Alternating Recursive Method for Q-matrix Learning

Yuan Sun
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Japan
yuan@nii.ac.jp

Shiwei Ye
School of Electronic and Communication Engineering, University of China Academy of Science, China
shwye@ucas.ac.cn

Shunya Inoue
Tokyo Kasei University
1-18-1 Kaga, Itabashi-ku, Japan
inoues@tokyo-kasei.ac.jp

Yi Sun
School of Computer and Control Engineering, University of China Academy of Science, China
sunyi@ucas.ac.cn

ABSTRACT

The key issue affecting Cognitive Diagnostic Models (CDMs) is how to specify attributes and the Q-matrix. In this paper, we first attempt to use the Boolean Matrix Factorization (BMF) method to express conjunctive models in CDMs. Because BMF is an NP-hard problem [2], we propose a recursive method that updates the attribute matrix (its rank equals to one) in each step. As Boolean algebra is irreversible, it requires time to recursively compute and update the matrix, especially when the number of attributes is large. To speed up computations, we use a Heaviside step function, which allows us to decompose the recursive computing process into normal non-negative matrices and get the results by mapping them back into a Boolean matrix. Two different algorithms are presented: a deterministic heuristic algorithm and a stochastic algorithm. Simulation results from an actual test show that the proposed method can learn the original Q-matrix well from item response data.

Keywords

Q-matrix, Cognitive Diagnostic Models (CDMs), Boolean Matrix Factorization (BMF), Conjunctive Models, Alternating Recursive Algorithm.

1. INTRODUCTION

Cognitive diagnostic assessment (CDA) has attracted a great deal of attention in the psychological and educational measurement fields. It not only reports students' total test scores, but also assesses students' mastery of attributes, which refers to their knowledge, skills or strategies, so that it can provide students or their teachers with diagnostic information on their strengths and weaknesses. A key issue in CDA is to correctly specify the so-called Q-matrix introduced by Tatsuoka [23], which associates the items and attributes of students a diagnostic test intends to assess.

To construct a Q-matrix, experts in the particular domain usually need to specify what attributes are key knowledge or skills for students to acquire. There are two approaches to constructing it. One is to develop test items and specify the Q-matrix for the items particular to cognitive diagnostic purposes. The other way is to apply diagnostic modeling to an existing test and specify the Q-matrix for the test items. Once the Q-matrix has been specified and items have been administered in a test, the items are calibrated to one of the cognitive diagnostic models, in which a known Q-matrix is typically assumed [24, 11, 13, 12, 15, 4]. However, if the Q-matrix is not specified appropriately, it could seriously affect the models' goodness of fit [21]. In that sense, the key issues affecting CDMs are how to define attributes and specify the Q-matrix. On the other hand, there is a big problem with the current manual method of generating the Q-matrix. When the domain or content of the tests is broader, it is an extremely difficult task for experts to specify attributes and the Q-matrix manually. This difficulty and their time-consuming nature could be reasons why CDMs are still not as popular as they should be in educational fields. Automatic and intelligent help to alleviate this difficulty is obviously desirable, and even necessary. During the past decade, the problem of how to map test items into latent skills based on students' test responses has become a hot topic in psychometrics and in educational data mining (see [16, 17, 14, 28, 7, 6, 5, 1, 3] for recent contributions).

In the previous studies [17] and [28], the proposed DINA model-based Q-matrix learning approach use EM algorithm to solve the Q-matrix. They need to involve a matrix T(Q) on a scale of $2^n \times 2^K$, wherein n, K is the number of items and attributes, respectively, which makes it extremely difficult to achieve for even a medium sized Q-matrix. In [7], an alternative least square method was proposed to solve the Q-matrix, where they use a matrix inverse operation that values may appear negative and estimated values for all the parameters are real instead of binary 0 or 1. To finally map the results into binary 0 or 1, there need to assume an appropriate threshold to truncate the values, but the threshold selected can't be automatically given, it can only be artificial and thus subjective. Moreover, the most serious problem for the previous methods so far ([7][17][28]) is that the number of attributes K have to be set in advance. They can't be determined by students' real response dataset R matrix, which is extremely critical for data driven Q-matrix learning approach.

Here we present new algorithms to learn the Q-matrix automatically from the students' item response matrix on the basis of the Boolean Matrix Factorization (BMF) technique and demonstrate how the methods can yield promising results from simulation data. This is the first attempt to apply BMF to Q matrix discovery.

2. CONJUNCTIVE MODELS AND BOOLEAN MATRIX FACTORIZATION

Various statistical models have been built around the Q-matrix [23, 15, 13, 11, 12, 26, 4, 21]. The applicable statistical model, such as conjunctive or compensatory models, will vary depending on whether there are hierarchical relations or interactions among the defined attributes. Conjunctive models assume that students can get "correct" for an item only if they have mastered all attributes required for that item and only a fraction of them results in a success probability equal to that of a student possessing none of the attributes. The DINA model (deterministic inputs, noisy-and-gate; [13]) is one of the simplest and most widely studied conjunctive models. This study makes the conjunctive assumption for dichotomously scored test items in CDMs.

