Estimating the Q-matrix for Cognitive Diagnosis Models in a Bayesian Framework

Meng-ta Chung

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2014

Abstract

Estimating the Q-matrix for Cognitive Diagnosis Models in a Bayesian Framework

Meng-ta Chung

This research aims to develop an MCMC algorithm for estimating the Q-matrix in a Bayesian framework. A saturated multinomial model was used to estimate correlated attributes in the DINA model and rRUM. Closed-forms of posteriors for guess and slip parameters were derived for the DINA model. The random walk Metropolis-Hastings algorithm was applied to parameter estimation in the rRUM. An algorithm for reducing potential label switching was incorporated into the estimation procedure. A method for simulating data with correlated attributes for the DINA model and rRUM was offered.

Three simulation studies were conducted to evaluate the algorithm for Bayesian estimation. Twenty simulated data sets for simulation study 1 were generated from independent attributes for the DINA model and rRUM. A hundred data sets from correlated attributes were generated for the DINA and rRUM with guess and slip parameters set to 0.2 in simulation study 2. Simulation study 3 analyzed data sets simulated from the DINA model with guess and slip parameters generated from $Uniform(0.1, 0.4)$. Results from simulation studies showed that the Q-matrix recovery rate was satisfactory. Using the fraction-subtraction data, an empirical study was conducted for the DINA model and rRUM. The estimated Q-matrices from the two models were compared with the expert-designed Q-matrix.

Keywords: Bayesian, Cognitive Diagnosis models, DINA, MCMC, Q-matrix, rRUM

Contents

List of Tables

Acknowledgements

I am grateful to everyone who has helped me. I am particularly grateful to my advisor Dr. Matthew S. Johnson for his kindness, patience and guidance for all these years. I sincerely thank him. I am grateful to my committee members for their suggestions. I am grateful to Dr. James E. Corter and Dr. Bryan Keller for giving me the opportunity of being their teaching assistant.

I am grateful to my friends Rui Xiang and Huacheng Li for their friendship. I am especially grateful to my dear friend and mentor Ernesto Power for all his help, support, encouragement and Friday conversations throughout the years.

My gratefulness to my family is beyond words. I dearly thank my wife Yu-chun Huang for her understanding, companionship and love. I thank my son Ping-yuan Chung for the joy and inspiration he brings to me. I thank my father Cheng-hsin Chung and mother Su-ching Chuang for always believing in me, and for always being there for me.

Mengta

Chapter 1 Introduction

An important objective of traditional exams is to make inferences about an examinee's general ability with reference to others in the normative group (Brown & Hudson, 2002). An aggregated total score that indicates an examinee's relative position on an ability continuum should be adequate for that objective. Item response theory (IRT) models provide a single, continuous estimate of an examinee's overall ability and is sufficient for this objective. IRT models focus on item-level analysis, and they typically possess a simple loading structure in the sense that each item only loads on one dimension that is usually construed as the ability of each examinee to each item.

While achieving an ordering of examinees remains an important goal in some settings of educational measurement, modern measurement methods focus on cognitive skill diagnoses that can provide feedback on strengths and weaknesses of specific learning objectives (Henson, Stout, Douglas, He, & Roussos, 2003; Huebner, 2010). Cognitive diagnostic assessment (CDA) is a new framework that aims to evaluate whether an examinee has mastered or possessed a particular cognitive skill called *attribute* (Leighton & Gierl, 2007). As a specific example, an examinee has to know how to add and subtract in order to correctly solve for x in the item $x = 1 + 3 - 2$. Addition and subtraction are two of the attributes being measured in this item. CDA provides useful information for examinees to improve their learning. Examinees' attribute states are valuable resources for educators to give or adjust their teaching.

Q-matrix method has been applied to CDA research, (eg., von Davier, 2005; Templin,

& Henson, 2006; Chiu, Douglas, & Li, 2009; de la Torre, 2011; DeCarlo, 2012; Liu, Xu, & Ying, 2012; Wang, Chang, & Douglas, 2012; Liu, Xu, & Ying, 2013), and it is the method of focus in this research.

1.1 Theoretical Background

An attribute is a discrete latent variable. Suppose there are I examinees taking the exam that measures K attributes. A binary matrix $\boldsymbol{A}_{I\times K}$ reveals the connection between examinees and attributes. The general entry of $\boldsymbol{A}$ is α_{ik}, $\boldsymbol{A}_{I\times K} = (\alpha_{ik})_{I\times K}$,

$$\alpha_{ik} = \begin{cases} 0 & \text{if examinee } i \text{ does not master attribute } k \\ 1 & \text{if examinee } i \text{ masters attribute } k \end{cases}. \tag{1.1}$$

For example, in an exam that measures three attributes, if the second examinee possesses only the first attribute, then $\alpha_{21} = 1$ and $\alpha_{22} = \alpha_{23} = 0$. If we define for examinee i a vector $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK})$ that represents the status of mastery on each of the K attributes, then $\boldsymbol{\alpha}_2 = (1, 0, 0)$.

In order to evaluate examinees with respect to their levels of competence in each attribute, CDA partitions items into attributes by using the Q-matrix (Tatsuoka, 1983), a binary matrix showing the relationship between exam items and attributes. Given an exam with J items that measure K attributes, the Q-matrix is represented as a $J \times K$ matrix, $\boldsymbol{Q}_{J\times K}$. The general entry in the j^{th} row and k^{th} column of $\boldsymbol{Q}_{J\times K}$ is written as q_{jk}, $\boldsymbol{Q}_{J\times K} = (q_{jk})_{J\times K}$. In a Q-matrix,

$$q_{jk} = \begin{cases} 0 & \text{if attribute } k \text{ is not required by item } j \\ 1 & \text{if attribute } k \text{ is required by item } j \end{cases}. \tag{1.2}$$

An example of a Q-matrix is given in Table 1.1. Item 1 and item 2 measure *addition and subtraction* attributes respectively. Item 3 tests whether an examinee has mastered both of the attributes. In other words, $q_{11} = q_{22} = q_{31} = q_{32} = 1$ and $q_{12} = q_{21} = 0$.

Table 1.1: Example of a Q-matrix

	Attribute	
Item	addition	subtraction
1. $1 + 3$	1	0
2. $8 - 5$	0	1
3. $6 + 2 - 1$	1	1

The Q-matrix reflects the design of a CDA and is the core element that determines the quality of the diagnostic feedback (Rupp & Templin, 2008). The Q-matrix is usually constructed by area experts during exam development. Based on the Q-matrix method, a variety of cognitive diagnosis models (CDMs) have been developed, including the deterministic-input, noisy "and" gate (DINA) model (Junker & Sijtsma, 2001), noisy-inputs, deterministic "and" gate (NIDA) model (Maris, 1999), and reduced reparameterized unified model (rRUM) (Hartz, 2002). These CDMs require the Q-matrix that specifies which latent attributes are measured and how they are interrelated. Each of these CDMs has an item response function (IRF) that predicts the probability of the correct response for each item, given the attribute status of an examinee on each attribute. The above three models are *conjunctive*, in the sense that a correct item response results from possessing all the required attributes. From the Q-matrix in table 1.1, a conjunctive model assumes that an examinee has to master both addition and subtraction in order to correctly answer item 3.

1.2 Motivation and Purpose of the Research

While some of the exams are written with the purpose of being CDAs, their Q-matrices are not specified during exam development and therefore have to be assigned after the fact. Even when the Q-matrix is specified during the stage of exam development, there are concerns that area experts might neglect some attributes, and that different experts might have different opinions. Therefore, it is of practical importance to develop an automated method that offers a more objective means of getting the correct attribute-to-item mapping (Desmarai, 2011). Despite the need of an automated Q-matrix discovery method, related research is still limited.

Using the reparameterized DINA (RDINA) and higher-order RDINA models, DeCarlo (2012) acquired the conclusion that posterior distributions from Bayesian estimation could be used to obtain information about questionable Q-matrix entries and suggested using the Bayesian method in more extensive simulations. Extending DeCarlo (2012), this research uses the Bayesian concept in a different procedure that estimates the whole Q-matrix. More specifically, DeCarlo (2012) treated some Q-matrix entries as Bernoulli variables and assumed the rest of the Q-matrix entries fixed, whereas this research uses a saturated multinomial model to estimate the entire Q-matrix. In addition, this research incorporates an algorithm, based on Erosheva and Curtis (2012), into the estimation procedure for reducing potential label switching.

Although in many cases attributes are independent, in other cases attributes might be correlated. A method for correlated attributes could be helpful, and it would be expected that a method working for correlated attributes is supposed to work for independent attributes. Some methods have been developed to account for correlated attributes. de la Torre and Douglas (2004) proposed the higher-order DINA model that includes an IRT model for the joint distribution of the attributes. The higher-order DINA model assumes that the cognitive attributes are dependent on one or some general abilities. Hartz (2002) assumed the distribution of attributes is multivariate normal with zero mean and estimated an unconstrained examinee attribute correlation matrix. Using a saturated multinomial model, this research develops an MCMC algorithm for estimating correlated attributes.

For the objective to be achieved, customized sampling algorithms are formulated, and they are implemented in base R (R development core team, 2013). Offering an R package is one of the goals in this research. Simulation studies using the DINA model and rRUM are conducted to evaluate the effectiveness of the algorithms. As preliminary investigations, simulation studies in this research use a complete Q-matrix that can identify all attribute patterns. The fraction subtraction data (Tatsuoka, 1990) is used in the empirical study. It should be noted that this research is not entirely exploratory, as the number of attributes is assumed to be known in the estimation.

1.3 Outline

The rest of the thesis is structured as follows. Chapter 2 reviews the theoretical foundations for this research. First, the DINA, NIDA models and the RUM are introduced. Section 2.2 presents existing Q-matrix studies, which are categorized as confirmatory or exploratory. Section 2.3 describes Bayesian computation using conjugate priors, inverse transform sampling, and Markov chain Monte Carlo algorithms. Some common issues in Q-matrix research are also provided in the last section of Chapter 2.

Chapter 3 explains the algorithms for estimating the Q-matrix. Sections 3.1 and 3.2 formulate the algorithms based on the DINA model and the rRUM, respectively. A possible solution for label switching and a measure of accuracy rate are outlined in section 3.3. A method for simulating data from correlated attributes is introduced in section 3.3, and the background of the fraction subtraction data to be used in empirical study is described in section 3.5. Chapter 4 presents the results of the estimated Q-matrices from the DINA model and rRUM. The results are evaluated by log-likelihood and the the accuracy rate defined in section 3.4.

Chapter 2 Review of Literature

2.1 Cognitive Diagnosis Models

Several extensive reviews of CDMs have appeared in the literature, including Junker (1999), Hartz (2002), DiBello, Roussos, and Stout (2007), Roussos, Templin, and Henson (2007), Fu and Li (2007) and Rupp and Templin (2008b). This section focuses on the DINA, NIDA models and the RUM, as they are the bases of this research. These three models are *conjunctive*, for they assume that solving an item requires the conjunction of each required attribute. A correct item response is produced when all attributes required by the item are mastered, and the mastery of an attribute cannot make up for the nonmastery of another attribute.

The DINA Model

In the DINA model, an examinee is viewed as either having or not having a particular attribute. Whether examinee i possesses attribute k is typically denoted as α_{ik}, a dichotomous latent response variable with values of 0 or 1 indicating absence or presence of a skill, respectively. The DINA model is conjunctive. That is, in order to correctly answer item j, examinee i must possesses all the required attributes. Whether examinee i is able to correctly answer item j is defined by another latent response variable η_{ij},

$$\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}}. \tag{2.1}$$

The latent response variable η_{ij} is related to observed item performance X_{ij} according to the guess parameter,

$$g_j = P(X_{ij} = 1|\eta_{ij} = 0), \tag{2.2}$$

and the slip parameter,

$$s_j = P(X_{ij} = 0|\eta_{ij} = 1). \tag{2.3}$$

In other words, g_j represents the probability of $X_{ij} = 1$ when at least one required attribute is lacking, and s_j denotes the probability of $X_{ij} = 0$ when all required attributes are present. $1 - s_j$ indicates the probability of a correct response for an examinee classified as having all required skills.The item response function (IRF) for item j is

$$P(X_{ij} = 1|\boldsymbol{\alpha}, s, g) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}}, \tag{2.4}$$

and, assuming local independence and independence among examinees, the joint likelihood function for all responses is

$$P(X_{ij} = x_{ij}, \forall\, i, j|\boldsymbol{\alpha}, s, g) = \prod_{i=1}^{I}\prod_{j=1}^{J}\Big((1 - s_j)^{x_{ij}} s_j^{1-x_{ij}}\Big)^{\eta_{ij}} \Big(g_j^{x_{ij}}(1 - g_j)^{1-x_{ij}}\Big)^{1-\eta_{ij}}. \tag{2.5}$$

The DINA model is one of the most parsimonious CDMs and is easy to interpret (de la Torre, 2008). However, one limitation is that the model might be too simple because it partitions the examinees into only two equivalence classes per item and missing one attribute is equivalent to missing all required attributes (Henson & Douglas, 2005). It is reasonable that an examinee lacking only one of the required attributes have a higher probability of a correct response than those lacking all of the required attributes. It should be also noted that the monotonicity constraint, $1 - s_j > g_j$, should be placed in order to enhance the interpretability of the model (Junker & Sijtsma, 2001).

The NIDA Model

The NIDA model is probably the simplest of the conjunctive models (Roussos, Templin, & Hensen, 2007). The NIDA model is also defined by slip and guess parameters; however, the slip and guess occur at the attribute level. The slip and guess parameters of attribute k are denoted by s_k and g_k. The slip and guess parameters have no subscript for items. They are the same for every item for which $q_{jk} = 1$. Because the NIDA model is conjunctive, all skills involved in an item must be mastered to succeed. With a latent variable η_{ijk} indicating whether the performance of examinee i in item j is consistent with possessing attribute k, g_k and s_k are defined as

$$g_k = P(\eta_{ijk} = 1 | \alpha_{ik} = 0, q_{jk} = 1) \tag{2.6}$$

and

$$s_k = P(\eta_{ijk} = 0 | \alpha_{ik} = 1, q_{jk} = 1). \tag{2.7}$$

The IRF of the NIDA model is

$$P(X_{ij} = 1 | \boldsymbol{\alpha}, s, g) = \prod_{k=1}^{K} \left((1 - s_k)^{\alpha_{ik}} g_k^{1-\alpha_{ik}} \right)^{q_{jk}}, \tag{2.8}$$

and the joint likelihood function for all responses is

$$\begin{aligned} &P(X_{ij} = x_{ij}, \forall i, j | \boldsymbol{\alpha}, s, g) \\ &= \prod_{i=1}^{I} \prod_{j=1}^{J} \left(\prod_{k=1}^{K} \left((1 - s_k)^{\alpha_{ik}} g_k^{1-\alpha_{ik}} \right)^{q_{jk}} \right)^{x_{ij}} \left(1 - \prod_{k=1}^{K} \left((1 - s_k)^{\alpha_{ik}} g_k^{1-\alpha_{ik}} \right)^{q_{jk}} \right)^{1-x_{ij}}, \end{aligned} \tag{2.9}$$

which assumes local independence and independence among examinees.

The NIDA model assumes that the probability of correct application of an attribute is the same for all items. This is a severe restriction, especially if the model is to evaluate the effective of the items (Roussos, Templin, & Hensen, 2007). The monotonicity constraint, $1 - s_k > g_k$, should also be placed in order to enhance the the interpretability of the model.

The RUM

Extending the NIDA model, Maris (1999) proposed a model that estimates the slip and guess parameters for different items. That is, the the slip and guess parameters have subscripts for both items and attributes. To improve the fit of the model to the data, Dibello, Stout, and Roussos (1995) suggested the unified model that incorporates a unidimensional ability parameter. However, these two models are not statistically identifiable. Hartz (2002) reparameterized the unified model so that the parameters of the model can be identified while retaining their interpretability. The RUM is one of the more complicated conjunctive CDM (Roussos, Templin, & Hensen, 2007). The RUM defines the probability of a correct response to an item as

$$\pi_j^* = \prod_{k=1}^{K} (1 - s_{jk})^{q_{jk}}, \tag{2.10}$$

and the penalty for each attribute no possessed as

$$r_{jk}^* = g_{jk}/1 - s_{jk}. \tag{2.11}$$

π_j^* is the probability that an examinee, having acquired all the attributes required for item j, will correctly apply these attributes in solving the item. Under this view, π^* is interpreted as an item difficulty parameter, and r_{jk}^* can be seen as an indicator of the diagnostic capacity of item j for attribute k. Also from the perspective of monotonicity, $1 - s_{jk}$ should be greater than g_{jk}. Specifically, r_{jk}^* should be constrained to the interval $(0, 1)$.

