item Banking. Journal of Educational Measurement, 21(4), 315–330.

교육부 (2022). 2022 개정 교육과정 총론 및 과학과 교육과정. 교육부 고시 제2022-33호. [별책 1], [별책 9].

박혜영, 김경희, 장의선, 김유향, 양영자, 김선희, 김현주 (2022). 고교학점제 도입에 따른 성취평가제 개선 방안. 한국교육과정평가원, 12(8).

김진숙 등 (2021). 고교학점제 도입을 위한 교육과정 개선 및 대입제도 개편 방향. 한국교육과정평가원, 1(3).

Bulut, O., & Kim, D. (2021). The use of data imputation when investigating dimensionality in sparse data from computerized adaptive tests. Journal of Applied Testing Technology.

Vale, C.D. (2004). Computerized item banking. In Downing, S.D., & Haladyna, T.M. (Eds.) The Handbook of Test Development. Routledge.

Lenz, W. (1920), "Beiträge zum Verständnis der magnetischen Eigenschaften in festen Körpern", Physikalische Zeitschrift, 21: 613–615.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. Zeitschrift für Physik, 31(1), 253-258.

Cai, L., Choi, K., Hansen, M., & Harrell, L. (2016). Item response theory. Annual Review of Statistics and Its Application, 3, 297-321.

Padilla-Camberos, E., Barragán-Álvarez, C. P., Diaz-Martinez, N. E., Rathod, V., & Flores-Fernández, J. M. (2018). Effects of Agave fructans (Agave tequilana Weber var. azul) on body fat and serum lipids in obesity. Plant foods for human nutrition, 73, 34-39.

Reckase, M. D. (2009). Multidimensional Item Response Theory. https://doi.org/10.1007/978-0-387-89976-3.

Selke, W. (1988). The ANNNI model—theoretical analysis and experimental application. Physics Reports, 170(4), 213-264.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Oakes, J. M. (2002). Risks and wrongs in social science research: An evaluator's guide to the IRB. Evaluation Review, 26(5), 443-479.

Akaike, H. (1974). A new look at the statistical model identification. IEEE Transactionson Automatic Control, 19 (6), 716–723. Math. Rev., 423716.

Kullback, S. and R. A. Leibler (1951). On information and sufficiency. Annals of Mathematical Statistics 22(1), 79–86.

Ying, X. (2019, February). An overview of overfitting and its solutions. In Journal of physics: Conference series (Vol. 1168, p. 022022). IOP Publishing.

조성민, 구남욱, 김현정, 이소연, 이인화. (2019. 12. 20). OECD 국제 학업성취도 평가 연구: PISA 2018 결과 보고서 (연구보고 RRE 2019-11). 한국교육과정평가원.

Cohen, O., Oberhardt, M., Yizhak, K., & Ruppin, E. (2016). Essential genes embody increased mutational robustness to compensate for the lack of backup genetic redundancy. Plos one, 11(12), e0168444.

M.F. Atiyah, The moment map in symplectic geometry.Durham Symposiumon Global Riemannian Geometry. Ellis Horwood Ltd. (1984), 43–51.

Gibbs, J. W. (1902). Elementary principles in statistical mechanics: developed with especial reference to the rational foundations of thermodynamics. C. Scribner's sons.

국 문 초 록

문항반응이론(Item Response Theory, IRT)은 문항과 사람 간 상호작용에 대한 일반적인 양상에 관한 이론이다. 문항반응이론은 문제은행 등 다양한 상황에서 활용된다. 뿐만 아니라 심리학 등 다양한 학문 영역에서 문항반응이론을 연구 방법론으로 채택하고 있다. 이처럼 문항반응이론은 학문적, 실용적 중요성을 지닌 것으로 평가할 수 있다.

문항반응이론은 실용성과 유연성의 측면에서 고전 시험 이론(Classical Test Theory, CTT)을 능가하는 것으로 평가할 수 있다. 다만, 문항반응이론은 문항과 사람 간 상호작용을 지나치게 단순화하였다는 점을 문제점으로 지적할 수 있다. 기존 문항반응이론은 시험 결과를 통한 학생의 수준 진단 신뢰도 및 추가 문제 추천 정확도 등에 한계를 가지고 있다. 이러한 한계로 인해 기존 문항반응이론이 평가-학생지도 간 연계성을 약화시킬 수 있다.

문항반응이론이 평가-학생지도 간 연계성을 강화시키기 위하여, 새로운 문항반응이론은 학생 수준 진단의 신뢰성 확보 및 결측치 예측(imputation) 성능 향상이 필요하다. 이를 위하여 본 연구는 문항-문항 간 상호작용에 주목하여 문항반응이론의 성능 향상을 도모하였다. 기존 문항반응이론은 문항-문항 간 상호작용을 간접적으로 반영하였으나, 새로운 문항반응이론은 문항-문항 간 상호작용을 직접 반영하였다. 이러한 상호작용을 본 연구에서 '응답정합성(response consistency)'으로 명

명하였다.

새로운 문항반응이론의 성능 향상 및 성능 검증을 위하여 기계학습 (machine learning) 방식을 도입하였다. 그 결과 새로운 문항반응이론은 응답정합성 도입을 통하여 더욱 일반화된 학생 수준 진단이 가능해졌다. 그리고 개선된 진단 결과를 바탕으로 더 높은 결측치 예측 성능을 보였다.

응답정합성은 문항-문항 간 상호작용을 통하여 정답을 아는 응답과 정답을 모르고 추측한 응답 간 변별력을 강화시킴으로써 문항반응이론의 성능을 향상시킨 것으로 평가할 수 있다. 한편, 응답정합성이 범주화 한 문제 묶음이 실제로 PISA 2018의 수준 체계 분류와 일치함을 확인할 수 있었다. 이로써 본 연구는 교육평가 영역에서도 데이터 기반 접근법 (data-driven approach) 도입의 가능성을 연 것으로 평가할 수 있다. 한편, 문항-문항 간 상호작용의 연구성과를 다문항 (multiple items) 간 상호작용으로 확대한다면, 응답정합성 개념 일반화에 한걸음 다가갈 수 있을 것으로 전망한다.

주요어: 응답정합성, 다차원 문항반응이론, 문제은행, 결측치 예측, 기계학습, 데이터 기반 접근법

학 번: 2021-28401