mathematics

MDPI

Article

A Deterministic Learning Algorithm Estimating the Q-Matrix for Cognitive Diagnosis Models

Meng-Ta Chung * and **Shui-Lien Chen**

Department of Management Sciences, Tamkang University, New Taipei City 251301, Taiwan; slchen@mail.tku.edu.tw
* Correspondence: 157333@mail.tku.edu.tw

Abstract: The goal of an exam in cognitive diagnostic assessment is to uncover whether an examinee has mastered certain attributes. Different cognitive diagnosis models (CDMs) have been developed for this purpose. The core of these CDMs is the Q-matrix, which is an item-to-attribute mapping, traditionally designed by domain experts. An expert designed Q-matrix is not without issues. For example, domain experts might neglect some attributes or have different opinions about the inclusion of some entries in the Q-matrix. It is therefore of practical importance to develop an automated method to estimate the Q-matrix. This research proposes a deterministic learning algorithm for estimating the Q-matrix. To obtain a sensible binary Q-matrix, a dichotomizing method is also devised. Results from the simulation study shows that the proposed method for estimating the Q-matrix is useful. The empirical study analyzes the ECPE data. The estimated Q-matrix is compared with the expert-designed one. All analyses in this research are carried out in R.

Keywords: Q-matrix; DINA; RRUM; CDM

check for updates

Citation: Chung, M-T..; Chen, S.-L. A Deterministic Learning Algorithm Estimating the Q-Matrix for Cognitive Diagnosis Models. *Mathematics* **2021**, *9*, 3062. https://doi.org/10.3390/math9233062

Academic Editor: Juan Ramón Torregrosa Sánchez

Received: 9 October 2021
Accepted: 22 November 2021
Published: 28 November 2021

1. Introduction

1.1. Cognitive Diagnosis Models

Cognitive diagnostic assessment (CDA) is a framework that intends to evaluate an examinee's mastery of a specific cognitive skill called an attribute [1]. A few cognitive diagnosis models (CDMs) have been developed, such as the deterministic input, noisy "and" gate (DINA) model [2], and the reduced reparameterized unified model (RRUM) [3,4].

The core of different CDMs is the Q-matrix [5]. Table 1 shows a simple example of the Q-matrix, which assesses whether an examinee has acquired addition and subtraction attributes. An examinee must possess an addition attribute in order to correctly answer the first item. Suppose that the attribute status of the examinee is $(1, 0)$. Without possessing the subtraction attribute, the examinee is expected to only answer item 1 correctly since items 2 and 3 require a subtraction attribute.

Table 1. A Q-matrix example.

Item	Attribute	
	Addition	Subtraction
(1). $4 + 5$	1	0
(2). $7 - 2$	0	1
(3). $6 + 3 - 1$	1	1

The DINA model and RRUM are two popular models based on the Q-matrix. The DINA model and RRUM are non-compensatory, assuming that examinees must have mastered all the required attributes to correctly answer an item.

Suppose the data is comprised of item responses from I examinees to J items that measure K attributes. In the DINA model, η_{ij} indicates whether examinee i $(i = 1, \cdots, I)$ can correctly answer item j $(j = 1, \cdots, J)$,

$$\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}},$$

where α_{ik} is the mastery of attribute k $(k = 1, \cdots, K)$ for examinee i and q_{jk} is the state of the jth item and kth attribute in the Q-matrix. The item response function (IRF) for the DINA model is:

$$P(X_{ij} = 1|\boldsymbol{\alpha}_i) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}},$$

where g_j and s_j are guess and slip parameters for item j, $\boldsymbol{\alpha}_i$ is the attribute status of examinee i. The guess parameter g_j represents the probability of $X_{ij} = 1$ when at least one required attribute is lacking, and the slip parameter s_j denotes the probability of $X_{ij} = 0$ when all required attributes are present.

Suppose the guess and slip parameters for item 1 in Table 1 are $g_1 = s_1 = 0.2$. If the attribute status of an examinee i is $\boldsymbol{\alpha}_i = (1, 0)$, then $\eta_{i1} = 1$. Therefore, the examinee's probability of correctly answering item 1 is $P\big(X_{ij} = 1|\boldsymbol{\alpha}_i = (1, 0)\big) = (1 - 0.2)^1 0.2^0 = 0.8$.

The RRUM is another popular CDM, especially in language assessment. The IRF of the RRUM is:

$$P(X_{ij} = 1|\boldsymbol{\alpha}_i) = \pi_j^* \prod_{k=1}^{K} (r_{jk}^{*(1-\alpha_{ik})})^{q_{jk}}, \tag{1}$$

where $\pi_j^* = \prod_{k=1}^{K} (1 - s_{jk})^{q_{jk}}$ and $r_{jk}^* = g_{jk}/1 - s_{jk}$. Note that g_{jk} and s_{jk} are guess and slip parameters of item j for attribute k. As in the DINA model, α_{ik} is the mastery of attribute k for examinee i and q_{jk} is the state of the jth item and kth attribute in the Q-matrix.

