Nonlinear Penalized Estimation of True Q-Matrix in Cognitive Diagnostic Models

Rui Xiang

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

COLUMBIA UNIVERSITY

2013

ABSTRACT

Nonlinear Penalized Estimation of True Q-Matrix in Cognitive Diagnostic Models

Rui Xiang

Cognitive assessment is a growing area in psychological and educational measurement, where tests are given to assess mastery/deficiency of attributes or skills. A key issue is the correct identification of attributes associated with items, in other words, the correct specification of item-attribute relationships. A widely used mathematical formulation is the well known Q-matrix introduced by Tatsuoka in 1983. The so-called Q-matrix is a J by K binary matrix establishing the relationship between responses and attributes by indicating the required attributes for each item. The entry in the j-th row and k-th column indicates if item j requires attribute k. Previous statistical analyses with such models typically assume a known Q-matrix provided by domain experts such as those who developed the questions. However, if the Q-matrix is not specified appropriately, it could seriously affect the model goodness of fit. Unfortunately, the estimation of Q-matrices is largely an unexplored area. As a result, the primary purpose of this research is to set up a mathematical framework to estimate the true Q-matrix based on item response data. The research also evaluates the method through simulation studies, and applies it to estimate Q from real item response data. However, as the optimization approaches are not common for discrete values, a probabilistic model with a penalized likelihood function is built. The model considers all the Q-matrix elements as parameters and estimates them through EM algorithm. However, as the estimates are continuous values between 0 and 1, cut-off points are used to transfer them to binary values. Two simulation designs are conducted to evaluate the feasibility and performance of the model. An empirical study is also addressed here to estimate the true Q-matrix from a secondary data of fraction subtraction item responses. The estimated Q-matrix is then compared with the one originally designed by test developers. The results conclude that our model performs well and is able to identify 60% to 90% of correct elements of Q-matrix. The model also indicates possible misspecifications of the designed Q-matrix in the fraction subtraction test.

Keywords: Q-matrix, cognitive diagnostic models, nonlinear estimations, penalized approach

Table of Contents

List of Figures

List of Tables

Acknowledgments

From the formative stages of this thesis, to the final draft, I owe an immense debt of gratitude to my supervisor, Dr. Matthew S. Johnson. His sound advice and careful guidance were invaluable as I attempted to examine the possibilities of pure Q-matrix estimation in cognitive diagnostic models.

Deepest gratitude are also due to the members of the supervisory committee, Dr. Lawrence T. DeCarlo, Dr. Young-Sun Lee, Dr. Hsu-Min Chiang, and Dr. Ken Cheung, without whose knowledge and assistance this study would not have been successful.

I would also like to thank my graduate friends, Mr. Meng-ta Chung, Mr. Jianzhou Zhang, Ms. Nan Jiang, Ms. Rong Cheng, Ms. Yunting Xiao, Mr. Huacheng Li and Mr. Jon-Paul Paolino, for sharing the literature, R programs and invaluable assistance, without your time and kind help time, this research would not be completed on time.

For their efforts and assistance, a special thanks as well to the staffs of the department office of Human Development, especially Ms. Diane V. Katanik, Ms. Laurie Behrman and Ms. Stephanie Phillips for accommodating everything.

Finally, I would be remiss without mentioning my respective parents who have given me the drive and discipline to tackle any task with enthusiasm and determination. And my Christian mentor, Dr. Ada C. Mui, whose sage instructions and patient encouragement will be remembered always.

This thesis is dedicated to my family who have been constant source of inspiration

Chapter 1

Introduction

Cognitive diagnostic modeling (CDM), which intends to diagnose subjects' mastery status of a group of discretely defined skills or attributes, has become a growing field of psychometric research over the past several years. The reason why CDM is so important largely accounts for the call for more formative assessments made by the No Child Left Behind Act of 2001, and this new psychometric model can provide students with more detailed information regarding their specific strengths and weaknesses [Huebner, 2010]. Rather than assigning a score on a continuous scale to the students representing a broadly defined latent ability, CDM attempts to assess in detail whether an examinee has mastered a group of specific skills or not, and these required skills in the test substitute a whole latent ability in a common item response theory (IRT) or Classical Test Theory (CTT) model. For example, a test of subtraction fraction may include the skills of 1) converting a whole number to a fraction, 2) separating a whole number from a fraction, 3) simplifying before subtracting, and so forth [de la Torre and Douglas, 2004]; and a reading test may require the attributes of 1) remembering details, 2) knowing fact from opinion, 3) speculating from contextual clues, and so on [McGlohen and Chang, 2008]. Thus, if people are more interested in knowing the mastery of skills by the students and design a skill-based test, CDM is superior to be used over traditional psychometric approaches such as IRT on skill assessments [Henson et al., 2009]. CDM may also potentially aid teachers to direct students to more individualized remediation and focus on the specific weaknesses [Huebner, 2010].

In CDM, the definition of skill mastery is usually binary, indicating that students are identified as masters or non-masters of each skill [Huebner, 2010]. A correct response on an item depends on mastery of multiple skills that required by the item. Thus, CDM assigns to each subject a vector of binary valued (0/1) $\vec{\alpha} = (\alpha_1, \ldots, \alpha_K)$ for K skills that are assessed in the test, where 1 denotes mastery of the skill and 0 denotes non-mastery. The total 2^K possible attribute patterns are referred to as the latent classes. Therefore, failing in a test is due to some skills that have not been mastered by a student, or the latent attribute pattern the student holds does not cover all the necessary skills that are required by the items. In sum, CDMs provide diagnostic information about the mastery/non-mastery of specific skills by mastery pattern [Henson et al., 2009], often by modeling probability of correctly answering an item as a function of an attribute mastery pattern [Henson and Douglas, 2005].