2.1 Item Response Matrix, Q-matrix, and Knowledge States Matrix

Suppose there are K attributes in a particular domain. Then a student's attribute patterns $\boldsymbol{\alpha_i} = (\alpha_{i1}, \alpha_{i2...}, \alpha_{iK})$, called knowledge states, indicate the student's mastery status in terms of the K attributes. α_{ik}=1 indicates the ith student's mastery of attribute k and α_{ik}=0 indicates non-mastery of the attribute. As stated above, the Q-matrix indicates the required attributes for each item. The entry of the Q-matrix (denoted q_{jk}) equals one if item j requires attribute k; and zero otherwise. The Q-matrix is used to establish a relationship between the students' responses and the attribute. It is assumed that the item responses are determined by the attributes involved in each item and the attributes mastered by each student.

For conjunctive models, based on the students' knowledge states and Q-matrix for items, an ideal item response matrix R can be generated, whose element r_{ij} is typically represented in the following form.

$$r_{i,j}(A,Q) = \xi_{ij} = \prod_{k=1}^{K} \alpha_{i,k}^{q_{j,k}} = \begin{cases} 1 & (\alpha_{i,k} \ge q_{j,k} : k = 1,..,K) \\ 0 & (\alpha_{i,k} < q_{j,k} : \exists k \in \{1,...,K\}) \end{cases} \quad \text{......} (1)$$

Here r_{ij} is the latent response variable of the ith student with the latent knowledge states $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{jK})$ to the jth item, indicating whether the student i has all the attributes required for item j. It represents a deterministic prediction of item response from each student's knowledge state.

An example of eight students' knowledge states and their ideal response patterns to seven items identified by a Q-matrix is illustrated. Given two matrices $\boldsymbol{A}$ and $\boldsymbol{Q}$, the ideal response matrix $\boldsymbol{R}$ will be obtained as follows.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{Q} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\boldsymbol{R} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

It is important to note that mapping of knowledge states to ideal response patterns is not a one-to-one correspondence; rather, given a particular set of items and a particular Q-matrix, two or more different knowledge states can result in the same ideal response pattern.

2.2 Boolean Matrix Factorization

A recent technique based on Boolean Matrix Factorization (BMF) has been shown to be extremely effective for getting valuable results in binary data analyses [8, 9, 10, 18, 19, 20, 25, 27].

Definition: If $P = (p_{i,j})^{m \times k} \in \{0,1\}^{m \times k}$ and $Q = (q_{i,j})^{k \times n} \in \{0,1\}^{k \times n}$, the Boolean product of P and Q is defined as

$$P \odot Q = (\mathop{\vee}_{s=1}^{k} p_{i,s} q_{s,j})^{m \times n} \in \{0,1\}^{m \times n}.$$

In our previous work involving BMF, we verified that an ideal response matrix R can be expressed in terms of the following Boolean relations of the knowledge states matrix $\boldsymbol{A}$ and the $\boldsymbol{Q}$-matrix (see [30] for details).

$$\boldsymbol{R} = \overline{\overline{\boldsymbol{A}} \odot \boldsymbol{Q}^T} \quad \text{..} (2)$$

Here, $\boldsymbol{A}$ and $\boldsymbol{Q}$ are a m-students by K-attributes binary mastery matrix and an n-items by K-attributes binary Q-matrix, for student i=1,..,m; item j=1,…,n; and attribute k=1,…,K. The bar notation in equation (2) represents logical NOT operation (i.e. $\overline{0} = 1, \overline{1} = 0$), and T represents the transpose of the matrix. The notation ⊙ represents the Boolean product. These notations will apply hereafter.

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_m \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1K} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mK} \end{pmatrix}$$

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_n \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1K} \\ q_{21} & q_{22} & \cdots & q_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nK} \end{pmatrix}$$

The process of Boolean matrix factorization is to determine the matrices $\boldsymbol{Q}$ and $\boldsymbol{A}$ from $\boldsymbol{R}$, and the goal of a factorization algorithm is to minimize the estimated R with R in equation (2). It is clear that equation (2) becomes much more powerful than the usual equation (1) as the number of latent attributes increases.