1.2. Model Based Estimation of the Q-Matrix

Most of the Q-matrices are not specified during exam development and are conventionally designed by domain experts after the fact. A more objective means to assign the Q-matrix is needed because domain experts might neglect some attributes or have different opinions. For instance, different modifications of the Q-matrix for the fraction subtraction data [6] have been suggested (e.g., Tatsuoka [6]; de la Torre [7]; de la Torre and Douglas [8]; de la Torre and Douglas [9]; Henson et al. [10]; Tatsuoka [11]) over the past three decades.

It is therefore of practical importance to develop an automated method for searching the Q-matrix. The purpose of this research is to develop a deterministic learning algorithm that estimates the Q-matrix for CDMs. It should be noted that this research attempts to develop a method to estimate the whole Q-matrix rather than to validate an existing one.

Estimating the whole Q-matrix is NP-hard [12]. The sample space of this discrete optimization can be huge. For example, simulation studies of this research try to recover a 15 by 4 Q-matrix (see Table 2), which gives $^{2^{60}}/_{4!}$ possible Q-matrices when column permutation is considered. Since the optimization problem can be simplified using model-based methods, most of the studies attempting to estimate the Q-matrix are CDM based.

For example, Chen et al. [13], Liu et al. [14], Liu et al. [15], and Xu and Shang [16] use the DINA model to estimate the Q-matrix. Chung [17] and Chung [18] show that both the DINA model and RRUM are both feasible models. Treating the Q-matrix as a parameter to estimate these studies assume that the true model is the DINA model or the RRUM. Although simulations in these studies suggest promising Q-matrix recovery rates, it is desirable to find non-model-based methods.

Table 2. Q-matrices for simulations.

Item	**Complete Q-Matrix**				**Incomplete Q-Matrix**			
	Attributes				**Attributes**			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
1	1	0	0	0	1	0	0	1
2	0	1	0	0	0	1	1	0
3	0	0	1	0	0	1	1	0
4	0	0	0	1	1	0	0	1
5	1	1	0	0	1	1	0	0
6	1	0	1	0	1	0	1	0
7	1	0	0	1	1	0	0	1
8	0	1	1	0	0	1	1	0
9	0	1	0	1	0	1	0	1
10	0	0	1	1	0	0	1	1
11	1	1	1	0	1	1	1	0
12	1	1	0	1	1	1	0	1
13	1	0	1	1	1	0	1	1
14	0	1	1	1	0	1	1	1
15	1	1	1	1	1	1	1	1

1.3. NMF Estimation of the Q-Matrix

Winters [19] was the first to apply non-negative matrix factorization (NMF) to explore the structure of the Q-matrix. Desmarais and his colleagues have also published a few studies (e.g., Desmarais [20]; Desmarais [21]; Desmarais and Naceur [22]; Desmarais et al. [23]) that apply NMF to estimating the Q-matrix. These studies applying NMF are inspiring, showing that Q-matrix estimation is in essence a matrix factorization.

While their findings are promising, some issues are worth considering. The primary concern is that the inclusion of an entry (1 or 0) is counted solely on visual inspection using heatmaps. While using visual inspection is desirable in some conditions, we would expect a more decisive means to avoid ambiguity. Another issue is that the way the initial values were retrieved is unclear, especially considering that NMF is very sensitive to initial values (Cichocki et al. [24]; Gond and Nadi [25]; Zheng et al. [26]). A more recent work by Casalino et al. [12] has shown that using the constraint alternating least square is a practical solution to a stable estimation.

Other issues are related to their simulation studies. No information is available for the different sample sizes and the correlations between attributes. In addition, each item measures at most two attributes in their Q-matrix for simulations when in reality it is not uncommon to have an items testing more than two attributes. We are also interested in whether NMF can recognize items measuring all attributes.

Inspired by Desmarais and his colleagues' research, this research offers a factorization algorithm for the Q-matrix, as well as refining simulation designs. Specifically, we use a maximum likelihood estimation and enforce a dichotomizing scheme in the estimation. We define a recovery rate to investigate the effect of the method on different sample sizes and correlations under complete and incomplete Q-matrix designs. In addition to the DINA model, the RRUM is also adopted in the simulation study.

2. A Deterministic Learning Algorithm for the Q-Matrix

2.1. Maximum Likelihood Estimation

Barnes et al. [27] and Desmarais [20] give the following equation that infers the item response ($\boldsymbol{X}$) as the product of the Q-matrix ($\boldsymbol{Q}$) and the attribute mastery matrix ($\boldsymbol{\alpha}$):

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{\alpha}^T, \tag{2}$$

where $\boldsymbol{X}$, $\boldsymbol{Q}$, and $\boldsymbol{\alpha}$ are binary with each element either 0 or 1. Estimating the Q-matrix turns into a factorization problem, which can be simplified if we temporarily treat $\boldsymbol{Q}$ as continuous.