The RUM allows for the possibility that not all required attributes have been explicitly specified in the Q-matrix by incorporating a general ability measure, $P_{c_j}(\theta_i)$. In the RUM, the probability of a correct response can be written as

$$P(X_{ij} = 1|\boldsymbol{\alpha}, r^*, \pi^*, \theta) = \pi_j^* \prod_{k=1}^{K} (r_{jk}^{*(1-\alpha_{ik})})^{q_{jk}} P_{c_j}(\theta_i). \tag{2.12}$$

$P_{c_j}(\theta_i)$ is the item characteristic curve in the Rasch model, where c_j is the difficulty parameter and θ_i is the general measure of an examinee's knowledge not specified by the Q-matrix.

Hartz (2002) further suggested a reduced version of the RUM, which has been used in the analysis of real data (e.g., Jang, 2005; McGlohen, Chang, & Miller, 2004; Templin, Henson, Templin, & Roussos, 2004; Templin, 2004; Templin & Douglas, 2004; Henson & Templin, 2007). The reduced reparameterized unified model (rRUM) sets $P_{c_j}(\theta_i) = 1$, assuming that the Q-matrix completely specifies the attributes required by the exam items. The IRF of the rRUM is

$$P(X_{ij} = 1|\boldsymbol{\alpha}, r^*, \pi^*) = \pi_j^* \prod_{k=1}^{K} (r_{jk}^{*(1-\alpha_{ik})})^{q_{jk}}, \tag{2.13}$$

and, based on the assumptions of local independence and independence among examinees the joint likelihood function for all responses in the rRUM is

$$\begin{aligned} &P(X_{ij} = x_{ij}, \forall i, j|\boldsymbol{\alpha}, r^*, \pi^*) \\ &\qquad = \prod_{i=1}^{I}\prod_{j=1}^{J}\left(\pi_j^* \prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{x_{ij}} \left(1 - \pi_j^* \prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{1-x_{ij}}. \end{aligned} \tag{2.14}$$

The rRUM is a generalization of the NIDA model (Junker & Sijtsma, 2001). Although different than the slip and guess parameters of the NIDA model, the parameters of the rRUM retain the model identifiable and allow the probabilities of slipping and guessing to vary across items. In practice, the rRUM is used more often then the RUM (Henson et al., 2007).

Other Models

In addition to the conjunctive models, compensatory models such as the compensatory RUM (Hartz, 2002) and the NIDO model (Templin, Henson, & Douglas, 2006) have been developed, where high competence on one attribute can compensate for low competence on other attributes. A disjunctive model, the DINO model (Templin & Henson, 2006), was also advanced and mainly applied in psychological assessment. A disjunctive model is an extreme version of the compensatory model, where high competence on one attribute is sufficient to correctly answer an item. In addition, more general models have been proposed, for example, the GDM (von Davier, 2005), the LCDM (Henson, Templin, & Willse, 2009) and the G-DINA model (de la Torre, 2011). It should be noted that all the models mentioned in this review require a Q-matrix.

2.2 Q-matrix

Research on the Q-matrix can be generally categorized as exploratory or confirmatory. Exploratory approaches intend to discover the Q-matrix from the data when the whole Q-matrix is unknown. Confirmatory approaches aim to refine a specified Q-matrix in which some of the Q-matrix elements are assumed to be known. Although an entirely exploratory approach obtains no information about the number of attributes in advance, an approach given the number of attributes is still regarded as exploratory here as long as it estimates the whole Q-matrix. As such, the method developed in this research is regarded as exploratory.

Confirmatory Approaches

Rupp and Templin (2008) conducted a simulation study using the DINA model with different misspecifications of the Q-matrix. Their results showed that the slip parameters were overestimated when attributes were incorrectly omitted in the Q-matrix while guess parameters were overestimated when the attributes were unnecessarily added to the Q-matrix. The overestimation appeared only in the items for which the Q-matrix was misspecified. Practically, large values of the slip and the guess parameters suggest empirical evidence for Q-matrix misspecification. Henson and Templin (2007) applied the same concept to the RUM. From their results, a high value of r^* suggests that nonmastery of the attribute has little influence on the probability of a correct response, and a low value of π^* indicates that many examinees classified as mastering all required attributes are still missing the item.

Using the DINA model, Templin and Henson (2006) adopted a Bayesian estimation procedure that specifies some, usually less than 20, Q-matrix elements of the Q-matrix as being random rather than as fixed. Simulations indicated that the procedure could recover the true structure when some elements of Q-matrix were not known with certainty. Henson and Templin (2006) also applied the same idea to the RUM and reported that even when 20% of the Q-matrix has been misspecified, nearly complete recovery of the true Q-matrix occurred.

Using the RDINA model, DeCarlo (2012) demonstrated that posterior distributions are useful for obtaining information about elements whose inclusion in the Q-matrix is uncertain. However, unlike the result from Templin and Henson (2006), the result from DeCarlo (2012) showed that the recovery rate was not always 100% and the recovery was poor under the situation of a complete uncertainty about an attribute. A new finding in DeCarlo (2012) was that recovery rates can be adversely affected for some uncertain elements when other elements of the Q-matrix were not correctly specified.

De la Torre (2008) suggested the sequential EM-based δ-method to find a validated Q-matrix. In conjunction with the DINA model, the method used a particular fit statistic δ that minimizes the sum of the average slip and guess parameters. While other statistics might be more useful or appropriate as stated in de la Torre (2008), a simulation study indicated that an incorrect Q-matrix resulted in more bias in the parameters and shrunken δ, showing the method is promising. However, DeCarlo (2012) noted that the sequential search algorithm might not lead to the best solution.

Chiu (2013) developed the Q-matrix refinement method, a nonparametric method for identifying and correcting the misspecified entries of a Q-matrix. Using the weighted Hamming distance, the Q-matrix refinement method operates by minimizing the residual sum of squares between the observed responses and the ideal responses to an exam item. Conceptually similar to the hill climbing algorithm promoted by Barnes (2003), the method does not rely on the estimation of model parameters and makes no assumptions other than those made by the cognitive diagnosis model supposed to underlie examinees' observed item responses.

Exploratory Approaches

The primary goal of the study is to develop an algorithm for Bayesian estimation of the Q-matrix. Some researchers have been trying to find automated approaches for searching the Q-matrix that best fits the data. A number of their studies are presented below.

Self-learning Q-matrix Theory. Liu, Xu, and Ying (2012, 2013) proposed the theory of self-learning Q-matrix that involves the following. Suppose a Q-matrix has J items and K attributes. Let the attribute vector be $\boldsymbol{A} = (A_1, A_2, \ldots, A_n, \ldots, A_k)$, where $A_n \in \{0, 1\}$ indicates possessing attribute A_n or not. $T(Q)$ is a non-linear function of the Q-matrix that provides a linear relationship between the attribute distribution and the response distribution. $\hat{p}_{ij}$ is defined as the proportion of examinees who acquire neither attribute i nor attribute j.

Table 2.1: Q-matrix

	Attribute	
Item	attribute 1	attribute 2
1	1	0
2	0	1
3	1	1

From the above Q-matrix (Table 2.1), $\hat{p}_{00}$, $\hat{p}_{01}$, $\hat{p}_{10}$, $\hat{p}_{11}$ can be calculated. With $\hat{p}_{00}$, $\hat{p}_{01}$, $\hat{p}_{10}$, $\hat{p}_{11}$, the following relationships can be drawn:

$$N(\hat{p}_{10} + \hat{p}_{11}) = N_{I_1};\ N(\hat{p}_{01} + \hat{p}_{11}) = N_{I_2};\ N\hat{p}_{11} = N_{I_3}.$$

With the natural constraint that $\sum_{ij} \hat{p}_{ij} = 1$, $\hat{\boldsymbol{p}}$ solves the linear equation

$$T(Q)\hat{\boldsymbol{p}} = \alpha, \tag{2.15}$$

where

$$T(Q) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},\ \alpha = \begin{bmatrix} N_{I_1}/N \\ N_{I_2}/N \\ N_{I_3}/N \end{bmatrix},\ \hat{\boldsymbol{p}} = (\hat{p}_{10},\ \hat{p}_{01},\ \hat{p}_{11}). \tag{2.16}$$

Let $S(Q) = \inf |T(Q)\boldsymbol{p} - \alpha|$, then it can be expected that if the empirical distribution $\hat{\boldsymbol{p}}$ minimizes $S(Q)$, the Q is a correctly specified Q-matrix. Based on the self-learning Q-matrix theory, Xiang (2013) suggested a non-linear penalized method for estimating the Q-matrix. The concept is to build a non-linear transformation function T to obtain $T(\boldsymbol{Q})$ from $\boldsymbol{Q}$, and then create a penalty function from (2.15). After the function is minimized, the optimized Q-matrix is obtained.

Non-negative Matrix Factorization. Winters (2006) used the non-negative matrix factorization (NMF) to explore the structure of a Q-matrix. Suppose $\boldsymbol{W}$ is a $J \times K$ Q-matrix, $\boldsymbol{H}$ is a $K \times N$ matrix that represents the attributes acquired by each of the N examinees, and $\boldsymbol{V}$ is a $J \times N$ matrix that denotes the exam outcome. The NMF decomposes $\boldsymbol{V}$ as the product of $\boldsymbol{W}$ and $\boldsymbol{H}$, that is, $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$. The constraint that $\boldsymbol{W}$ and $\boldsymbol{H}$ are non-negative suggests that the K attributes are additive causes that contribute to the success of items, and implies that they only increase the probability of success. However, it should be noted that there could be more than one solution to $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$. Desmarais (2011) showed that the NMF method obtained desirable results for a Q-matrix in which each item contains only one attribute. Nevertheless, more investigation is needed, as it is not uncommon for an item to measure more than one attribute.

Hill-climbing Algorithm. Barnes (2003) used a hill-climbing algorithm that generates a matrix representing the relationship between attributes and items directly from examinees' responses. The algorithm begins by setting the number of attributes K to one, and then creates a random Q-matrix with values ranging from zero to one. Examinees' responses are clustered according to attribute patterns, and the total number of errors associated with assigning responses to attribute patterns is computed. If the total number of errors from this Q-matrix is minimized, the change is saved.

This process is repeated for all values in the Q-matrix entries until the number of errors in the Q-matrix is not reduced. After a Q-matrix is obtained in this fashion, the algorithm is run again with a new random starting point several times. The Q-matrix with the fewest number of errors is then saved. This algorithm is repeated for increasing values of K to determine the best number of attributes to use in the Q-matrix. The final Q-matrix is selected when adding an additional attribute does not significantly decrease the overall errors. The number of attributes should be less than the number of items.

2.3 Bayesian Computation

Bayesian inferences are based on the posterior distribution, which summarizes our knowledge of the parameter given the data actually observed (Gelman, Carlin, Stern, & Rubin, 2004). Consider a general problem of inferring a distribution for a parameter θ given the observed data $\boldsymbol{y}$. The prior distribution $p(\theta)$ is our information about the uncertain parameter before the data are seen. The information about the parameter in the observed data is contained in the likelihood function $p(\boldsymbol{y}|\theta)$. With the prior $p(\theta)$ and the likelihood $p(\boldsymbol{y}|\theta)$, we can obtain the posterior distribution $p(\theta|\boldsymbol{y})$,

$$p(\theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\theta)p(\theta)}{\int p(\boldsymbol{y}|\theta)p(\theta)d\theta}. \tag{2.17}$$

From (2.17), a closed form for the integral in the denominator only exists for some particular conditions such as conjugate distributions. If the posterior distribution is in the same family as the prior distribution, the prior and posterior are called conjugate distributions and the prior is called a conjugate prior. The posterior distribution can be found analytically using simple updating formulas.

For other cases when conjugate priors are not available, one approach to approximate the integral is the quadrature method. When the prior is specified on a dense grid of points spanning the range of θ, the posterior can be numerically generated by summing across the discrete values. Alternatives to approximation methods are sampling methods, such as inverse transform sampling and Markov chain Monte Carlo (MCMC) algorithms including the Gibbs sampling and Metropolis-Hastings algorithm. Conjugate priors, inverse transform sampling and MCMC constitute the algorithms employed in this research, and they are delineated in the following.

Conjugacy

A conjugate prior typically simplifies computations, in that its posterior gives a closed form that can be found analytically (Gelman, Carlin, Stern, & Rubin, 2004). The generalization of the well known fact that the conjugate prior for a binomial distribution is a Beta distribution is that the conjugate prior for a multinomial distribution is a Dirichlet distribution.

In a multinomial distribution, each trial from n independent trials results in one of the k categories. Suppose $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$ represent the frequencies falling into the k categories with probabilities $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$. The probability mass function (PMF) of the multinomial distribution is

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \binom{n}{x_1 x_2 \cdots x_k} \prod_{i=1}^{k} \theta_i^{x_i}, \tag{2.18}$$

where $n = \sum_{i=1}^{k} x_i$ and $\sum_{i=1}^{k} \theta_i = 1$. Suppose that $\boldsymbol{\theta}$ follows a Dirichlet distribution,

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{k} \Gamma(a_i) \theta_1^{a_1 - 1} \ldots \theta_k^{a_k - 1}. \tag{2.19}$$

We can see that the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ is also a Dirichlet distribution, as

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \propto \prod_{i=1}^{k} \theta_i^{a_i - 1} \prod_{i=1}^{k} \theta_i^{x_i} = \prod_{i=1}^{k} \theta_i^{a_i + x_i - 1}. \tag{2.20}$$

A Dirichlet distribution can be constructed using Gamma distributions. Let $w_i \sim Gamma(a_i, 1)$ be independent for $i = 1, 2, \ldots, k$ and set $\theta_i = w_i/\tau$, where $\tau = w_1 +$

$w_2 + \cdots + w_k$. The probability density function (PDF) for $\boldsymbol{w} = (w_1, w_2, \ldots, w_k)$ is

$$p(\boldsymbol{w}) = \prod_{i=1}^{k} \frac{1}{\Gamma(a_i)} w_i^{a_i - 1} e^{-w_i}, \tag{2.21}$$

where $\Gamma(\cdot)$ is a Gamma function,

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt. \tag{2.22}$$

Since $\sum_{i=1}^{k} \theta_i = 1$, $\boldsymbol{\theta}$ has only $k-1$ dimensions. The PDF of $\boldsymbol{\theta}$ can be derived by a multivariate change of variable from $(w_1, w_2, \ldots, w_k)$ to $(\theta_1, \theta_2, \ldots, \theta_{k-1}, \tau)$. The Jacobian of this transformation is derived as $J = \tau^{k-1}$. Thus it can be obtained that

$$p(\theta_1, \theta_2, \ldots, \theta_{k-1}, \tau) = \tau^{k-1} \prod_{i=1}^{k} \frac{(\theta_i \tau)^{a_i - 1}}{\Gamma(a_i)} e^{-\theta_i \tau} = \tau^{(a_1 + a_2 + \cdots + a_k) - 1} e^{-\tau} \prod_{i=1}^{k} \frac{\theta_i^{a_i - 1}}{\Gamma(a_i)}. \tag{2.23}$$

Integrating (2.23) with respect to τ gives

$$p(\theta_1, \theta_2, \ldots, \theta_{k-1}) = \prod_{i=1}^{k} \frac{\theta_i^{a_i - 1}}{\Gamma(a_i)} \int_0^{\infty} \tau^{(a_1 + a_2 + \cdots + a_k) - 1} e^{-\tau} d\tau = \frac{1}{B(\boldsymbol{a})} \prod_{i=1}^{k} \theta_i^{a_i - 1}, \tag{2.24}$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)$. As $\sum_{i=1}^{k} \theta_i = 1$, θ_k is determined by $\theta_1, \theta_2, \ldots, \theta_{k-1}$. Therefore $p(\theta_1, \theta_2, \ldots, \theta_{k-1}) = p(\theta_1, \theta_2, \ldots, \theta_k) = p(\boldsymbol{\theta})$,

$$p(\boldsymbol{\theta}) = \frac{1}{B(\boldsymbol{a})} \prod_{i=1}^{k} \theta_i^{a_i - 1}, \tag{2.25}$$

which is $Dirichlet(\boldsymbol{a})$. This shows that $\boldsymbol{\theta}$ follows the Dirichlet distribution with the parameter $\boldsymbol{a}$. When $k = 2$, a Dirichlet distribution simplifies to a Beta distribution and therefore the conjugate prior for a binomial distribution is a Beta distribution.

