Q-matrix Learning and DINA Model Parameter Estimation

Yuan Sun
National Institute of Informatics
Tokyo, Japan
Email: yuan@nii.ac.jp

Shiwei Ye, Guiping Su
University of China Academy
of Science, Beijing, China
Email: {shwye,sugp}@ucas.ac.cn

Yi Sun
University of China Academy
of Science, Beijing, China
Email:sunyi@ucas.ac.cn

Abstract

The DINA model is one of the most widely used models in cognitive and skills diagnosis, and several algorithms have been developed for estimating the model parameters. However, since the parameter space is very large and has a mix of binary variables, even medium-sized testing is extremely challenging. To make the model practical, a fast optimization algorithm for parameter estimation is needed. In this study, we converted the deterministic Q-matrix learning problem into a Boolean matrix factorization (BMF) problem and developed a recursive algorithm to find an approximate solution while solving the uncertainty parameters analytically using maximum likelihood estimation (MLE). We proved that the MLE is equivalent to the minimum information entropy of the DINA model. Simulation results demonstrated that our proposed algorithm converges rapidly to the optimal solution under suitable initial values of $skill - item$ association and is insensitive to the initial values of the uncertainty parameters.

I. Introduction

In educational and psychological testing and many other disciplines, cognitive diagnostic models (CDMs) have been attracting considerable interest [1]. Instead of using a single total score or several sub-scores to evaluate student performance, CDMs are designed to diagnose students' strengths and weaknesses and to provide specific information in the form of attribute mastery profiles. These not only provide more accurate measurement of learning and progress but also help to improve instruction methods and suggest possible interventions to address individual and group needs. Central to many CDMs is the DINA model, which is one of the most widely used models in cognitive diagnostic assessment [2]. The model variables include the well-known Q-matrix (the deterministic part), which specifies the item-attribute relationships; the student knowledge state matrix A, which specifies the student-attribute mapping; and two noise variables (the random part) related to item response functions, termed the slip s_i and guessing g_i parameters, which indicate that a student has or lacks the attributes required by an item i but nevertheless fails or succeeds in answering the item correctly[3]. Several algorithms using the EM algorithm and maximum likelihood estimation (MLE) have been proposed for learning both deterministic Q-matrix parameters and random parameters in the whole parameter space [4][5]. However, when the parameter space is very large, this mixing of binary variables makes the achievement of even a medium-sized Q-matrix extremely challenging.

This paper proposes a new method in which the Q-matrix and uncertainty parameters (slip and guessing) are derived separately. The deterministic skill mapping (Q-matrix) has one-to-many mapping between individual skills and one or more associated assessment items. The optimization of the DINA model can therefore be disaggregated, with Q-matrix learning and the uncertainty slip and guessing parameters derived separately through an observable real response matrix R. In particular, we converted the deterministic Q-matrix learning problem into a Boolean matrix factorization (BMF) problem based on an ideal response matrix $\mathcal{R}$. We then solved the model uncertainty parameters analytically, based on an observable real response matrix R, by minimizing the system entropy for the DINA model. Based on the above assumption, we proved that the MLE of the DINA model is equivalent to the minimum information entropy (MIE) of the system. Because BMF is an NP-hard problem [6][7][8][9][10], we proposed a recursive approach to find an approximate solution for Q-matrix learning in the attribute space.

The rest of this paper is structured as follows. Section II gives a brief introduction to the DINA model and BMF. Section III presents the fast MLE-based recursive algorithm proposed for Q-matrix learning and uncertainty parameter estimation. In Section IV, we present the results of simulations of the proposed algorithm. Finally, in Section V, we discuss the findings and present our conclusions.

II. DINA Model and Boolean Matrix Factorization

In DINA model[11], the probability of a correct response to an item is determined by two error probabilities(the guessing probability (g_j) and the slip probability (s_j)) and one latent response variable. Assuming there are K attributes in a particular domain, a student's attribute patterns $\alpha_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iK})$, called knowledge states, indicate the student's mastery of the K attributes. $\alpha_{ik} = 1$ indicates that the ith student has mastered attribute k, while zero otherwise. As noted above, the Q-matrix gives the required attributes for each item. The entry in the Q-matrix (denoted q_{jk}) equals one if item j requires attribute k, and zero otherwise.

A recent development of BMF has been shown to be an extremely effective approach for deriving usable results from binary data [12][13][14][15].The goal of BMF is to decompose a Boolean matrix ($\mathcal{R}$) into two Boolean matrices, where one of the matrices, the concept matrix, can be viewed as a set of meaningful concepts, while the second , called the combination matrix, describes each observed record. So proved by study, the problem of CDM could be explained by BMF.