This research proposes to freely derive $\boldsymbol{Q}$ by maximum likelihood estimation and then dichotomize the estimate to produce a binary Q-matrix. The advanced procedure is explained as the following. Suppose that a corresponding continuous version of $\boldsymbol{Q}$ is $\boldsymbol{R}$. The problem becomes to find a valid factorization of $\boldsymbol{X}$ that could provide an estimate of $\boldsymbol{R}$:

$$\boldsymbol{X} = \boldsymbol{R}\boldsymbol{\alpha}^T. \tag{3}$$

If the transformation in (3) is linear and nonsingular, then $\boldsymbol{\alpha}^T = \boldsymbol{R}^{-1}\boldsymbol{X}$. Using a Jacobian transformation, we can derive the pdf of $\boldsymbol{X}$:

$$p_X(\boldsymbol{X}) = \frac{1}{|\text{det}\boldsymbol{R}|} p_\alpha\left(\boldsymbol{\alpha}^T\right). \tag{4}$$

Let $\boldsymbol{M} = \boldsymbol{R}^{-1}$, and let p_i denote the pdf of $\boldsymbol{\alpha}_i$. The pdf of $\boldsymbol{X}$ in (4) becomes:

$$p_X(\boldsymbol{X}) = |\text{det}\boldsymbol{M}| p_\alpha\left(\boldsymbol{\alpha}^T\right) = |\text{det}\boldsymbol{M}| \prod_i p_i\left(\boldsymbol{\alpha}_i^T\right).$$

Suppose the Q-matrix measures K attributes. That is, $\boldsymbol{M} = (\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_K)^T$, and therefore:

$$p_X(\boldsymbol{X}) = |\text{det}\boldsymbol{M}| \prod_{i=1}^{K} p_i\left(\boldsymbol{m}_i^T \boldsymbol{X}\right). \tag{5}$$

Suppose that we have N observations, denoted $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots \boldsymbol{X}_N$. From the pdf shown in (5), the likelihood $L(\boldsymbol{M})$ is therefore:

$$L(\boldsymbol{M}) = \prod_{n=1}^{N} \prod_{i=1}^{K} p_i\left(\boldsymbol{m}_i^T \boldsymbol{X}_n\right) |\text{det}\boldsymbol{M}|,$$

and therefore the log-likelihood is:

$$\log L(\boldsymbol{M}) = \sum_{n=1}^{N} \sum_{i=1}^{K} \log p_i\left(\boldsymbol{m}_i^T \boldsymbol{X}_n\right) + N \log |\text{det}\boldsymbol{M}|. \tag{6}$$

Dividing the log-likelihood of (6) by N yields:

$$\frac{1}{N} \log L(\boldsymbol{M}) = E\left(\sum_{i=1}^{K} \log p_i\left(\boldsymbol{m}_i^T \boldsymbol{X}\right) + \log |\text{det}\boldsymbol{M}|\right). \tag{7}$$

It should be noted that the expectation will eventually be replaced by sample averages.

To find the learning algorithm for $\boldsymbol{M}$, we can maximize (7) by taking partial derivative with respect to $\boldsymbol{M}$:

$$\frac{1}{N} \frac{\partial \log L(\boldsymbol{M})}{\partial \boldsymbol{M}} = \frac{\partial E\left(\sum_{i=1}^{K} \log p_i\left(\boldsymbol{m}_i^T \boldsymbol{X}\right)\right)}{\partial \boldsymbol{M}} + \frac{\partial E(\log |\text{det}\boldsymbol{M}|)}{\partial \boldsymbol{M}}.$$

To simplify notation, we let $u_i = \boldsymbol{m}_i^T \boldsymbol{X}$ so that $\frac{\partial E\left(\sum_{i=1}^{K} \log p_i\left(\boldsymbol{m}_i^T \boldsymbol{X}\right)\right)}{\partial \boldsymbol{M}} = \frac{\partial \sum_{i=1}^{K} \log p_i(u_i)}{\partial \boldsymbol{M}}$. If we let $p(\boldsymbol{u}) = \prod_{i=1}^{n} p_i(u_i)$ and define $\varphi(\boldsymbol{u}) = -\frac{\frac{\partial p(\boldsymbol{u})}{\partial \boldsymbol{u}}}{p(\boldsymbol{u})}$ where $\varphi(\cdot)$ is the negative score function, Lee et al. [28] shows that:

$$\frac{\partial \sum_{i=1}^{K} \log p_i(u_i)}{\partial \boldsymbol{M}} = -\varphi(\boldsymbol{u})\boldsymbol{X}^T.$$

We now evaluate the other term, $\frac{\partial \log |\det \boldsymbol{M}|}{\partial \boldsymbol{M}}$. From Grossman [29], we know that:

$$\frac{1}{\boldsymbol{M}} = \frac{1}{\det \boldsymbol{M}} \mathrm{adj}(\boldsymbol{M}),$$

where $\mathrm{adj}(\boldsymbol{M})$ is the adjoint of $\boldsymbol{M}$. We can express $\det \boldsymbol{M}$ in terms of cofactors,

$$\det \boldsymbol{M} = \sum_{k=1}^{n} b_{ik} M_{ik}. \tag{8}$$

Taking partial derivative of (8) with respect to m_{ij} gives $\frac{\partial \det \boldsymbol{M}}{\partial m_{ij}} = m_{ij}$, which suggests that:

$$\frac{\partial \det \boldsymbol{M}}{\partial b_{ij}} = \mathrm{adj}(\boldsymbol{M})^{T}.$$