Inverse Transform Sampling

Under the condition of sampling from a univariate distribution, if the posterior is a distribution to which conjugacy is applicable, the posterior can be sampled directly. However, conjugacy is not realistic, but if F^{-1} of the univariate distribution can be derived analytically, then the inverse transform sampling is an alternative. To generate random numbers from a probability distribution, we can sample random numbers from $\mathit{Uniform}(0,1)$ and then transform these values by the inverse of its cumulative distribution function (CDF) (Ross, 2013).

Let F be the CDF of a continuous random variable X, and U be a random variable distributed as $\mathit{Uniform}(0,1)$. Since

$$P(X \le x) = P(F^{-1}(U) \le x) = P(U \le F(x)) = P(U \le u), \tag{2.26}$$

X can be generated from the CDF F by first generating a random number from U and then setting $X = F^{-1}(U)$.

Inverse transform sampling can also be applied to *any* discrete variables. Suppose a discrete random variable X has PMF,

$$P(X = x_j) = p_j,\ j = 1, \ldots, \sum_j p_j = 1, \tag{2.27}$$

and U is distributed as $\mathit{Uniform}(0,1)$. As $P(a \le U < b) = b - a$ when $0 < a < b < 1$, thus

$$P(X = x_j) = P\left(\sum_{i=1}^{j-1} p_i \le U < \sum_{i=1}^{j} p_i\right) = p_j. \tag{2.28}$$

Therefore, sampling from a discrete distribution can be accomplished by first generating a random number U from $Uniform(0, 1)$ and then setting

$$X = \begin{cases} x_1 & \text{if} \quad U < p_1 \\ x_2 & \text{if} \quad p_1 \leq U < p_1 + p_2 \\ \vdots & \\ x_j & \text{if} \quad \sum_{i=1}^{j-1} p_i \leq U < \sum_{i=1}^{j} p_i \\ \vdots & \end{cases} . \tag{2.29}$$

Gibbs Sampling

The inverse transform sampling does not work with multivariate distributions. A remedy for multivariate distributions is made by Gibbs sampling. Gibbs sampling (Geman & Geman, 1984) is an MCMC algorithm for obtaining a sequence of random samples from a multivariate probability distribution. Suppose the joint distribution $p(\boldsymbol{\theta})$ is the posterior distribution we want to sample from, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$. Given a starting position $\boldsymbol{\theta}^0 = (\theta_1^0, \theta_2^0, \ldots, \theta_k^0)$, Gibbs sampling performs the following steps at iteration t:

1. sample θ_1^t from $p(\theta_1 | \theta_2^{t-1}, \ldots, \theta_{k-1}^{t-1}, \ldots, \theta_k^{t-1}, \boldsymbol{y})$

2. sample θ_2^t from $p(\theta_2 | \theta_1^t, \theta_3^{t-1}, \ldots, \theta_k^{t-1}, \boldsymbol{y})$

   $\vdots$

3. sample θ_k^t from Gibbs sampling can sample from the joint posterior distribution if the full conditional distribution $p(\theta_j | \boldsymbol{\theta}_{\setminus j}, \boldsymbol{y})$ of each parameter is known, where $\boldsymbol{\theta}_{\setminus j}$ denotes the vector $\boldsymbol{\theta}$ excluding θ_j. As θ_j is conditional on all the other parameters and the data, the full conditional distribution is

$$p(\theta_j | \boldsymbol{\theta}_{\setminus j}, \boldsymbol{y}) = p(\theta_j | \theta_1^t, \theta_2^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^t, \ldots, \theta_{k-1}^t, \theta_k^t, \boldsymbol{y}), \; j = 1, 2, \ldots, k. \tag{2.30}$$

Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm (Metropolis & Ulam, 1949; Hastings, 1970) works for both univariate and multivariate distributions. Suppose there are k parameters, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$. Let $q(\boldsymbol{\theta}^{(*)}, \boldsymbol{\theta})$ be the candidate density when the chain is at $\boldsymbol{\theta}$, and let $p(\boldsymbol{\theta}|\boldsymbol{y})$ be the conditional posterior density. The reversibility condition will be $p(\boldsymbol{\theta}|\boldsymbol{y})q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(*)}) = p(\boldsymbol{\theta}^{(*)}|\boldsymbol{y})q(\boldsymbol{\theta}^{(*)}, \boldsymbol{\theta})$ for all states. There will be some $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{(*)}$ where the reversibility condition does not hold. The balance can be restored by introducing the probability of moving which is given by

$$\varphi = \min\left(1, \frac{p(\boldsymbol{\theta}^{(*)}|\boldsymbol{y})q(\boldsymbol{\theta}^{(*)}, \boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{y})q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(*)})}\right). \tag{2.31}$$

A special case when $q(\boldsymbol{\theta}^{(*)}|\boldsymbol{\theta}) = q(|\boldsymbol{\theta}^{(*)} - \boldsymbol{\theta}|)$ is known as a random walk, which is a local exploration of the neighborhood of the current value of the Markov chain. The concept is to select candidate point $\boldsymbol{\theta}^{(*)}$ at iteration t according to $\boldsymbol{\theta}^{(*)} = \boldsymbol{\theta}^{(t-1)} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a random perturbation with a distribution independent of $\boldsymbol{\theta}^{(t-1)}$, such as $Uniform(-\delta, \delta)$ or $Normal(0, \sigma^2)$. That is, $\boldsymbol{\theta}^{(*)}$ can be sampled from $Uniform(\boldsymbol{\theta}^{(t-1)} - \delta, \boldsymbol{\theta}^{(t-1)} + \delta)$ or $Normal(\boldsymbol{\theta}^{(t-1)}, \sigma^2)$.

Although not necessarily the most efficient solution, random walk Metropolis-Hastings is often regarded as a generic algorithm that caters in most cases and a natural approach for the construction of a Metropolis-Hastings algorithm (Robert & Casella, 2004). For a random walk, the candidate is drawn from a symmetric distribution centered at the current value. The candidate density is given by $q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(*)}) = q(\theta_1^{(*)} - \theta_1, \ldots, \theta_k^{(*)} - \theta_k)$, where $q(, \ldots,)$ is symmetric about 0 for each of its arguments. So the candidate density can be written as $q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(*)}) = q(\boldsymbol{\theta}^{(*)} - \boldsymbol{\theta})$. Therefore, the acceptance probability is

$$\varphi = \min\left(1, \frac{p(\boldsymbol{\theta}^{(*)}|\boldsymbol{y})q(\boldsymbol{\theta}^{(*)}, \boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{y})q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(*)})}\right) = \min\left(1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}^{(*)})p(\boldsymbol{\theta}^{(*)})}{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}\right). \tag{2.32}$$

The Metropolis-Hastings within Gibbs sampling algorithm (Tierney, 1994) is a hybrid algorithm that combines the Metropolis-Hastings algorithm and Gibbs sampling. This algorithm retains the idea of sequential sampling but uses a Metropolis-Hastings step on some or all variables rather than attempting to sample from the exact conditional distribution. That is to say, each step in a cycle of the Gibbs sampling is itself a Metropolis-Hastings step. In each step of the Metropolis within Gibbs algorithm, a candidate value $\theta_j^{(*)}$ of the j^{th} component of $\boldsymbol{\theta}$ is proposed by $q(\theta_j^{(*)}|\boldsymbol{\theta})$ and updated with probability

$$\varphi = \min\left(1, \frac{p(\boldsymbol{y}|\theta_j^{(*)}, \boldsymbol{\theta}_{\setminus j})p(\theta_j^{(*)}, \boldsymbol{\theta}_{\setminus j})q(\theta_j|\theta_j^{(*)}, \boldsymbol{\theta}_{\setminus j})}{p(\boldsymbol{y}|\theta_j, \boldsymbol{\theta}_{\setminus j})p(\theta_j, \boldsymbol{\theta}_{\setminus j})q(\theta_j^{(*)}|\theta_j, \boldsymbol{\theta}_{\setminus j})}\right). \tag{2.33}$$

2.4 Discussion and Some Issues

Conjunctive CDMs assume that solving an item requires the possession of each required attribute. The DINA model and the rRUM are of focus in this study. In the DINA model, guess and slip happen at the item level. In the NIDA model, guess and slip occur at the attribute level. The severe restriction that the probability of correct application of an attribute is the same for all items hinders the practicality of the NIDA model, especially if the need is to evaluate the effective of items. A generalization of the NIDA model is the rRUM, a reduced version of the RUM that has parameters at both item and attribute levels.

A number of studies have proposed automated methods to search the Q-matrix. The hill-climbing algorithm (Barnes, 2003) was shown to perform at least as well as principal component analysis for skill clustering analysis. However, the algorithm often terminates with the estimated Q-matrix entries having values between 0 and 1, limiting the usefulness of the estimated Q-matrix (Chiu, 2013). In addition, the algorithm does not scale well to a Q-matrix that comprises 20 or more items (Desmarais, 2011). Using the weighted Hamming distance, the Q-matrix refinement method (Chiu, 2013) can be regarded as a confirmatory reformation of the hill-climbing algorithm. Although the purpose of the Q-matrix refinement method is to *refine* an existing Q-matrix, the method can also be implemented under an exploratory condition.

The NMF method requires that each item in the Q-matrix has a high level attribute, and each item only belongs to that attribute. The interpretation of the result from NMF is different from the standard interpretation of the Q-matrix for the conjunctive models.

The self-learning Q-matrix theory offered a model based method to estimate the Q-matrix from item responses. It requires a saturated T-matrix or known guessing parameters in DINA model. The method also needs complicated numerical methods, which makes it not easy to implement in real data analysis.

Some general issues raised in Q-matrix research are discussed here. First, verifying the number of attributes is an important issue in the Q-matrix estimation. Although the number of attributes is assumed to be known in the present research, the number of attributes is not predetermined for most of the exams. Even when the number of attributes is predetermined, the concept that represents each of the attribute might be different from different experts.

The second issue is that it is impossible to estimate the entire vector of attribute pattern probabilities under some Q-matrices. For example, some attribute patterns are not estimable in the DINA model with unconstrained attribute pattern distributions (Johnson, 2009). This can be illustrated by the Q-matrix in table 3.3. With 2 attributes, there are 4 possible attribute patterns, $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$. However, we can only estimate the probability for three sets of patterns, in that the ideal responses for $(0, 0)$ and $(0, 1)$ are the same, i.e., $(0, 0, 0, 0)$. It is impossible to distinguish the attribute patterns $(0, 0)$ and $(0, 1)$ from one another.

Table 2.2: Example of Estimability

Q-matrix			Attribute Patterns		Ideal Responses			
item	attribute 1	attribute 2	attribute 1	attribute 2	item 1	item 2	item 3	item 4
1	1	0	0	0	***0***	***0***	***0***	***0***
2	1	0	0	1	***0***	***0***	***0***	***0***
3	1	1	1	0	1	1	0	0
4	1	1	1	1	1	1	1	1

Estimability is equivalence in Q-matrices is the third issue. From the Q-matrix in Table 3.4, if no examinee possesses attribute pattern $(1, 0, 0)$, then changing the attribute state for attribute 2 in item 2 from 0 to 1 makes no difference in the ideal response pattern (Table 3.5). In other words, the two Q-matrices are equivalent. They are not distinguishable based on data.

Table 2.3: Example of Q-matrix Equivalence

	Attribute		
Item	attribute 1	attribute 2	attribute 3
1	0	0	1
2	1	***0***	0
3	0	1	0
4	1	1	1

Table 2.4: Ideal Response Patterns

Attribute Patterns			Ideal Responses			
attribute 1	attribute 2	attribute 3	item 1	item 2	item 3	item 4
0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	1	1	0	0	0
1	1	0	0	1	1	0
1	0	1	1	1	0	0
0	1	1	1	0	1	0
1	1	1	1	1	1	1

Chapter 3 Methodology

This research aims to develop Q-matrix estimation algorithms for the DINA model and rRUM in a Bayesian framework. An algorithm for reducing label switching is provided in section 3.3. A method for simulating data is described in section 3.4. Simulated data are generated using an artificial Q-matrix shown in table (3.1). The algorithms for the Bayesian method are used to recover the Q-matrix from simulated data sets. The setting for chapter 3 is comprised of responses from I examinees to J items that measure K attributes.

3.1 Estimating the Q-matrix for the DINA Model

The Bayesian formulation of estimating the Q-matrix for the DINA model is introduced in this section. The model and the sampling algorithm for the full condition posterior distributions are introduced here and explained in 3 steps below. The model is the following,

$$\left.\begin{array}{c} y_{ij} \sim Bernoulli(p(\boldsymbol{\alpha}_i) \\ p(\boldsymbol{\alpha}_i) = (1-s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}} \\ \boldsymbol{\alpha}_i|\boldsymbol{\theta}_i \sim Multinomial(2^K, \boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i \sim Dirichlet(a_1, a_2, \ldots, a_{2^K}) \\ \boldsymbol{q}_j|\boldsymbol{\phi}_j \sim Multinomial(2^K-1, \boldsymbol{\phi}_j) \\ g_j \sim Beta(a,b) \\ s_j \sim Beta(c,d) \end{array}\right\}, \tag{3.1}$$

for $i = 1, \ldots, I$, and $j = 1, \ldots, J$.

In model (3.1), $\boldsymbol{\alpha}_i$ indicates the 2^K possible attribute patterns with underlying probability vector $\boldsymbol{\theta}_i$ for examinee i, $\boldsymbol{q}_j$ represents the $2^K - 1$ possible Q-matrix patterns having underlying probability vector $\boldsymbol{\phi}_j$ for item j.

The full conditional posterior distributions of the parameters for the DINA model are expressed as follows:

$$p(\boldsymbol{\alpha}_i|\boldsymbol{y}, \boldsymbol{\alpha}_i, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}_i, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q})p(\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i) \tag{3.2}$$

$$p(\boldsymbol{g}|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{q}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q})p(\boldsymbol{g}) \tag{3.3}$$

$$p(\boldsymbol{s}|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{q}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q})p(\boldsymbol{s}) \tag{3.4}$$

$$p(\boldsymbol{q}_j|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q}_j) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q}_j)p(\boldsymbol{q}_j|\boldsymbol{\phi}_j) \tag{3.5}$$

Specifically, with the likelihood from (2.5), the full conditional posterior distributions are

$$\begin{aligned} p(\boldsymbol{\alpha}_j|\boldsymbol{y}, \boldsymbol{\alpha}_i, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha}_i, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q})p(\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i) \\ &= \left[\prod_{i=1}^{I}\prod_{j=1}^{J}\left((1-s_j)^{y_{ij}} s_j^{1-y_{ij}}\right)^{\eta_{ij}} \left(g_j^{y_{ij}}(1-g_j)^{1-y_{ij}}\right)^{1-\eta_{ij}}\right] \\ &\times \left[\binom{n}{y_1 y_2 \cdots y_{2^K}} \prod_{i=1}^{2^K} \theta_i^{y_i}\right]\left[\frac{1}{B(\boldsymbol{a})}\prod_{i=1}^{2^K}\theta_i^{a_i-1}\right], \end{aligned} \tag{3.6}$$

$$\begin{aligned} p(\boldsymbol{g}|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{q}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q})p(\boldsymbol{g}) \\ &= \left[\prod_{i=1}^{I}\prod_{j=1}^{J}\left((1-s_j)^{y_{ij}} s_j^{1-y_{ij}}\right)^{\eta_{ij}} \left(g_j^{y_{ij}}(1-g_j)^{1-y_{ij}}\right)^{1-\eta_{ij}}\right]\left[\frac{\boldsymbol{g}^{a-1}(1-\boldsymbol{g})^{b-1}}{B(a,b)}\right], \end{aligned} \tag{3.7}$$

$$\begin{aligned} p(\boldsymbol{s}|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{q}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q})p(\boldsymbol{s}) \\ &= \left[\prod_{i=1}^{I}\prod_{j=1}^{J}\left((1-s_j)^{y_{ij}} s_j^{1-y_{ij}}\right)^{\eta_{ij}} \left(g_j^{y_{ij}}(1-g_j)^{1-y_{ij}}\right)^{1-\eta_{ij}}\right]\left[\frac{\boldsymbol{s}^{c-1}(1-\boldsymbol{s})^{d-1}}{B(c,d)}\right], \end{aligned} \tag{3.8}$$

$$p(\boldsymbol{q}_j|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{q}_j) \propto p(\boldsymbol{y}|\boldsymbol{g}, \boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{q}_j)p(\boldsymbol{q}_j|\boldsymbol{\phi}_j) \tag{3.9}$$
$$= \left[\prod_{i=1}^{I}\prod_{i=1}^{I}\left((1-s_j)^{y_{ij}} s_j^{1-y_{ij}}\right)^{\eta_{ij}}\left(g_j^{y_{ij}}(1-g_j)^{1-y_{ij}}\right)^{1-\eta_{ij}}\right]p(\boldsymbol{q}_j|\boldsymbol{\phi}_j).$$

Whether examinee i masters all the required attributes for item j can be determined by η_{ij},

$$\eta_{ij} = \begin{cases} 0 & \text{if} \quad \boldsymbol{\alpha}_i'\boldsymbol{q}_j \neq \boldsymbol{q}_j'\boldsymbol{q}_j \\ 1 & \text{if} \quad \boldsymbol{\alpha}_i'\boldsymbol{q}_j = \boldsymbol{q}_j'\boldsymbol{q}_j \end{cases} . \tag{3.10}$$

The Q-matrix for the DINA model is estimated using the following 3-step algorithm. The steps are performed sequentially at iteration t, $t = 1, \ldots, T$.