CDMs have been developed into various types of models, with additional parameters such as slipping or guessing, or different model assumptions, and widely used in different areas of educational measurement, especially standardized large-scale tests of educational skills. Liu et al. [2011b] had listed a short list of popular CDMs, including rule space method [Tatsuoka, 1983, 2009, 1985, 1990], the reparameterized unified/fusion model (RUM) [DiBello et al., 1995; Hartz, 2002; Templin et al., 2003; Roussos et al., 2007a], the conjunctive (noncompensatory) DINA and NIDA models [de la Torre and Douglas, 2004; de la Torre, 2011; Junker and Sijtsma, 2001; Templin, 2006; Maris, 1999], the disjunctive (compensatory) DINO and NIDO models [Templin and Henson, 2006; Templin, 2006], the attribute hierarchy method [Leighton et al., 2004], and clustering methods [Chiu et al., 2009].

Tatsuoka [1983, 2009, 1985, 1990] developed the rule space methodology to address the problems associated with diagnosis of mastery of underlying dimensions, or attributes, of an item. But attributes are not generated by rule space but by a domain expert. The statistical idea is to classify students' responses to a set of items into one (or more) prespecified attribute-mastery patterns. Within rule space, specialized functions, called Boolean Description Functions (BDF), are used systematically to determine the knowledge states of interest and to map them into ideal item-response patterns. Rule space then plots the ideal item-response patterns in terms of two variables: θ and ζ, which are the ability continuum derived from the IRT theory. Im and Corter [2011] had indicated that Rule Space Model was seen as one of the most viable alternatives to the traditional unidimensional models of IRT. The College Board has used the model to report diagnostic, or scores broken down into relatively specific achievement areas and cognitive subskills, with SAT scores.

The RUM is another effective psychometric and statistical approach for practical skills diagnostic testing. It is designed to provide practical tools for standardized testing that can include effective skills diagnosis in testing. The idea of the model is to consider both skills-based item parameters and skills-based examinee parameters, with additional parameters to improve the fit of the model to the data [Roussos et al., 2007a]. The initial research of the foundational modeling work was conducted by DiBello et al. [1995], and subsequentially developed by Hartz [2002] and Templin et al. [2003]. Roussos et al. [2007b] applied a Bayesian version of the RUM to the math section of American College Testing (ACT) assessment about the skills involved in successfully answering the math items of the test. Hartz [2002] applied RUM to the 60-item PSAT in order to inform students of the skills they should master prior to taking the SAT. The study conducted by Templin in his 2006 NCME workshop was using data from the Trends in International Math and Science Study (TIMSS) and compared skill mastery between countries.

In addition, it is important to note that CDMs have been classified as either conjunctive or disjunctive, or similarly, compensatory or non-compensatory. Normally conjunctive comes with non-compensatory and disjunctive is interchangeable to compensatory. Models are conjunctive (non-compensatory) if all the required attributes are necessary for successful completion of the item, and models are disjunctive (compensatory) if the absence of one attribute can be made up for by the presence of other attributes [de la Torre, 2009a]. The RUM method can be designed as either compensatory or non-compensatory. The deterministic inputs, noisy "and" gate (DINA) model [Junker and Sijtsma, 2001] is another example of a conjunctive (non-compensatory) model. It is determined by a latent response ξ_{ij}, a slipping parameter s_j and a guessing parameter g_j, on the ith persons and jth tasks. DINA has particularly enjoyed much attention due to its simplicity of estimation and interpretation [Huebner, 2010]. Another stochastic conjunctive model is the noisy inputs, deterministic, "and" gate (NIDA) model which was introduced by Maris [1999]. The difference between NIDA and DINA is that DINA has item-level parameters but NIDA has attribute-level parameters [de la Torre, 2009a], in other words, NIDA includes one more item-attribute information in the determination of the three parameters in DINA. Both models had been used in past studies for simulation research and real data analysis such as the fraction subtraction data done by de la Torre and Douglas [2004].

The deterministic input, noisy "or" gate model (DINO) is a disjunctive (compensatory) model developed by Templin and Henson [2006] and has been used to diagnose pathological gambling. The DINO is defined in a similar manner as DINA but now the equivalence latent deterministic aspect of the model is based on a disjunctive factor ω_{ij} which divides individuals into a group that satisfied at least one necessitated criterion and a group that have not satisfied any necessitated criteria. Another disjunctive model, the noisy input, deterministic "or" gate model (NIDO) is the compensatory analog of the NIDA model. The model specifies two parameters per attribute, one representing examinees who have mastered the attribute and the other representing examinees who are lacking mastery the attribute [Templin, 2006].

The attribute hierarchy method (AHM) was introduced by Leighton et al. [2004] and represented a variation of Tatsuoka's rule space method. The assumption is that attributes are organized in a hierarchical way to form a cognitive model for task performance. The model was applied to the domain of syllogistic reasoning to evaluate the cognitive competencies required in a higher-level thinking task. Besides AHM, cluster analysis [Chiu et al., 2009] provides an alternative way to cluster subjects who posses the same skills into one group by K-means or hierarchical agglomerative clustering without an item response model. The English language skills were assessed by Chiu et al. [2009] using the clustering methods from the Examination for the Certificate of Proficiency in English (ECPE) conducted by the University of Michigan English Language Institute.

However, despite the vast majority of CDMs, most of the models need a way to demonstrate an item-by-attribute relationship, thus utilizing a relationship mapping matrix referred to as a Q-matrix first introduced by Tatsuoka [Tatsuoka, 1985]. The Q-matrix is an efficient way to represent specific skills that are needed to answer each item correctly. Under the setting of Q-matrix, J items (tasks) are measuring the K attributes, so the Q-matrix is a $J \times K$ binary matrix $(q_{jk})_{J\times K}$ with elements 0 and 1 indicating whether the jth item requires the kth attribute or not, where $j = 1, \ldots, J$ and $k = 1, \ldots, K$. Each Q-matrix element q_{jk} is then used in the construction of

CDMs as the most important information factor.