As $\mathrm{adj}(\boldsymbol{M})^{T} = (\det \boldsymbol{M})\left(\boldsymbol{M}^{T}\right)^{-1}$, we get:

$$\frac{\partial |\det \boldsymbol{M}|}{\partial \boldsymbol{M}} = \frac{\det \boldsymbol{M}}{\boldsymbol{M}^{T}},$$

so that:

$$\frac{\partial \log |\det \boldsymbol{M}|}{\partial \boldsymbol{M}} = \frac{1}{det|\boldsymbol{M}|} \frac{\partial |\det \boldsymbol{M}|}{\partial \boldsymbol{M}} = \frac{1}{\boldsymbol{M}^{T}}.$$

The gradient of the log-likelihood in (7) is therefore:

$$\begin{aligned}\frac{1}{N} \frac{\partial \log L(\boldsymbol{M})}{\partial \boldsymbol{M}} &= \frac{\partial E\left(\sum_{i=1}^{n} \log p_i\left(\boldsymbol{m}_i^{T} \boldsymbol{X}\right)\right)}{\partial \boldsymbol{M}} + \frac{\partial E(\log |\det \boldsymbol{M}|)}{\partial \boldsymbol{M}}.\\ &= -E\left(\varphi(\boldsymbol{M}\boldsymbol{X})\boldsymbol{X}^{T}\right) + \frac{1}{\boldsymbol{M}^{T}}\end{aligned}$$

For only one data point, the expectation is omitted. The learning algorithm is given by:

$$\Delta \boldsymbol{M} = -\varphi(\boldsymbol{M}\boldsymbol{X})\boldsymbol{X}^{T} + \frac{1}{\boldsymbol{M}^{T}}.$$

2.2. Rotation and Dichotomization

Inverse $\boldsymbol{M}$ in each iteration, and we can obtain $\boldsymbol{R}$. To enhance the interpretability, $\boldsymbol{R}$ entails an orthogonal rotation by varimax. The rotated $\boldsymbol{R}$ is denoted as $\boldsymbol{Q}$. As the Q-matrix is binary, a method to dichotomize $\boldsymbol{Q}$ is imperative. We devise the following dichotomizing scheme for converting $\boldsymbol{Q}$ to the binary Q-matrix.

Suppose the Q-matrix uses J items to measure K attributes. Each row in $\boldsymbol{Q}_{J \times K}$ is an item. Item j can be presented as $\boldsymbol{q}_j = (q_{j1}, q_{j2}, \cdots, q_{jK})$. Let the entry with the highest value in item j be q_{jmax}, namely, $q_{jmax} = max(q_{j1}, q_{j2}, \cdots, q_{jK})$. The value of each attribute in the item is then decided by the relative magnitude of its value to the highest one. That is, $q_j/q_{jmax} = (q_{j1}/q_{jmax}, q_{j2}/q_{jmax}, \cdots, q_{jK}/q_{jmax})$. For entry q_{jk} in item j, if $q_{jk}/q_{jmax} \geq a$, set q_{jk} to 1. If $q_{jk}/q_{jmax} < a$, then set q_{jk} to 0.

For example, suppose a Q-matrix measures 4 attributes, and values for certain item j is $\boldsymbol{q}_j = (0.111, 0.222, 0.333, 0.444)$. From $\boldsymbol{q}_j$, we find $q_{jmax} = 0.444$ and therefore $q_j/q_{jmax} = (0.111/0.444, 0.222/0.444, 0.333/0.444, 0.444/0.444) = (0.25, 0.5, 0.75, 1)$. If a is set to 0.6, then the derived Q-matrix state for item j is $(0, 0, 1, 1)$.

Different values of a (i.e., 0.4, 0.5, 0.6) are tested in this research. After this procedure is applied to every item, the whole Q-matrix is derived because $\boldsymbol{Q}_{J \times K}$ is now binary.

3. Simulation Study

We examine whether our proposed algorithm can recover the Q-matrix from simulated data. Settings for the simulation are the same as those in Chung [18]. The following simulation study is carried out using customized R codes.

3.1. Q-Matrix for Simulation

This simulation study adopts the artificial Q-matrix (the complete Q-matrix in Table 2) acquired from Rupp and Templin [30]. Fifteen items that measure four attributes comprise the Q-matrix, which is constructed such that each attribute appears alone from items 1 to 4, in a pair from items 5 to 10, or in a triad from items 11 to 14. It should be noted that item 15 measures all 4 attributes.

The complete Q-matrix in Table 2 contains at least one item devoted solely to each attribute [31]. That is, each attribute in the complete Q matrix in Table 2 is individually measured by at least one item (e.g., attribute 1 is individually tested by item 1).

This research also examines the effectiveness of our proposed algorithm in recovering an incomplete Q-matrix (the incomplete Q-matrix in Table 2), in which an item measures more than one attribute. Each item in the incomplete Q-matrix in Table 2 tests at least two attributes (e.g., item 1 tests attributes 1 and 4).