Step 1: Updating Attributes

It is not uncommon for attributes to be correlated with one another (Hartz, 2002; de la Torre, 2004; Templin et al., 2008; Feng, Harbin, & Huebner, 2014), and the more correlated the attributes are, the more difficult it is to estimate the Q-matrix (Liu, Xu, & Ying, 2012). While attributes could be independent in many cases, we benefit from a method that assumes correlated attributes because it should work for both conditions.

In updating an examinee's attribute state, a saturated multinomial model is used that assumes no restrictions on the probabilities of the attribute patterns (see Maris, 1999). With K attributes, there are a total of 2^K possible attribute patterns for examinee i. An attribute pattern is a K-bit binary number, which can be converted to a decimal number. For example, the 5-bit binary number $(10010)_2$ can be converted to the decimal number 9 with the following conversion,

$$(b_n b_{n-1} \cdots b_0)_2 = b_n(2)^n + b_{n-1}(2)^{n-1} + \cdots + b_0(2)^0, \tag{3.11}$$

where $(b_n b_{n-1} \cdots b_0)_2$ denotes a binary number.

Let $\boldsymbol{E}_{2^K \times K} = (\varepsilon_{nk})_{2^K \times K}$ be the matrix of possible attribute patterns for examinee i. $\boldsymbol{E}$ has 2^K rows, and each row of $\boldsymbol{E}$ represents a possible attribute pattern. If an exam measures two attributes ($K = 2$), then the number of possible attribute patterns is 4 ($2^K = 4$). The matrix of possible attribute patterns $\boldsymbol{E}$ is

$$\boldsymbol{E}_{2^2 \times 2} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{3.12}$$

From (3.12), each row of $\boldsymbol{E}$ can be converted to a decimal number. After the conversion, these 2^K possible attribute patterns become a multinomial distribution. That is,

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}. \tag{3.13}$$

Assuming a Dirichlet prior, the model is

$$\boldsymbol{\alpha}_i | \boldsymbol{\theta}_i \sim Multinomial(2^K, \boldsymbol{\theta}_i), \tag{3.14}$$

$$\boldsymbol{\theta}_i \sim Dirichlet(a_1, a_2, \ldots, a_{2^K}), \tag{3.15}$$

where $\boldsymbol{\theta}_i$ is the underlying probability vector with 2^K elements.

From (3.6), we need to calculate $p(\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)$. Because the conjugate prior for a multinomial distribution is a Dirichlet distribution, $p(\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)$ is also a Dirichlet distribution. If the Dirichlet prior $Dirichlet(1, 1, \ldots, 1)$ is used, then the conditional posterior is distributed as

$$\boldsymbol{\theta}_i | \boldsymbol{\alpha}_i \sim Dirichlet(1 + y_1, 1 + y_2, \ldots, 1 + y_{2^K}), \tag{3.16}$$

where y_l, $l = 1, \ldots, 2^K$, is the number examinees possessing the l^{th} attribute pattern obtained from iteration $t - 1$. As no function in base R can be used to sample from Dirichlet distribution, Gamma distributions are used to construct the Dirichlet distribution. Suppose that $w_1, \ldots, w_{2^K}$ are distributed as $Gamma(a_1, 1), \ldots, Gamma(a_{2^K}, 1)$, and that $\tau = w_1 + \cdots + w_{2^K}$, then $(w_1/\tau, w_2/\tau, \ldots, w_{2^K}/\tau)$ is distributed as $Dirichlet(a_1, a_2, \ldots, a_{2^K})$. For each of the 2^K possible attribute patterns, calculate the total number of examinees $(y_1, y_2, \ldots, y_{2^K})$ falling into the attribute pattern, and then sample from $Gamma(1+y_1, 1) = w'_1, Gamma(1+y_2, 1) = w'_2, \ldots, Gamma(1+y_{2^K}, 1) = w'_{2^K}$. Let $\tau' = w'_1+w'_2+\cdots+w'_{2^K}$, then

$$p(\boldsymbol{\theta}_i | \boldsymbol{\alpha}_i) = (w'_1/\tau', w'_2/\tau', \ldots, w'_{2^K}/\tau'). \tag{3.17}$$

It can be seen from (3.6) that with (3.17) and the likelihood of each attribute pattern, attribute patterns for examinee i can be sampled from the full conditional posterior by using the discrete version of inverse transform sampling.

Let the posterior $(p_1, p_2, \ldots, p_{2^K})$ be the PMF of the 2^K possible attribute patterns. The CDF is computed by adding up the probabilities for the 2^K points of the distribution. To sample from this discrete distribution, partition $(0, 1)$ into 2^K subintervals $(0, p_1)$, $(p_1, p_1 + p_2), \ldots, (\sum_{k=0}^{K} p_{2^k-1}, \sum_{k=0}^{K} p_{2^k})$, and then generate a value u from $Uniform(0, 1)$. Updating the attribute state of examinee i is achieved by first checking which subinterval the value u falls into and then converting the subinterval number to the binary number using (3.11). After the algorithm is applied to each examinee, attribute states for all examinees are obtained for iteration t.

Step 2: Updating g and s Parameters

With the estimated attribute states from step 1, guess and slip parameters are updated in step 2. Because the conjugate prior for a binomial distribution is a Beta distribution, the full conditional posteriors of the guess and slip parameters are also Beta distributions if Beta priors are assumed for the two parameters. $Beta(1, 1)$, which is equal to $Uniform(0, 1)$, is chosen as the prior for both guess and slip parameters.

Closed-forms of the full conditional posteriors for guess and slip parameters are derived as follows. In the DINA model, for examinee i answering item j, *guess* occurs when $\eta_{ij} = 0$ but $y_{ij} = 1$ and *slip* happens when $\eta_{ij} = 1$ but $y_{ij} = 0$. Consequently, in estimating g_j, the total number of successes is $\sum_{i=1}^{I}(1 - \eta_{ij})y_{ij}$, and the total number of failures is $\sum_{i=1}^{I}(1-\eta_{ij})(1-y_{ij})$. As $g_j \sim Beta(1, 1)$ and $s_j \sim Beta(1, 1)$, the full conditional posterior distribution for g_j is

$$g_j | s_j, \boldsymbol{\alpha}, \boldsymbol{y}, \boldsymbol{q} \sim Beta\left(1 + \sum_{i=1}^{I}(1 - \eta_{ij})y_{ij}, 1 + \sum_{i=1}^{I}(1 - \eta_{ij})(1 - y_{ij})\right). \tag{3.18}$$

In estimating s_j, the total number of successes is $\sum_{i=1}^{I}\eta_{ij}(1 - y_{ij})$, and the total number of failures is $\sum_{i=1}^{I}\eta_{ij}y_{ij}$. Therefore, the full conditional posterior distribution for s_j is

$$s_j | g_j, \boldsymbol{\alpha}, \boldsymbol{y}, \boldsymbol{q} \sim Beta\left(1 + \sum_{i=1}^{I}\eta_{ij}(1 - y_{ij}), 1 + \sum_{i=1}^{I}\eta_{ij}y_{ij}\right). \tag{3.19}$$

The monotonicity constraint indicates that the probability of answering an item correctly is supposed to be higher for an examinee who possesses all the required attributes than for one who lacks at least one attribute, that is, $1 - s_j > g_j$. Junker and Sijtsma (2001) observed that the monotonicity did not always hold for the DINA model if no constraint was imposed. To achieve monotonicity, we can use inverse transform sampling to sample from a truncated Beta distribution. The g_j and s_j parameters are sampled from $Uniform(0, 1 - s_j)$ and $Uniform(0, 1 - g_j)$, and then inverted to Beta distributions.

Step 3: Updating the Q-matrix

With the updated $\boldsymbol{\alpha}$, $\boldsymbol{g}$ and $\boldsymbol{s}$ from steps 1 and 2, this step updates the Q-matrix. Similar to step 1, step 3 uses a saturated model for the Q-matrix. There are $2^K - 1$ possible Q-matrix patterns for item j. The Q-matrix pattern with all 0's is excluded because an item has to measure at least one attribute. After each binary pattern is converted to a decimal number by (3.11), the $2^K - 1$ possible Q-matrix patterns are distributed as a multinomial distribution. In updating the Q-matrix for item j, the model is

$$\boldsymbol{q}_j | \boldsymbol{\phi}_j \sim Multinomial(2^K - 1, \boldsymbol{\phi}_j). \tag{3.20}$$

An entry in the Q-matrix is denoted as q_{jk}. Let $p(q_{jk} = 1) = \phi_{jk}$ and $p(q_{jk} = 0) = 1 - \phi_{jk}$. The conjugate prior for a Bernoulli distribution is a Beta distribution. If $Beta(1, 1)$ is chosen as the prior, $\phi_{jk} \sim Beta(1, 1)$, then the conditional posterior for ϕ_{jk} is

$$\phi_{jk} | q_{jk} \sim Beta(1 + q_{jk}, 2 - q_{jk}). \tag{3.21}$$

The posterior mean is $2/3$ for $q_{jk} = 1$, and $1/3$ for $q_{jk} = 0$.

Similar to step 1, step 3 uses $\boldsymbol{E}_{2^K \times K} = (\varepsilon_{nk})_{2^K \times K}$ as the matrix of possible Q-matrix patterns for item j. $\boldsymbol{E}$ has 2^K rows, and each row of $\boldsymbol{E}$ represents a possible Q-matrix pattern. Containing all 0's, the first row of $\boldsymbol{E}$ is excluded in this step. That is, from (3.12), there are only 3 possible Q-matrix patterns. Let $\boldsymbol{\phi}_j = (\boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_{2^K})$, then the likelihood of each possible Q-matrix pattern for item j is

$$p(\boldsymbol{q}_j | \boldsymbol{\phi}_j) = \left(\prod_{n=1}^{K} \phi_{2n}^{\varepsilon_{2n}} (1 - \phi_{2k})^{1 - \varepsilon_{2n}}, \ldots, \prod_{n=1}^{K} \phi_{2^K n}^{\varepsilon_{2^K n}} (1 - \phi_{jn})^{1 - \varepsilon_{2^K n}} \right). \tag{3.22}$$

Each element in $p(\boldsymbol{q}_j|\boldsymbol{\phi}_j)$ is the probability of a possible Q-matrix pattern. Therefore, from (3.9), with the likelihood for item j from each of the $2^K - 1$ possible patterns and $p(\boldsymbol{q}_j|\boldsymbol{\phi}_j)$, the Q-matrix for item j can be sampled from the full conditional posterior distribution. After the procedure is applied to every item, the whole Q-matrix is derived for iteration t. It should be noted that if a procedure can apply the algorithm to all items at once, the estimated Q-matrix will be still the same because the items are conditionally independent to one another.

Initial Values

The initial values for the DINA model are generated as the following:

$$\theta_i \sim Uniform(0, 0.1) \tag{3.23}$$

$$q_{jk} \sim Bernoulli(0.5) \tag{3.24}$$

$$\phi_{jk} \sim Uniform(0, 1) \tag{3.25}$$

$$s_j \sim Uniform(0.1, 0.5) \tag{3.26}$$

$$g_j \sim Uniform(0.1, 0.5) \tag{3.27}$$

3.2 Estimating the Q-matrix for the rRUM

Because the estimating procedures for $\boldsymbol{\alpha}$ and $\boldsymbol{q}$ in the rRUM are similar to those in the DINA model, they are only briefly presented in this section. The sampling algorithm for the rRUM differs from that for the DINA model mainly in updating the $\boldsymbol{\pi}^*$ and $\boldsymbol{r}^*$ parameters. As no conjugate prior could be found for $\boldsymbol{\pi}^*$ and $\boldsymbol{r}^*$, random walk Metropolis-Hastings algorithm is used to update the two parameters. The Bayesian formulation of estimating the Q-matrix for the rRUM is represented as the following,

$$\left.\begin{array}{c} y_{ij} \sim Bernoulli(p(\boldsymbol{\alpha}_i)) \\ p(\boldsymbol{\alpha}_i) = \pi_j^* \prod_{k=1}^{K} (r_{jk}^{*(1-\alpha_{ik})})^{q_{jk}} \\ \boldsymbol{\alpha}_i|\boldsymbol{\theta}_i \sim Multinomial(2^K, \boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i \sim Dirichlet(a_1, a_2, \ldots, a_{2^K}) \\ \boldsymbol{q}_j|\boldsymbol{\phi}_j \sim Multinomial(2^K - 1, \boldsymbol{\phi}_j) \\ \pi_j^* \sim Beta(a, b) \\ r_{jk}^* \sim Beta(c, d) \end{array}\right\}. \tag{3.28}$$

The full conditional posterior distributions of the parameters for the rRUM are:

$$p(\boldsymbol{\alpha}_i|\boldsymbol{y}, \boldsymbol{r}^*, \boldsymbol{\pi}^*, \boldsymbol{q}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}_i, \boldsymbol{\pi}^*, \boldsymbol{r}^*, \boldsymbol{q})p(\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i) \tag{3.29}$$

$$p(\boldsymbol{\pi}^*|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{r}^*, \boldsymbol{q}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{\pi}^*, \boldsymbol{r}^*, \boldsymbol{q})p(\boldsymbol{\pi}^*) \tag{3.30}$$

$$p(\boldsymbol{r}^*|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{\pi}^*, \boldsymbol{q}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{\pi}^*, \boldsymbol{r}^*, \boldsymbol{q})p(\boldsymbol{r}^*) \tag{3.31}$$

$$p(\boldsymbol{q}_j|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{\pi}^*, \boldsymbol{r}^*) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{\pi}^*, \boldsymbol{r}^*, \boldsymbol{q}_j)p(\boldsymbol{q}_j|\boldsymbol{\phi}_j) \tag{3.32}$$

Explicitly, with the likelihood from (2.14), the full conditional posterior for $\boldsymbol{\alpha}_i$ is

$$
\begin{aligned}
p(\boldsymbol{\alpha}_i|\boldsymbol{y}_i,\boldsymbol{\theta}_i,\boldsymbol{g},\boldsymbol{s},\boldsymbol{q}) &\propto p(\boldsymbol{y}_i|\boldsymbol{\alpha}_i,\boldsymbol{\theta}_i,\boldsymbol{g},\boldsymbol{s},\boldsymbol{q})p(\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i) \\
&= \left[\prod_{j=1}^{J}\left(\pi_j^*\prod_{k=1}^{K}r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{y_{ij}}\left(1-\pi_j^*\prod_{k=1}^{K}r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{1-y_{ij}}\right] \\
&\times\left[\binom{n}{y_1 y_2\cdots y_{2^K}}\prod_{i=1}^{2^K}\theta_i^{y_i}\right]\left[\frac{1}{B(\boldsymbol{a})}\prod_{i=1}^{2^K}\theta_i^{a_i-1}\right],
\end{aligned}
\tag{3.33}
$$

and the full conditional posterior for $\boldsymbol{\pi}^*$ and $\boldsymbol{r}^*$ given the rest of the parameters and the data is

$$
\begin{aligned}
p(\boldsymbol{\pi}^*,\boldsymbol{r}^*|\boldsymbol{\alpha},\boldsymbol{q},\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha},\boldsymbol{\pi}^*,\boldsymbol{r}^*,\boldsymbol{q})p(\boldsymbol{\pi}^*)p(\boldsymbol{r}^*) \\
&= \prod_{i=1}^{I}\prod_{j=1}^{J}\left(\pi_j^*\prod_{k=1}^{K}r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{y_{ij}}\left(1-\pi_j^*\prod_{k=1}^{K}r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{1-y_{ij}}p(\boldsymbol{\pi}^*)p(\boldsymbol{r}^*).
\end{aligned}
\tag{3.34}
$$

The sampling procedure is discussed as follows. At iteration t, $t = 1,\ldots,T$, run the following steps:

Step 1: Updating Attributes

The procedure for updating attributes in the rRUM is similar to that in the DINA model. A saturated model is used to cope with correlated attributes. The model is

$$\boldsymbol{\alpha}_i|\boldsymbol{\theta}_i \sim Multinomial(2^K,\boldsymbol{\theta}_i), \tag{3.35}$$

$$\boldsymbol{\theta}_i \sim Dirichlet(a_1,a_2,\ldots,a_{2^K}). \tag{3.36}$$

After converted to decimal numbers, binary possible attribute patterns are distributed as a multinomial distribution. A Dirichlet prior is used because it is a conjugate prior for a multinomial distribution. Therefore, the posterior is also a Dirichlet distribution. Gamma distributions are used to construct the posterior Dirichlet distribution.