In CDM, it is necessary to find a decomposition that meets a certain objective. For example, we may choose to find the decomposition that minimizes the number of attributes k. However, this problem corresponds to the minimum tiling problem originally studied by Geerts et al., which has been proven to be NP-hard[13]. Using the definition of BMF, we confirmed that an ideal response matrix $\mathcal{R}$ can be expressed in terms of the following Boolean relations of the knowledge state matrix A and the Q-matrix in CDM.

$$\mathcal{R} = \overline{\overline{A} \odot Q^T} \tag{1}$$

Here, A and Q represent an m-student by K-attribute binary mastery matrix and an n-item by K-attribute binary Q-matrix, where student $i = 1, .., m$, item $j = 1, .., n$, and attribute $k = 1, .., K$. The bar notation in equation (1) represents a logical NOT operation (i.e., $\overline{0} = 1, \overline{1} = 0$). The goal of BMF is to determine the matrices Q and A from $\mathcal{R}$, and that of the factorization algorithm is to minimize the estimated real response matrix R with the ideal response matrix $\mathcal{R}$ from equation (1). It is clear that equation (1) is more powerful when the number of latent attributes increases. In addition, equation (1) adds flexibility to the data-driven learning of the Q-matrix.

III. BMF and Recursive Algorithm for DINA Model

Let $R(i, j)$ and $\mathcal{R}(i, j)$ be the real response matrix and ideal response matrix of the ith student to the jth item, respectively. The DINA model gives the following conditional probability formula:

$$P\left(R(i,j)|\mathcal{R}(ij)\right) = \begin{cases} 1 - s_j & when R(i,j) = 1; \mathcal{R}(i,j) = 1 \\ s_j & when R(i,j) = 0; \mathcal{R}(i,j) = 1 \\ g_j & when R(i,j) = 1; \mathcal{R}(i,j) = 0 \\ 1 - g_j & when R(i,j) = 0; \mathcal{R}(i,j) = 0 \end{cases}$$

where $\mathcal{R}(i,j) = \overline{\vee_{l=1}^{k} \overline{A}(i,l) Q(j,l)}$. For simplification, we set $B = \overline{A}, C(i,j) = \overline{\mathcal{R}}(i,j)$. The DINA model then allows equation (2) to be rewritten as

$$\begin{aligned} P\left(R(i,j)|B,Q\right) &= \prod_{i=1}^{m}\prod_{j=1}^{n} (1-s_j)^{R(i,j)\overline{C}(i,j)} s_j^{\overline{R}(i,j)\overline{C}(i,j)} \\ &\times \; g_j^{R(i,j)C(i,j)} (1-g_j)^{\overline{R}(i,j)C(i,j)} \end{aligned}$$

where $C = B \odot Q^T$. The optimization problem consists of finding the maximum likelihood of conditional probability:

$$\begin{aligned} E(B,Q) &= \sum_{i=1}^{m}\sum_{j=1}^{n} lnP\left(R(i,j)|B,Q\right) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} R(i,j)\overline{C}(i,j) ln(1-s_j) + \overline{R}(i,j)\overline{C}(i,j) ln(s_j) \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{n} \overline{R}(i,j) C(i,j) ln(1-g_j) + R(i,j) C(i,j) ln(g_j) \end{aligned} \tag{3}$$

When C is fixed, by maximizing the likelihood function $E(s, g)$ with respect to slip s and guessing g, we obtain

$$s_j = \frac{\lambda_j}{\lambda_j + \gamma_j} \tag{4}$$

Here,

$$\gamma_j = \sum_{i=1}^{m} R(i,j)\overline{C}(i,j), \quad \lambda_j = \sum_{i=1}^{m} \overline{R}(i,j)\overline{C}(i,j). \tag{5}$$

Similarly,

$$g_j = \frac{\tau_j}{\rho_j + \tau_j} \tag{6}$$

where

$$\tau_j = \sum_{i=1}^{m} \overline{R}(i,j) C(i,j), \quad \rho_j = \sum_{i=1}^{m} R(i,j) C(i,j) \tag{7}$$

Substituting the results from (4) and (5) into equation (3), we obtain

$$\begin{aligned} E(B,Q) &= \sum_{j=1}^{n} \left[\gamma_j ln \frac{\gamma_j}{\gamma_j + \lambda_j} + \lambda_j ln \frac{\lambda_j}{\gamma_j + \lambda_j} \right. \\ &+ \left. \tau_j ln \frac{\tau_j}{\rho_j + \tau_j} + \rho_j ln \frac{\rho_j}{\rho_j + \tau_j} \right] \end{aligned} \tag{8}$$