3.2. Generating Correlated Attributes

Chung [18] suggests generating correlated attributes using a copula with the Choleski decomposition. Suppose the Q-matrix (Q) uses J items to measure K attributes for I examinees. That is, $Q_{J\times K} = \left(q_{jk}\right)_{J\times K}$ and $\boldsymbol{\alpha}_{I\times K} = (\alpha_{ik})_{I\times K}$. Let $\boldsymbol{\vartheta}$ be the I by K underlying probability matrix of $\boldsymbol{\alpha}$, and let column k of $\boldsymbol{\vartheta}$ be vector $\boldsymbol{\vartheta}_k$, $k = 1,\ldots,K$. That is, $\boldsymbol{\vartheta} = (\boldsymbol{\vartheta}_1,\ldots,\boldsymbol{\vartheta}_K)$. The correlation coefficient for each pair of columns in $\boldsymbol{\vartheta}$ takes the constant value ρ , and the correlation matrix is represented as $\boldsymbol{\Sigma}$. Each entry in $\boldsymbol{\Sigma}$ corresponds to the correlation coefficient between two columns in $\boldsymbol{\vartheta}$. $\boldsymbol{\Sigma}$ can be further decomposed as $\boldsymbol{\Sigma} = \boldsymbol{\nu}^T\boldsymbol{\nu}$ using the Choleski decomposition, where $\boldsymbol{\nu}$ is an upper triangular matrix.

After $\boldsymbol{\nu}$ is derived, create an $I \times K$ matrix $\boldsymbol{\tau}$, in which each entry is generated from $N(0,1)$. $\boldsymbol{\tau}$ is then transformed to γ by using $\gamma = \boldsymbol{\tau}\boldsymbol{\nu}$, so that **fl** and $\boldsymbol{\Sigma}$ will have the same correlation structure. Set $\Phi(\gamma) = \boldsymbol{\vartheta}$, where $\Phi(\cdot)$ is the CDF of the standard normal distribution. $\boldsymbol{\alpha}$ is generated using inverse transform sampling. Create a matrix $\boldsymbol{\Theta}_{I\times K} = (\theta_{ik})_{I\times K}$, where each element is generated from $Uniform(0,1)$. If $\vartheta_{ik} \geq \theta_{ik}$, set α_{ik} to 1, and if $\vartheta_{ik} < \theta_{ik}$, set α_{ik} to 0.

3.3. Generating Data from the DINA Model and RRUM

With the Q-matrix and dependent attributes, data were simulated using the DINA model and the RRUM. Values for all guess and slip parameters are set to 0.2 for both the DINA model ($s_j = g_j = 0.2$) and RRUM ($s_{jk} = g_{jk} = 0.2$), and the data are then created using the inverse transform sampling from two points, in which the probability is obtained from the IRF of the DINA model and RRUM. Note that from Equation (1) for the RRUM, we simulate data using s_{jk} and g_{jk} instead of using reparameterized parameters π_j^* and r_{jk}^*.

We follow the following settings that appear in Chung [18] . Examinees in groups of 500, 1000, and 2000 were simulated with the correlation between each pair of attributes set to 0.1, 0.3, and 0.5 for both the DINA model and RRUM. One hundred datasets were simulated for each combination of sample size and correlation.

3.4. Evaluation

Evaluating how well the proposed method recovers the true Q-matrix $\boldsymbol{q}$, the recovery rate Δ suggested by [14] is defined as:

$$\Delta = \frac{1}{M}\sum_{m=1}^{M}\left(1 - \frac{\left|\hat{\boldsymbol{q}}^{(m)} - \boldsymbol{q}\right|}{JK}\right),$$

where $M = 100$, $|\cdot|$ takes the absolute value and $\hat{\boldsymbol{q}}^{(m)}$ stands for the estimated Q-matrix for mth dataset.

3.5. Results from Simulation

For both the DINA model and RRUM, the optimal cutoff value a is 0.5 (see Table 3). It should be noted that 0.5 is not an arbitrary cutoff. The value of 0.5 here means a half of the highest value. The following has stated Section 2.2: If $q_{jk}/q_{jmax} \geq a$, set q_{jk} to 1. If $q_{jk}/q_{jmax} < a$, then set q_{jk} to 0. The value of an attribute has to be more than a half of the highest value to be regarded as 1 in the Q-matrix.

Table 3. Average Δ from different a.

DINA		RRUM	
a	Δ	a	Δ
0.4	0.983	0.4	0.944
0.5	0.994	0.5	0.962
0.6	0.991	0.6	0.954

As $a = 0.5$ is optimal, Table 4 shows the recovery rate for the complete Q-matrix when $a = 0.5$. For the complete Q-matrix with $a = 0.5$, average Δ from all combinations in the DINA model is 0.994, suggesting an adequate result when the Q-matrix is complete although average Δ drops to 0.962 for the more complicated RRUM.

In terms of different sample sizes and correlations, Δ rises along with the increase of the sample size and drops when the correlation between attributes rises. Note that the number of attributes is supposed to be known in advance.

For different combinations of sample size and correlation, Δ declines when the correlation between attributes goes up, while it increases along with the increase of the sample size. Δ is higher when the data are simulated from the RRUM.