Step 2: Updating $\boldsymbol{r}^*$ and $\boldsymbol{\pi}^*$ Parameters

Random walk Metropolis-Hastings algorithm is used to update r^*_{jk} and π^*_j. As $\boldsymbol{\pi}^*$ and $\boldsymbol{r}^*$ are assumed to be independent of each other, $p(\boldsymbol{\pi}^*, \boldsymbol{r}^*) = p(\boldsymbol{\pi}^*)p(\boldsymbol{r}^*)$. The algorithm uses $Beta(1,1)$ as the prior for π^*_j and r^*_{jk},

$$\pi^*_j \sim Beta(1,1), \tag{3.37}$$

$$r^*_{jk} \sim Beta(1,1). \tag{3.38}$$

At iteration t, the algorithm samples candidate values for $\boldsymbol{r}^*$ from $Uniform(\boldsymbol{r}^{*(t-1)} - \delta, \boldsymbol{r}^{*(t-1)} + \delta)$, and for $\boldsymbol{\pi}^*$ from $Uniform(\boldsymbol{\pi}^{*(t-1)} - \delta, \boldsymbol{\pi}^{*(t-1)} + \delta)$. Note that candidate values for both $\boldsymbol{r}^*$ and $\boldsymbol{\pi}^*$ have to be restricted to the interval $(0,1)$, and δ is adjusted so that the acceptance rate is between 25% and 40% (see Gilks et al., 1996). After calculating the acceptance probability φ for candidates for $\boldsymbol{r}^*$ and $\boldsymbol{\pi}^*$,

$$\varphi = \frac{p(\boldsymbol{y}|\boldsymbol{\alpha}^{(t)}, \boldsymbol{r}^{*(*)}, \boldsymbol{\pi}^{*(*)}, \boldsymbol{q}^{(t-1)})p(\boldsymbol{r}^{*(*)})p(\boldsymbol{\pi}^{*(*)})}{p(\boldsymbol{y}|\boldsymbol{\alpha}^{(t)}, \boldsymbol{r}^{*(t-1)}, \boldsymbol{\pi}^{*(t-1)}, \boldsymbol{q}^{(t-1)})p(\boldsymbol{r}^{*(t-1)})p(\boldsymbol{r}^{*(t-1)})}, \tag{3.39}$$

set

$$\boldsymbol{r}^{*(t)} = \begin{cases} \boldsymbol{r}^{*(*)} & \text{with probability } \min(1,\varphi) \\ \boldsymbol{r}^{*(t-1)} & \text{otherwise} \end{cases}, \tag{3.40}$$

and

$$\boldsymbol{\pi}^{*(t)} = \begin{cases} \boldsymbol{\pi}^{*(*)} & \text{with probability } \min(1,\varphi) \\ \boldsymbol{\pi}^{*(t-1)} & \text{otherwise} \end{cases}. \tag{3.41}$$

Step 3: Updating the Q-matrix

The procedure for updating the Q-matrix is also the same in the rRUM. A saturated model is used in updating the Q-matrix. With the likelihood of each possible Q-matrix pattern and the probability of each of the $2^K - 1$ patterns, the normalized probability for each possible pattern is obtained. The Q-matrix pattern for item j is sampled from this discrete distribution.

Each entry in the Q-matrix is denoted as q_{jk}. Let $p(q_{jk} = 1) = \phi_{jk}$ and $p(q_{jk} = 0) = 1 - \phi_{jk}$. $Beta(1, 1)$ is used as the prior for ϕ_{jk}. As the conjugate prior for a Bernoulli distribution, the posterior is $Beta(1 + q_{jk}, 2 - q_{jk})$.

In updating the Q-matrix for item j, a saturated model is used,

$$\boldsymbol{q}_j|\boldsymbol{\phi}_j \sim Multinomial(2^K - 1, \boldsymbol{\phi}_j). \tag{3.42}$$

As in (3.22), the distribution of the $2^K - 1$ possible Q-matrix patterns is

$$p(\boldsymbol{q}_j|\boldsymbol{\phi}_j) = \left(\prod_{n=1}^{K} \phi_{2n}^{\varepsilon_{2n}}(1 - \phi_{2k})^{1-\varepsilon_{2n}}, \ldots, \prod_{n=1}^{K} \phi_{2^K n}^{\varepsilon_{2^K n}}(1 - \phi_{jn})^{1-\varepsilon_{2^K n}}\right). \tag{3.43}$$

For item j, the full conditional posterior distribution is

$$\begin{aligned} p(\boldsymbol{q}_j|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{g}, \boldsymbol{s}) &\propto p(\boldsymbol{y}|\boldsymbol{q}_j, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{\alpha})p(\boldsymbol{q}_j|\boldsymbol{\phi}_j) \\ &= \left[\prod_{i=1}^{I}\left(\pi_j^* \prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{y_{ij}}\left(1 - \pi_j^* \prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{jk})q_{jk}}\right)^{1-y_{ij}}\right] p(\boldsymbol{q}_j|\boldsymbol{\phi}_j). \end{aligned} \tag{3.44}$$

From (3.43) and (3.44), we can sample from the posterior distribution. Convert the decimal number to binary number, and the Q-matrix pattern for item j is derived. After applying the procedure to each item, we can obtain the whole Q-matrix for iteration t.

Initial Values

The initial values for the rRUM are generated as the following:

$$\theta_i \sim Uniform(0, 0.1) \tag{3.45}$$

$$q_{jk} \sim Bernoulli(0.5) \tag{3.46}$$

$$\phi_{jk} \sim Uniform(0, 1) \tag{3.47}$$

$$\pi_j^* \sim Uniform(0, 1) \tag{3.48}$$

$$r_{jk}^* \sim Uniform(0, 1) \tag{3.49}$$

3.3 Label Switching

One concern in Bayesian Q-matrix estimation, as in Bayesian factor analysis, is label switching, which arises when components of the Bayesian model are switched multiple times on different iterations during one run of an MCMC sampler (Jasra, Holmes, & Stephens, 2005). Since the label sampled is assigned at each step of the sampler, the assignment of the particular label is unique only up to the permutation group. For example, the following two Q-matrices are equivalent although column 1 and column 3 are switched,

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \tag{3.50}$$

With 3 attributes, there are $3! = 6$ equivalent Q-matrices. Therefore, if label switching happens during a run of MCMC, posterior summaries will be biased and have inflated variance, although the result may match after attribute names are relabeled.

Erosheva and Curtis (2012) proposed a method to deal with label switching in Bayesian confirmatory factor analysis. Without posing any constraints in the analysis, the concept of their method is relabel the factors after the fact. Inspired by Erosheva and Curtis (2012), this research develops an algorithm to account for label switching by reordering the columns of attributes in the Q-matrix at each iteration.

The logic of this algorithm is that if the underlying probability of the Q-matrix ϕ converges, the shortest Euclidean distance to this arbitrary matrix should be from only one permutation of ϕ, if each entry of the arbitrary matrix is a random decimal. The algorithm is outlined as the following steps:

Step 1

Let $\boldsymbol{P}_{J\times K} = (\phi_{jk})_{J\times K}$ be the underlying probability of the Q-matrix $\boldsymbol{Q}_{J\times K}$, and create an arbitrary matrix $\boldsymbol{D}_{J\times K} = (d_{jk})_{J\times K}$, with each entry d_{jk} generated from $Uniform(0, 1)$.

Step 2

At iteration t, permute the columns of $\boldsymbol{P}$ estimate, $\boldsymbol{P}_{est}^{(t)}$, and calculate the Euclidean distance between every permutation of $\boldsymbol{P}_{est}^{(t)}$ and $\boldsymbol{D}$.

Step 3

The permutation of $\boldsymbol{P}_{est}^{(t)}$ with the shortest Euclidean distance to $\boldsymbol{D}$ is used at iteration $t+1$ to estimate the Q-matrix, $\boldsymbol{q}_{est}^{(t+1)}$. The final Q-matrix estimate is the average of these relabeled Q-matrix estimates from all iterations excluding burn-ins.

3.4 Simulated Data

Setup

To understand the effectiveness of the algorithm developed in this research, data sets are simulated for the DINA model and rRUM from an artificial Q-matrix. Simulation studies are conducted to see if the true Q-matrix could be recovered. The Q-matrix in table (3.1) is obtained from de la Torre (2008). Thirty items that measure 5 attributes comprise the Q-matrix, which is constructed in a way that each attribute appears alone, in a pair, or in triple the same number of times as other attributes. This Q-matrix also appears to have a clear pattern that implies main effects from items 1 to 10, two-way interactions from items 11 to 20 and three-way interactions from items 21 to 30. This Q-matrix is complete, containing at least one item devoted solely to each attribute (Zhang, DeCarlo, & Ying, 2014).

A hundred datasets are generated using this Q-matrix and the average Q-matrix estimates are calculated for both models. To measure how well the algorithm recovers the true Q-matrix, the measure of accuracy defined in section 3.4 is reported.

Computations are conducted on a laptop computer running Ubuntu 12.10 Linux with 2.4 GHz CPU and 6 GB RAM. For each simulation study, corresponding R programs are run 100,000 iterations after 10,000 burn-ins. Simulated data are generated using the following steps.

Step 1: Decomposing Correlation Matrix Σ

The procedure for generating attributes is applied to both the DINA model and the rRUM. The setting is similar to that in Chiu, Douglas, and Li (2009) and Liu, Xu, and Ying (2013). Attributes are assumed to be correlated with one another in the simulation. Let $\boldsymbol{\vartheta} = (\boldsymbol{\vartheta}_1, \ldots, \boldsymbol{\vartheta}_K)$ be the underlying probability of $\boldsymbol{\alpha}$. A copula is used to generate intercorrelated $\boldsymbol{\vartheta}$ (see Ross, 2013). The correlation coefficient ρ takes a constant value for

Table 3.1: Q-matrix for Simulation Studies

	Attribute						Attribute				
Item	1	2	3	4	5	Item	1	2	3	4	5
1	1	0	0	0	0	16	0	1	0	1	0
2	0	1	0	0	0	17	0	1	0	0	1
3	0	0	1	0	0	18	0	0	1	1	0
4	0	0	0	1	0	19	0	0	1	0	1
5	0	0	0	0	1	20	0	0	0	1	1
6	1	0	0	0	0	21	1	1	1	0	0
7	0	1	0	0	0	22	1	1	0	1	0
8	0	0	1	0	0	23	1	1	0	0	1
9	0	0	0	1	0	24	1	0	1	1	0
10	0	0	0	0	1	25	1	0	1	0	1
11	1	1	0	0	0	26	1	0	0	1	1
12	1	0	1	0	0	27	0	1	1	1	0
13	1	0	0	1	0	28	0	1	1	0	1
14	1	0	0	0	1	29	0	1	0	1	1
15	0	1	1	0	0	30	0	0	1	1	1

each pair of columns in $\boldsymbol{\theta}$, and the correlation matrix $\boldsymbol{\Sigma}$ is

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & & \rho \\ & \ddots & \\ \rho & & 1 \end{bmatrix}. \tag{3.51}$$

Each entry in $\boldsymbol{\Sigma}$ corresponds to the correlation coefficient between two columns in $\boldsymbol{\vartheta}$. Symmetric with all the eigenvalues positive, $\boldsymbol{\Sigma}$ is a real symmetric positive-definite matrix, which can be decomposed as $\boldsymbol{\Sigma} = \boldsymbol{C}^T\boldsymbol{C}$ using Choleski decomposition, where $\boldsymbol{C}$ is an upper triangular matrix.

Step 2: Generating Correlated $\boldsymbol{\alpha}$

After obtaining $\boldsymbol{C}$, create an $I \times K$ matrix $\boldsymbol{U}$, each entry of which is generated from $N(0, 1)$. $\boldsymbol{U}$ is further transformed to $\boldsymbol{R}$ by using $\boldsymbol{R} = \boldsymbol{UC}$, so that $\boldsymbol{R}$ and $\boldsymbol{\Sigma}$ will have the same correlation structure. Set $\Phi(\boldsymbol{R}) = \boldsymbol{\vartheta}$, where $\Phi(\cdot)$ is the cumulative standard normal distribution function, and determine $\boldsymbol{\alpha}_i = (\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, \ldots, \boldsymbol{\alpha}_{iK})$ by

$$\alpha_{ik} = \begin{cases} 1 & \text{if } \vartheta_{ik} \geq \Phi^{-1}(\frac{k}{K+1}) \\ 0 & \text{otherwise} \end{cases}, \tag{3.52}$$

where $k = 1, 2, \ldots, K$.

Step 3: Generating Data

For the DINA model, examinees' $\boldsymbol{\eta}$ are determined by (3.10). When the data is simulated with slip and guess parameters for all items setting to 0.2, the probability of correctly answering an item is 0.8 for examinees whose $\eta = 1$, and the probability of correctly answering an item is 0.2 for those whose $\eta = 0$. An $N \times J$ probability matrix is thus formed, with each of the elements representing the probability of an examinee correctly answering an item.

The inverse transform sampling for two categories, 0 and 1, is conducted to generate the data. The concept is create another $N \times J$ probability matrix, with each element generated from $Uniform(0, 1)$, and then compare the two $N \times J$ matrices. For $\eta = 1$, if the corresponding value from $Uniform(0, 1)$ is greater then 0.8, then change the η from 1 to 0, meaning slip; if the corresponding value from $Uniform(0, 1)$ is less than 0.8, then η remains 1. For $\eta = 0$, if the corresponding value from $Uniform(0, 1)$ is less then 0.2, then convert the η from 0 to 1, meaning guess; if the corresponding value from $Uniform(0, 1)$ is greater than 0.2, the η remains 0.

For the rRUM, $\boldsymbol{\alpha}$ is generated from steps 1 and 2. g_{jk} and s_{jk} are set to 0.2, and π_j^* and r_{jk}^* are obtained from (2.10) and (2.11). The data is then generated using inversion sampling method from two points, in which the probability is obtained from (2.13)

Measure of Accuracy Δ

After the Q-matrix is derived, we need to evaluate how well the algorithm recovers the true Q-matrix. A measure of discrepancy between the estimated Q-matrix and the true Q-matrix is defined as the following. Suppose we have M datasets, Let $\hat{\boldsymbol{Q}}^{(m)} = (\hat{q}_{jk}^{(m)})_{J\times K}$ and $\boldsymbol{Q} = (q_{jk})_{J\times K}$ represents the estimated Q-matrix from mth dataset and true Q-matrix, respectively. The measure of discrepancy δ is defined as

$$\delta = \frac{1}{M}\sum_{m=1}^{M}\frac{|\hat{\boldsymbol{Q}}^{(m)} - \boldsymbol{Q}|}{JK},\ m = 1, 2, \ldots, M \tag{3.53}$$

where the $|\cdot|$ is the absolute value. A measure of accuracy Δ is further defined as $\Delta = 1-\delta$, which is confined between 0 and 1. The higher the Δ the better the estimate.

3.5 Simulation Design

Scenarios of the three simulation studies are as follows.

Simulation Study 1

Twenty simulated data for simulation study 1 were generated from independent attributes for the DINA model and rRUM. That is, each entry of $\boldsymbol{\alpha}$ was generated from $Bernoulli(0.5)$. g_j and s_j were set to 0.2 for each item in the DINA model. g_{jk} and s_{jk} were set to 0.2 for each item in the rRUM. Sample size for study 1 is 2,000.

Simulation Study 2

In generating $\boldsymbol{\alpha}$, simulation study 2 set ρ to 0.3. Cutoff point for $\boldsymbol{\alpha}$ were set using (3.52). g_j and s_j were set to 0.2 for each item in the DINA model. g_{jk} and s_{jk} were set to 0.2 for each item in the rRUM. A hundred data sets were generated for the DINA model and rRUM. Sample size for study 2 is 2,000.