Based on the DINA model, B and Q are unknown but fixed parameters, whose uncertainties arise from the slip and guess variables for each item in the test. Because

$$\begin{aligned} \gamma_j + \lambda_j &= \sum_{i=1}^{m} \overline{C}(i,j) = mP(\mathcal{R}(:,j) = 1) \\ \tau_j + \rho_j &= \sum_{i=1}^{m} C(i,j) = mP(\mathcal{R}(:,j) = 0) \end{aligned} \tag{2}$$

So, we can rewrite equation (8) as follows:

$$E(B,Q) = -m \times H(B,Q)$$

$$\begin{aligned} &H(B,Q) \\ &= \sum_{j=1}^{n} P(\mathcal{R}(:,j) = 1) \times \left(s_j ln \frac{1}{s_j} + (1-s_j) ln \frac{1}{1-s_j} \right) \\ &= \sum_{j=1}^{n} \left(P(\mathcal{R}(:,j) = 1) H(s_j) + P(\mathcal{R}(:,j) = 0) H(g_j) \right) \end{aligned} \tag{9}$$

Here, $H(p)$is the information entropy of the system. The maximum likelihood function $E(s,g)$ is equivalent to minimizing the information entropy, which represents the uncertainty of the system. Clearly, in the case $s_j = g_j = 0$, where there is no noise, the entropy function $H(s,g)$ is minimized. However, in a real system, the presence of noise means that $s_j, g_j \neq 0$. A previous study[6] proposed a method for Q-matrix learning based on minimizing $\sum_j (s_j + g_j)$, and demonstrated through numerical experiments that this was able to correct errors under certain conditions. However, the approach is a heuristic one and lacks a theoretical basis when applied to Q-matrix learning. When the maximum and guessing parameters $s_j and g_j$ are less than half, $H(s_j)$ and $H(g_j)$ are monotonically increasing functions of parameter s_j or g_j, and minimizing $s_j + g_j$ will produce reasonable results. As noted above, the MIE method is equivalent to MLE, which is convergent for parameter estimation in DINA in a statistical sense. In what follows, $E(B,Q)$ is chosen as the objective function. By setting $r_j = \sum_{j=1}^{m} R(i,j)$ and $c_j = \sum_{j=1}^{m} C(i,j)$, the following equations can be verified:

$$\begin{aligned} \rho_j &= c_j - r_j + \gamma_j \\ \lambda_j &= m - c_j - \gamma_j \\ \tau_j &= r_j - \gamma_j \end{aligned} \tag{10}$$

After solving the slip and guessing probabilities g_j and s_j, the objective function is set to the maximum $E(B,Q)$ in B and Q space. We propose a heuristic algorithm for recursively optimizing the B and Q matrices. In $C = B \odot Q^T$, we denote $B = (b_1, b_2, \cdots, b_k), Q = (q_1, q_2, \cdots, q_k)$, $B^l = (b_1, b_2, \cdots, b_{l-1}, b_{l+1}, \cdots, b_k), Q^l = (q_1, q_2, \cdots, q_{l-1}, q_{l+1}, \cdots, q_k)$, and $C^l = B^l (Q^l)^T$, so that $C = C^l \vee b_l (q_l)^T$. For a fixed C, we randomly select the l-th column from the B and Q matrices and fix the B^l and Q^l matrices such that the values of b_l and q_l are optimized. From the definition $C(i,j) = C^l(i,j) + \overline{C}^l B(i,l) Q(j,l)$ and substituting $C(i,j)$ into equation (5), we obtain

$$\begin{aligned} \gamma_j &= \sum_{i=1}^{m} R(i,j)\overline{C}(i,j) \\ &= \sum_{i=1}^{m} \left[R(i,j) C^l(i,j) + R(i,j)\overline{C}^l(i,j) B(i,l) Q(j,l) \right] \\ &= \gamma_j^l + Q(j,l) \sum_{i=1}^{m} R(i,j)\overline{C}^l(i,j) B(i,l) \end{aligned}$$

Here, $\gamma_j^l = \sum_{i=1}^{m} R(i,j) C^l(i,j)$. Similarly, we have

$$\begin{aligned} c_j &= \sum_{i=1}^{m} \left[C^l(i,j) + \overline{C}^l(i,j) B(i,l) Q(j,l) \right] \\ &= c_j^l + Q(j,l) \sum_{i=1}^{m} \overline{C}^l(i,j) B(i,l) \end{aligned}$$