The analysis of CDMs normally assumes a known Q-matrix and the development of the Q-matrix becomes one of the most important steps of CDMs. The basic methods of Q-matrix construction include simple inspection of the items; multiple rater methods; and iterative procedures based on item parameters [Henson, 2009]. Simple inspection means that the domain experts evaluate the items and determine which attributes are required by each one. Multiple-raters is more likely to be used where several experts and researchers are working together on the determination of Q-matrix. The last method is the refinement based on item parameters, which is done after model fit and not normally used. However, even though so much care has been placed in determining an initial Q-matrix, it is still possible that the matrix is incorrectly identified [Henson, 2009]. Because the results of CDMs and model fits are very sensitive to the construction of Q-matrix, if a prior Q-matrix provided by experts is identified correctly, it will surely very helpful to the model estimation and identification of latent attributes, but a misspecified Q-matrix could seriously affect the goodness of fit of the model and the results will not be trustable [Liu et al., 2011b].

Due to the concern, some studies have been conducted to examine the statistical consequences of misspecification of attributes in Q-matrix. For example, Rupp and Templin [2008] did a study on the effects of Q-matrix misspecification on parameter estimates and classification accuracy in DINA model by changing one ”0” or ”1” for each item in an assessment. Results indicated high overestimation of slipping and guessing parameters and misclassification for attribute classes on students. Thus how to construct a correct specification of Q-matrix is becoming an important issue in CDMs.

Statistically, a good way to estimate Q-matrix is based on empirical data rather than subjective judgments of experts. However, the estimation problem is largely an unexplored area [Liu et al., 2011b]. The only paper found was done by Liu et al. [2011b,a] presenting the estimation procedures for Q-matrix in DINA and DINO models. The reason why people has not done much about this area is possibly that the inference and estimation of the Q-matrix are very challenging. First of all the Q-matrix is on a discrete space with binary elements of 0 and 1. Estimation for discrete variables increases computation complexity because calculus tools are not applicable. Secondly, the Q-matrix is latent and nonidentifiable [Liu et al., 2011b]. It is entirely possible that multiple Q-matrices lead to an identical response distribution, which means two Q-matrices from the same equivalence class may not be distinguishable based on data. Last but not least, CDMs make assumptions on the distributions of unobserved latent attributes. The responses of items based on attributes via Q-matrix created a highly nonlinear discrete linking function. The nonlinearity of linking function also adds to the difficulty of the estimation [Liu et al., 2011b].

Considering these difficulties, Liu et al. [2011b,a] defined an estimator of the Q-matrix and talked about regularity conditions under which desirable theoretical properties were established. They continued to complete the estimation of the Q-matrix under the DINA model specification with known and unkonwn slipping and guessing parameters, and extended the estimation procedure along with the consistency results to the DINO model. Their research provided an estimation procedure on the Q-matrix with sufficient conditions under which a consistent estimator exists, and a parallel analysis for the DINA and DINO model. Liu et al. [2011a] also stated that their estimation procedure was able to be implemented to NIDA and NIDO models, with modifications on the theoretical properties under such model specifications.

Despite the contribution done by Liu et al. [2011b,a, 2012], their theoretical methods required a lot of assumptions to prove the theories of their estimation results with a pre-defined criteria. These assumptions had to be made due to the difficulties of discrete and nonlinear estimation. For example, the Q-matrix needs to be complete which means each attribute there exists an item only requiring that attribute, and a nonlinear transformation of Q-matrix should be saturated which means the transformed matrix has to contain all combinations of positive responses to items. However, it is not always the case in real situations and their methods are hard to apply into real data analysis. This dissertation is to establish an alternative way, hopefully a better way, to estimate Q-matrix based on fewer assumptions. The fundamental difference between this dissertation and the previous literature is considering the Q-matrix elements as probabilities of requiring an attribute by an item, and estimate the Q-matrix on a continuous space. The procedure becomes much easier and multiple ways could be used to estimate the continuous latent variables. Numerical methods for unconstrained optimization and nonlinear equations, and expectation-maximization (EM) algorithm which is an iterative method for finding maximum likelihood estimates, are good options of methods for the estimation. Moreover, a penalized technical can help build restrictions and push the matrix elements asymptotically to 0 or 1 as close as possible for the further recovery back to the discrete Q-matrix.

In sum, this dissertation is going to establish a purely exploratory method to estimate the whole Q-matrix from item response data. The primary research questions include whether the methods are able to estimate the Q-matrix, how the method performs on estimation accuracy, and what the differences would be between estimated Q-matrix and designed Q-matrix in a real situation. The secondary research questions seek to find out if estimated Q-matrices are in fact identical to true or designed Q-matrices, and how their latent class sizes are distributed. The primary research questions can be answered by the proportion of correctly identified elements of estimated Q-matrices in simulation studies, and the proportion of identical elements between estimated Q-matrix and the designed Q-matrix in an empirical study. The secondary research questions can be answered by the comparison of unique ideal item response patterns between Q-matrices, and the comparison of estimated results on latent class sizes. The researcher have constructed multiple evaluation criteria based on data mining theories for a precise and robust comparison. The second chapter is a literature review on the past Q-matrix related issues, the third one is the detailed method descriptions developed to estimate the Q-matrix, including the designs of both simulation and empirical studies. Chapter four shows the results from all types of studies and the last chapter will be the discussion on the strength, weakness and future development of the research.

Chapter 2

Literature review

This chapter reviews the development of Q-matrix in CDMs in more detail. Although most of current studies are concerned about CDM applications and Q-matrix misspecifications, some useful information regrading why Q-matrix estimation is necessary and how to possibly solve the estimation problems could still be found. The first part of the section is about attribute space and Q-matrix defined in CDMs, then some popular CDMs based on attributes and Q-matrix will be discussed. The next topic is a review on the diagnostics on misspecification of Q-matrices in CDMs, followed by the last part of possible ways that could estimate Q-matrix done by other researchers.

2.1 Attribute space and Q-matrix

The analysis of most CDMs is based on an item-attribute incidence matrix called a Q-matrix [Tatsuoka, 1983]. The diagnostic power of CDMs relies on the construction of a Q-matrix with attributes that is theoretically appropriate and empirically supported [Lee and Sawaki, 2009]. Apparently the quality of final inference results from the CDMs is heavily influenced by how the attributes and Q-matrix are defined. How to specify attributes and Q-matrix becomes the most fundamental step in CDMs, and how to define attribute space comes before the establishment of Q-matrix. It is usually subjective to choose attributes that represent the latent cognitive process being assessed. Sometimes the target attributes we are interested in for student evaluation can serve as the basis to develop the attribute space and Q-matrix. Normally content or domain experts are responsible for this step. Nevertheless, the attributes and Q-matrix identified by experts are not guaranteed to be true or most appropriate. Systematic research efforts have not been done enough to investigate the appropriateness of cognitive attributes identified in the context of CDMs.