Table 4. Recovery rates of complete Q-matrix when $a = 0.5$.

N	DINA			RRUM		
	ρ			ρ		
	0.1	**0.3**	**0.5**	**0.1**	**0.3**	**0.5**
500	0.997	0.990	0.975	0.942	0.932	0.913
1000	0.999	0.998	0.992	0.985	0.974	0.954
2000	1.000	1.000	0.998	0.995	0.988	0.976

Note: N and ρ respectively stand for sample size and correlation coefficient.

When the Q-matrix is incomplete, Δ is unsatisfactory (see Table 5), ranging from 0.754 to 0.772. Such low recovery rates from an incomplete Q-matrix are similar to the findings in Chung [18].

Table 5. Recovery rates of incomplete Q-matrix when $a = 0.5$.

N	DINA			RRUM		
	ρ			ρ		
	0.1	**0.3**	**0.5**	**0.1**	**0.3**	**0.5**
500	0.792	0.772	0.751	0.781	0.761	0.740
1000	0.838	0.813	0.772	0.825	0.804	0.762
2000	0.849	0.831	0.797	0.834	0.822	0.789

Note: N and ρ respectively stand for sample size and correlation coefficient.

4. Empirical Study

Obtained from the CDM package in R [32], the Examination for the Certificate of Proficiency in English (ECPE) data that consists of 2922 examinees is analyzed using the proposed method described in Section 2. The ECPE is a test developed and scored by the English Language Institute of the University of Michigan [33].

The ECPE data have appeared in different studies, such as Buck and Tatsuoka [34], Henson and Templin [35], Templin and Hoffman [33], Feng, Habing, and Huebner [36], and Templin and Bradshaw [37].

Buck and Tatsuoka [34] suggest a Q-matrix consisting of 28 items that measure 3 attributes: Morphosyntactic rules, Cohesive rules, and Lexical rules (expert Q-matrix in Table 5). In addition, a parallel analysis was tentatively used to determine the number of attributes in the Q-matrix. The result also suggests 3 attributes.

We consequently decided to evaluate a 3-attribute Q-matrix solution. Exhibited in Table 6, the expert-designed and estimated Q-matrices are denoted Q_X and Q_E respectively. An inspection of Q_X and Q_E shows that each of them is a complete Q-matrix, and no two columns are identical. Comparing $Q_X$with Q_E, we find that 11 items are identical. Overall, 76.2% of the entries are identical. Like the expert designed Q-matrix, the estimated Q-matrix is complete. As for model fit, AICs for Q_X and Q_E are respectively 85,812.92 and 85,709.47, suggesting that the estimated Q-matrix (Q_E) better fits the data.

Table 6. Q-matrix estimate from the empirical study.

Item	Expert Q-Matrix (Q_X)			Estimated Q-Matrix (Q_E)		
	Attribute			Attribute		
	1	2	3	1	2	3
1	1	1	0	1	1	1
2	0	1	0	0	1	1
3	1	0	1	1	0	0
4	0	0	1	0	0	1
5	0	0	1	0	1	1
6	0	0	1	0	0	1
7	1	0	1	1	0	1
8	0	1	0	1	1	1
9	0	0	1	0	0	1
10	1	0	0	1	0	0
11	1	0	1	1	0	1
12	1	0	1	0	0	1
13	1	0	0	0	0	1
14	1	0	0	0	0	1
15	0	0	1	0	1	1
16	1	0	1	0	0	1
17	0	1	1	0	0	1
18	0	0	1	0	1	1
19	0	0	1	0	0	1
20	1	0	1	1	0	1
21	1	0	1	0	1	1
22	0	0	1	0	0	1
23	0	1	0	0	1	1
24	0	1	0	0	1	0
25	1	0	0	1	0	0
26	0	0	1	0	1	1
27	1	0	0	1	0	1
28	0	0	1	0	0	1

5. Discussion

The last 10 years have seen the development of a few CDM-based methods for extracting the Q-matrix, whereas non-CDM based approaches have been rarely seen. One of the non-CDM based methods makes use of NMF. Desmarais and his colleagues revealed that NMF is a useful method in deriving the Q-matrix. From the result of NMF, Q-matrix estimation is a matrix factorization. Inspired by NMF, this study demonstrates the practicality of our proposed deterministic learning algorithm in generating the Q-matrix.

As the Q-matrix is regarded as a factor model with binary factor loadings, a rotation using varimax after the estimation is necessary to aid interpretation. After a rotation, an objective way to dichotomize continuous estimates is needed to derive the final Q-matrix. Arbitrarily using a fixed value such as 0.5 might generate fallacious estimates. For example, if the factor loadings for an item j is (0.111, 0.222, 0.333, 0.444), then a cutoff 0.5 will result in (0, 0, 0, 0). In this research, we enforce a dichotomizing scheme using a proportion of the highest loading (a) on the estimates after they are rotated. Results from the simulation study suggests that setting a to 0.5 is optimal.