Simulation Study 3

Simulation study 3 used only the DINA model. Twenty data sets were generated using $\rho = 0.15$ for $\boldsymbol{\alpha}$. g_j and s_j were both generated from $Uniform(0.1, 0.4)$. Using the same values of g_j and s_j, another 20 data sets were generated using $\rho = 0.3$ for $\boldsymbol{\alpha}$. Cutoff points for $\boldsymbol{\alpha}$ were set using (3.52). Sample size for simulation study 3 is 1,000.

3.6 Empirical Study

In addition to simulations, this research estimated the Q-matrix from real data. The fraction-subtraction data (Tatsuoka, 1990) that consists of 536 examinees has been widely analyzed (e.g., DeCarlo, 2011, 2012; de la Torre, 2008, 2009; de la Torre & Douglas, 2008; Henson, Templin, & Willse, 2009; C. Tatsuoka, 2002; K. K. Tatsuoka, 1990), and virtually every researcher who has used the data has suggested possible modifications of the Q-matrix (DeCarlo, 2012).

DeCarlo (2011) analyzed the 20-item version of the fraction-subtraction data using the RDINA model and showed that there are some issues with respect to the classification of examinees and the latent class sizes. Specifically, it is shown that if latent class size estimates for one or more skills are close to unity, then it is likely that the Q-matrix has been misspecified. The problems are largely associated with the specification of the Q-matrix, and DeCarlo (2011) suggested further research on effects of the Q-matrix specification on the classification of examinees and estimation of the latent class sizes.

De la Torre (2008) suggested a 15-item version of the fraction-subtraction data. The Q-matrix (Table 3.2) suggested by de la Torre (2008) consists of 5 attributes: (1) performing basic fraction-subtraction operation, (2) simplifying/reducing, (3) separating whole number from fraction, (4) borrowing one from whole number to fraction, and (5) converting whole number to fraction. Although the small sample size of 536 is a concern, this research uses the 15-item version of the fraction-subtraction data to estimate its Q-matrix. The estimated Q-matrix is compared with the Q-matrix suggested by de la Torre (2008).

Table 3.2: Q-matrix for the Fraction-Subtraction Data

Item		Attribute 1	2	3	4	5
1	$\frac{3}{4} - \frac{3}{8}$	1	0	0	0	0
2	$3\frac{1}{2} - 2\frac{3}{2}$	1	1	1	1	0
3	$\frac{6}{7} - \frac{4}{7}$	1	0	0	0	0
4	$3 - 2\frac{1}{5}$	1	1	1	1	1
5	$3\frac{7}{8} - 2$	0	0	1	0	0
6	$4\frac{4}{12} - 2\frac{7}{12}$	1	1	1	1	0
7	$4\frac{1}{3} - 2\frac{4}{3}$	1	1	1	1	0
8	$\frac{11}{8} - \frac{1}{8}$	1	1	0	0	0
9	$3\frac{4}{5} - 3\frac{2}{5}$	1	0	1	0	0
10	$2 - \frac{1}{3}$	1	0	1	1	1
11	$4\frac{5}{7} - 1\frac{4}{7}$	1	0	1	0	0
12	$7\frac{3}{5} - \frac{4}{5}$	1	0	1	1	0
13	$4\frac{1}{10} - 2\frac{8}{10}$	1	1	1	1	0
14	$4 - 1\frac{4}{3}$	1	1	1	1	1
15	$4\frac{1}{3} - 1\frac{5}{3}$	1	1	1	1	0

(1) performing basic fraction-subtraction operation (2) simplifying/reducing (3) separating whole number from fraction (4) borrowing one from whole number to fraction (5) converting whole number to fraction

Chapter 4 Results

4.1 Simulation Studies

Simulation Study 1

Although the purpose of the research is to explore whether the algorithms can recover the true Q-matrix, parameter recovery for the DINA model and rRUM appeared to be good when the true Q-matrix is given.

Twenty data sets that assumed independent attributes were generated. Each entry of $\boldsymbol{\alpha}$ was generated from $Bernoulli(0.5)$. Results from study 1 showed that the true Q-matrix could be always fully recovered when the cutoff point is set to 0.5.

Simulation Study 2

The DINA model. Table 4.1 shows the mean Q-matrix estimates from 100 simulated data sets for the DINA model. The right part of table 4.1 is the discrepancy between true and estimated Q-matrices $|\hat{q}_{jk} - q_{jk}|$, where $|\cdot|$ denotes an absolute value. After checking the trace plots for each of the estimated Q-matrix, no label switching was identified and the chain became stable after about 10,000 iterations. Convergence was also checked using trace plots, and no bad mixing is observed. The variance of Q-matrix estimates from the 100 data sets is reported in appendix I.

Defined in section 3.4, the measure of accuracy Δ is 0.966 for the mean estimated Q-matrix. For individual attribute, Δ's for attributes 1 to 5 are 0.888, 0.971, 0.993, 0.997 and 0.983, respectively. The optimized true Q-matrix shows a pattern that suggests main effects from items 1 to 10, two-way interactions from items 11 to 20 and three-way interactions from items 21 to 30. Δ's for these three parts of items are 0.974, 0.967 and 0.958, respectively. Intuitively, we would expect that Δ for items 21 to 30 be smaller. Although there seems to be a trend that Δ is smaller as the number of attributes being measured increases, there is really no significant difference among them.

If the cutoff point is set to 0.5, the mean estimated Q-matrix is shown to fully recover the true Q-matrix. As can be seen, the result seems to be promising, demonstrating the algorithm for the DINA model works well. Nevertheless, it should be noted that the algorithm was not effective enough to accurately estimate attribute 1, due to the more complicated setting of cutoff point for simulating attributes by (3.52).

The rRUM. Table 4.2 presents the mean Q-matrix estimates from 100 simulated data sets for the rRUM. The right part of table 4.2 is the discrepancy between true and estimated Q-matrices. As in the DINA model, convergence was checked using trace plots, and no bad mixing was observed. The trace plots for each of the 100 estimated Q-matrices suggested no label switching. The variance of the estimates is included in appendix II.

The measure of accuracy Δ is 0.876 for the mean estimated Q-matrix. Δ's for attributes 1 to 5 are 0.815, 0.852, 0.901, 0.904 and 0.907, respectively. As can be seen, Δ for attribute 1 is relatively low. Δ's for items 1 to 10, items 11-20, items 21-30 are 0.881, 0.860 and 0.887, respectively. As in the DINA model, the Δ's for the three parts of items show no prominent difference.

If the cutoff point is set to 0.5, two entries in attribute 1, $\hat{q}_{20,1}$ and $\hat{q}_{28,1}$, are missed. Considering that rRUM is a more complicated model with more parameters to be estimated, it is not surprising to see that compared with the DINA model, the rRUM yielded a less accurate estimate. Nevertheless, the result is satisfactory, showing that the algorithm for the rRUM is useful. In sum, from simulation studies 1 and 2, algorithms based on the DINA model and the rRUM performed well, and no label switching was observed.

Simulation Study 3

g_j and s_j in simulation study 3 were generated from $Uniform(0.1, 0.4)$, and the same set values of g_j and s_j were used in each data set. Table (4.3) shows the result for $\rho = 0.15$, and table (4.4) presents the result for $\rho = 0.3$. It is expected that the recovery rate be lower in study 3, in that there are more variability in guess and slip parameters and the sample size is smaller. When ρ is 0.15, Δ is 0.929. If cutoff is set to 0.5, there are 3 entries missed. When ρ is 0.3, Δ is 0.904. If cutoff is set to 0.5, there are 9 entries missed. As can be seen, when ρ is smaller, the recovery rate is higher.

Summary of Simulation Studies

It should be noted that the columns of estimated Q-matrices above were reordered so that the permutation with smallest Euclidean distance to the true Q-matrix was saved for final analysis. From the result of simulation study 1, where attributes are independent, both the DINA model and the rRUM can fully recover the true Q-matrix when cutoff point is set to 0.5.

When attributes are correlated as in simulation studies 2 and 3, recovery rate deteriorates to some extent. From simulation study 2, the recovery rate is higher for the DINA model than for the rRUM. From simulation study 3, it can be seen that recovery rates decline for both models, compared to simulation study 2. Sample size, degree of correlation, and the way to set cutoff point for attributes have different impacts on the recovery rate. It is worth noting that in simulation study 3, although the higher the attributes correlate, the lower the recovery rate, the difference is not distinct.

Table 4.1: Q-matrix from Study 2 for the DINA Model ($N = 2000,\ \rho = 0.3,\ g = s = 0.2$)

	Estimated Q-matrix						Discrepancy				
	Attribute						Attribute				
Item	1	2	3	4	5	Item	1	2	3	4	5
1	0.976	0.030	0.000	0.000	0.001	1	0.024	0.030	0.000	0.000	0.001
2	0.041	0.999	0.003	0.013	0.001	2	0.041	0.001	0.003	0.013	0.001
3	0.029	0.005	0.990	0.000	0.020	3	0.029	0.005	0.010	0.000	0.020
4	0.035	0.021	0.000	1.000	0.031	4	0.035	0.021	0.000	0.000	0.031
5	0.295	0.059	0.016	0.007	0.986	5	0.295	0.059	0.016	0.007	0.014
6	0.977	0.030	0.000	0.001	0.000	6	0.023	0.030	0.000	0.001	0.000
7	0.041	0.999	0.003	0.013	0.010	7	0.041	0.001	0.003	0.013	0.010
8	0.029	0.005	0.990	0.000	0.020	8	0.029	0.005	0.010	0.000	0.020
9	0.036	0.021	0.000	1.000	0.031	9	0.036	0.021	0.000	0.000	0.031
10	0.281	0.059	0.014	0.008	0.988	10	0.281	0.059	0.014	0.008	0.012
11	0.970	0.999	0.003	0.013	0.001	11	0.030	0.001	0.003	0.013	0.001
12	0.987	0.021	0.990	0.000	0.020	12	0.013	0.021	0.010	0.000	0.020
13	0.962	0.021	0.000	1.000	0.030	13	0.038	0.021	0.000	0.000	0.030
14	0.936	0.054	0.011	0.002	0.989	14	0.064	0.054	0.011	0.002	0.011
15	0.057	0.990	0.990	0.012	0.020	15	0.057	0.010	0.010	0.012	0.020
16	0.076	0.989	0.000	1.000	0.029	16	0.076	0.011	0.000	0.000	0.030
17	0.247	0.998	0.011	0.001	0.988	17	0.247	0.002	0.011	0.001	0.012
18	0.034	0.019	0.990	1.000	0.041	18	0.034	0.019	0.010	0.000	0.041
19	0.262	0.051	1.000	0.000	0.997	19	0.262	0.051	0.000	0.000	0.003
20	0.247	0.094	0.011	1.000	0.984	20	0.247	0.094	0.011	0.000	0.016
21	0.980	0.990	0.990	0.012	0.015	21	0.020	0.010	0.010	0.012	0.015
22	0.989	0.985	0.000	1.000	0.028	22	0.011	0.015	0.000	0.000	0.028
23	0.940	0.998	0.011	0.000	0.988	23	0.060	0.002	0.011	0.000	0.012
24	0.948	0.019	0.990	1.000	0.039	24	0.052	0.019	0.010	0.000	0.039
25	0.880	0.056	1.000	0.005	0.995	25	0.120	0.056	0.000	0.005	0.005
26	0.844	0.082	0.011	1.000	0.986	26	0.156	0.082	0.011	0.000	0.014
27	0.090	0.989	0.990	1.000	0.033	27	0.090	0.011	0.010	0.000	0.033
28	0.258	0.987	1.000	0.009	0.994	28	0.258	0.013	0.000	0.009	0.006
29	0.351	0.975	0.023	1.000	0.986	29	0.351	0.025	0.023	0.000	0.014
30	0.354	0.122	0.999	1.000	0.992	30	0.354	0.122	0.001	0.000	0.008

Table 4.2: Q-matrix from Study 2 for the rRUM ($N = 2000,\ \rho = 0.3,\ g = s = 0.2$)

	Estimated Q-matrix						Discrepancy				
	Attribute						Attribute				
Item	1	2	3	4	5	Item	1	2	3	4	5
1	0.971	0.135	0.074	0.074	0.071	1	0.029	0.135	0.074	0.074	0.071
2	0.123	0.971	0.100	0.082	0.160	2	0.123	0.029	0.100	0.082	0.160
3	0.162	0.132	0.994	0.082	0.104	3	0.162	0.132	0.006	0.082	0.104
4	0.208	0.250	0.157	0.980	0.104	4	0.208	0.250	0.157	0.020	0.104
5	0.260	0.221	0.166	0.156	0.966	5	0.260	0.221	0.166	0.156	0.034
6	0.974	0.111	0.073	0.076	0.083	6	0.026	0.111	0.073	0.076	0.083
7	0.166	0.978	0.071	0.120	0.103	7	0.166	0.022	0.071	0.12	0.103
8	0.189	0.162	0.994	0.133	0.079	8	0.189	0.162	0.006	0.133	0.079
9	0.264	0.130	0.106	0.988	0.110	9	0.264	0.130	0.106	0.012	0.110
10	0.395	0.265	0.134	0.150	0.978	10	0.395	0.265	0.134	0.15	0.022
11	0.975	0.976	0.114	0.132	0.125	11	0.025	0.024	0.114	0.132	0.125
12	0.973	0.137	0.995	0.284	0.122	12	0.027	0.137	0.005	0.284	0.122
13	0.971	0.440	0.118	0.984	0.169	13	0.029	0.44	0.118	0.016	0.169
14	0.972	0.233	0.359	0.152	0.982	14	0.028	0.233	0.359	0.152	0.018
15	0.153	0.978	0.994	0.144	0.108	15	0.153	0.022	0.006	0.144	0.108
16	0.226	0.977	0.125	0.984	0.334	16	0.226	0.023	0.125	0.016	0.334
17	0.310	0.975	0.171	0.147	0.953	17	0.310	0.025	0.171	0.147	0.047
18	0.244	0.142	0.991	0.981	0.156	18	0.244	0.142	0.009	0.019	0.156
19	0.346	0.392	0.994	0.237	0.985	19	0.346	0.392	0.006	0.237	0.015
20	0.506	0.338	0.187	0.986	0.977	20	0.506	0.338	0.187	0.014	0.023
21	0.971	0.975	0.994	0.146	0.128	21	0.029	0.025	0.006	0.146	0.128
22	0.959	0.982	0.187	0.983	0.199	22	0.041	0.018	0.187	0.017	0.199
23	0.971	0.979	0.189	0.177	0.980	23	0.029	0.021	0.189	0.177	0.020
24	0.969	0.208	0.993	0.980	0.179	24	0.031	0.208	0.007	0.02	0.179
25	0.938	0.281	0.964	0.183	0.984	25	0.062	0.281	0.036	0.183	0.016
26	0.942	0.268	0.226	0.984	0.978	26	0.058	0.268	0.226	0.016	0.022
27	0.276	0.974	0.992	0.984	0.188	27	0.276	0.026	0.008	0.016	0.188
28	0.536	0.974	0.955	0.212	0.985	28	0.536	0.026	0.045	0.212	0.015
29	0.384	0.973	0.267	0.986	0.981	29	0.384	0.027	0.267	0.014	0.019
30	0.378	0.301	0.992	0.984	0.981	30	0.378	0.301	0.008	0.016	0.019

Table 4.3: Q-matrix from Study 3 for the DINA Model ($N = 1000,\ \rho = 0.15$)