When developing the attribute space, it is also important to be aware of whether the attributes interact with each other, such as a correlation or a nature of interactions [DiBello et al., 2007]. This case leads us to the issue of conjunctive (non-compensatory) versus disjunctive (compensatory). Conjunctive attribute space requires all necessary attributes of an item to perform it correctly and lack of any one would lead to a failure. Compensatory interaction of attributes might have a chance that a high enough level of competence on one skill can compensate for a low level of competence on another skill and results in successful task performance. These different assumptions of attribute interactions will lead to different types of cognitive diagnostic models based on even the same Q-matrices.

An example of attributes defined in a cognitive assessment test is the mathematical contents in the fraction subtraction data being used in some previous studies [de la Torre and Douglas, 2004; de la Torre, 2009b; DeCarlo, 2011]. The data consisted of responses to 40 items involving subtraction of fractions by 536 examinees firstly used and described by Tatsuoka [1990]. The eight attributes required were: (A1) Convert a whole number to a fraction, (A2) Separate a whole number from a fraction, (A3) Simplify before subtracting, (A4) Find a common denominator, (A5) Borrow from

Table 2.1: A simple Q-matrix example

Item	A1	A2	A3	A4	A5	A6	A7	A8
$\frac{3}{4}-\frac{3}{8}$	0	0	0	1	0	0	1	0
$\frac{6}{7}-\frac{4}{7}$	0	0	0	0	0	0	1	0
$3\frac{1}{2}-2\frac{3}{2}$	0	1	1	0	1	0	1	0
$3-2\frac{1}{5}$	1	1	0	0	0	0	1	0

whole number part, (A6) Column borrow to subtract the second numerator from the first, (A7) Subtract numerators, and (A8) Reduce answers to simplest form.

After the attributes are identified, the Q-matrix specifies which attributes are needed to solve each item [Tatsuoka, 1983, 1990]. The $J \times K$ Q-matrix $(q_{jk})_{J\times K}$ of zeros and ones indicates whether the jth item requires the kth attribute or not, where there are J items and K attributes. If $q_{jk} = 1$ then the jth item at least needs the knowledge of kth attribute to answer the question correctly. If the item does not need the knowledge of kth attribute, then $q_{jk} = 0$. An example of a small part of the Q-matrix of the fraction subtraction items designed by previous experts is shown as an example in the Table 2.1 [de la Torre and Douglas, 2004]. From the Q-matrix example, we see that in order to solve the math problem of $\frac{6}{7} - \frac{4}{7}$, the students are considered to master the skill of (A7) subtract numerators; to solve the problem of $3\frac{1}{2} - 2\frac{3}{2}$, students are supposed to master multiple skills of (A2) separate a whole number from a fraction, (A3) simplify before subtracting, (A5) borrow from whole number part, and (A7) subtract numerators. As long as the attributes and Q-matrix are established, cognitive diagnostic models are able to be developed.

2.2 Current cognitive diagnostic models

After the attribute space and Q-matrix are established, the next step is attempting to discover the latent attributes the examinees possess from the test items. Cognitive diagnostic models have been developed for this purpose by researchers in the past several years [DeCarlo, 2011]. One important difference among the models is based on whether the proficiency variables are discrete or continuous depending on the purpose of the assessment, and another distinction lies on the attribute interaction manner [DiBello et al., 2007]. However, fundamentally all the function of CDMs specifies the probability of a particular correct item response with the attribute pattern of the subject and also the item characteristics. Here the discussion will concentrate on these models with an explicit Q matrix: DINA, NIDA, DINO and NIDO models, due to their simplification and fewer assumptions. DINA and NIDA are based on conjunctive attribute space, while DINO and NIDO are based on compensatory attribute space.

2.2.1 DINA model

The deterministic input, noisy "and" gate (DINA) model [Junker and Sijtsma, 2001] is considered the foundation of the other three models and some other CDMs, and it is also one of the least complex models [Rupp and Templin, 2008]. In DINA model, the probability of responding to an item correctly is determined by two error probabilities and one latent response variable. The guessing probability (g_j)represents the probability of getting a correct response on the jth item when at least one required attribute is lacking. The slipping probability (s_j) represents the probability of getting a wrong response on the jth item when all required attributes are present. The latent dichotomous variable ξ_{ij} indicates whether the ith respondent possess all required attributes to answer the jth item correctly or not whereas the value "1" in this case and ”0” otherwise (deterministic input) [Rupp and Templin, 2008].

Let X_{ij} be the binary score of respondent i to jth item (1 means correct and 0 means incorrect), q_{jk} represent the element in Q-matrix for item j and attribute k (jth row and kth column), α_{ik} indicate whether ith respondent possess attribute k, and $\vec{\alpha}_{j\cdot} = (\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jK})'$ denote the vector of total K skills that are needed to solve the jth item.

$$P(X_{ij} = 1|\xi_{ij}) = (1 - s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}} \tag{2.1}$$

where

$$\xi_{ij} = \prod_{k:q_{jk}=1} \alpha_{ik} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}} \tag{2.2}$$

indicating whether the ith respondent has all the attributes required for the jth item. The latent vector $\vec{\alpha}_{i\cdot} = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK})$ are called *knowledge states*, and the vectors $\vec{\xi}_{i\cdot} = (\xi_{i1}, \xi_{i2}, \ldots, \xi_{iJ})$ are called *ideal response patterns* [Junker and Sijtsma, 2001]. The above represents the deterministic input part of the model that indicates a deterministic prediction of task performance from each respondent's knowledge state.

And

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

$$g_j = P(X_{ij} = 1|\xi_{ij} = 0) \tag{2.4}$$

where s_j and g_j are error probabilities: false negative (slipping) and false positive rates (guessing).