The simulation study demonstrates that the proposed deterministic learning algorithm is capable of extracting the Q-matrix from data, with the results showing that recovery rates from different conditions are all above 0.9. Both sample sizes and correlations between attributes are influential in the Q-matrix recovery. When the Q-matrix is incomplete, the results are not as good as the settings with a complete Q-matrix. The low recovery rate is due to the non-identifiability of model parameters under the incomplete Q-matrix [38].

One limitation in this research is that the number of attributes is assumed to be known in the estimation. Future research could use a parallel analysis to deal with the issue. Another limitation is that data were simulated only from the DINA model and RRUM. It is desirable to test the effectiveness of the proposed method on data generated from more general CDMs, such as the GDINA model. Further research using the proposed method is also recommended to examine the recovery rate from data simulated from non-compensatory models, such as the DINO model.

More extensive research is needed to compare the result from factor models with the result from CDM-based methods. A further comparison could investigate the effect of model misspecification on guess and slip parameters in the DINA model and RRUM.

In the empirical study, our Q-matrix estimate is very different from the expert designed Q-matrix, although 76.2% of the Q-matrix entries are identical. It is suggested that further research applies validation approach, setting those identical entries as fixed and estimating the rest of the entries.

In conclusion, this research demonstrated that the Q-matrix is intrinsically a factor model. Different dimensionality reduction techniques, such as principal component analysis and singular value decomposition, incorporated with the dichotomizing scheme advanced in this research should be able to derive the Q-matrix.

Author Contributions: Conceptualization, M.-T.C.; methodology, M.-T.C.; software, M.-T.C.; validation, M.-T.C. and S.-L.C.; formal analysis, M.-T.C.; investigation, M.-T.C. and S.-L.C.; resources, M.-T.C. and S.-L.C.; data curation, M.-T.C. and S.-L.C.; writing—original draft preparation, M.-T.C.; writing—review and editing, S.-L.C.; supervision, S.-L.C.; project administration, S.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ECPE data can be obtained from the CDM R package.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Leighton, J.P.; Gierl, M.J. *Cognitive Diagnostic Assessment for Education. Theory and Applications*, 1st ed.; Cambridge University Press: Cambridge, UK, 2008; pp. 1–2.
2. Junker, B.W.; Sijtsma, K. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Appl. Psychol. Meas.* **2001**, *25*, 258–272. [CrossRef]
3. DiBello, L.V.; Stout, W.F. Unified cognitive psychometric assessment likelihood-based classification techniques. In *Diagnostic Assessment*; Paul, D.N., Susan, F.C., Robert, L.B., Eds.; Routledge: New York, NY, USA, 1995; pp. 361–389.
4. Hartz, S. A Bayesian Framework for the Unified Model for Assessing Cognitive Abilities: Blending Theory with Practicality. Ph.D. Thesis, University of Illinois, Champaign, IL, USA, 2002.
5. Tatsuoka, K.K. Rule space: An approach for dealing with misconceptions based on item response theory. *J. Educ. Meas.* **1983**, *20*, 345–354. [CrossRef]
6. Tatsuoka, K.K. Toward an integration of item-response theory and cognitive error diagnosis. In *Diagnostic Monitoring of Skill and Knowledge Acquisition*; Frederiksen, R., Glaser, L., Shafto, M., Eds.; Routledge: Hillsdale, MI, USA, 1990; pp. 453–488.
7. de la Torre, J. An empirically-based method of Q-matrix validation for the DINA model: Development and applications. *J. Educ. Meas.* **2008**, *45*, 343–362. [CrossRef]
8. de La Torre, J.; Douglas, JA. Higher-order latent trait models for cognitive diagnosis. *Psychometrika* **2004**, *69*, 333–353. [CrossRef]
9. de La Torre, J.; Douglas, J. Model evaluation and multiple strategies in cognitive diagnosis: An analysis of fraction subtraction data. *Psychometrika* **2008**, *73*, 595. [CrossRef]
10. Henson, R.A.; Templin, J.L.; Willse, J.T. Defining a Family of Cognitive Diagnosis Models Using Log-Linear Models with Latent Variables. *Psychometrika* **2009**, *74*, 191–210. [CrossRef]
11. Tatsuoka, C. Data analytic methods for latent partially ordered classification models. *Appl. Stat.* **2002**, *51*, 337–350. [CrossRef]
12. Casalino G., Castiello, C.; Del Buono, N.; Esposito, F.; Mencar, C. Q-matrix Extraction from Real Response Data Using Nonnegative Matrix Factorizations. In *Computational Science and Its Applications—ICCSA 2017*; Gervasi, O., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10404._15. [CrossRef]
13. Chen, Y.; Culpepper, S.A.; Douglas, J.A. Bayesian Estimation of the DINA Q matrix. *Psychometrika* **2018**, *83*, 89–108. [CrossRef]
14. Liu, C.W.; Andersson, B.; Skrondal, A. A Constrained Metropolis–Hastings Robbins–Monro Algorithm for Q Matrix Estimation in DINA Models. *Psychometrika* **2020**, *85*, 322–357. [CrossRef] [PubMed]
15. Liu, J.; Xu, G.; Ying, Z. Data-driven learning of Q-matrix. *Appl. Psychol. Meas.* **2012**, *36*, 609–618. [CrossRef] [PubMed]
16. Xu, G.; Shang, Z. Identifying Latent Structures in Restricted Latent Class Models. *J. Am. Stat. Assoc.* **2018**, *113*, 1284–1295. [CrossRef]
17. Chung, M. Estimating the Q-Matrix for Cognitive Diagnosis Models in a Bayesian Framework. Ph.D. Thesis, Columbia University, New York, NY, USA, 2014.
18. Chung, M. A Gibbs sampling algorithm that estimates the Q-matrix for the DINA model. *J. Math. Psychol.* **2019**, *93*, 102275. [CrossRef]
19. Winters, T. Educational Data Mining: Collection and Analysis of Score Matrices for Outcomes-Based Assessment. Ph.D. Thesis, University of California, Riverside, CA, USA, 2006.
20. Desmarais, M.C. Conditions for effectively deriving a Q-Matrix from data with Non-negative Matrix Factorization. In Proceedings of the Educational Data Mining 2011, Eindhoven, The Netherland, 6–8 July 2011.
21. Desmarais, M.C. Mapping question items to skills with non-negative matrix factorization. *ACM SIGKDD Explor. Newsl.* **2012**, *13*, 30–36. [CrossRef]
22. Desmarais, M.C.; Naceur, R. A Matrix Factorization Method for Mapping Items to Skills and for Enhancing Expert-Based Q-matrices. In Proceedings of the 16th Conference on Artificial Intelligence in Education, AIED2013, Memphis, TN, USA, 9–12 July 2013.
23. Desmarais, M.C.; Beheshti, B.; Naceur, R. Item to skills mapping: Deriving a conjunctive Q-matrix from data. In Proceedings of the 11th International Conference Intelligent Tutoring Systems, ITS2012, Chania, Greece, 14–18 June 2012.
24. Cichocki, A.; Zdunek, R.; Phan, A.-H.; Amari, S. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*; Wiley: Hoboken, NJ, USA, 2009; pp. 236–239.
25. Gong, L.; Nandi, A.K. An enhanced initialization method for non-negative matrix factorization. In *Machine Learning for Signal Processing (MLSP)*; 2013 IEEE International Workshop; IEEE: Southampton, UK, 2013.
26. Zheng, Z.; Yang, J.; Zhu, Y. Initialization enhancer for nonnegative matrix factorization. *Eng. Appl. Artif. Intell.* **2007**, 20, 101–110. [CrossRef]
27. Barnes, T.; Bitzer, D.; Vouk, M. Experimental analysis of the q-matrix method in knowledge discovery. In *Foundations of Intelligent Systems: Lecture Notes in Computer Science*; Hacid, M.-S., Murray, V., Raś, W., Tsumoto, S., Eds.; Springer: Heidelberg, Germany, 2005; Voume 3488, pp. 603–611.
28. Lee, T.W.; Girolami, M.; Sejnowski, T.J. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Comput.* **1999**, *11*, 417–441. [CrossRef] [PubMed]
29. Grossman, S.I. *Elementary Linear Algebra*, 5th ed.; Brooks Cole: Pacific Grove, CA, USA, 1994.
30. Rupp, A.; Templin, J. Effects of Q-matrix misspecification on parameter estimates and misclassification rates in the DINA model. *Educ. Psychol. Meas.* **2008**, *68*, 78–98. [CrossRef]

