Data-driven Learning of Q-matrix

Jingchen Liu, Gongjun Xu, and Zhiliang Ying

Columbia University

Abstract

The recent surge of interests in cognitive assessment has led to developments of novel statistical models for diagnostic classification. Central to many such models is the well-known Q-matrix, which specifies the item-attribute relationships. This paper proposes a data-driven approach to identification of the Q-matrix and estimation of related model parameters. A key ingredient is a flexible T-matrix that relates the Q-matrix to response patterns. The flexibility of the T-matrix allows construction of a natural criterion function as well as a computationally amenable algorithm. Simulations results are presented to demonstrate usefulness and applicability of the proposed method. Extension to handling of the Q-matrix with partial information is presented. The proposed method also provides a platform on which important statistical issues, such as hypothesis testing and model selection, may be formally addressed.

Keywords: Cognitive diagnosis, DINA model, DINO model, latent traits, model selection, multidimensionality, optimization, self-learning, statistical estimation.

1 Introduction

Diagnostic classification models (DCM) are an important statistical tool in cognitive diagnosis and can be employed in a number of disciplines, including educational assessment and clinical psychology (Rupp and Templin, 2008b; Rupp, Templin, and Henson, 2010). A key component in many such models is the so-called Q-matrix, which specifies item-attribute relationships, so that responses to items can reveal the attribute configurations of the respondents. Tatsuoka (1983, 2009) proposed the simple and easy-to-use rule space method for Q-matrix based classifications.

Different DCMs can be built around the Q-matrix. One simple and widely studied model among them is the DINA model (Deterministic Input, Noisy output "AND" gate; see Macready

and Dayton, 1977; Junker and Sijtsma, 2001). Other important developments can be found in Tatsuoka (1985); DiBello, Stout, and Roussos (1995); Junker and Sijtsma (2001); Hartz (2002); Tatsuoka (2002); Leighton, Gierl, and Hunka (2004); von Davier (2005); Templin (2006); Templin and Henson (2006); Chiu, Douglas, and Li (2009). Rupp et al. (2010) contains a comprehensive summary of many classical and recent developments.

There is a growing literature on the statistical inference of Q-matrix based DCMs that addresses the issues of item parameters estimation when the Q-matrix is prespecified (Rupp, 2002; Henson and Templin, 2005; Roussos, Templin, and Henson, 2007; Stout, 2007). Having a correctly specified Q-matrix is crucial both for parameter estimation (such as the slipping, guessing probability, and the attribute distribution) and for the identification of subjects' underlying attributes. As a result, these approaches are sensitive to the choice of the Q-matrix (Rupp and Templin, 2008a; de la Torre, 2008; de la Torre and Douglas, 2004). For instance, a misspecified Q-matrix may lead to substantial lack of fit and, consequently, erroneous attribute identification. Thus, it is desirable to be able to detect misspecification and to obtain a data driven Q-matrix.

In this paper, we consider the estimation problem of the Q-matrix. In particular, we introduce an estimator of the Q-matrix under the setting of the DINA model. The proposed estimator only uses the information of dependence structure of the responses (to items) and does not rely on information about the attribute distribution, or the slipping, or guessing parameters. The definition of these concepts will be provided in the text momentarily. Nonetheless, if additional information is available such as a parametric form of the attribute distribution or partial information about the Q-matrix, the estimation procedure is flexible enough to incorporate those structures. Such information, if correct, can be substantially improve the efficiency of the estimator, enhance the identifiability of the Q-matrix, and reduce the computational complexity. In addition to the construction of the estimator, we also provide computational algorithms and simulation studies to assess the performance of the proposed procedure.

It is worth pointing out that our method is in fact generic in the sense that it can be adapted to cover a large class of DCMs besides the DINA model. In particular, the procedure is implementable to the DINO (Deterministic Input, Noisy output "OR" gate) model, the NIDA (Noisy Inputs, Deterministic "And" Gate) model, and the NIDO (Noisy Inputs, Deterministic "Or" Gate), model among others. In addition to the estimation of the Q-matrix, we emphasize that the main idea

behind the derivations forms a principled inference framework. For instance, during the course of the description of the estimation procedure, necessary conditions for a correctly specified Q-matrix are naturally derived. Such conditions can be used to form appropriate statistics for hypothesis testing and model diagnosis. In connection to that, additional developments (e.g. the asymptotic distributions of the corresponding statistics) are needed, but they are not the focus of the current paper. Therefore, the proposed framework can potentially serve as a principled inference tool for the Q-matrix in diagnostic classification models.

This paper is organized as follows. Section 2 is a presentation of the estimation procedures and the corresponding algorithms. Section 3 includes simulation studies to assess the performance of the proposed estimation methods.

2 Estimation of the Q-matrix

We will be concerned with the situation that N subjects taking a test consisting of J items. The responses are binary, so that the data will be an $N \times J$ matrix with entries being 0 or 1. The diagnostic classification model to be considered for such data envisions K attributes that are related to both the subjects and the items.

2.1 Setup and notation

The following notation and specifications are needed to describe the diagnostic classification models.

Responses to items: There are J items and we use $\mathbf{R} = (R^1, ..., R^J)^\top$ to denote the vector of responses to them, where, for each j, R^j is a binary variable taking 0 or 1, and superscript $\top$ denotes transpose.

Attribute profile: There are K attributes and we use $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)^\top$ to denote the vector of attributes, where $\alpha_k = 1$ or 0, indicating the presence or absence of the k-th attribute, $k = 1, \ldots, K$.

Note that both $\boldsymbol{\alpha}$ and $\mathbf{R}$ are subject-specific. Throughout this paper, we assume that the number of attributes K is known and that the number of items J is observed.

Q-matrix: This describes the link between the items and the attributes. In particular, $Q = (Q_{jk})_{J\times K}$ is an $J \times K$ matrix with binary entries. For each j and k, $Q_{jk} = 1$ indicates that item j requires attribute k and $Q_{jk} = 0$ otherwise.