3. ALTERNATE RECURSIVE METHOD FOR Q-MATRIX LEARNING

3.1 Approach to Response Matrix through Attribute Latent Space Perturbation

Instead of approximating the item response matrix $\boldsymbol{R}$, for simplicity, we will approach its complementary matrix $\overline{\boldsymbol{R}}$ by using matrix $\boldsymbol{H}$=$\boldsymbol{X}$ ⊙ $\boldsymbol{Y}^T$. From equation (2), it is implicit that $X = \overline{A}$ and $Y = Q$, where $\boldsymbol{A}$ and $\boldsymbol{Q}$ are the initial knowledge state matrix and Q-matrix, respectively. In order to approximate $\overline{\boldsymbol{R}}$, we need to perturb $\boldsymbol{X}$ and $\boldsymbol{Y}$ by adding one-column vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ to the latent attribute space (the operations are denoted as $\boldsymbol{X}_{\text{new}}$=[$\boldsymbol{X}$, $\boldsymbol{x}$], and $\boldsymbol{Y}_{\text{new}}$=[$\boldsymbol{Y}$, $\boldsymbol{y}$]). Thus, $\boldsymbol{H}_{\text{new}}$=$\boldsymbol{X}_{\text{new}}$⊙ $(\boldsymbol{Y}_{\text{new}})^T$, which minimizes the following objective error function:

$$E(\boldsymbol{x},\boldsymbol{y}) = (\| \overline{\boldsymbol{R}} - \boldsymbol{X}_{\text{new}} \odot (\boldsymbol{Y}_{\text{new}})^T \|_F)^2$$

where $\| \ \|_F$ indicates the Frobenius norm of the matrix. Without loss of generality, we can fix $\boldsymbol{y}$ and choose $\boldsymbol{x}$ to minimize the error function $E(\boldsymbol{x},\boldsymbol{y})$:

$$\left\| \overline{R} - X_{\text{new}} \odot Y_{\text{new}}^T \right\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (\overline{r}_{ij} - h_{ij} \vee x_i y_j)^2$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij} + h_{ij} \vee x_i y_j - 2\bar{r}_{ij}(h_{ij} \vee x_i y_j))$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij} + h_{ij} + \bar{h}_{ij} x_i y_j - 2\bar{r}_{ij}(h_{ij} + \bar{h}_{ij} x_i y_j))$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij} - \bar{r}_{ij} h_{ij} + h_{ij} - \bar{r}_{ij} h_{ij}) + \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{h}_{ij} x_i y_j - 2\bar{r}_{ij}\bar{h}_{ij} x_i y_j)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij}(1-h_{ij}) + h_{ij}(1-\bar{r}_{ij})) + \sum_{i=1}^{m} x_i (\sum_{j=1}^{n}(1-\bar{r}_{ij})\bar{h}_{ij} y_j - \sum_{j=1}^{n}\bar{r}_{ij}\bar{h}_{ij} y_j)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij}\bar{h}_{ij} + r_{ij} h_{ij}) + \sum_{i=1}^{m} x_i (\sum_{j=1}^{n} r_{ijij}\bar{h}_{ij} y_j - \sum_{j=1}^{n}\bar{r}_{ij}\bar{h}_{ij} y_j)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij}\bar{h}_{ij} + r_{ij} h_{ij}) + \sum_{i=1}^{m} x_i \sum_{j=1}^{n} r_{ij}\bar{h}_{ij} y_j - \sum_{i=1}^{m} x_i \sum_{j=1}^{n}\bar{r}_{ij}\bar{h}_{ij} y_j \quad \cdots (3)$$

When $u,v \in \{0,1\}$, the $\max\{u,v\}=u+v-uv= v+u\bar{v}$ is used to obtain equation (3). The three terms in equation (3) consist of the original ***H*** matrix error and two perturbation errors. We express them as $E_{orig}=\sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{r}_{ij}\bar{h}_{ij}+r_{ij}h_{ij})$, $E_P^{+}(i)=\alpha_i=\sum_{j=1}^{n} r_{ij}\bar{h}_{ij} y_j$, and $E_P^{-}(i) =\beta_i = \sum_{j=1}^{n}\bar{r}_{ij}\bar{h}_{ij} y_j$. Obviously, E_{orig} is the already existing error term which is independent of how ***x*** and ***y*** are optimized. Therefore, in what follows, we focus on the α_i and β_i error terms.

Observation 1: For i=1,2,…,m, if $\bar{r}_{ij}$ = 0 and h_{ij}=0, when x_i=1 the total error of the function E(***x***,***y***) increases by a positive factor $E_P^{+}(i)=\alpha_i=\sum_{j=1}^{n} r_{ij}\bar{h}_{ij} y_j$. Although the original approaching matrix ***H*** doesn't have any error with respect to matrix $\bar{\boldsymbol{R}}$, the new approaching matrix is such that $1=\boldsymbol{H}_{new}(i,j) = h_{ij} \vee x_i y_j > \bar{r}_{ij} =0$; hence, the total error increases.

Observation 2: For i=1,2,…,m, if $\bar{r}_{ij}$ =1 and h_{ij}=0, when x_i=1 the total error of E(***x***,***y***) decreases by a positive factor $E_P^{-}(i) =\beta_i = \sum_{j=1}^{n}\bar{r}_{ij}\bar{h}_{ij} y_j$. Although the original approaching matrix ***H*** has an error with respect to matrix $\bar{\boldsymbol{R}}$, the new approaching matrix is such that $1=\boldsymbol{H}_{new}(i,j) = h_{ij} \vee x_i y_j = \bar{r}_{ij} =1$; hence, the total error decreases.