Here, $c_j^l = \sum_{i=1}^{m} C^l(i,j)$. Setting $u_j = \sum_{i=1}^{m} R(i,j)\overline{C}^l(i,j) B(i,l)$ and $v_j = \sum_{i=1}^{m} \overline{C}^l(i,j) B(i,l)$ gives

$$\gamma_j = \gamma_j^l + Q(j,l) u_j, \quad c_j = c_j^l + Q(j,l) v_j$$

Integrating with equation(10), we get

$$\gamma_j ln \frac{\gamma_j}{\gamma_j + \lambda_j} = \gamma_j^l ln \frac{\gamma_j^l}{m - c_j^l} + Q(j,l) w_{j,1}$$

where

$$w_{j,1} = (\gamma_j^l + u_j) ln \frac{\gamma_j^l + u_j}{m - c_j^l - v_j} - \gamma_j^l ln \frac{\gamma_j^l}{m - c_j^l}$$

Analogously,

$$\begin{aligned} \lambda_j ln \frac{\lambda_j}{\gamma_j + \lambda_j} &= \lambda_j^l ln \frac{\lambda_j^l}{c_j^l} + Q(j,l) w_{j,2} \\ \rho_j ln \frac{\rho_j}{\rho_j + \tau_j} &= \rho_j^l ln \frac{\rho_j^l}{c_j^l} + Q(j,l) w_{j,3} \\ \rho_j ln \frac{\tau_j}{\tau_j + \tau_j} &= \rho_j^l ln \frac{\tau_j^l}{c_j^l} + Q(j,l) w_{j,4} \end{aligned}$$

where

$$\begin{aligned} w_{j,2} &= (\lambda_j^l + u_j + v_j) ln \frac{\gamma_j^l + u_j + v_j}{m - c_j^l - v_j} - \lambda_j^l ln \frac{\lambda_j^l}{m - c_j^l} \\ w_{j,3} &= (\rho_j^l + u_j + v_j) ln \frac{\rho_j^l + u_j + v_j}{c_j^l + v_j} - \rho_j^l ln \frac{\rho_j^l}{c_j^l} \\ w_{j,4} &= (\tau_j^l - u_j) ln \frac{\tau_j^l - u_j}{c_j^l + v_j} - \tau_j^l ln \frac{\tau_j^l}{c_j^l} \end{aligned}$$

respectively. By substituting the above equations into (8), we obtain

$$E(B,Q) = E(B^l, Q^l) + \sum_{j=1}^{n} Q(j,l)(w_{j,1} + w_{j,2} + w_{j,3} + w_{j,4}) \tag{11}$$

We now present a two-step recursive algorithm for maximizing the likelihood function $E(B,Q)$. First, we fix $B(:,l)$. To maximize the value of $E(B,Q)$ with respect to $Q(j,l)$ of the Boolean value, we use the following updating formula for $Q(j,l)$:

$$Q(j,l) = \theta(w_{j,1} + w_{j,2} + w_{j,3} + w_{j,4}) \tag{12}$$

Here, $\theta(\cdot)$is defined as follows:

$$\theta(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

If $Q(:,l)$ is fixed, we can rewrite u_j as

$$u_j = \sum_{i=1}^{m} R(i,j)\overline{C}^l(i,j) B(i,l) = u_j^i + R(i,j)\overline{C}^l(i,j) B(i,l)$$

Here,

$$u_j^i = \sum_{k=1, k\neq i}^{m} R(i,j)\overline{C}^l(k,j) B(k,l)$$

Similarly, for v_j,

$$u_j = \sum_{i=1}^{m} \overline{C}^l(i,j)B(i,l) = v_j^i + \overline{C}^l(i,j)B(i,l)$$

where

$$v_j^i = \sum_{k=1,k\neq i}^{m} \overline{C}^l(k,j)B(k,l)$$

Therefore, the distance parameter w can be written as,

$$w_{j,1} = G_1(i,j) + B(i,l)H_1(i,j) - F_1(i,j)$$

where

$$\begin{aligned} F_1(i,j) &= \gamma_j^l ln\frac{\gamma_j^l}{m-c_j^l} \\ G_1(i,j) &= (\gamma_j^l + u_j^i) ln\frac{\gamma_j^l + u_j^i}{m-c_j^l - v_j^i} \end{aligned}$$

and

$$\begin{aligned} H_1(i,j) &= (\gamma_j^l + u_j^i + R(i,j)\overline{C}^l(i,j)) \\ &\times \; ln\frac{\gamma_j^l + u_j^i + R(i,j)\overline{C}^l(i,j)}{m-c_j^l - v_j^i - \overline{C}^l(i,j)} - G_1(i,j) \end{aligned}$$