Let $\xi^j(\boldsymbol{\alpha}, Q)$ denote the *ideal response*, which indicates if a subject possessing attribute profile $\boldsymbol{\alpha}$ is capable of providing a positive response to item j if the item-attribute relationship is specified by matrix Q. Different ideal response structures give rise to different DCMs. For instance,

$$\xi^j_{DINA}(\boldsymbol{\alpha}, Q) = \mathbf{1}(\alpha_k \geq Q_{jk} \text{ for all } k = 1, ..., K) \tag{1}$$

is associated with the DINA model, where $\mathbf{1}$ is the usual indicator function. The DINA model assumes conjunctive relationship among attributes, that is, it is necessary to possess all the attributes indicated by the Q-matrix to be capable of providing a positive response to an item. In addition, having additional unnecessary attributes does not compensate for a lack of the necessary attributes. To simplify the notation, we write $\xi^j_{DINA} = \xi^j$.

The last ingredient of the model specification is related to the so-called slipping and guessing parameters (Junker and Sijtsma, 2001). The concept is due to Macready and Dayton (1977) for mastery testing; see also van der Linden (1978). The slipping parameter is the probability that a subject (with attribute profile $\boldsymbol{\alpha}$) responds negatively to an item if the ideal response to that item $\xi(\boldsymbol{\alpha}, Q) = 1$; similarly, the guessing parameter refers to the probability that a subject responds positively if his or her ideal response $\xi(\boldsymbol{\alpha}, Q) = 0$. We use s to denote the slipping probability and g to denote the guessing probability (with corresponding subscript indicating different items). In the discussion, it is more convenient to work with the complement of the slipping parameter. Therefore, we define $c = 1 - s$ to be the probability of answering correctly, with s_j and c_j being the corresponding item-specific notation. Given a specific subject's profile $\boldsymbol{\alpha}$, the response to item j under the DINA model follows a Bernoulli distribution

$$P(R^j = 1 | Q, \boldsymbol{\alpha}, c_j, g_j) = c_j^{\xi^j(\boldsymbol{\alpha}, Q)} g_j^{1-\xi^j(\boldsymbol{\alpha}, Q)}. \tag{2}$$

In addition, conditional on $\boldsymbol{\alpha}$, $(R^1, ..., R^J)$ are jointly independent.

Lastly, we use subscripts to indicate different subjects. For instance, $\mathbf{R}_r = (R^1_r, ..., R^J_r)^\top$ is the response vector of subject r. Similarly, $\boldsymbol{\alpha}_r$ is the attribute vector of subject r. With N subjects, we observe $\mathbf{R}_1, ..., \mathbf{R}_N$ but not $\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_N$. We further assume that the attribute profiles are i.i.d.

so that

$$P(\boldsymbol{\alpha}_r = \boldsymbol{\alpha}) = p_{\boldsymbol{\alpha}}$$

and let $\mathbf{p} = (p_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \{0,1\}^K)$, $\mathbf{c} = (c_1, ..., c_J)$, and $\mathbf{g} = (g_1, ..., g_J)$. Thus, we have completed our model specification.

2.2 Estimation of the Q-matrix

Intuition. The attribute profiles of examinees are not directly observed. Thus, the estimator of the Q-matrix is built only on the information contained in response vectors, $\mathbf{R}_1$,..., $\mathbf{R}_N$. The estimation of the Q-matrix is based on an assessment of how well a given matrix Q fits the data. Throughout the discussion, we use Q to denote the true matrix that generates the data and Q' to denote a generic J by K matrix with binary entries. In particular, each Q-matrix along with the corresponding parameters $(Q', \mathbf{p}, \mathbf{c}, \mathbf{g})$ determines the distribution of the response vector $\mathbf{R}$ given by

$$P(\mathbf{R}|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} \prod_{j=1}^{J} P(R^j|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g}). \tag{3}$$

We further consider the (observed) empirical distribution

$$\hat{P}(\mathbf{R}) = \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{R}_i = \mathbf{R}). \tag{4}$$

If the Q-matrix and the other parameters, $(Q', \mathbf{p}, \mathbf{c}, \mathbf{g})$, are correctly specified the empirical distribution in (4) eventually converges to (3) as the sample size (the number of subjects) becomes large. The estimator is then constructed based on this observation.

The T-matrix. The T-matrix is central to the construction of our estimator. It is another representation of the Q-matrix and serves as a connection between the observed response distribution and the model structure. In particular, it sets up a linear dependence between the attribute distribution and the response distribution. It is a tool that allows the expression of the probabilities in (3) in terms of matrix products. We first specify each row vector of the T-matrix. For each item j, recall that

$$P(R^j = 1|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} P(R^j = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g}), \tag{5}$$

where $P(R^j = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g}) = (c_j - g_j)\xi^j(\boldsymbol{\alpha}, Q') + g_j$. If we create a row vector $B_{Q',\mathbf{c},\mathbf{g}}(j)$ of length 2^K containing the probabilities $P(R^j = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g})$ for all $\boldsymbol{\alpha}$'s and arrange those elements in an appropriate order, then for all j we can write (5) in the form of a matrix product

$$\sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} P(R^j = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g}) = B_{Q',\mathbf{c},\mathbf{g}}(j)\mathbf{p},$$

where $\mathbf{p}$ is the column vector containing the probabilities $p_{\boldsymbol{\alpha}}$. Similarly, for each pair of items, we may establish that the probability of responding positively to both items j_1 and j_2 is

$$\begin{aligned} P(R^{j_1} = 1, R^{j_2} = 1|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) &= \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} P(R^{j_1} = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g}) P(R^{j_2} = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g}) \\ &= B_{Q',\mathbf{c},\mathbf{g}}(j_1, j_2)\mathbf{p}, \end{aligned}$$

where $B_{Q',\mathbf{c},\mathbf{g}}(j_1, j_2)$ is a row vector containing the probabilities $P(R^{j_1} = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g})P(R^{j_2} = 1|Q', \boldsymbol{\alpha}, \mathbf{c}, \mathbf{g})$ for each $\boldsymbol{\alpha}$. Note that each element of $B_{Q',\mathbf{c},\mathbf{g}}(j_1, j_2)$ is the product of the corresponding elements of $B_{Q',\mathbf{c},\mathbf{g}}(j_1)$ and $B_{Q',\mathbf{c},\mathbf{g}}(j_2)$. With a completely analogous construction, we have that

$$P(R^{j_1} = 1, ..., R^{j_l} = 1|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) = B_{Q',\mathbf{c},\mathbf{g}}(j_1, ..., j_l)\mathbf{p},$$

for each combination of distinct $(j_1, ..., j_l)$. Similarly, $B_{Q',\mathbf{c},\mathbf{g}}(j_1, ..., j_l)$ is the element-by-element product of $B_{Q',\mathbf{c},\mathbf{g}}(j_1)$,...,$B_{Q',\mathbf{c},\mathbf{g}}(j_l)$. From a computational point of view, one only needs to construct the $B_{Q',\mathbf{c},\mathbf{g}}(j)$'s for each individual item j and then take products to obtain the corresponding combinations.