The above two observations imply that if $\alpha_i \le \beta_i$, setting x_i=1 will decrease the total error of E(***x***,***y***), and if $\alpha_i > \beta_i$, we obviously have to set x_i=0 directly to ensure that the total error does not increase. To control the error in the case of $\alpha_i \le \beta_i$ and setting x_i=1, we exploit two heuristic methods by using a determinate function or by using a sigmoid function based on adjusting coefficients ρ and τ, which are constant parameters bigger than zero, and satisfies $\alpha_i \le \rho \cdot \beta_i$, $0<\rho\le 1$. The two heuristic algorithms based on the BMF approach are illustrated in Figure 1.

Step 1. Input the response matrix $\bar{\boldsymbol{R}} \in \{0,1\}^{m\times n}$, Initialize ***H***=0, ***X***=***Φ***, ***Y***=***Φ***; k=1; for given K

Step 2. Compute $\boldsymbol{P} = (r_{ij}\bar{h}_{ij})_{m\times n}$, $\boldsymbol{Q} = (\bar{r}_{ij}\bar{h}_{ij})_{m\times n}$, $y(j)$=1, when $j=k$; otherwise, $y(j)$=0; j=1,…,n

Step 3. Calculate $\boldsymbol{\alpha} =\boldsymbol{Py}$, $\boldsymbol{\beta} = \boldsymbol{Qy}$, and update ***x*** *based on one of the following two methods*:

1) (*update* method *I*) $\boldsymbol{x}=\theta(\rho\cdot\boldsymbol{\beta}-\boldsymbol{\alpha})$, $0<\rho\le 1$ is a given parameter

2) (*update method II*) Prob $(\boldsymbol{x}(i)=1|\alpha_i,\beta_i)=\dfrac{1}{1+e^{-\tau(\beta_i-\alpha_i)}}$, where τ is a given parameter

Step 4. Compute $\boldsymbol{u} =\boldsymbol{P}^T\boldsymbol{x}$, $\boldsymbol{v} = \boldsymbol{Q}^T\boldsymbol{x}$ and select *one of the following approaches* to update ***y***:

1) $\boldsymbol{y}=\theta(\rho\cdot\boldsymbol{v} -\boldsymbol{u})$, where $0<\rho\le 1$ is a given parameter;

2) Prob $(\boldsymbol{y}(j)=1|u_j,v_j)=\dfrac{1}{1+e^{-\tau(v_i-u_i)}}$ where τ is a given parameter;

Step 5. Repeat Step 3 and Step 4, till ***x*** and ***y*** converge (*update method I*) or the distributions of x and y become stable (*update method II*)

Step 6. Set $\boldsymbol{H} = \boldsymbol{H} \vee (\boldsymbol{xy}^T)$, ***X***=[***X***,***x***], and ***Y***=[***Y***,***y***], k=k+1, if k>K, output ***H***, ***X***, ***Y*** and break,

Else go to step2

Figure 1. Perturbation Approaching Algorithms (PAA)

For PAA algorithm we have the following convergence theorem 1.

Theorem 1.

1) For PPA deterministic algorithms, suppose H be the original matrix and Hnew be the updated matrix, then $\|\bar{\boldsymbol{R}} - X_{new} \odot (Y_{new})^T\|_F \le \|\bar{\boldsymbol{R}} - H\|_F$. holds.

2) By the same token, for PPA stochastic algorithm, it converges each step at the final equilibrium distribution $P(x,y)=(1/Z)\exp(-x^T(P-Q)y/\tau)$, wherein Z is a normalized constant for probability distribution .

Proof: Define local energy function as $Eloc(x,y)= x^T(P-Q)y$.

1) For PPA deterministic updating algorithm, if we see it as Hopfield neural network bidirectional associative computing updating algorithm [29], it holds according to Hopfield convergence theorem of bidirectional associative memory (BAM).

2) For the PPA stochastic algorithm, if we see it as a limitation Boltzmann machine in neural network [29] , according to the Boltzmann convergence theorem, every step of the final x, y equilibrium distribution will be $P(x,y)=(1/Z)\exp(-x^T(P-Q)y/\tau)$, wherein Z is a probability distribution of a normalization factor , τ is called Boltzmann machine annealing temperature which should be gradually reduced during the iteration . QED.

As each step of the optimization process are discrete variables {0,1} quadratic optimization problem , they are all NP-hard problem according to the computational complexity theory. Therefore there will always be a local optimum, which is also the reason why sometimes PPA stochastic algorithm is used. While deterministic algorithm can always guarantee that each time one attribute is added, the error of approaching response matrix with the real response matrix R can be reduced accordingly, although it may eventually fall into local minimum of the local error Eloc. The stochastic algorithm uses that random error energy can increase certain probability in an iterative process, so when the annealing temperature decreases slowly enough, the conclusions based on simulated annealing algorithm, it is possible to converge to the probability of a global optimum.