Each ξ_{ij} acts as an ”and” gate with the deterministic inputs $\alpha_{ik}^{q_{jk}}$, and each X_{ij} is modeled as a noisy observation of each ξ_{ij} [Junker and Sijtsma, 2001]. The final item response function of DINA model is

$$P(X_{ij} = x_{ij}, \forall i, j|\xi, s, g) = \prod_{i=1}^{N} \prod_{j=1}^{J} [(1 - s_j)^{x_{ij}} s_j^{1-x_{ij}}]^{\xi_{ij}} [g_j^{x_{ij}} (1 - g_j)^{1-x_{ij}}]^{1-\xi_{ij}} \tag{2.5}$$

Where $x_{ij} = 1$ or 0.

The DINA model has been fit within a fully Bayesian framework using Markov chain Monte Carlo (MCMC) methods or maximum-likelihood estimation (MLE) [DeCarlo, 2011]. In fact the following models are using the same or similar methods for parameter estimation. Reparameterized DINA and higher order DINA model are further developments of the DINA model.

2.2.2 NIDA model

The noisy inputs, deterministic "and" gate model (NIDA) was first discussed by Maris [1999]. Unlike DINA model, the slips and guessing in NIDA model happen at the attribute level instead of the item level. $\eta_{ijk} = 1$ or 0 is defined as whether the ith respondent's performance on the jth item is consistent with possessing attribute k. Thus η_{ijk} is related to the ith respondent's attribute space α_i.

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

$$g_k = P(\eta_{ijk} = 1|\alpha_{ik} = 0, Q_{jk} = 1) \tag{2.7}$$

and

$$P(\eta_{ijk} = 1|\alpha_{ik} = a, Q_{jk} = 0) = 1 \tag{2.8}$$

regardless of the value a (0 or 1) of α_{ik}. Observed item performance is related to the latent response variable η_{ijk} through

$$X_{ij} = \prod_{k:q_{jk}=1} \eta_{ijk} = \prod_{k=1}^{K} \eta_{ijk} \tag{2.9}$$

So the item response function is

$$P(X_{ij} = 1|\alpha, s, g) = \prod_{k=1}^{K} P(\eta_{ijk} = 1|\alpha_{ik}, q_{jk}) = \prod_{k=1}^{K} [(1 - s_k)^{\alpha_{ik}} g_k^{1-\alpha_{ik}}]^{q_{jk}} = \prod_{k=1}^{K} (\frac{1 - s_k}{g_k})^{\alpha_{ik} q_{jk}} \prod_{k=1}^{K} g_k^{q_{jk}} \tag{2.10}$$

The noisy inputs η_{ijk} which show the attributes α_{ik} in respondents are combined in a deterministic ”and” gate X_{ij} [Junker and Sijtsma, 2001]. The joint item response function is

$$P(X_{ij} = x_{ij}, \forall i, j|\alpha, s, g) = \prod_{i=1}^{N} \prod_{j=1}^{J} \{\prod_{k=1}^{K} [(1 - s_k)^{\alpha_{ik}} g_k^{1-\alpha_{ik}}]^{q_{jk}}\}^{x_{ij}} \{1 - \prod_{k=1}^{K} [(1 - s_k)^{\alpha_{ik}} g_k^{1-\alpha_{ik}}]^{q_{jk}}\}^{1-x_{ij}} \tag{2.11}$$

2.2.3 DINO model

The deterministic, noisy ”or” gate model is the compensatory analog of the DINA model and is defined in a similar way [Templin, 2006]. The latent ”or” gate now is determined by a binary disjunctive model ω_{ij} instead of ξ_{ij}.

$$\omega_{ij} = 1 - \prod_{k=1}^{K} (1 - \alpha_{ik})^{q_{jk}} \tag{2.12}$$

If $\omega_{ij} = 1$ then the ith individual has satisfied at least one Q-matrix necessitated attribute of the jth item. If $\omega_{ij} = 0$ then the ith individual has not occupied any necessitated attribute needed for the jth item [Templin and Henson, 2006]. Thus the probability of a positive response based on ω_{ij} will be:

$$P(X_{ij} = 1|\omega_{ij}) = (1 - s_j)^{\omega_{ij}} g_j^{1-\omega_{ij}} \tag{2.13}$$

where $s_j = P(X_{ij} = 0|\omega_{ij} = 1)$ is the slipping parameter and $g_j = P(X_{ij} = 1|\omega_{ij} = 0)$ is the guessing parameter, $1 - s_j > g_j$. Compared with the DINA model, the item response function will be almost the same expect the substitution of ω_{ij} on ξ_{ij}. However, the meanings of slipping and guessing are slightly different. In DINO model, slipping is the probability of failure on the item although examinees have mastered one or more specified attributes for the item (In DINA the examinees have to master all the needed attributes), while guessing is the probability of getting the item right when examinees are lacking mastery of every of the Q-matrix specified attributes for the item (In DINA the examinees lack at least one of the specified attributes) [Templin, 2006].

2.2.4 NIDO model

The NIDO model (Noisy Inputs, Deterministic "or" gate) is the compensatory analog of the NIDA model for cognitive diagnosis [Templin, 2006]. The model construction has different notations and definitions on slipping and guessing. Two parameters per attribute are specified in the model, one representing examinees who have mastered the attribute (called the beta parameter) and one representing examinees who are lacking mastery the attribute (called the tau parameter). The set of attribute parameters for the NIDO model are the same for each item (item discrimination is equal for all items) [Templin, 2006]. The NIDO model has not been widely used in the past literatures.

$$P(X_{ij} = 1|\alpha_{ij}) = [1 + \exp(\sum_{k=1}^{K} (\tau_k + \beta_k \alpha_{ij}) q_{jk})]^{-1} \tag{2.14}$$

The above four models are the basis for more complicated cognitive diagnostic models such as higher order, hierarchical, or reparameterized models. After the model is implemented, the check on goodness of fit is the next important step. which is particularly related back to the specification of Q-matrix. However, the Q-matrix is usually determined by expert judgments so that there might be some mistakes, or uncertainty about its elements. Fortunately people have realized this problem and some researches have been conducted to detect the uncertainty and misspecification of Q-matrix.