The T-matrix has 2^K columns. Each row vector of the T-matrix is one of the vectors $B_{Q',\mathbf{c},\mathbf{g}}(j_1, ..., j_l)$, i.e., the T-matrix is a stack of B-vectors

$$T_{\mathbf{c},\mathbf{g}}(Q') = \begin{pmatrix} B_{Q',\mathbf{c},\mathbf{g}}(1) \\ \vdots \\ B_{Q',\mathbf{c},\mathbf{g}}(J) \\ B_{Q',\mathbf{c},\mathbf{g}}(1,2) \\ \vdots \end{pmatrix}.$$

By the definition of the B-vectors, we have that

$$T_{\mathbf{c},\mathbf{g}}(Q')\mathbf{p} = \begin{pmatrix} P(R^1 = 1|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) \\ \vdots \\ P(R^J = 1|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) \\ P(R^1 = 1, R^2 = 1|Q', \mathbf{p}, \mathbf{c}, \mathbf{g}) \\ \vdots \end{pmatrix} \tag{6}$$

is a vector containing the corresponding probabilities associated with a particular set of parameters $(Q', \mathbf{c}, \mathbf{g}, \mathbf{p})$. We further define β to be the vector containing the probabilities (corresponding to those in (6)) of the empirical distribution, e.g., the first element of β is $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(R_i^1 = 1)$ and the $(J+1)$-th element is $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(R_i^1 = 1, R_i^2 = 1)$. With a large sample and a set of correctly specified parameters $(Q, \mathbf{c}, \mathbf{g}, \mathbf{p})$, we have that

$$\beta \to T_{\mathbf{c},\mathbf{g}}(Q)\mathbf{p} \tag{7}$$

almost surely as $N \to \infty$.

An illustrative example. To aid the understanding of the T-matrix, we provide one simple example. Suppose that we are interested in testing two attributes. The population is naturally divided into four strata. The corresponding contingency table of attributes is

	Attribute 2	
Attribute 1	p_{00}	p_{01}
	p_{10}	p_{11}

Let vector $\mathbf{p} = (p_{00}, p_{10}, p_{01}, p_{11})^\top$ contain all the corresponding probabilities in this particular order. Consider an exam containing three problems and admitting the following Q-matrix,

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{8}$$

To simplify the discussion, we consider the case that $c_i = 1$ and $g_i = 0$, that is, there is no chance of slipping or guessing. Thus, the response $\mathbf{R}$ is completely determined by the attribute profile $\boldsymbol{\alpha}$. Under this simplified situation, if the Q-matrix is correctly specified, we should be able to obtain the following identities

$$\begin{aligned} p_{10} + p_{11} &= N_1/N, \\ p_{01} + p_{11} &= N_2/N, \\ p_{11} &= N_3/N, \end{aligned}$$

where $N_j = \sum_{r=1}^{N} I(R_i^j = 1)$ is the total number correct responses to item j. We then create the corresponding T-matrix and β-vector as follows

$$T_{\mathbf{c},\mathbf{g}}(Q) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \beta = \begin{pmatrix} N_1/N \\ N_2/N \\ N_3/N \end{pmatrix}. \tag{9}$$

The first column of $T_{\mathbf{c},\mathbf{g}}(Q)$ corresponds to the zero attribute profile; the second corresponds to $\boldsymbol{\alpha} = (1, 0)$; the third corresponds to $\boldsymbol{\alpha} = (0, 1)$; and the last corresponds to $\boldsymbol{\alpha} = (1, 1)$. The first row of $T(Q)$ corresponds to item one, the second to two, and the third to three. Note that the T-matrix changes as the Q-matrix changes. For instance, for an alternative matrix

$$Q' = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

the corresponding T-matrix would be

$$T_{\mathbf{c},\mathbf{g}}(Q') = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}; \tag{10}$$

while the β-vector remains.

To illustrate our idea, consider the matrix in (8). With a correctly specified Q-matrix, we can establish that

$$T_{\mathbf{c},\mathbf{g}}(Q)\mathbf{p} = \beta. \tag{11}$$

Note that the β-vector is directly observed and the attribute distribution $\mathbf{p}$ is not. Thus, the above display suggests that a necessary condition for a correctly specified Q-matrix is that the above linear equations (with $\mathbf{p}$ being the variable) has a solution subject to the natural condition that $\sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} = 1$.

If for any misspecified Q-matrix, the equation (11) does not have any solution, then the Q-matrix is identifiable. Otherwise, we may include more constraints in the T-matrix to enhance the identifiability. For instance, we may further consider the combination of items one and two, that is,

$$p_{11} = N_{1\wedge 2}/N, \tag{12}$$

where $N_{1\wedge 2} = \sum_{i=1}^{N} \mathbf{1}(R_i^1 = 1, R_i^2 = 1)$. The above identity also suggests that people who are able to solve problem 3 must have both attributes and therefore are able to solve both problems 1 and 2, that is, $N_3 = N_{1\wedge 2}$. Certainly, this is not necessarily respected in the real data, though it is a logical conclusion. The slipping and guessing parameters are introduced to account for such disparities. With the additional constraint in (12) included, the corresponding T-matrix and β-vector should be

$$T_{\mathbf{c},\mathbf{g}}(Q) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \beta = \begin{pmatrix} N_1/N \\ N_2/N \\ N_3/N \\ N_{1\wedge 2}/N \end{pmatrix}. \tag{13}$$

Similarly, one may include other (linear) constraints in the T-matrix that correspond to combinations of distinct items.