3.2 A Fast Alternating Recursive Algorithm

Due to the monotony of the Boolean matrix product, the algorithm we proposed in 3.1 often relapses into a local optimal solution. We suggested an alternative recursive algorithm to improve the accuracy of approaching matrix $\boldsymbol{H}$. For given $\boldsymbol{H}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_K) \odot (\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_K)^T$, let $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_K)$, $\boldsymbol{Y}=(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_K)$, $\boldsymbol{X}^i=(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},\ldots,\boldsymbol{x}_K)$, $\boldsymbol{Y}^i=(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_{i-1},\boldsymbol{y}_{i+1},\ldots,\boldsymbol{y}_K)$, $\boldsymbol{H}^i=\boldsymbol{X}\odot(\boldsymbol{Y}^i)^T$. Therefore, we have $\boldsymbol{H}=\boldsymbol{H}^i\vee \boldsymbol{x}_i(\boldsymbol{y}_i)^T$. The index i can be chosen by using random or deterministic methods. Using the results of Perturbation Approaching Algorithms (PAA) as initial values, one can iteratively optimize the perturbation vector $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ for fixed $\boldsymbol{X}^i$ and $\boldsymbol{Y}^i$. In practice, when the dimension of attribute space K is quite large, we need to reuse the previous results for the next iteration on the perturbation vectors $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ in order to reduce the time complexity of the algorithm. However, because of the special characteristics of the Boolean matrix product, we cannot make use of $\boldsymbol{H}^j$ in the previous round, since $\boldsymbol{H}^j\neq\boldsymbol{H}^i\vee \boldsymbol{x}_i(\boldsymbol{y}_i)^T \wedge(\bar{\boldsymbol{x}}_j\bar{\boldsymbol{y}}_j^T)$. Luckily, the algorithm can be sped up by using the general matrix product instead of the Boolean matrix product if we introduce the Heaviside step function as follows.

Define $\theta(x)$ as the Heaviside step function.

$$\theta(x)=\begin{cases}1 & x>0\\ 0 & x\le 0\end{cases} \quad (4)$$

Given a matrix $B=(b_{i,j})^{m\times n}\in R^{m\times n}$, we define its Heaviside step function as $\theta(B)=(\theta(b_{i,j}))^{m\times n}\in R^{m\times n}$; namely, $\theta(x)$ acts on every element of matrix B.

Property 1. $P=(p_{i,j})^{m\times n}\in R_+^{m\times n}$ and $Q=(q_{i,j})^{m\times n}\in R_+^{m\times n}$ Here, $R_+=\{x,x\ge 0\}$. If $\lambda\ge 1$ exists such that $P\ge\lambda Q$, we have $\theta(P)=\theta(P-Q)$.

Proof: If $q_{i,j}=0$, it immediately follows that

$\theta(p_{i,j})=\theta(p_{i,j}-q_{i,j})=0$.

On the other hand, if $q_{i,j}>0$, from the assumptions, we have

$p_{i,j}-q_{i,j}>(\lambda-1)q_{i,j}$

Therefore, $\theta(p_{i,j})=\theta(p_{i,j}-q_{i,j})=1$,

Hence, we have $\theta(P)=\theta(P-Q)$. QED.

Property 2. $P=(p_{i,j})^{m\times n}\in R_+^{m\times n}$ and $Q=(q_{i,j})^{m\times n}\in R_+^{m\times n}$, here $R_+=\{x,x\ge 0\}$. Then we have $\theta(PQ)=\theta(P)\odot\theta(Q)$

Proof: Let $G=(g_{i,j})^{n\times k}=PQ$. If $g_{i,j}=0$, then $\theta(g_{i,j})=0$. While if $g_{i,j}>0$, then $\theta(g_{i,j})=1$.

In case of $g_{i,j}=\sum_{t=1}^{k}p_{i,t}q_{t,j}=0$, because P and Q are nonnegative matrices, we have $p_{i,t}q_{t,j}=0$ for all $1\le t\le n$, which leads to

$\theta(p_{i,t}q_{t,j})=0$.

From the definition of Heaviside step function θ, i.e. formula (4), we have

$\theta(p_{i,t})\theta(q_{t,j})=0$.

Therefore, we have

$\bigvee_{t=1}^{n}\theta(p_{i,t})\theta(q_{t,j})=\theta(g_{i,j})=0$.

On the other hand, in case of $g_{i,j}=\sum_{t=1}^{k}p_{i,t}q_{t,j}>0$, because P and Q are nonnegative matrices, there exists $p_{i,t}q_{t,j}>0$ for $1\le t\le n$, which leads to

$\theta(p_{i,t}q_{t,j})=1$.

From definition of θ, we have $\theta(p_{i,t})\theta(q_{t,j})=1$.

Therefore, we have

$\bigvee_{t=1}^{n}\theta(p_{i,t})\theta(q_{t,j})=1$

Considering the above two cases, we have

$\theta(PQ)=\theta(P)\odot\theta(Q)$

hold for $P=(p_{i,j})^{m\times n}\in R_+^{m\times n}$ and $Q=(q_{i,j})^{m\times n}\in R_+^{m\times n}$. QED.