Analogously, we have

$$w_{j,t} = G_t(i,j) + B(i,l)H_t(i,j) - F_t(i,j), \quad t = 1,2,3,4$$

whose variables F_t, G_t, H_t can be concretely expressed as

$$F_2(i,j) = \lambda_j^l ln\frac{\lambda_j^l}{m-c_j^l}, F_3(i,j) = \rho_j^l ln\frac{\rho_j^l}{c_j^l}, F_4(i,j) = \tau_j^l ln\frac{\tau_j^l}{c_j^l}$$

$$\begin{aligned} G_2(i,j) &= (\lambda_j^l + u_j^i + v_j^i) ln\frac{\lambda_j^l + u_j^i + v_j^i}{m-c_j^l - v_j^i} \\ G_3(i,j) &= (\rho_j^l + u_j^i + v_j^i) ln\frac{\rho_j^l + u_j^i + v_j^i}{c_j^l + v_j^i} \\ G_4(i,j) &= (\tau_j^l - u_j^i) ln\frac{\tau_j^l - u_j^i}{c_j^l + v_j^i} \end{aligned}$$

and

$$\begin{aligned} H_2(i,j) &= \left(\lambda_j^l + u_j^i + (1+R(i,j))\overline{C}^l(i,j)\right) \\ &\times \; ln\frac{\lambda_j^l + u_j^i + (1+R(i,j))\overline{C}^l(i,j)}{m-c_j^l - v_j^i - \overline{C}^l(i,j)} - G_2(i,j) \end{aligned}$$

$$\begin{aligned} H_3(i,j) &= \left(\rho_j^l + u_j^i + (1+R(i,j))\overline{C}^l(i,j)\right) \\ &\times \; ln\frac{\rho_j^l + u_j^i + (1+R(i,j))\overline{C}^l(i,j)}{c_j^l + v_j^i + \overline{C}^l(i,j)} - G_3(i,j) \end{aligned}$$

and correspondingly,

$$\begin{aligned} H_4(i,j) &= \left(\tau_j^l - u_j^i - R(i,j)\overline{C}^l(i,j)\right) \\ &\times \; ln\frac{\tau_j^l - u_j^i - R(i,j)\overline{C}^l(i,j)}{c_j^l + v_j^i + \overline{C}^l(i,j)} - G_4(i,j) \end{aligned}$$

Based on this analysis, equation (11) can be rewritten as follows:

$$\begin{aligned} E(B,Q) &= E(B^l,Q^l) + \sum_{j=1}^{n}\sum_{k=1}^{4} Q(j,l)(G_k(i,j) - F_k(i,j)) \\ &+ B(i,l)\sum_{j=1}^{n}\sum_{k=1}^{4} Q(j,l)H_k(i,j) \end{aligned} \tag{13}$$

Using the maximum $E(B,Q)$ value and recognizing that $B(j,l)$ is also a Boolean value, the updating formula for $B(j,l)$ becomes

$$B(i,l) = \theta\left(\sum_{j=1}^{n}\sum_{k=1}^{4} Q(j,l)\overline{C}^l(i,j)H_k(i,j)\right) \tag{14}$$

The fast MLE-based parameter estimation algorithm based on the DINA model is shown as follows: Each iteration step in the above algorithm identifies a local maximum. The convergence of the fast MLE-based parameter estimation algorithm is demonstrated by the follow theorem.

Algorithm 1 Fast MLE-based parameter estimation algorithm.

Input: Initializing response matrix $R \in \{0,1\}^{m\times n}$ and $A, \overline{A} \in \{0,1\}^{m\times k}$, attribute Matrix $Q \in \{0,1\}^{n\times k}$,

Step 1: Computing $G = BQ^T$

Step 2: Random(or deterministic) selecting $1 \le l \le k$, updating the l-column of matrix $\overline{A}$ and Q

2.1 Computing $G^l = B(:,l)Q(:,l)^T, G^l = G - G^l$, $C^l = \theta(G^l)$,

2.2 updating $Q(:,l)$ based on formula (12)

2.3 sequence updating $B(:,l)$ based on formula (14)

2.3.1 computing $u_j \leftarrow \sum_{i=1}^{m} R(i,j)\overline{C}^l(i,j)B(i,l)$
$v_j = \sum_{i=1}^{m} \overline{C}^l(i,j)B(i,l)$

2.3.1 For $i = 1, \cdots, m$ Updating u and v as follows

i) $u_j^i \leftarrow u_j - R(i,l)\overline{C}^l(i,j)B(i,l)$
$v_j^i \leftarrow v_j - \overline{C}^l(i,j)B(i,l)$

ii) sequence update $B(:,l)$ based on formula (14)

iii) $u_j \leftarrow u_j^i + R(i,l)\overline{C}^l(i,j)B(i,l)$
$v_j \leftarrow v_j^i + \overline{C}^l(i,j)B(i,l)$

2.3.3 repeat step 2.3.2 until $B(:,l)$ convergence

2.4 repeating step 2.2 till $Q(:,l)$ and $B(:,l)$ convergence:

2.5 updating $B : B(:,l) \leftarrow B(:,l)$;
$Q : Q(:,l) \leftarrow Q(:,l)$; $G : G = G^l + B(:,l)Q(:,l)^T$;

Step 3: repeating step 2 till approach error immobility or less than a given threshold.