Objective function and estimation of the Q-matrix. Based on the above construction and the discussions, we introduce an objective function

$$S_{\mathbf{c},\mathbf{g},\mathbf{p}}(Q) = |T_{\mathbf{c},\mathbf{g}}(Q)\mathbf{p} - \beta|, \tag{14}$$

where $|\cdot|$ is the Euclidean distance. If all the parameters are correctly specified, we expect that $S_{\mathbf{c},\mathbf{g},\mathbf{p}}(Q) \to 0$ as $N \to \infty$. A natural estimator of the Q-matrix would be

$$\hat{Q} = \arg\inf_{Q'} S_{\mathbf{c},\mathbf{g},\mathbf{p}}(Q').$$

Dealing with the unknown parameters. Most of the time, the parameters $(\mathbf{c}, \mathbf{g}, \mathbf{p})$ are unknown. Under these situations, we consider the profiled objective functions

$$S(Q') = \inf_{\mathbf{c},\mathbf{g},\mathbf{p}} S_{\mathbf{c},\mathbf{g},\mathbf{p}}(Q'), \tag{15}$$

where the minimization is subject to the natural constraints that $c_i, g_i, p_{\boldsymbol{\alpha}} \in [0,1]$ and $\sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} = 1$. Then, the corresponding estimator is

$$\hat{Q} = \arg\inf_{Q'} S(Q'). \tag{16}$$

The minimization of $\mathbf{p}$ in (15) consists of a quadratic optimization with linear constraints, and therefore can be done efficiently. The minimization with respect to $\mathbf{c}$ and $\mathbf{g}$ is usually not straightforward. One may alternatively replace the minimization by other estimators (such as the maximum likelihood estimator) $(\hat{\mathbf{c}}(Q'), \hat{\mathbf{g}}(Q'), \hat{\mathbf{p}}(Q'))$. Thus, the objective function becomes

$$\hat{S}(Q') = S_{\hat{\mathbf{c}}(Q'),\hat{\mathbf{g}}(Q'),\hat{\mathbf{p}}(Q')}(Q'). \tag{17}$$

The corresponding estimator is

$$\tilde{Q} = \arg\inf_{Q'} \hat{S}(Q'). \tag{18}$$

This alternative allows certain flexibility in the estimation procedure. The S-function in (17) is usually easier to compute. Therefore, we often work with the estimator (18) and a hill-climbing algorithm to compute $\tilde{Q}$ is given in the next subsection.

Remark 1 *Conceptually, we may include all the combinations $(j_1, ..., j_l)$ for $l = 1, ..., J$ in the T-matrix, which results in a T-matrix of $2^J - 1$ rows. We call such a T-matrix* saturated. *The corresponding vector β contains all the information of the observed responses. However, from a*

practical point of view, in order to ensure a convergence of (7) *when the* T*-matrix is saturated, it is necessary to have sample size* $N >> 2^J$*. That is, the sample size needs to be sufficiently large so that the count in each cell of the* J*-way contingency table is non-zero. Unfortunately, such a large sample is usually not achievable even for a reasonable number of items, e.g.,* $J = 20$*. Furthermore, to construct a matrix of* 2^J *row typically induces a substantial computational overhead.*

With this concern, we typically do not include all combinations of items in the T*-matrix. A practical suggestion is to include, in an ascending order,* 1*-way,* 2*-way combinations,... until the number of rows of the* T*-matrix reaches* $N/10$*. Generally speaking we include combinations of fewer items first then include those of more items. In the simulation study presented later, a* T*-matrix including at least up to* $(K+1)$*-way combinations performs well empirically. For the corresponding theoretically analysis, see Liu et al. (2011).*

2.3 Computations

In this subsection, we consider the computation of the estimator. In particular, we consider the estimator in (18) and the objective function (17). Let $(\hat{\mathbf{c}}, \hat{\mathbf{g}}, \hat{\mathbf{p}})$ be the maximum likelihood estimator (MLE). The computation of the MLE's can be done efficiently by the EM algorithm (Dempster, Laird, and Rubin (1977); de la Torre (2009)). Furthermore, we consider the optimization of (16) and (18).

The optimization of a general nonlinear discrete function is a very challenging problem. A simple-minded search of the entire space consists of evaluating the function S up to $2^{J\times K}$ times. In our current setting, an *a priori* Q-matrix, denoted by Q_0, is usually available. We expect that Q_0 is reasonably close to the true matrix Q.

For each Q', let $U_j(Q')$ be the set of $J \times K$ matrices that are identical to Q' except for the j-th row (item). Then our algorithm is described as follows.

Algorithm 1 *Choose a starting point* $Q(0) = Q_0$*. For each iteration* m*, given the matrix from the previous iteration* $Q(m-1)$*, we perform the following steps.*

1. *Let*

$$Q_j = \arg \inf_{Q' \in U_j(Q(m-1))} S(Q'). \tag{19}$$

2. *Let* $j_* = \arg\inf_j S(Q_j)$.

3. *Let* $Q(m) = Q_{j*}$.

Repeat steps 1-3 until $Q(m) = Q(m-1)$.

At each step m, the algorithm considers updating one of the J items. In particular, if the j-th item is updated, the Q-matrix for the next iteration would be Q_j. Then, $Q(m)$ is set to be the Q_{j_*} that admits the smallest objective function among all the Q_j's. The optimization (19) consists of evaluating the function S 2^K times. Thus, the total computation complexity of each iteration is $J \times 2^K$ evaluations of S.

Remark 2 *The simulation study in Section 3 shows that if* Q_0 *is different from* Q *by 3 items (out of 20 items) Algorithm 1 has a very high chance of recovering the true matrix with reasonably large samples.*

3 Simulation

In this section, we conduct simulation studies to illustrate the performance of the proposed method. We generate the data from the DINA model under different settings and compare the estimated Q-matrix and the true Q-matrix.

3.1 Estimation of the Q-matrix with no special structure

The simulation setting. To start with, we consider a 20×3 Q-matrix ($J = 20$ items and $K = 3$ attributes), denoted by Q_1, given by

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{20}$$

We further generate the attributes from a uniform distribution, i.e.,

$$p_{\boldsymbol{\alpha}} = 2^{-K}.$$

The slipping parameters and the guessing parameters are set to be $s_i = g_i = 0.2$ for all items. In addition, for each sample size $N = 500$, 1000, 2000, and 4000, one hundred data sets were generated under such a setting.