For given matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, we generate $\boldsymbol{G}=\boldsymbol{X}\boldsymbol{Y}^T$, $\boldsymbol{G}_i=\boldsymbol{X}_i(\boldsymbol{Y}_i)^T$ by taking the general matrix product. According to the above properties, $\boldsymbol{H}=\theta(\boldsymbol{G})$ and $\boldsymbol{H}_i=\theta(\boldsymbol{G}_i)$ hold. We update $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ after updating $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ in the previous round. Instead of computing the whole matrix $\boldsymbol{H}_j$, we calculate

$$\boldsymbol{G}^j=\boldsymbol{G}^i+\boldsymbol{x}_i\boldsymbol{y}_i^T-\boldsymbol{x}_j\boldsymbol{y}_j^T \quad (5)$$

$$\boldsymbol{H}^j=\theta(\boldsymbol{G}^j) \quad (6)$$

In equation (5), $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the updated values and $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ are the original values. Equation (5) and (6) enable us to compute matrix $\boldsymbol{H}^j$ quickly. The alternating approach algorithm is as follows in Figure 2.

For AIA algorithms, we have the following convergence theorem 2.

Theorem 2.

1) For AIA deterministic algorithms, suppose H be the original matrix and $\boldsymbol{H}_{new}$ be the updated matrix, then

$$\|\bar{\boldsymbol{R}}-X_{\text{new}}\odot(\boldsymbol{Y}_{\text{new}})^T\|_F\le\|\bar{\boldsymbol{R}}-H\|_F.$$

2) By the same token, for AIA stochastic algorithm, it converges each step at the final equilibrium distribution

$$P(x,y)=(1/Z)\exp(-\|\bar{\boldsymbol{R}}-X\odot Y^T\|_F/\tau),$$

wherein Z is a normalized constant for probability distribution.

Proof: Define local energy function as Eloc (x, y) $=\|\bar{\boldsymbol{R}}-X\odot Y^T\|_F$.
1) For AIA deterministic updating algorithm, if we see its 3rd step as Hopfield neural network bidirectional associative computing

updating algorithm [29], it holds according to Hopfield convergence theorem of bidirectional associative memory.

2) For the AIA stochastic algorithm, if we see its 3rd step as a simulated annealing algorithm [29], according to the conclusions of simulated annealing, the final equilibrium distribution $P(x,y)=(1/Z)\exp(-\|\bar{R}-X\odot Y^T\|_F/\tau)$ can be obtained, wherein Z is a probability distribution of a normalization constant , τ is called Boltzmann machine annealing temperature, which should be gradually reduced during the iteration. QED.

> Step 1. Input the response matrix $\bar{\boldsymbol{R}} \in \{0,1\}^{m\times n}$ and $\boldsymbol{X}\in\{0,1\}^{m\times k}$, $\boldsymbol{Y}\in\{0,1\}^{n\times k}$
>
> Step 2. Compute $\boldsymbol{G}=\boldsymbol{X}\boldsymbol{Y}^T$.
>
> Step 3. Randomly select (or deterministically select) l, $1\le l\le K$, and update the l columns of $\boldsymbol{X}$ and $\boldsymbol{Y}$.
>
> 3.1 Compute $\boldsymbol{G}_l=\boldsymbol{X}(:,l)\boldsymbol{Y}(:,l)^T$, $\boldsymbol{G}^l=\boldsymbol{G}-\boldsymbol{G}_l$, $\boldsymbol{H}=\theta(\boldsymbol{G}^l)$, $U=(r_{ij}\bar{h}_{ij})_{m\times n}$, $V=(\bar{r}_{ij}\bar{h}_{ij})_{m\times n}$;
>
> 3.2 Set $\boldsymbol{y}=\boldsymbol{Y}(:,l)$;
>
> 3.3 Calculate $\boldsymbol{\alpha}=\boldsymbol{U}\boldsymbol{y}$, $\boldsymbol{\beta}=\boldsymbol{V}\boldsymbol{y}$, and update $\boldsymbol{x}$ *based on one of the following methods*:
>
> 1) (*update* method *I*) $\boldsymbol{x}=\theta(\rho\cdot\boldsymbol{\beta}-\boldsymbol{\alpha})$, where $0<\rho\le1$ is a given parameter;
>
> 2) (*update method II*) Prob $(\boldsymbol{x}(i)=1|\alpha_i,\beta_i)=\dfrac{1}{1+e^{-\tau(\beta_i-\alpha_i)}}$, where τ is a given parameter;
>
> 3.4 Compute $\gamma=\boldsymbol{U}^T\boldsymbol{x}$, $\delta=\boldsymbol{V}^T\boldsymbol{x}$, and update $\boldsymbol{y}$ *based on one of the following methods*:
>
> 1) $y=\theta(\rho\delta-\gamma)$, where $0<\rho\le1$ is a given parameter;
>
> 2) $\Pr ob(y(j)=1)|\gamma_j,\delta_j)=\dfrac{1}{1+e^{-\tau(\delta_j-\gamma_j)}}$, where τ is a given parameter;
>
> 3.5 Repeat steps 3.3 and 3.4, till $\boldsymbol{x}$ and $\boldsymbol{y}$ converge (*update method I*) or the distributions of x and y become stable (*update method II*).
>
> 3.6 Update $\boldsymbol{X}$: $\boldsymbol{X}(:,l)=\boldsymbol{x}$;
>
> Update $\boldsymbol{Y}$: $\boldsymbol{Y}(:,l)=\boldsymbol{y}$;
>
> Update $\boldsymbol{G}$: $\boldsymbol{G}=\boldsymbol{G}^l+\boldsymbol{x}\boldsymbol{y}^T$;
>
> Step 4. Repeat step 3 till the change of values less than a given threshold.