2.3 Diagnostics on misidentification of Q-matrix

Rupp and Templin [2008] researched Q-matrix misspecification and its effects on item parameter estimates and respondent classification accuracy for the DINA model. A Q-matrix for an assessment with 15 possible attribute patterns based on four independent attributes was misspecified by changing one "0" or "1" for each item. As a result, certain attribute combinations were completely deleted from the Q-matrix, and certain incorrect dependency relationships between attributes were represented. Their results showed clear evidences that included an item specific overestimation of slipping parameters when attributes were deleted from the Q-matrix, an item-specific overestimation of guessing parameters when attributes were added to the Q-matrix, and high misclassification rates for attribute classes that contained attribute combinations that were deleted from the Q-matrix.

Im and Corter [2011] investigated the statistical consequences of attribute misspecification in the rule space method for cognitively diagnostic measurement. The two types of attribute misspecifications were exclusion of an essential attribute (which affected problem-solving performance) and inclusion of a superfluous attribute (which did not). Their results showed that exclusion of an essential attribute tended to lead to underestimation of examinees' mastery probabilities for the remaining attributes, whereas inclusion of a superfluous attribute generally led to overestimation of attribute mastery probabilities for the other attributes. In addition, order relations among attributes induced by superset/subset relationships affected the biases in the estimated attribute mastery probabilities in systematic ways. These results underscored the importance of correct attribute specification in cognitively diagnostic assessment and delineate some specific effects of using incorrect attribute sets.

DeCarlo [2011] applied DINA model in the fraction subtraction data and revealed some problems on the classification of respondents. For example, examinees who got all of the items incorrect were classified as having most of the skills. Some respondents were classified as having a higher level skill but not having a lower level skill. Again some of the latent class sizes of the attributes were very large. Obtaining large estimates of the latent class sizes can indicate misspecification of the Q-matrix, such as the inclusion of an irrelevant skill. Analytical studies and simulations were able to find out these problems that largely associated with the structure of Q-matrix.

Another approach to check the Q-matrix appropriateness was stated by de la Torre [2008]. It proposed an empirical based method of validating a Q-matrix used in the DINA model by minimizing the sum of the average slip and guess parameters. The correct row vector (item vector) of q_j in Q-matrix is based on

$$q_j = \arg\max_{\alpha}[1 - s_j - g_j] = \arg\max_{\alpha}[\delta_j] \tag{2.15}$$

where $\delta_j = 1 - s_j - g_j$. This sequential EM-Based δ-Method intended to improve model-data fit by selecting the optimal q vectors. But there are still some potential problems such as the slipping and guessing parameters are assumed to be known.

Since misspecification of Q-matrix has became a huge problem on cognitive diagnostic models, it is important to find out a way to get an accurate specification of Q-matrix before fitting the model. However, besides expert judgments, few studies have been done on empirical estimation on Q-matrix. Liu et al. [2011b] have implemented a theoretical analysis on the learnability of the underlying Q-matrix which was the milestone literature on the estimation issue to my perspective. Some other possible methods will also be discussed in the next topic.

2.4 Possible Q-matrix estimation methods

Theoretically the uncertainty of Q-matrix can be recognized by using a Bayesian approach where some elements of the Q-matrix are specified as being randomly binomial distributed but not all elements are missing. The posterior distribution of a hyperparameter can then be used to obtain information about each element [DeCarlo, 2011, 2012]. However, the method requires a large number of hyperparameters to be specified in the model. Another approach is to consider a bunch of possible Q-matrices with each fitted an associated model. When the models are fitted, we compare the indices of relative goodness of fit, such as the Bayesian information criterion (BIC) or Akaike information (AIC)[Rupp and Templin, 2008; DeCarlo, 2012; Cen et al., 2005; de la Torre and Douglas, 2008]. However it is hard to decide the number and elements of possible Q-matrices used for comparison at the very beginning, and it still requires some of the Q-matrix elements are already known.

Tiffany Barnes [Romero et al., 2011] talked about a computation Q-matrix algorithm in her paper *Novel Derivation and Application of Skill Matrices: The Q-Matrix Method* selected by the Book *Handbook of Educational Data Mining* to extract skill matrices from student problem-solving data and use these derived skill matrices in novel ways to automatically assess, understand, and correct student knowledge. The algorithm which is called ”hill-climbing” algorithm creates a matrix representing relationships between concepts and questions directly from student response data by minimizing the total error for all students among the number of attributes, values of Q-matrix, and the answers to all questions. The algorithm is a nice try on Q-matrix establishment from a computer programming perspective, and provides an idea that the estimation can be based on a selection criteria such as the total error. A detailed description of her algorithm is stated below.

The algorithm first sets the number of concepts to one and then generates a random Q-matrix, then calculates the ideal response vector (IDR) and compares it to each student response, and assigns the response to the closest IDR and concept state, with an "error" being the distance from the response to the IDR. The total Q-matrix error is the sum of these errors over all students. Then hill-climbing is performed by adding or subtracting a small fixed delta to a single Q-matrix value, and recomputing its error. If the overall Q-matrix error is improved, the change is saved. This process is repeated for all the values in the Q-matrix several times until the error in the q-matrix is not changing significantly. After a Q-matrix is computed in this fashion, the algorithm is run again with a new random initial Q-matrix several times, and the Q-matrix with minimum error is saved. To determine the best number of skills or attributes to use in the Q-matrix, this algorithm is repeated for increasing the number of attributes, until a stopping criterion is met: either when the Q-matrix error falls below a pre-set threshold, such as that of less than 1 per student as used here, or by looking for a decrease in the marginal reduction of error by adding more concepts.