To reduce the computational complexity, the T-matrix contains combinations of up to four items. More generally, the simulation study shows that a T-matrix containing all the $(K+1)$ (and lower) combinations delivers good estimates. We implement Algorithm 1 with a starting Q-matrix Q_0 specified as follows. The Q_0 is constructed based on the true matrix Q by misspecifying three items. In particular, we randomly selected 3 items out of the total 20 items without replacement. For each of the selected items, the corresponding row of Q_0 is sampled uniformly from all the possible K dimensional binary vectors excluding the true vector (of Q) and the zero vector. That is, each of these rows is a uniform sample of $2^K - 2$ vectors. Thus, it is guaranteed that Q_0 does not have zero-vectors and is different form Q by precisely three items. The simulation results are given by the first row of Table 1. The columns "$\hat{Q} = Q$" and "$\hat{Q} \neq Q$" contains the frequencies of

the events "$\hat{Q} = Q$" and "$\hat{Q} \neq Q$" respectively. Based on 100 independent simulations, $\hat{Q}$ recovers the true Q-matrix 98 times when the sample size is 500. For larger samples, the estimate $\hat{Q}$ never misses the true Q-matrix.

Insert Table 1 about here

We further simulate the data from Q-matrices with 4 and 5 attributes

$$Q_2 = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\\ 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\\ 1&1&0&0\\ 1&0&1&0\\ 1&0&0&1\\ 0&1&1&0\\ 0&1&0&1\\ 0&0&1&1\\ 1&1&1&0\\ 1&1&0&1\\ 1&0&1&1\\ 0&1&1&1\\ 1&1&1&1\\ 1&1&1&1 \end{pmatrix}, \quad Q_3 = \begin{pmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 1&1&0&0&0\\ 1&0&1&0&0\\ 1&0&0&1&0\\ 1&0&0&0&1\\ 0&1&1&0&0\\ 0&1&0&1&0\\ 0&1&0&0&1\\ 0&0&1&1&0\\ 0&0&1&0&1\\ 0&0&0&1&1 \end{pmatrix}. \tag{21}$$

With exactly the same settings, the results are given by the corresponding rows in the Table 1. The estimator performs well except for the cases where $N = 500$. This is mainly because the sample size is small relative to the dimension K.

An improved estimation procedure for small samples. We further investigate the data sets generated according to Q_2 and Q_3 with $N = 500$ when the estimator $\hat{Q}$ did not perform as well as other situations. In particular, we look into the cases when $\hat{Q} \neq Q$. We observe that Q-matrices with more misspecified entries do not necessarily admit larger S values. In some cases, Q does not minimize the objective function S; nonetheless, $S(Q)$ is not much larger than the global minimum $\inf_{Q'} S(Q')$. Figures 1 and 2 show two typical cases. For each of the two figures, two plots are provided. The x-axis shows the number of iterations of the optimization algorithm. The y-axis of

the left plot shows the number of misspecified entries of $Q(m)$ at iteration m; the plot on the right shows the objective function $S(Q(m))$. For the case shown in Figure 1, the algorithm just misses the true Q-matrix by one entry; for the case in Figure 2, the algorithm in fact passes the true Q-matrix and move to another one. Both cases show that the true Q-matrix does not minimize the objective function S. In fact, the values of the S function have basically dropped to a very low level after three iterations. The algorithm tends to correct one misspecified item at each of the first 2 iterations. After iteration 3, the reduction of the S function is marginal, and there are several Q-matrices that fits the data approximately equally well. For such situations where there are several matrices whose S values are close to the global minimum, we recommend careful investigation of all those matrices and selection of the most sensible one from a practical point of view.

Insert Figure 1 about here

Insert Figure 2 about here

Motivated by this, we consider a modified algorithm with an early stopping rule, i.e., we stop the algorithm when the reduction of the S-function value is below some threshold. In particular, we choose a threshold value of 4.5% of $S(Q_{m-1})$ at the m-th iteration. With this early stopping rule, the estimator for Q_2 and Q_3 can be improved substantially. The results based on the *same* samples as in Table 1 are shown in Table 2 which is show much high frequency of recovering the true Q-matrix.

Insert Table 2 about here

When attribute profile $\boldsymbol{\alpha}$ follows a non-uniform distribution. We consider the situation where the attribute profile $\boldsymbol{\alpha}$ follows a non-uniform distribution. We adopt a similar setting as in Chiu et al. (2009), where attributes are correlated and unequal prevalence. We assume a multivariate probit model. In particular, for each subject, let $\theta = (\theta_1, \cdots, \theta_k)$ be the underlying ability following a multivariate normal distribution $MVN(0, \Sigma)$, where the covariance matrix Σ has unit variance and common correlation ρ taking values of $0.05, 0.15$ and 0.25. Then the attribute profile

$\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)$ is determined by

$$\alpha_k = \begin{cases} 1 & \text{if } \theta_k \geq \Phi^{-1}\left(\frac{k}{K+1}\right) \\ 0 & \text{otherwise.} \end{cases}$$

The other settings are similar as the previous simulations. We consider the true Q-matrix given as in (20) and $K = 3$. The slipping and guessing parameters are set to be 0.2. 100 independent datasets are generated. Table 3 shows the frequency of Q_1 being recovered by the estimator (after applying the early stopping method introduced in the above subsection). We can find that the more correlated the attributes are, the more difficult it is to estimate a Q-matrix. This is mostly because the samples are unevenly distributed over the 2^K possible attribute profiles and thus the "effective sample size" becomes smaller.

Insert Table 3 about here

3.2 Estimation of the Q-matrix with partial information

In this subsection, we consider the situation where partial knowledge is available for the Q-matrix. We consider one of the situations discussed in Section 4. Consider a $J \times K$ Q-matrix where, among the total J items, the attribute requirements of $J-1$ items are known. Of interest is learning the J-th item. In this simulation we let $J = 2K + 1$. The first $2K$ rows of Q are known to form two complete matrices, i.e.,

$$Q = \begin{pmatrix} I_K \\ I_K \\ V_J \end{pmatrix},$$

where I_K is the identity matrix of dimension K and V_J is the row corresponding to the J-th item to be learnt. The the corresponding estimator becomes

$$\hat{Q} = \arg \inf_{Q' \in U_J(Q)} S(Q'),$$

where $U_J(Q)$ is defined in Section 2.3, as the set of Q-matrices identical to Q for the first $J-1$ rows.