			Estimated Q-matrix					Discrepancy				
			Attribute					Attribute				
Item	g	s	1	2	3	4	5	1	2	3	4	5
1	0.337	0.169	0.962	0.049	0.004	0.004	0.002	0.038	0.049	0.004	0.004	0.002
2	0.346	0.265	0.011	1.000	0.000	0.000	0.000	0.011	0.000	0.000	0.000	0.000
3	0.221	0.201	0.053	0.000	1.000	0.000	0.000	0.053	0.000	0.000	0.000	0.000
4	0.188	0.265	0.055	0.000	0.000	1.000	0.000	0.055	0.000	0.000	0.000	0.000
5	0.346	0.186	0.446	0.168	0.068	0.003	1.000	0.446	0.168	0.068	0.003	0.000
6	0.362	0.128	0.963	0.051	0.002	0.001	0.001	0.037	0.051	0.002	0.001	0.001
7	0.227	0.315	0.043	1.000	0.000	0.000	0.000	0.043	0.000	0.000	0.000	0.000
8	0.201	0.322	0.110	0.000	1.000	0.000	0.000	0.110	0.000	0.000	0.000	0.000
9	0.289	0.243	0.116	0.000	0.000	1.000	0.000	0.116	0.000	0.000	0.000	0.000
10	0.338	0.329	0.524	0.246	0.215	0.055	1.000	0.524	0.246	0.215	0.055	0.000
11	0.252	0.316	0.948	1.000	0.000	0.000	0.000	0.052	0.000	0.000	0.000	0.000
12	0.292	0.254	0.929	0.002	1.000	0.000	0.000	0.071	0.002	0.000	0.000	0.000
13	0.363	0.367	0.688	0.106	0.017	1.000	0.001	0.312	0.106	0.017	0.000	0.001
14	0.301	0.337	0.733	0.330	0.099	0.033	1.000	0.267	0.330	0.099	0.033	0.000
15	0.373	0.355	0.186	0.960	1.000	0.000	0.000	0.186	0.040	0.000	0.000	0.000
16	0.264	0.164	0.047	1.000	0.000	1.000	0.000	0.047	0.000	0.000	0.000	0.000
17	0.284	0.367	0.445	0.676	0.054	0.067	1.000	0.445	0.324	0.054	0.067	0.000
18	0.159	0.283	0.114	0.002	1.000	1.000	0.000	0.114	0.002	0.000	0.000	0.000
19	0.202	0.279	0.332	0.187	0.999	0.000	1.000	0.332	0.187	0.001	0.000	0.000
20	0.153	0.244	0.474	0.295	0.027	1.000	1.000	0.474	0.295	0.027	0.000	0.000
21	0.238	0.121	0.929	0.991	1.000	0.000	0.000	0.071	0.009	0.000	0.000	0.000
22	0.113	0.372	0.954	1.000	0.000	1.000	0.000	0.046	0.000	0.000	0.000	0.000
23	0.186	0.124	0.943	0.931	0.000	0.000	1.000	0.057	0.069	0.000	0.000	0.000
24	0.332	0.277	0.829	0.127	0.910	0.999	0.000	0.171	0.127	0.090	0.001	0.000
25	0.146	0.172	0.842	0.108	0.999	0.000	1.000	0.158	0.108	0.001	0.000	0.000
26	0.125	0.352	0.691	0.161	0.047	1.000	1.000	0.309	0.161	0.047	0.000	0.000
27	0.336	0.181	0.480	0.971	0.999	1.000	0.000	0.480	0.029	0.001	0.000	0.000
28	0.225	0.222	0.463	0.849	0.997	0.001	1.000	0.463	0.151	0.003	0.001	0.000
29	0.302	0.299	0.634	0.715	0.253	0.928	0.999	0.634	0.285	0.253	0.072	0.001
30	0.148	0.257	0.503	0.217	0.995	0.996	1.000	0.503	0.217	0.005	0.004	0.000

Table 4.4: Q-matrix from Study 3 for the DINA Model ($N = 1000,\ \rho = 0.3$)

			Estimated Q-matrix					Discrepancy				
			Attribute					Attribute				
Item	g	s	1	2	3	4	5	1	2	3	4	5
1	0.337	0.169	0.985	0.031	0.021	0.027	0.021	0.015	0.031	0.021	0.027	0.021
2	0.346	0.265	0.017	1.000	0.000	0.000	0.000	0.017	0.000	0.000	0.000	0.000
3	0.221	0.201	0.030	0.000	1.000	0.000	0.000	0.030	0.000	0.000	0.000	0.000
4	0.188	0.265	0.118	0.146	0.000	1.000	0.000	0.118	0.146	0.000	0.000	0.000
5	0.346	0.186	0.645	0.507	0.116	0.014	1.000	0.645	0.507	0.116	0.014	0.000
6	0.362	0.128	0.998	0.008	0.002	0.001	0.001	0.002	0.008	0.002	0.001	0.001
7	0.227	0.315	0.008	1.000	0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.000
8	0.201	0.322	0.061	0.000	1.000	0.000	0.000	0.061	0.000	0.000	0.000	0.000
9	0.289	0.243	0.164	0.178	0.000	1.000	0.000	0.164	0.178	0.000	0.000	0.000
10	0.338	0.329	0.642	0.409	0.302	0.193	1.000	0.642	0.409	0.302	0.193	0.000
11	0.252	0.316	0.780	1.000	0.000	0.000	0.000	0.220	0.000	0.000	0.000	0.000
12	0.292	0.254	0.896	0.017	1.000	0.000	0.000	0.104	0.017	0.000	0.000	0.000
13	0.363	0.367	0.540	0.159	0.044	1.000	0.001	0.460	0.159	0.044	0.000	0.001
14	0.301	0.337	0.795	0.375	0.231	0.102	1.000	0.205	0.375	0.231	0.102	0.000
15	0.373	0.355	0.367	0.921	0.993	0.003	0.001	0.367	0.079	0.007	0.003	0.001
16	0.264	0.164	0.293	0.997	0.000	1.000	0.000	0.293	0.003	0.000	0.000	0.000
17	0.284	0.367	0.532	0.821	0.101	0.053	1.000	0.532	0.179	0.101	0.053	0.000
18	0.159	0.283	0.168	0.159	1.000	1.000	0.000	0.168	0.159	0.000	0.000	0.000
19	0.202	0.279	0.552	0.219	0.997	0.001	1.000	0.552	0.219	0.003	0.001	0.000
20	0.153	0.244	0.521	0.349	0.010	1.000	1.000	0.521	0.349	0.010	0.000	0.000
21	0.238	0.121	0.919	0.900	0.900	0.000	0.000	0.081	0.100	0.100	0.000	0.000
22	0.113	0.372	0.937	0.965	0.000	1.000	0.000	0.063	0.035	0.000	0.000	0.000
23	0.186	0.124	0.807	0.942	0.001	0.000	1.000	0.193	0.058	0.001	0.000	0.000
24	0.332	0.277	0.741	0.067	0.997	1.000	0.001	0.259	0.067	0.003	0.000	0.001
25	0.146	0.172	0.818	0.125	1.000	0.000	1.000	0.182	0.125	0.000	0.000	0.000
26	0.125	0.352	0.782	0.494	0.077	1.000	1.000	0.218	0.494	0.077	0.000	0.000
27	0.336	0.181	0.315	0.950	1.000	1.000	0.000	0.315	0.050	0.000	0.000	0.000
28	0.225	0.222	0.665	0.904	0.985	0.000	1.000	0.665	0.096	0.015	0.000	0.000
29	0.302	0.299	0.561	0.690	0.283	0.953	1.000	0.561	0.310	0.283	0.047	0.000
30	0.148	0.257	0.540	0.179	0.996	1.000	1.000	0.540	0.179	0.004	0.000	0.000

4.2 Empirical Study

Tables (4.5) and (4.6) report the Q-matrix estimates for the fraction subtraction data from the DINA model and rRUM, respectively. Columns of the estimated Q-matrices were reordered to match the expert designed Q-matrix. 76% of the estimated Q-matrix entries from the DINA model are consistent with the the expert designed Q-matrix, and 72% from the rRUM. The two estimated Q-matrices seem to be somewhat close to the expert designed Q-matrix. Although most of the items have at least one attribute inconsistent with expert designed Q-matrix, if the cutoff is set to 0.5, items 3, 6 and 12 from the DINA model and items 4, 5, 6, 8 and 14 from the rRUM are consistent with the expert designed Q-matrix. Items 8, 11, 14, 15 from the DINA model, and items 3, 7, 13 from the rRUM have one attribute missed. In brief, 7 items from the DINA model and 8 items from the rRUM miss one or no attribute, when compared with the expert designed Q-matrix.

Examining the attribute level, we can see that attribute 1 has only one miss and attributes 2 and 5 has 2 misses from the rRUM. From the DINA model, attribute 5 has just one miss. Although the two estimated Q-matrices do not agree with each other for most of the items, they both indicated that attributes 3 and 4 are doubtful. It is worth noting that if we took out attributes 3 and 4, then 10 out of the 15 items from the rRUM were consistent with the expert designed Q-matrix.

Convergence was assessed using the Raftery and Lewis diagnostic (Raftery & Lewis, 1992) from the CODA R package (Plummer et al., 2006). The general advice is to pay attention to dependence factors exceeding 5, as it might be due to influential starting values, high correlations, or poor mixing (Gill, 2007). The convergence diagnostic for each entry ϕ_{jk}, the underlying probability of q_{jk}, is shown in Appendix III for the DINA model and in Appendix IV for the rRUM. Overall, convergence was achieved for most of the Q-matrix entries; however 5 entries in the DINA model and 3 entries in the rRUM have dependence factors exceeding 5.

Obtained from the NPCD R package (Zhang & Chiu, 2014), the log-likelihood of the estimated Q-matrix and expert designed Q-matrix for the DINA model are -2280 and -2673; for the rRUM, the log-likelihood of the estimated Q-matrix and expert designed Q-matrix are -2386 and -2748. The estimated Q-matrix from the DINA model appears to fit the fraction-subtraction data better.

All in all, from the simulation studies, both the DINA model and rRUM performed well. In the empirical study using the fraction-subtraction data, the two models suggested two different Q-matrices. Although the Q-matrix estimate from the DINA model is closer to the expert designed Q-matrix, the rRUM yields more accurate items if we take the expert designed Q-matrix as the true Q-matrix.

Table 4.5: Q-matrix for the Fraction-Subtraction Data from the DINA Model

Expert Designed Q-matrix							Estimated Q-matrix				
		Attribute					Attribute				
Item		1	2	3	4	5	1	2	3	4	5
1	$\frac{3}{4}-\frac{3}{8}$	1	0	0	0	0	0.438	1.000	0.000	0.000	0.444
2	$3\frac{1}{2}-2\frac{3}{2}$	1	1	1	1	0	0.102	0.912	1.000	0.166	0.000
3	$\frac{6}{7}-\frac{4}{7}$	1	0	0	0	0	1.000	0.000	0.000	0.000	0.000
4	$3-2\frac{1}{5}$	1	1	1	1	1	0.415	0.999	0.000	0.054	1.000
5	$3\frac{7}{8}-2$	0	0	1	0	0	0.540	0.001	0.044	0.001	0.989
6	$4\frac{4}{12}-2\frac{7}{12}$	1	1	1	1	0	0.679	0.936	1.000	0.884	0.000
7	$4\frac{1}{3}-2\frac{4}{3}$	1	1	1	1	0	0.989	0.355	1.000	0.998	0.000
8	$\frac{11}{8}-\frac{1}{8}$	1	1	0	0	0	1.000	0.000	0.000	0.000	0.000
9	$3\frac{4}{5}-3\frac{2}{5}$	1	0	1	0	0	0.839	0.886	0.942	0.915	0.471
10	$2-\frac{1}{3}$	1	0	1	1	1	1.000	0.000	0.000	0.000	0.000
11	$4\frac{5}{7}-1\frac{4}{7}$	1	0	1	0	0	1.000	0.000	0.000	0.000	0.000
12	$7\frac{3}{5}-\frac{4}{5}$	1	0	1	1	0	0.997	0.008	1.000	1.000	0.000
13	$4\frac{1}{10}-2\frac{8}{10}$	1	1	1	1	0	1.000	0.001	0.000	1.000	0.000
14	$4-1\frac{4}{3}$	1	1	1	1	1	0.875	0.948	1.000	0.138	1.000
15	$4\frac{1}{3}-1\frac{5}{3}$	1	1	1	1	0	0.996	0.015	1.000	1.000	0.000

(1) performing basic fraction-subtraction operation (2) simplifying/reducing (3) separating whole number from fraction (4) borrowing one from whole number to fraction (5) converting whole number to fraction

Table 4.6: Q-matrix for the Fraction-Subtraction Data from the rRUM

Expert Designed Q-matrix							Estimated Q-matrix				
		Attribute					Attribute				
Item		1	2	3	4	5	1	2	3	4	5
1	$\frac{3}{4}-\frac{3}{8}$	1	0	0	0	0	0.835	0.099	0.987	1.000	0.078
2	$3\frac{1}{2}-2\frac{3}{2}$	1	1	1	1	0	0.167	1.000	0.242	0.999	0.391
3	$\frac{6}{7}-\frac{4}{7}$	1	0	0	0	0	1.000	0.044	0.339	0.962	0.091
4	$3-2\frac{1}{5}$	1	1	1	1	1	0.700	0.563	0.829	1.000	1.000
5	$3\frac{7}{8}-2$	0	0	1	0	0	0.299	0.146	1.000	0.109	0.135
6	$4\frac{4}{12}-2\frac{7}{12}$	1	1	1	1	0	0.747	1.000	0.841	1.000	0.212
7	$4\frac{1}{3}-2\frac{4}{3}$	1	1	1	1	0	0.996	1.000	0.118	0.982	0.484
8	$\frac{11}{8}-\frac{1}{8}$	1	1	0	0	0	1.000	0.881	0.224	0.324	0.085
9	$3\frac{4}{5}-3\frac{2}{5}$	1	0	1	0	0	0.840	1.000	1.000	1.000	0.235
10	$2-\frac{1}{3}$	1	0	1	1	1	1.000	0.122	0.155	0.980	0.117
11	$4\frac{5}{7}-1\frac{4}{7}$	1	0	1	0	0	1.000	0.529	0.913	0.724	0.067
12	$7\frac{3}{5}-\frac{4}{5}$	1	0	1	1	0	1.000	1.000	0.770	0.646	0.948
13	$4\frac{1}{10}-2\frac{8}{10}$	1	1	1	1	0	1.000	1.000	0.424	1.000	0.088
14	$4-1\frac{4}{3}$	1	1	1	1	1	0.871	0.984	0.918	0.998	1.000
15	$4\frac{1}{3}-1\frac{5}{3}$	1	1	1	1	0	0.999	1.000	0.376	0.622	0.991

(1) performing basic fraction-subtraction operation (2) simplifying/reducing (3) separating whole number from fraction (4) borrowing one from whole number to fraction (5) converting whole number to fraction

Chapter 5 Discussion

5.1 Summary

Estimating the Q-matrix for cognitive diagnosis models has drawn attention in recent years, and various methods have been proposed (e.g., Barnes, 2003; Chiu, 2013; DeCarlo, 2012; de la Torre, 2008; Henson & Templin, 2006; Liu, Xu, & Ying, 2012; Templin & Henson, 2006; Winters, 2006). DeCarlo (2012) has laid the foundations for Bayesian Q-matrix research. This research applied the Bayesian method in a more exploratory manner. An MCMC algorithm for Bayesian estimation was proposed based on the DINA model and rRUM, and was implemented in base R.

It is not uncommon to have correlated attributes in an exam. Liu, Xu, and Ying (2012) noticed that the more correlated attributes are, the more difficult it is to estimate attributes. The sampling algorithm in this research used a saturated multinomial model to account for correlated attributes in the estimation. In estimating the parameters for the DINA model, closed-form posteriors for the guess and slip parameters were derived. In the rRUM, the random walk Metropolis-Hastings algorithm was applied to parameter estimation. Based on Erosheva and Curtis (2012), an algorithm for dealing with potential label switching was also advanced. The basic concept of this relabeling algorithm is to reorder columns of the estimated Q-matrix at each iteration during a run of MCMC.

The algorithm developed in this research appeared to be feasible in Q-matrix estimation and was accurate in parameter estimation for the DINA model and rRUM. Three simulation studies were conducted to evaluate the algorithm for Bayesian estimation. Results from simulation studies showed that the Q-matrix recovery rate was satisfactory. Sample size, degree of correlation and variability in guess and slip parameters all affected the recovery rate. Recovery rate was higher from a data set with a larger sample size and less correlated attributes.

5.2 Implications for Practice

Although the results of the simulation studies seemed to be promising, the empirical study using the fraction-subtraction data clearly showed that the estimated Q-matrix deviates from the expert designed Q-matrix. Although more than 70% of the estimated Q-matrix entries for the DINA model and rRUM correspond to the expert designed Q-matrix after the estimated Q-matrices are reordered, the meaning of estimated Q-matrices is unclear. We need to exercise caution in their interpretations.

It is always helpful to bring in experts in related fields to better understand the estimated Q-matrix. Nevertheless, area experts might neglect some attributes, and different experts might have different opinions. Therefore, although the estimated Q-matrix may not be appropriate to entirely replace the expert designed Q-matrix, the method can serve to validate existing knowledge about the Q-matrix and provides unnoticed information about the data.

5.3 Limitations and Suggestions

Unlike Liu, Xu, and Ying (2012) , Chen et al. (2013) reported that the recovery rate is higher when attributes are more correlated. In simulation study 3 of this research, although the Q-matrix recovery rate when $\rho = 0.15$ is better than that when $\rho = 0.3$, the difference is not apparent. The magnitude of correlation does not seem to significantly affect the recovery rate. Further research using higher correlated attributes is needed to investigate the issue.