Figure 2. Alternating Iteration Algorithms (AIA)

According to the conclusions of simulated annealing the AIA stochastic algorithm can obtain global optimal Q-matrix and A matrix when attributes fixed, as long as we assure the decreasing rate of annealing temperature being slow enough.

4. EXPERIMENTAL RESULTS

Q-matrix Reproduction from Real Response Data

To show how the proposed methods work, real responses to an actual test we developed is used. The test is a fraction diagnostic test comprised of 35 items in terms of the conjunctive assumption. Eight attributes have been specified and developed by content experts as essential skill required in solving fraction problem according to the "Japanese government curriculum guidelines for teaching" (for details, see [22]). We administered the test to 144 sixth grade students in an elementary school in Tokyo and the real response data of 144 students to 35 items are used as ***R*** matrix.

We use the proposed algorithms to approach this response matrix ***R*** and reproduce the ***Q***-matrix and knowledge states matrix ***A***. Specifically we use the PAA algorithm illustrated in Figure 1 to optimize latent matrices ***Q*** (denoted as ***X***) and ***A*** (denoted as ***Y*** for complementary ***A***) first. By setting the results as initial values we use the other AIA algorithm illustrated in Figure 2 then to iteratively optimize the perturbation and get the final estimated matrices ***Q***, ***A*** and approaching matrix ***H*** of the complementary matrix ***R***. As shown in the above section, there are two update methods in each algorithm but here we only show the results of update method I.

It is indicated that calculations converge within 15-20 iterations for all ρ parameter ranging from 0 to 1. An example for ρ=0.5 is shown in Figure 3. The vertical axis represents the coverage rate indicating how well the generated approaching matrix has reproduced the original response ***R*** matrix. We can see that the initial value estimated by the PAA algorithm covers a little more than 85% of the original ***R*** and by the AIA algorithm the coverage rate increases rapidly at first 5-6 iterations moving towards a stable value (Figure 3). The convergence plots for all ρ parameter have the same trends, although the coverage rate for each one is a slight different.

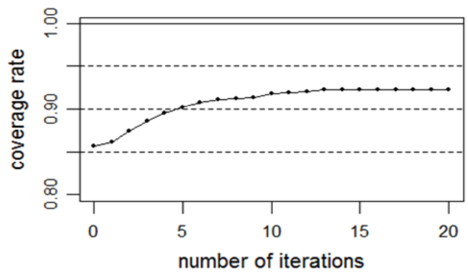

Figure 3. Converging of coverage rate (ρ=0.5)

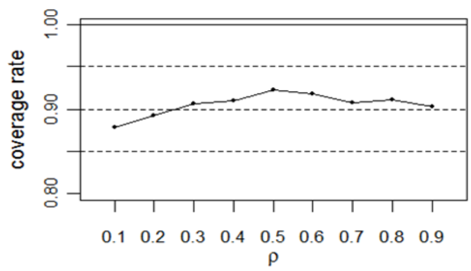

Figure 4. Coverage rates along with different ρ

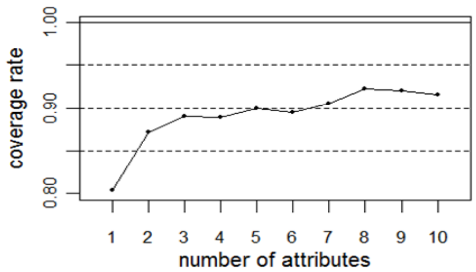

Figure 5. Coverage rates with number of attributes (ρ=0.5)

The coverage rate at 20 iterations for each ρ is given in Figure 4. We can see that despite of different ρ the coverage rates are as high as around 90%, which indicates our algorithms are valid and give good optimization and reproduction from the original response data.

Figure 5 shows how well the original response ***R*** matrix is reproduced by different number of attributes. The coverage rate, starting from 80% by single attribute, has been increasing as the number of attributes increases reaching the peak at eight attributes. It is interesting to notice that "eight" are the number the content experts specified and considered to be appropriate for the particular test in our previous study [22].