With a similar setting to the previous simulations, the slipping and guessing parameters are set to be 0.2 and the population is set to be uniform, i.e., $p_{\boldsymbol{\alpha}} = 2^{-K}$. For each combination of $K = 3$, 4, and 5, we consider different V_J's. 100 independent datasets are generated. Table 4 shows the frequency of V_J being recovered by the estimator. One empirical finding is that the more "1"'s V_J contains, the more difficult it is to estimate V_J.

Insert Table 4 about here

4 Discussions

Estimation of the Q-matrix for other DCM's. The differences among DCM's lie mostly in their ideal response structures and the distribution of the response vectors implied by the Q-matrices. The distribution of response vector $\mathbf{R}$ takes an additive form such as that in (3) if responses to different items are conditionally independent given the attribute profile $\boldsymbol{\alpha}$. With such a structure, one can construct the corresponding B-vectors that contain the corresponding conditional probabilities of the response vectors given each attribute profile $\boldsymbol{\alpha}$. Furthermore, a T-matrix is constructed by stacking all the B-vectors and an S-function is defined as the L^2 distance between the observed frequencies and those implied by the Q matrix. An estimator is then obtained by minimizing the S-function. Thus, this estimation procedure can be applied to other DCM's. For instance, one immediate extension of the current estimation procedure is to the DINO model.

Incorporating available information in the estimation procedure. Sometimes partial information is available for the parameters $(Q, \mathbf{c}, \mathbf{g}, \mathbf{p})$. For instance, it is often reasonable to assume that some entries of the Q-matrix are known. Suppose we can separate the attributes into "hard" and "soft" ones. By "hard", we mean those that are concrete and easily recognizable in a given problem and, by "soft", we mean those that are subtle and not obvious. We can then assume that entries in columns which correspond to "hard" attributes are known. Alternatively, there may be a subset of items whose attribute requirements are known, while the item-attribute relationships of all other items need to be learnt, for example, when new items need to be calibrated according to the existing ones. Furthermore, even if an estimated Q-matrix may not be an appropriate replacement of the *a priori* Q-matrix provided by the "expert" (such as exam makers), it can serve as

validation as well as a method of calibration using existing knowledge about the Q-matrix. When such information is available and correct, computation can be substantially reduced. This is because the optimization, for instance that in (18), can be performed subject to existing knowledge of the Q-matrix. In particular, once the attribute requirements of a subset of $J-1$ items are known, one can calibrate other items, one at a time, using those known items. More specifically, consider a $J \times K$ matrix Q^*, the first $J-1$ items of which are known. We estimate the last item by $\hat{Q} = \arg\sup_{Q' \in U_J(Q^*)} S(Q')$, i.e., we minimize the S-function subject to our knowledge about the first $J-1$ items. Note that this optimization requires 2^K evaluations of the S-function and is therefore efficient. Thus, to calibrate M items, the total computation complexity is $O(M \times 2^K)$, which is typically of a manageable order.

Information about other parameters such as $\mathbf{c}$, $\mathbf{g}$, and $\mathbf{p}$ can also be included in the estimation procedure. For instance, the attribute population is typically modeled to admit certain parametric form such as a log-linear model with certain interactions (von Davier and Yamamoto, 2004; Henson and Templin, 2005; Xu and von Davier, 2008). This type of information can be incorporated in to the definition of (15) and (17), where the minimization and estimation of $(\mathbf{c}, \mathbf{g}, \mathbf{p})$ can be subject to additional parametric form or constraints. Such addition information is helpful enhancing the identifiability of the Q-matrix.

Theoretical properties of the estimator. Under restrictive conditions, the theoretical properties of the proposed methods have been established in Liu, Xu, and Ying (2011), which assumes the following conditions hold. First, the guessing parameters for all items are known. In the definition of the objective function (17), $\hat{\mathbf{g}}$ is replaced by the true guessing parameters. Second, the true Q-matrix is complete, that is, for each attribute k there exists an item that only requires this particular attribute. Equivalently, there exist K rows in Q such that the corresponding sub-matrix is diagonal. Together with a few other technical conditions, it is shown that with probability converging to 1, $\hat{Q}$ (and $\tilde{Q}$) is the same as the true matrix Q up to a column permutation. We write two matrices $Q_1 \sim Q_2$ if they differ only by a column permutation. Permuting the columns of a Q-matrix is equivalent to relabeling the attributes. The data does not contain information about the specific meaning of the attributes. In this sense, we do not expect to distinguish two matrices Q_1 and Q_2 based on the data if $Q_1 \sim Q_2$. Therefore, results of the form $P(\hat{Q} \sim Q) \to 1$ are the

strongest type that one may expect. In addition, the binary relationship "$\sim$" is an equivalence relationship. The corresponding quotient set is the finest resolution possibly identifiable based on the data.

Under weaker conditions, such as absence of completeness in the Q-matrix or the presence of unknown guessing parameter, the identifiability of the Q-matrix may be weaker, which corresponds to a coarser quotient set. One empirical finding is that Q-matrices with more diversified items tend to be easier to identify. For instance, one simple yet surprising example of a non-identifiable Q-matrix is that

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

with slipping and guessing probabilities being 0.2 for all items and $p_{\boldsymbol{\alpha}} = 1/4$ for all $\boldsymbol{\alpha}$. This Q-matrix cannot be distinguished from

$$Q' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix},$$

that is, one can find another set of slipping, guessing probabilities and $p'_{\boldsymbol{\alpha}}$ that implies the same distribution of the response vector.

Model Validation The proposed framework is applicable to not only the estimation of the Q-matrix but also the validation of an existing Q-matrix. If the Q-matrix is correctly specified and the assumptions of the DINA model are in place, then one may expect

$$|\beta - T_{\hat{\mathbf{c}},\hat{\mathbf{g}}}(Q)\mathbf{p}| \to 0$$

in probability as $N \to \infty$. The above convergence requires no additional conditions to establish the consistency of $\hat{Q}$ and $\tilde{Q}$ (such as completeness or diversified attribute distribution). In fact, it suffices that the responses are conditionally independent given the attributes and $(\hat{\mathbf{c}}, \hat{\mathbf{g}})$ are

consistent estimators of $(\mathbf{c}, \mathbf{g})$. Then, one may expect that

$$\hat{S}(Q) \to 0.$$

If the convergence rate of the estimators $(\hat{\mathbf{c}}, \hat{\mathbf{g}})$ is known, for instance, $(\hat{\mathbf{c}} - \mathbf{c}, \hat{\mathbf{g}} - \mathbf{g}) = O_p(N^{-1/2})$, then a necessary condition for a correctly specified Q-matrix is that $S_{\hat{\mathbf{c}},\hat{\mathbf{g}}}(Q) = O_p(N^{-1/2})$. The asymptotic distribution of S depends on the specific form of $(\hat{\mathbf{c}}, \hat{\mathbf{g}})$. Consequently, checking the closeness of S to zero forms a procedure for validation of the existing knowledge of the Q-matrix and the DINA model assumption.