Theorem: The fast MLE-based parameter estimation algorithm is convergent after a finite number of iteration steps.

Proof: The proof is divided into two steps, as follows:

1) During updating of the algorithm in step 2.2, the likeli-

hood function is not reduced. From equation 11, we have

$$E(B,Q) = E(B^l,Q^l) + \sum_{j=1}^{n} Q(j,l)(w_{j,1}+w_{j,2}+w_{j,3}+w_{j,4})$$

Since $w_{j,k}$ is independent of $Q(j,l), j = 1, \cdots n$,

$$\begin{aligned} E(B_{old},Q_{old}) &= E(B^l,Q^l) + \sum_{j=1}^{n} Q_{old}(j,l) \sum_{t=1}^{4} w_{j,t} \\ E(B_{new},Q_{new}) &= E(B^l,Q^l) + \sum_{j=1}^{n} Q_{new}(j,l) \sum_{t=1}^{4} w_{j,t} \end{aligned}$$

Therefore,

$$\begin{aligned} \Delta E &= E(B_{new},Q_{new}) - E(B_{old},Q_{old}) \\ &= \sum_{j=1}^{n} (Q_{new}(j,l) - Q_{old}(j,l)) \sum_{t=1}^{4} w_{j,t} \end{aligned}$$

For $\forall \alpha \in R$ and $\delta \in \{0.1\}$, $\theta(\alpha) \geq \delta\alpha$, and hence, $\Delta E \geq 0$.

2) As the right-hand side of formula (13) is independent of $B(i,l)$ for $1 \leq i \leq m$ at each iteration step,

$$\Delta E = E(B_{old}(i,l), Q_{old}(i,l)) \sum_{j=1}^{n} \sum_{k=1}^{4} (Q(j,l)\overline{C}^l H_k(i,j)$$

For the same reason, $\Delta E \geq 0$. Because the value of $E(B,Q)$ has a finite bound, the algorithm converges after a finite number of iterations.

IV. Simulation Results

A. Suitable and validated objective function for Q-matrix Learning

We first confirmed that MLE is a fine objective function for use in Q-matrix learning. For simplicity and to facilitate comparison with previous studies [5], we fixed the knowledge state matrix A-matrix as A_0 in all simulations. Initially, we defined Q_0 as a true Q-matrix. The ideal response pattern $\mathcal{R}$ can be generated from $\mathcal{R} = \overline{A_0} \odot Q_0^T$. For the given slip and guessing probabilities, s and g, we obtained a real response R-matrix R_0 from $\mathcal{R}$ by equation (2). In the absence of prior knowledge, if a brute force search in the entire A and Q space can recover the Q_0 directly from the generated real response R_0 by maximizing the likelihood function in equation (4), the MLE method can be considered a valid objective function for Q-matrix learning. To verify this, we took a 5×3 Q-matrix Q, computed all the different 5×3 Q using equation (4), and obtained the optimal Q_{opt} as $Q_{opt} = \arg_{Q \in \{0,1\}^{5\times 3}} E(B,Q)$. The simulation was run 100 times with a different true Q_0 set as the input to the Q-matrix in each run. All experimental values of Q_{opt} were exactly equal to Q_0, which strongly supported our claim that this objective function is appropriate for a small-sized Q-matrix. We then addressed larger Q_0s. Because it is not practical to search all possible Q-matrices, to reduce computational complexity, we narrowed down the search space by changing one or two element(s) of the Q_0. We used a 20×3 Q_0 as a case study. When changing a single element, the search space became $\binom{60}{1} = 60$, and when changing two elements, it became $\binom{60}{2} = 1770$. These results confirmed that Q_0 is also a local optimal solution when analyzing a larger-sized matrix.

B. Performance of Fast MLE-based parameter estimation algorithm

In this section, we compared the estimated Q-matrix and the true Q-matrix under various settings, using two datasets from an earlier paper [5]. Initially, the same Q-matrices of 20×3 (20 items by 3 attributes) and 20×4 (20 items by 4 attributes) were used. These are denoted as Q_1 and Q_2.