5. CONCLUSION AND FUTURE WORK

We used Boolean Matrix Factorization (BMF) to express conjunctive models in CDMs and proposed recursive algorithms for updating the matrix in latent attributes space (its rank equals one) at each step in order to get optimal solutions. We also used a Heaviside step function to decompose the recursive computing process into normal non-negative matrices and get results by mapping them back into a Boolean matrix, which makes our approximation algorithms faster. Two different algorithms were presented: a deterministic heuristic algorithm and a stochastic algorithm. We presented examples demonstrating applications of one of these algorithms based on an actual test dataset. The results indicate that Q-matrix learned can reproduce item response data with more than 90% coverage rate, which suggests that our algorithms are valid.

As the next step we will compare the Q-matrix learned from the data by the proposed methods with the one created by experts in our previous research [22]. We will also introduce statistical parameters, such as "guessing" and "slip", to make our methods more applicable for real data.

6. REFERENCES

[1] Barnes, T. 2010. Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on Educational Data Mining*, 159-172.

[2] 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

[3] Carpineto, C., Romano, G. 2004. *Concept Data Analysis. Theory and Applications*. Wiley.

[4] Chiu, C.-Y., Douglas, J. A., and Li, X. 2009. Cluster analysis for cognitive diagnosis: Theory and applications. *Psychometrika*, 74(4), 633-665.

[5] 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(4), 343-362.

[6] DeCarlo, L.T. 2011. On the analysis of fraction subtraction data: the DINA model, classification, latent class sizes, and the Q-matrix. Applied Psychological Measurement 35, 8-26.

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

[8] Desmarais, M.C., Beheshti, B., Naceur, R. Item to skills mapping: Deriving a conjunctive Q-matrix from data. *Proceeding of ITS'12 Proceedings of the 11th international conference on Intelligent Tutoring Systems*, 454-463.

[9] Desmarais, M.C.; Naceur, R. 2013. A Matrix Factorization Method for Mapping Items to Skills and for Enhancing Expert-Based Q-matrices. *Artificial Intelligence in Education, Lecture Notes in Computer Science,* 7926, 441-450.

[10] Desmarais, M.C. 2011. Mapping Question Items to Skills with Non-negative Matrix Factorization. *ACM SIGKDD Explorations Newsletter archive*, 13(2), 30-36.

[11] DiBello, L. V., Roussos, L. A., and Stout, W. 2007. Review of cognitively diagnostic assessment and a summary of psychometric models. *Handbook of statistics*. 26(31), 1-52.

[12] Hartz, S. M. 2002. *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality*. PhD thesis, University of Illinois at Urbana-Champaign.

[13] Junker, B. W. and Sijtsma, K. 2001. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(1), 258-273.

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

[15] Leighton, J. P., Gierl, M. J., and Hunka, S. M. 2004. The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement*, 41(3), 205-237.

[16] Li, N., Cohen, W.W., Matsuda, N., 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.

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

[18] Miettinen, P., Mielik?inen, T., Gionis, A., Das, G., Mannila, H. 2006. The discrete basis problem, in: Proc. PKDD 2006, in: Lecture Notes in Artificial Intelligence, 4213, 335-346.

[19] Miettinen, P., Mielikainen, T., Gionis, A., Das, G., Mannila, H. 2008. The Discrete Basis Problem. *IEEE Transactions on Knowledge and Data Engineering*, 20(10), 1348-1362.

[20] Neruda, R., Snasel,V., Platos,J., Kromer,P., Husek, D. 2008. Implementing Boolean Matrix Factorization. *ICANN 2008, Part I, LNCS* 5163, 543-552.

[21] 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.

[22] Takahashi, T., Sun, Y., Kakinuma, S. 2011. Development of attributes of cognitive diagnostic test in fraction calculations, in *Proceeding of 2011 conference of the Japan Association for Research on Testing*, 220-221.

[23] Tatsuoka, K. K. 1983. Rule space: an approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4), 345-354.

[24] Tatsuoka, K. K. 1985. A probabilistic model for diagnosing misconceptions in the pattern classification approach. *Journal of Educational Statistics*, 10(1), 55-73.

[25] Vaidya, J. 2012. Boolean Matrix Decomposition Problem: Theory, Variations and Applications to Data Engineering. *IEEE 28th International Conference on Data Engineering*

[26] von Davier, M. 2005. A general diagnostic model applied to language testing data. *ETS Research Report RR-05-16*. Princeton: Educational Testing Service

[27] Wille, R. 1982. Restructuring lattice theory: An approach based on hierarchies of concepts, in: I. Rival (Ed.), *Ordered Sets*, Reidel, Dordrecht/Boston, 445-470.

[28] Xiang, R. 2013. *Nonlinear Penalized Estimation of True Q-Matrix in Cognitive Diagnostic Models*. PhD thesis, Columbia University.

[29] Haykin, S. 2009. Neural networks and learning machines, 3rd ed., Prentice Hall, NJ.

[30] Ye, S., Sun, Y., Inoue, S., Sun, Y. 2014. Matrix extending methods for Q-matrix learning (work paper in progress).