Sample size. As the simulation results shows that the estimator miss the true Q-matrix with non-ignorable probability (over 50%). This probability is substantially reduced (to 2%) when the sample size is increased to $N = 1000$. This suggests that a practically large sample N should be at least 30×2^K. Note that the K binary attributes partition the population into 2^K groups. In order to have the estimator yield reasonably accurate estimate there should be on average at least 30 samples in each group. In addition, performance of the estimator maybe further affected by the underlying attribute distribution. For instance, if the attributes are very correlated, the probabilities of certain attributes will be substantially smaller than others. For such cases, estimation for some rows in the Q-matrix (those corresponding to the small probability attributes) will be less accurate. For such situations, the "effective sample size" is even smaller.

Computation. The optimization of $S(Q)$ over the space of $J \times K$ binary matrices is a nontrivial problem. This is a substantial computational load if J and K are reasonably large. This computation might be reduced by splitting the Q-matrix into small sub-matrices. For typical statistical models, dividing the parameter space is usually not possible. The Q-matrix adopts a particular structure with which there is certain independence among items so that splitting the Q-matrix is valid. Similar techniques have been employed in the literature, such as Chapter 8.6 in the Tatsuoka (2009) with large scale empirical studies in that chapter. In particular, for instance if there are 100 items, one can handle such a situation as follows. First, split the 100 items into 10 groups (possibly with overlapping items between groups if necessary); then apply the estimator to each of the 20 groups of items respectively. This is equivalent to breaking a big $100 \times K$ Q-matrix into 20 smaller

matrices and estimating each of them separately. Lastly, combine the 20 estimated sub-matrices together to form a single estimate. Given that the computation for smaller scale matrices is much easier than those big ones, the splitting approach reduces the computation overhead. Nonetheless, developing a fast computation algorithm is an important line of future research.

Summary. As a concluding remark, we emphasize that learning the Q-matrix based on the data is an important problem even if *a priori* knowledge is sometimes available. In this paper, we propose an estimation procedure of the Q-matrix under the setting of the DINA model. This method can also be adapted to the DINO model that is considered as the dual model of the DINA model. Simulation study shows that the estimator performs well when the sample size is reasonably large.

References

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

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.

de la Torre, J. (2009). Dina model and parameter estimation: a didactic. *Journal of Educational and Behavioral Statistics 34*, 115–130.

de la Torre, J. and J. Douglas (2004). Higher order latent trait models for cognitive diagnosis. *Psychometrika 69*, 333–353.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via em algorithm. *Journal of the Royal Statistical Society Series B-Methodological 39*(1), 1–38.

DiBello, L., W. Stout, and L. Roussos (1995). Unified cognitive psychometric assessment likelihood-based classification techniques. *Cognitively diagnostic assessment. Hillsdale, NJ: Erlbaum*, 361–390.

Hartz, S. (2002). A bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality. *Doctoral Dissertation, University of Illinois, Urbana-Champaign*.

Henson, R. and J. Templin (2005). Hierarchical log-linear modeling of the skill joint distribution. *External Diagnostic Research Group Technical Report*.

Junker, B. and K. Sijtsma (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement 25*, 258–272.

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

Liu, J., G. Xu, and Z. Ying (2011). Theory of self-learning Q-matrix. *Bernoulli to appear*.

Macready, G. and C. Dayton (1977). The use of probabilistic models in the assessment of mastery. *Journal of Educational Statistics 2*(2), 99–120.

Roussos, L. A., J. L. Templin, and R. A. Henson (2007). Skills diagnosis using IRT-based latent class models. *Journal of Educational Measurement 44*, 293–311.

Rupp, A. (2002). Feature selection for choosing and assembling measurement models: A building-block-based organization. *Psychometrika 2*, 311–360.

Rupp, A. and J. Templin (2008a). Effects of q-matrix misspecification on parameter estimates and misclassification rates in the dina model. *Educational and Psychological Measurement 68*, 78–98.

Rupp, A. and J. Templin (2008b). Unique characteristics of diagnostic classification models: A comprehensive review of the current state-of-the-art. *Measurement: Interdisciplinary Research and Perspective 6*, 219–262.

Rupp, A., J. Templin, and R. A. Henson (2010). *Diagnostic Measurement: Theory, Methods, and Applications*. Guilford Press.

Stout, W. (2007). Skills diagnosis using IRT-based continuous latent trait models. *Journal of Educational Measurement 44*, 313–324.

Tatsuoka, C. (2002). Data-analytic methods for latent partially ordered classification models. *Applied Statistics (JRSS-C) 51*, 337–350.

Tatsuoka, K. (1983). Rule space: an approch for dealing with misconceptions based on item response theory. *Journal of Educational Measurement 20*, 345–354.

Tatsuoka, K. (1985). A probabilistic model for diagnosing misconceptions in the pattern classification approach. *Journal of Educational Statistics 12*, 55–73.

Tatsuoka, K. (2009). *Cognitive assessment: an introduction to the rule space method*. CRC Press.

Templin, J. (2006). CDM: cognitive diagnosis modeling with mplus [computer software]. (available from http://jtemplin.myweb.uga.edu/cdm/cdm.html).

Templin, J. and R. Henson (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods 11*, 287–305.

van der Linden, W. (1978). Forgetting, guessing, and mastery: The macready and dayton models revisited and compared with a latent trait approach. *Journal of Educational Statistics 3*(4), 305–317.

von Davier, M. (2005). *A general diagnosis model applied to language testing data.* Educational Testing Service, Research Report.

von Davier, M. and K. Yamamoto (2004). A class of models for cognitive diagnosis. (slides available from www.von-davier.com). *the Spearman Conference*.