31. Chen, Y.; Liu, J.; Xu, G.; Ying, Z. Statistical analysis of Q-matrix based diagnostic classification models. *J. Am. Stat. Assoc.* **2015**, *110*, 850–866. [CrossRef]
32. Robitzsch, A.; Kiefer, T.; George, A.C.; Ünlü, A. CDM: Cognitive Diagnosis Modeling. R Package Version 7.5–15. 2020. Available online: https://CRAN.R-project.org/package=CDM (accessed on 9 Januray 2021).
33. Templin, J.; Hoffman, L. Obtaining diagnostic classification model estimates using Mplus. *Educ. Meas. Issues Pract.* **2013**, *32*, 37–50. [CrossRef]
34. Buck, G.; Tatsuoka, K.K. Application of the rule-space procedure to language testing: Examining attributes of a free response listening test. *Lang. Test.* **1998**, *15*, 119–157. [CrossRef]
35. Henson, R.; Templin, J. Importance of Q-matrix construction and its effects cognitive diagnosis model results. In Proceedings of the Annual Meeting of the National Council on Measurement in Education, Chicago, IL, USA, 10–12 April 2007.
36. Feng, Y.; Habing, B.; Huebner, A. Parameter estimation of the reduced RUM using the EM algorithm. *Appl. Psychol. Meas.* **2014**, *38*, 137–150. [CrossRef]
37. Templin, J.; Bradshaw, L. Hierarchical diagnostic classification models: A family of models for estimating and testing attribute hierarchies. *Psychometrika* **2014**, *79*, 317–339. [CrossRef] [PubMed]
38. Xu, G.; Zhang, S. Identifiability of Diagnostic Classification Models. *Psychometrika* **2016**, *81*, 625–649. [CrossRef] [PubMed]