The research conducted by Liu et al. [2011b] is the pioneer study on the empirical estimation of Q-matrix based on response data. Their idea is similar to Barnes' that minimize a criteria of total error. They introduce a central quantity the T-matrix which connected the Q-matrix with the response and attribute distributions. The non-linear transformation matrix $T(Q)$ has $2^K - 1$ (total K attributes) columns each of which corresponds to one nonzero attribute vector $\alpha \in \{0,1\}^K \setminus \{(0,\ldots,0)\}$. Each row of $T(Q)$ corresponds to positive response on one item or one "and" combination of items. Here Let I_j be a generic notation for positive responses to item j, and let "$\wedge$" stand for "and" combination so that $I_{j_1} \wedge I_{j_2}$ denotes positive responses to both items j_1 and j_2. Then the rows of $T(Q)$ stand for I_{j_1},$I_{j_1} \wedge I_{j_2}$,or $I_{j_1} \wedge I_{j_2} \wedge I_{j_3}$,.... If the rows of $T(Q)$ contain all positive responses to the single items and all "and" combinations, then the number of rows equal to $2^J - 1$ where J is the total number of items. And the $T(Q)$ is defined as *saturated*. Thus each element of $T(Q)$ indicates whether the attribute vector would possibly get the positive responses to the item combination. The next step is to build a column vector the length of which equals to the number of rows of $T(Q)$ and each element corresponds to the proportion of number of people who have positive response to the item combinations. Let the column vector be p we will have

$$T(Q)\hat{\mathbf{P}} = p \tag{2.16}$$

where $\hat{\mathbf{P}}$ contains the estimated proportions of respondents with each attribute profile. As a result for any binary matrix Q', let

$$S(Q') = \inf_{\hat{\mathbf{P}} \in [0,1]^{2^K - 1}} |T(Q')\hat{\mathbf{P}} - p| \tag{2.17}$$

and

$$\hat{Q} = \arg\inf_{Q'} S(Q') \tag{2.18}$$

then $\hat{Q}$ is an estimator of Q-matrix.

If a Q-matrix is *complete* (for each attribute there exists an item only requiring that attribute), and $T(Q)$ is saturated, Liu et al. [2011b] has proved the existence of best Q that can be drawn from the empirical response data mathematically together with two more conditions.

With the established theoretical framework, Liu et al. [2011b] implements the model into cognitive diagnostic models and consider slipping and guessing parameters into the estimation of Q-matrix. They have proved the consistence in DINA model with known or unknown slipping parameter and a known guessing parameter. They extended their theories to DINO model in their next paper and talked about the estimation of Q-matrix again under the condition of no slipping or guessing, and nonzero slipping and guessing probabilities in DINA model [Liu et al., 2011a]. They have also claimed that their results were consistent in all the four models mentioned above.

The estimation methods discussed by Liu et al. [2011b,a, 2012] do require lots of assumptions such as complete Q-matrix, saturated T-matrix, or known guessing parameters in DINA model, and the computation is so difficult and it is not practical to apply to real response data situation. However, it does provide a new and reliable idea to possibly estimate Q-matrix based on the real response data from subjects. Inspired by their studies, the dissertation comes up a new way, and hopefully a better and more applied method, that could possibly estimate Q-matrix elements.

Chapter 3

Methods

This section formally introduces model establishment and estimation approach on Q-matrix. As the elements of Q-matrix are dichotomous and the estimation process of discrete variables is extremely hard and complicated, we consider the components of Q-matrix as continuous variables within (0,1) indicating the probabilities that an item requires a specific attribute. With penalized techniques we are able to push the estimated values as close to 0 or 1 as possible and use cutoffs to get back to the discrete Q-matrix. In order to reduce computation complexity the dissertation considers the model conjunctive (non-compensatory) and first consider that item responses are free from guessing and slipping. In other words, guessing or slipping parameters are not considered and item responses are totally determined by the mastery of skills. Adding guessing and slipping parameters or disjunctive models will be discussed in the future studies. Detailed model assumptions will be discussed below.

3.1 Model specification

Some definitions have to be introduced first at the very beginning, although most of them have been talked about in the previous sections.

- *Attribute:* The certain skills that a subject masters or not. Assume that all items of the test require K attributes, $\vec{\alpha} = (\alpha_1, ..., \alpha_K)$ is the vector of attributes. $\alpha_k \in \{0, 1\}$ indicates the presence or absence of the kth attribute, $k \in \{1, 2, \ldots, K\}$.

$$\alpha_k = \begin{cases} 1 & \text{if a subject holds the } k\text{th attribute/skill} \\ 0 & \text{if a subject does not hold the } k\text{th attribute/skill} \end{cases}$$

- *Responses:* The binary responses that a subject gets an item right or not. Assume that there are J items in the test, $\mathbf{R} = (R_1, ..., R_J)$ is the vector of item responses. $R_j \in \{0, 1\}$ indicates whether a subject gets the jth item right or not, $j \in \{1, 2, \ldots, J\}$.

$$R_j = \begin{cases} 1 & \text{if a subject gets the } j\text{th item right} \\ 0 & \text{if a subject gets the } j\text{th item wrong} \end{cases}$$

- *Q-matrix:* Q-matrix is defined as the link between attributes and the items. A traditional $J \times K$ matrix $(q_{jk})_{J\times K}$ has binary elements $q_{jk} \in \{0, 1\}$ which tell us whether the jth item requires the kth attribute or not.

$$q_{jk} = \begin{cases} 1 & \text{if the } j\text{th item requires the } k\text{th attribute} \\ 0 & \text{if the } j\text{th item does not require the } k\text{th attribute} \end{cases}$$

The next step is to establish connections between the Q-matrix and the item responses. Some assumptions have to be restated here: the item responses are completely determined by the attributes; no slipping or guessing exists in the response determination; all required attributes together in each item from the Q-matrix are necessary and sufficient to provide a positive response ($R_j = 1$) to the specific item; lacking any of required attribute would lead to failure of answering the item, and possessing additional attributes does not compensate for the absence of necessary attributes. In sum, it could be considered as a conjunctive (non-compensatory) model without slipping or guessing. The appropriate mathematical demonstration for the ith subject on the jth item response is

$$R_{ij} = \xi_{ij} = \prod_{k=1}^{K} {\alpha_{ik}}^{q_{jk}} = \mathbf{1}(\alpha_k \geq q_{jk} : k = 1, \ldots, K, j = 1, \ldots, J) \tag{3.1}$$

Imagine that the ith respondent has a combination of the latent K attributes $\vec{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{iK})$, then according to the Q-matrix, there would be a theoretical item response pattern $\vec{\mathbf{R}}_i$ for ith subject, where $\vec{\mathbf{R}}_i = (R_{i1}, \ldots, R_{iJ})$ and each of the element R_{ij} is calculated according to the above equation.