Xu, X. and M. von Davier (2008). Fitting the structured general diagnostic model to NAEP data (rp-08-27). *Princeton, NJ: Educational Testing Service*.

	N=500		N=1000		N=2000		N=4000	
	$\hat{Q} = Q$	$\hat{Q} \neq Q$	$\hat{Q} = Q$	$\hat{Q} \neq Q$	$\hat{Q} = Q$	$\hat{Q} \neq Q$	$\hat{Q} = Q$	$\hat{Q} \neq Q$
Q_1	94	6	100	0	100	0	100	0
Q_2	82	18	100	0	100	0	100	0
Q_3	38	62	98	2	100	0	100	0

Table 1: Numbers of correctly estimated Q-matrices out of 100 simulations with $N = 500$, 1000, 2000, and 4000 for Q_1, Q_2, and Q_3.

	N=500	
	$\hat{Q} = Q$	$\hat{Q} \neq Q$
$Q = Q_2$	94	6
$Q = Q_3$	70	30

Table 2: The results of algorithm with an early stopping rule for Q_2 and Q_3 based on the same samples as in Table 1.

	N=1000		N=2000		N=4000	
	$\hat{Q} = Q_1$	$\hat{Q} \neq Q_1$	$\hat{Q} = Q_1$	$\hat{Q} \neq Q_1$	$\hat{Q} = Q_1$	$\hat{Q} \neq Q_1$
$\rho = 0.05$	78	22	98	2	100	0
$\rho = 0.15$	71	29	94	6	99	1
$\rho = 0.25$	41	59	76	24	95	5

Table 3: Numbers of correctly estimated Q_1 out of 100 simulations with $N = 500$, 1000, 2000, and 4000 for different ρ values.

	N=250		N=500		N=1000		N=2000	
V_J	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$
(1 0 0)	91	9	98	2	100	0	100	0
(1 1 0)	82	18	97	3	99	1	100	0
(1 1 1)	70	30	83	17	100	0	100	0

	N=500		N=1000		N=2000		N=4000	
V_J	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$
(1 0 0 0)	91	9	98	2	100	0	100	0
(1 1 0 0)	84	16	94	6	100	0	100	0
(1 1 1 0)	71	29	87	13	99	1	100	0
(1 1 1 1)	39	61	62	38	94	6	100	0

	N=1000		N=2000		N=4000		N=8000	
V_J	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$	$\hat{Q}=Q$	$\hat{Q}\neq Q$
(1 0 0 0 0)	95	5	100	0	100	0	100	0
(1 1 0 0 0)	88	12	99	1	100	0	100	0
(1 1 1 0 0)	77	23	98	2	100	0	100	0
(1 1 1 1 0)	47	53	76	24	92	8	100	0
(1 1 1 1 1)*	29	71	37	63	56	44	88	12

Table 4: Numbers of correctly estimated Q-matrices out of 100 simulations with $K = 3, 4, 5$. $*$ In the case of (1 1 1 1 1), $\hat{Q}$ recovers Q 100 times when $N = 12000$.

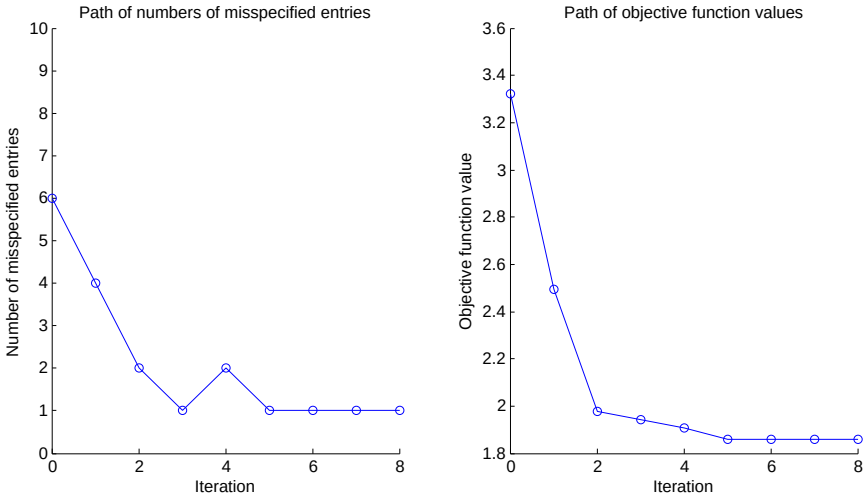

Figure 1: Results of a simulated data set with $N = 500$ and $K = 5$, for which the estimated Q-matrix does not pass the true one.

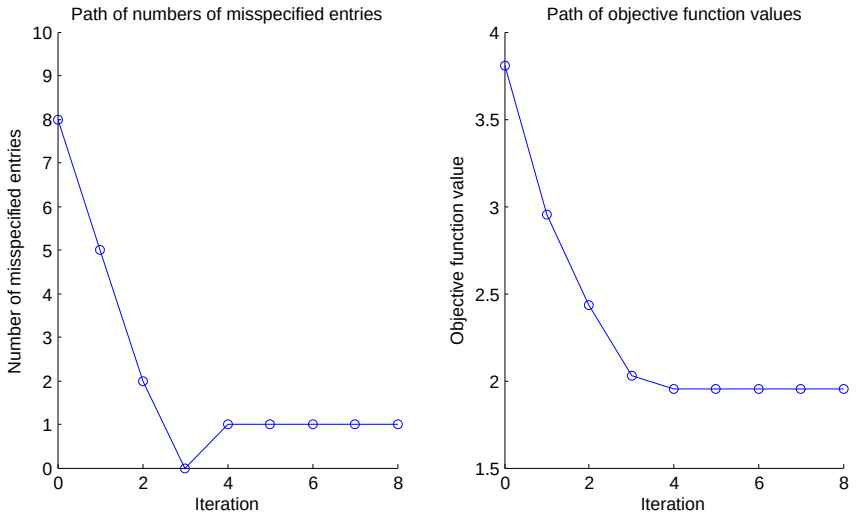

Figure 2: Results of a simulated data set with $N = 500$ and $K = 5$, for which the estimated Q-matrix passes the true one but does not converge to it.