There are some limitations in this research. First, correlation for each pair was fixed for each of the simulations. More complicated correlation structure is needed to examine how the correlation structure affects the Q-matrix recovery. Applying Dirichlet prior to estimating the Q-matrix might be a possible way to better understand the correlation structure among attributes, and this might also make the algorithm more efficient.

Second, only a complete Q-matrix was used in the simulation studies. While it has been established that a complete Q-matrix is sufficient and necessary to consistently identify all attribute patterns (Chen et al., 2013), an exam will not guarantee having a complete Q-matrix; the Q-matrix is usually more complicated in real life. More complicated Q-matrices should be used in simulations in order to examine how the algorithm performs.

Third, this research is based on the DINA model and rRUM. However, how well the two models fit the fraction-subtraction data is questionable, especially when they are compared with IRT models. Moreover, because of the conjunctive nature of the two models that divides examinees only into either mastery or non-mastery category, further research might apply the estimation procedure to more general models, such as the G-DINA model, that can identify the probability of different attribute patterns.

Furthermore, the estimation procedure was not entirely exploratory because the number of attributes in the Q-matrix was assumed to be known in this research. The empirical study assumed the fraction subtraction data has 5 attributes; however, the number of attributes is usually not known in real situations. Using log-likelihood might be able to reveal how the estimated Q-matrix with certain number of attributes fits the data.

In addition, there is a fundamental identifiability problem because we do not know how the attributes in the estimated Q-matrix correspond to true Q-matrix in real data. Even so, relabeling the attributes does not change the model. In estimating the Q-matrix, the data does not contain information about the specific meaning of each attribute; therefore, we are not able to differentiate estimated Q-matrix from true Q-matrix solely based on data if they are identical up to a column permutation (Chen et al., 2013).

A relatively small sample size of the fraction-subtraction data is a also a concern, because some attribute patterns might be too sparse to estimate accurately. Also, the fraction-subtraction data from 536 middle school students might not be very appropriate, as the data is clearly bimodal. It might be worth noting that one data set was generated using the fraction-subtraction Q-matrix from table 3.2. Guess and slip parameters were obtained from de la Torre (2008), correlation between each pair of attributes was set to 0.3, the cut-off point for generating attributes was determined by (3.52), and the sample size was 1000. The algorithm for the DINA model was used to extract the Q-matrix. When the cutoff point was set to 0.5, the estimated Q-matrix had only one miss. However, this was just the result from one data set, more data sets and thorough investigations are needed to make further inferences.

A remark on computation is that although taking logarithm will generally make numerical computations more stable (Patz & Junker, 1999), an error of numerical overflow might still happen. One approach to avoid the issue is rescale the values by subtracting the maximum value before taking exponential (see Gelman, Carlin, Stern, & Rubin, 2004). Estimating the Q-matrix is usually computationally intensive. As Chiu (2013) mentioned, estimating the Q-matrix is complicated and high-quality software is necessary; however such software tends to be proprietary and expensive to obtain. This research used base R to implement the algorithm. It is one of the goals of this research to offer a free R package for estimating CDMs. Along with this research, R programs for the DINA, NIDA models and rRUM have been developed. As the algorithm worked well in R programming environment, it is worth the effort to convert it to other lower-level programming language, such as C or Java, to facilitate its efficiency.

References

Barnes, T.M. (2003). *The Q-matrix Method of Fault-tolerant Teaching in Knowledge Assessment and Data Mining* (Doctoral Dissertation). North Carolina State University.

Brown, J. D., & Hudson, T. (2002). Criterion-referenced language testing. Cambridge: Cambridge University Press.

Chen, Y., Liu, J., Xu G., & Ying, Z. (2013). Statistical Analysis of Q-matrix Based Diagnostic Classification Models. Unpublished Manuscript.

Chiu, C. Y. (2013). Statistical Refinement of the Q-matrix in Cognitive Diagnosis. *Applied Psychological Measurement*, 37(8), 598-618.

Chiu, C. Y., & Douglas, J. A. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. *Journal of Classification* 30(2), 225-250.

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

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.

DeCarlo, L. T. (2012). Recognizing uncertainty in the Q-matrix via a Bayesian extension of the DINA model. *Applied Psychological Measurement*, 36, 447-468.

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

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. (2011). The generalized DINA model framework. *Psychometrika*, 76, 179-199.

Desmarais, M.C. (2011). Mapping Question Items to Skills with Non-negative Matrix Factorization ACM KDD-Explorations, 13(2), p. 30-36.

DiBello, L., Roussos, L. A., & Stout, W. (2007). Review of cognitively diagnostic assessment and a summary of psychometric models. In C. V. Rao & S. Sinharay (Eds.), *Handbook of statistics* (Vol. 26, Psychometrics, pp. 979-1027). Amsterdam, the Netherlands: Elsevier.

DiBello, L. V., Stout, W. F., & Roussos, L. A. (1995). Unified cognitive psychometric assessment likelihood-based classification techniques, chapter Cognitively diagnostic assessment, pages 361-390. Hillsdale, NJ: Erlbaum.

Erosheva, E. A., & Curtis, S. M. (2012, May). *A relabeling procedure for dealing with rotational invariance in Bayesian confirmatory factor analysis*. Paper presented at

Joint Statistical Meetings, San Diego, CA.

Fu, J., & Li, Y. (2007, April). *An integrated review of cognitively diagnostic psychometric models*. Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago, IL.

Gelman, A., Carlin, J. B., Stem, H. S., & Rubin, D. B. (2004). *Bayesian Data Analysis*. London: Chapman & Hall.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741.

Gilks, W.R., S. Richardson, & D.J. Spiegelhalter (eds.). (1996). Markov Chain Monte Carlo in Practice. London: Chapman and Hall.

Gill, J. (2002). *Bayesian Methods: A Social and Behavioral Sciences Approach*. Boca Raton, FL: Chapman & Hall/CRC.

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.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97-109.

Henson, R. A., Douglas J., Stout, W., Roussos, L., & He, X. (2003) Cognitive diagnostic reliability, ETS External Team Technical Report.

Henson, R., & Douglas, J. (2005). Test construction for cognitive diagnosis. *Applied Psychological Measurement*, 29, 262-277.

Henson, R. A., Roussos, L., Douglas, J., & He, X. (2008). Cognitive diagnostic attribute-level discrimination indices. *Applied Psychological Measurement*, 32, 275-288.

Henson, R., & Templin, J. (2006, April). *Bayesian Q-matrix inference using the Reparameterized Unified Model*. Paper presented during the annual meeting of the National Council on Measurement in Education in San Francisco, California.

Henson, R., & Templin, J. (2007, April). *Importance of Q-matrix construction and its effects cognitive diagnosis model results*. Paper presented at the annual meeting of the National Council on Measurement in Education in Chicago, Illinois.

Henson, R.A., Templin, J.L., & Willse, J.T. (2009). Defining a Family of Cognitive Diagnosis Models Using Log-Linear Models with Latent Variables. *Psychometrika*, 74(2):191-210.

Huebner, A. (2010). An overview of recent developments in cognitive diagnostic computer adaptive assessments. *Practical Assessment, Research and Evaluation*, 15(3).

Jang, E. E. (2005). *A validity narrative: Effects of reading skills diagnosis on teaching and learning in the context of NG-TOEFL* (Doctoral Dissertation). University of Illinois, Urbana-Champaign.

Jasra, A., Holmes, C.C., & Stephens D.A. (2005). Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20, 50–67.

Johnson, M. S. (2009). A note on the estimable attribute sets in cognitive diagnostic models. Unpublished Manuscript.

Junker, B. (1999). Some statistical models and computational methods that may be useful for cognitively-relevant assessment. Prepared for the Committee on the Foundations of Assessment, National Research Council, November 30, 1999: Department of Statistics, Carnegie Mellon University, Pittsburgh, PA.

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

Leighton, J.P. & Gierl, M.J. (Eds.). (2007). *Cognitive diagnostic assessment for education. Theory and applications*. Cambridge, MA: Cambridge University Press.

Liu, J., Xu, G., & Ying, Z. (2012). Data-driven learning of Q-matrix. *Applied Psychological Measurement*, 36, 609-618.

Liu, J., Xu, G., & Ying, Z. (2013) Theory of self-learning Q-matrix. *Bernoulli*, 19, 1790-1817.

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, 64, 187–212.

McGlohen, M., Chang, H. & Miller, E. (2004, April). *Joining diagnostic assessment with large-scale standardized testing*. Paper presented at Annual Meeting of American Educational Research Association, San Diego,CA.

Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44, 335-341.

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC, R News, vol 6, 7-11

R Development Core Team. (2013). R: A language and environment for statistical computing [Computer software]. Vienna, Austria: R Foundation for Statistical Computing. Available from http://www.r-project. org

Raftery, A.E. & Lewis, S.M. (1992). One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical Science*, 7, 493-497.

Robert, C., & Casella, G. (2004), *Monte Carlo Statistical Methods*. Springer Texts in Statistics, 2nd ed., Springer-Verlag, New York.

Ross, S. M. (2006), *Simulation*. 4th ed., Academic Press, San Diego.

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

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

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

Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnosis. In N. Frederiksen, R. Glaser, A. Lesgold, & M. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 453-488). Hillsdale, NJ: Erlbaum.

Tatsuoka, K.K., Corter, J., & Tatsuoka, C. (2004). Patterns of Diagnosed mathematical content and process skills in TIMSS-R across a sample of 20 countries. *American Educational Research Journal*, 41 (4):901–926.

Templin, J. (2004). Estimation of the RUM without alpha tilde: a general model for the proficiency space of examinee ability. External Diagnostic Research Group Technical Report.

Templin, J., & Douglas, J. (2004). Higher order RUM. External Diagnostic Research Group Technical Report

Templin, J., & Henson, R. (2006, April). *A Bayesian method for incorporating uncertainty into Q-matrix estimation in skills assessment*. Paper presented at the annual meeting of the National Council on Measurement in Education, San Francisco, CA.

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

Templin, J. L., Henson, R. A., & Douglas, J. (2006). General theory and estimation of cognitive diagnosis models: Using Mplus to derive model estimates. Unpublished manuscript.

Templin, J., Henson, R., Templin, S., & Roussos, L. (2004). Robustness of unidimensional hierarchical modeling of discrete attribute association in cognitive diagnosis models. External Diagnostic Research Group Technical Report.

von Davier, M. (2005). A general diagnostic model applied to language testing data. ETS Research Report: RR-05-16, Educational Testing Service, Princeton, NJ.

Winters, T. (2006). *Educational Data Mining: Collection and Analysis of Score Matrices for Outcomes-Based Assessment* (Doctoral dissertation). University of California, Riverside.

Xiang, R. (2013). *Nonlinear Penalized Estimation of True Q-Matrix in Cognitive Diagnostic Models* (Doctoral dissertation). Columbia University, New York.

Zhang, S. S., DeCarlo. L. T. , Ying, Z. (2013). Non-identifiability, equivalence classes, and attribute-specific classification in Q-matrix based Cognitive Diagnosis Models. arXiv:1303.0426.

Zheng, Y., & Chiu C.-Y. (2014). NPCD: Nonparametric Methods for Cognitive Diagnosis. R package version 1.0-5. http://CRAN.R-project.org/package=NPCD

Appendix I

	Attribute						Attribute				
Item	1	2	3	4	5	Item	1	2	3	4	5
1	0.009	0.017	0.000	0.000	0.000	16	0.054	0.010	0.000	0.000	0.027
2	0.039	0.000	0.001	0.011	0.000	17	0.122	0.000	0.010	0.000	0.010
3	0.022	0.001	0.010	0.000	0.020	18	0.014	0.019	0.010	0.000	0.039
4	0.023	0.02	0.000	0.000	0.029	19	0.105	0.040	0.000	0.000	0.001
5	0.152	0.052	0.013	0.004	0.011	20	0.092	0.063	0.010	0.000	0.013
6	0.008	0.017	0.000	0.000	0.000	21	0.008	0.010	0.010	0.011	0.011
7	0.039	0.000	0.001	0.011	0.010	22	0.003	0.012	0.000	0.000	0.025
8	0.022	0.001	0.010	0.000	0.020	23	0.015	0.000	0.010	0.000	0.010
9	0.023	0.020	0.000	0.000	0.029	24	0.036	0.018	0.010	0.000	0.035
10	0.141	0.050	0.011	0.005	0.010	25	0.050	0.046	0.000	0.003	0.002
11	0.017	0.000	0.001	0.011	0.000	26	0.061	0.059	0.010	0.000	0.011
12	0.004	0.012	0.01	0.000	0.019	27	0.049	0.010	0.010	0.000	0.030
13	0.031	0.020	0.000	0.000	0.027	28	0.086	0.003	0.000	0.009	0.003
14	0.047	0.048	0.01	0.000	0.010	29	0.102	0.011	0.017	0.000	0.011
15	0.049	0.010	0.01	0.011	0.019	30	0.085	0.066	0.000	0.000	0.003

Appendix II

Item	Attribute 1	2	3	4	5	Item	Attribute 1	2	3	4	5
1	0.004	0.031	0.003	0.002	0.001	16	0.020	0.002	0.013	0.002	0.104
2	0.005	0.004	0.011	0.004	0.018	17	0.039	0.003	0.037	0.015	0.014
3	0.020	0.006	0.001	0.001	0.005	18	0.024	0.011	0.001	0.003	0.026
4	0.038	0.039	0.052	0.003	0.010	19	0.037	0.079	0.001	0.038	0.002
5	0.016	0.033	0.039	0.032	0.006	20	0.083	0.054	0.030	0.002	0.003
6	0.004	0.008	0.004	0.005	0.005	21	0.003	0.003	0.001	0.017	0.007
7	0.010	0.003	0.004	0.020	0.004	22	0.004	0.001	0.045	0.003	0.019
8	0.016	0.039	0.001	0.014	0.004	23	0.003	0.002	0.051	0.019	0.003
9	0.062	0.015	0.013	0.001	0.036	24	0.004	0.014	0.001	0.003	0.024
10	0.073	0.065	0.011	0.019	0.003	25	0.012	0.034	0.034	0.017	0.002
11	0.003	0.003	0.032	0.027	0.013	26	0.007	0.028	0.034	0.002	0.003
12	0.003	0.007	0.001	0.084	0.010	27	0.027	0.003	0.001	0.002	0.017
13	0.004	0.121	0.016	0.002	0.038	28	0.092	0.002	0.035	0.026	0.002
14	0.003	0.049	0.090	0.025	0.002	29	0.041	0.004	0.064	0.002	0.002
15	0.007	0.002	0.001	0.010	0.004	30	0.031	0.049	0.001	0.002	0.002

Appendix III

	Dependence Factors for the DINA model				
	Attribute				
Item	1	2	3	4	5
1	1.010	0.999	1.010	0.993	1.010
2	1.010	0.987	1.010	0.994	1.010
3	1.030	2.100	2.150	0.988	1.000
4	1.000	0.995	1.010	0.995	0.995
5	1.000	3.130	1.010	1.000	0.998
6	0.991	11.40	0.999	1.010	2.160
7	1.000	0.993	1.010	0.996	1.000
8	1.000	1.010	3.360	1.000	1.070
9	1.010	1.000	0.996	1.010	1.010
10	0.994	1.010	3.170	0.998	1.000
11	3.030	1.010	5.680	0.998	1.010
12	10.50	0.996	1.000	1.010	1.000
13	2.090	0.988	6.440	1.010	1.990
14	4.500	1.010	8.840	1.010	0.990
15	3.120	0.997	0.996	0.989	1.020

Appendix IV

Item	Dependence Factors for the rRUM				
	Attribute				
	1	2	3	4	5
1	1.010	1.010	1.010	0.991	1.010
2	1.000	1.000	2.160	0.993	1.000
3	0.999	1.010	5.290	1.000	0.993
4	1.010	1.020	1.010	1.000	0.999
5	0.996	1.010	1.000	1.000	0.992
6	0.999	1.010	1.000	0.996	1.010
7	1.010	2.060	1.010	0.996	1.020
8	1.010	0.998	0.994	1.000	0.996
9	4.090	0.999	1.010	1.000	2.160
10	1.010	10.20	1.020	1.020	1.080
11	0.997	0.992	0.998	1.010	0.993
12	1.000	2.200	2.050	1.010	1.050
13	0.999	7.710	3.410	1.020	1.010
14	1.010	2.140	1.030	0.998	1.010
15	0.999	3.180	1.060	1.010	1.000