The slip and guessing parameters were set as $s = g = 0.2$ for all items, and 100 datasets were generated at sample sizes of M = 100, 300, 500, and 1000. The MLE-based Q-matrix learning algorithm was implemented with a starting matrix Q, specified as follows. Following the assumption in the original paper that that three items were misspecified [5], Q was constructed by randomly misspecifying nine elements for Q_1, which represents the maximum possible number of elements in the three misspecified items. In our setting, the nine elements were randomly selected and not necessarily limited to the three items. The simulation results under the condition $s = g = 0.2$ are shown in Table 1. $\hat{Q} = Q_0$ gives the frequency with which Q was correctly estimated, from the 100 independent simulations with different numbers of students. The first row shows that $\hat{Q}$ recovered the true Q_1 36 times at a sample size of 100, and that $\hat{Q}$ never failed to derive the true Q_1 at sample sizes of 300 or more. In contrast, in the results presented in [5], $\hat{Q}$ could only recover the true Q-matrix 98 times even at a sample size of 500. This demonstrates the superior performance of our novel algorithm.

The results for Q_2 under the same conditions are presented in the second row of Table 1. It can be seen that when the same number of nine elements was randomly misspecified, the Q_2 estimator did not perform as well as Q_1.

TABLE I
NUMBER OF CORRECTLY ESTIMATED Q-MATRICES IN 100 SIMULATIONS WITH STUDENT NUMBERS OF $M = 100, 300, 500,$ AND 1000 FOR $Q1$ AND $Q2$ ($s = g = 0.20$).

	$\frac{M=100}{\hat{Q}=Q_0}$	$\frac{M=300}{\hat{Q}=Q_0}$	$\frac{M=500}{\hat{Q}=Q_0}$	$\frac{M=1000}{\hat{Q}=Q_0}$
$Q1$	36/100	100/100	100/100	100/100
$Q2$	3/100	64/100	94/100	100/100

Next, we investigated the relationship between s, g and the number of correctly estimated Q-matrices. Our example used Q_2 and four conditions in which $s = g$ was equal to 0.10, 0.15, 0.20, and 0.25, as shown in Table 2. The number of correctly estimated Qs was shown to increase as s and g increased. A comparison of the same Q_2 in Tables 1 and 2 shows that at a sample size of M=100, the number of correctly estimated Q increased significantly from 3 at $s = g = 0.2$ to 56 at $s = g = 0.1$. At M=300, the number of correctly estimated Q

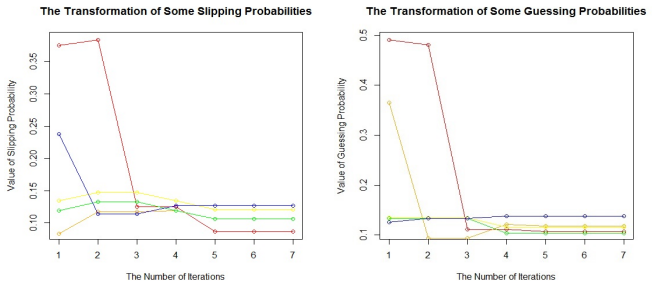

Fig. 1. Convergence of slip and guessing parameter.

was 64 from 100 at $s = g = 0.2$, but all 100 were correctly estimated at $s = g = 0.1$.

TABLE II
NUMBER OF CORRECTLY ESTIMATED Q-MATRICES IN 100 SIMULATIONS WITH STUDENT NUMBERS OF $M = 100, 300, 500$, AND 1000 FOR $Q2$ AT DIFFERENT VALUES OF s,g.

s, g	$M = 100$ $\hat{Q} = Q_0$	$M = 300$ $\hat{Q} = Q_0$	$M = 500$ $\hat{Q} = Q_0$	$M = 1000$ $\hat{Q} = Q_0$
0.10	56/100	100/100	100/100	100/100
0.15	29/100	98/100	100/100	100/100
0.20	3/100	64/100	94/100	100/100
0.25	1/100	25/100	55/100	92/100

Table 3 compares the estimation and the number of misspecified elements of Q_2 at $s = g = 0.2$. The results showed that under the same conditions of M and s, g, the more similar the Q matrix was to the true Q_0, the more likely it was that the Q was estimated correctly. The 12 randomly misspecified elements in Table 3 represent the maximum possible number of elements in the three misspecified items of Q_2, which corresponds to the results reported in [5]. When 12 elements of the Q were misspecified, a larger sample size was needed to recover the true Q_0. However, at M=500, $\hat{Q}$ recovered the true Q-matrix 94 times, compared with 82 times when the existing method was used.

TABLE III
NUMBER OF CORRECTLY ESTIMATED Q-MATRICES OF 100 SIMULATIONS WITH STUDENT NUMBERS OF $M = 100, 300, 500$, AND 1000 FOR $Q2$ WITH DIFFERENT NUMBERS OF MISSPECIFIED ELEMENTS IN THE Q-MATRIX ($s = g = 0.20$).

number	$M = 100$ $\hat{Q} = Q_0$	$M = 300$ $\hat{Q} = Q_0$	$M = 500$ $\hat{Q} = Q_0$	$M = 1000$ $\hat{Q} = Q_0$
3	7/100	76/100	98/100	100/100
6	6/100	77/100	94/100	99/100
9	3/100	64/100	94/100	100/100
12	2/100	58/100	94/100	100/100

On the other hand, the simulation results demonstrated that our algorithm also performed well when estimating the slip and guessing parameters. When the true values of both s and g parameters were set at 0.10 and different initial values were given to s and g for different items, our algorithm generated estimators that converged at the true value. The s and g estimations for five items are given in Figures 1, respectively.

V. CONCLUSIONS

In this paper, we proposed a fast MLE-based recursive algorithm capable of deriving the Q-matrix and uncertainty parameters for a DINA model. Specifically, we converted the deterministic Q-matrix learning problem to a BMF problem and used a recursive algorithm to find an approximate solution while solving the uncertainty parameters analytically through MLE. Simulation results confirmed that our proposed algorithm converged rapidly to the optimal solution under suitable initial conditions and outperformed the conventional method[5]. It was demonstrated that the convergence of the uncertainty parameters was insensitive to the initial value setting. Ongoing work suggests that because the information entropy is a convex function of the variables, which depends on the inner product of R and $\mathcal{R}$, a lower time complexity algorithm can be derived from the convex property of the objective function.

REFERENCES

[1] Rupp, A., Templin, J. and Henson, R. A. 2010. Diagnostic measurement: theory, methods, and applications. Guilford Press.

[2] Haertel, E.H. 1989. Using restricted latent class models to map the skill structure of achievement items. Journal of Educational Measurement, 26, 301-323.

[3] de la Torre, J. 2008. An empirically-based method of Q-matrix validation for the DINA model: Development and applications. Journal of Educational Measurement, 45, 343-362.

[4] Sun, Y., Ye, S., Shi, H., Wang, H. and Sun, Y. 2014. Maximum likelihood estimation based dina model and q-matrix learning. Proceedings of the International Conference on Behavior, Economic and Social Computing (BESC'2014), 1-6.

[5] Liu, J., Xu, G., Ying, Z. 2012. Data-driven learning of q-matrix. Applied Psychological Measurement, 36(7), 548-564.

[6] Sun, Y., Ye, S., Inoue, S. and Sun, Y. 2014. Alternating recursive method for q-matrix learning. Proceeding of the 7th International Conference on Educational Data Mining (EDM 2014), 14-20.

[7] Sun, Y., Ye, S., Sun, Y. and Kameda, T. 2015. Improved algorithms for exact and approximate Boolean matrix decomposition. 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA'2015).

[8] Belohlavek, R., Vychodi, V. 2010. Discovery of optimal factors in binary data via a novel method of matrix decomposition. Journal of Computer and System Sciences, 76, 3-20

[9] Li, N., Cohen, W.W., Matsuda, N. and Koedinger, K.R. 2011. A machine learning approach for automatic student model discovery. In Proceedings of the 4th International Conference on Educational Data Mining, 31-40.

[10] Koedinger, K.R., McLaughlin, E.A. and Stamper, J.C. 2012. Automated student model improvement. In Proceedings of the 5th International Conference on Educational Data Mining.

[11] Rupp, A. A. and Templin, J. 2008. The effects of Q-matrix misspecification on parameter estimates and classification accuracy in the DINA model. Educational and Psychological Measurement, 68(6), 78-96.

[12] Desmarais, M.C. 2011. Mapping question items to skills with non-negative matrix factorization. ACM KDD-Explorations, 13(2), 30-36.

[13] Miettinen, P., Mielikainen, T., Gionis, A., Gautam Das and Heikki Mannila, H. 2008. The discrete basis problem. IEEE Transactions on Knowledge and Data Engineering, 20(10), 1348-1362.

[14] Neruda, R., Snasel,V., Platos,J., Kromer,P. and Husek, D. 2008. Implementing Boolean matrix factorization. ICANN 2008, Part I, LNCS 5163, 543-552.

[15] Vaidya, J. 2012. Boolean matrix decomposition problem: theory, variations and applications to data engineering. IEEE 28th International Conference on Data Engineering.