Total K attributes would have 2^K possible attribute patterns, thus leading to 2^K latten classes. Now another matrix can be defined as the link between the attribute combinations and item responses.

- *D-Matrix:* The $J \times 2^K$ matrix $(d_{jl})_{J \times 2^K}$ has binary elements $d_{jl} \in \{0, 1\}$ which indicate the ideal response for the jth item when a subject has a lth combination of attributes.

$$d_{jl} = \begin{cases} 1 & \text{get the } j\text{th item right with the } l\text{th combination of attributes} \\ 0 & \text{fail to get the } j\text{th item right with the } l\text{th combination of attributes} \end{cases}$$

So far all the matrices here have binary elements 0 and 1. Estimating and optimizing discrete variables are much more difficult to achieve and time consuming, but continuous variables are easier to be estimated. Moreover, with penalization techniques it is possible to restrict the estimations close to 0 and 1 and get back to the binary matrices using reasonable cutoffs.

For this purpose, we define continuous Q-matrix as follows: A $J \times K$ matrix $(q_{jk})_{J\times K}$ has proportional elements $q_{jk} \in (0, 1)$ which tells us the probability of q_{jk} that the jth item requires the kth attribute. Similarly, *D-Matrix* is a $J \times 2^K$ matrix $(d_{jl})_{J\times 2^K}$ with proportional elements $d_{jl} \in (0, 1)$ which indicate the probability of d_{jl} to get positive response on the jth item when a subject has the lth combination of attributes.

Here comes out two possible methods to build criterions and estimate elements of Q-matrix. The first method is inspired and followed by Liu et al. [2011b] to construct a total error through a "regression" liked function. The second approach is based on item response function and maximum likelihood estimation method, to find out the best Q that maximize the likelihood function from the latent model.

3.2 Estimation approaches

3.2.1 Model 1: matrix transformation

The D-matrix is not enough to build up an equation and estimate the Q-matrix from item response data. The proportions of positive responses on each item is supposed to be used for estimation. However, a subject who gets one item right is possible to get another one right too, thus making each item accuracy proportion not exclusive; that is, intersections exist among proportions of correct responses for each item. The response data is hard to be categorized by the overlapping categories. The problem could be solved by one more step further: building the relationship between attribute patterns and the item response patterns. In this case, each response pattern will contain a unique group of subjects. People who fall into one response pattern are not able to fall into another. As a result, the probabilities of all item response patterns we get from the response data are mutually exclusive.

Because item responses are binary too, J items will lead to 2^J possible item response patterns. Now a similar T-matrix $T(Q)$ to Liu et al. [2011b] can be constructed below.

- *T-matrix* The $2^J \times 2^K$ T-matrix $(t_{ml})_{2^J \times 2^K}$ has binary elements $(t_{ml}) \in \{0, 1\}$ which indicate that whether an attribute pattern $\vec{\alpha}$ could get a response pattern $\vec{\mathbf{R}}$.

$$t_{ml} = \begin{cases} 1 & \text{get the } m\text{th item response pattern with the } l\text{th attribute pattern} \\ 0 & \text{not get the } m\text{th item response pattern with the } l\text{th attribute pattern} \end{cases}$$

The T-matrix is the central quantity that connects the Q-matrix with the response and attribute distributions. The 2^K columns each corresponds to one possible attribute vector $\vec{\alpha}$ and the 2^J rows each corresponds to one possible item response vector $\vec{\mathbf{R}}$. Instead of labeling the rows and columns of $T(Q)$ by ordinal numbers, the method follows Liu et al. [2011b,a]'s notation and label them by the vectors of attribute pattern and response pattern. For instance, the $\vec{\alpha}_l$-th column of $T(Q)$ is the column that corresponds to attribute $\vec{\alpha}_l$, and the $\mathbf{R}_m$-th row of $T(Q)$ is the row that corresponds to item response $\vec{\mathbf{R}}_m$.

Similarly the proportional continuous T-matrix can be defined as a $2^J \times 2^K$ matrix $(t_{ml})_{2^J \times 2^K}$ with proportional elements $t_{ml} \in (0, 1)$ which indicate the probability of

t_{ml} if an attribute pattern $\vec{\alpha}_l$ could get an ideal response pattern $\vec{\mathbf{R}}_m$.

Now it is time to build relationships between established matrices and the real item response data of the subjects.

- *y-vector.* Let $\vec{y}$ be a 2^J column vector the length of which equals to the number of rows of T-matrix $T(Q)$. Each element corresponds to one row vector item response pattern $\vec{\mathbf{R}}_m$ of $T(Q)$ and indicates the proportion of the sample that has the response pattern $\vec{\mathbf{R}}_m$.

Let the total number of subjects be N, we will have the definition of y-vector $(y_m)_{1\times 2^J}$ in the following mathematical way.

$$y_m = N_{\vec{\mathbf{R}}_m}/N \tag{3.2}$$

Where $N_{\vec{\mathbf{R}}_m} = \sum_{i=1}^{N} \mathbf{1}(\vec{\mathbf{R}}_{im} = 1)$.

- *$\hat{p}$-vector.* We let $\hat{\mathbf{p}}$ be a 2^K column vector the length of which equals to the number of columns of T-matrix $T(Q)$. Each element corresponds to one column vector attribute pattern $\vec{\alpha}_l$ of $T(Q)$ and indicates the proportion of the sample that has the attribute combination pattern $\vec{\alpha}_l$.

Similarly the mathematical definition of $\hat{p}$-vector $(\hat{p}_l)_{1\times 2^k}$ is

$$\hat{p}_l = N_{\vec{\alpha}_l}/N \tag{3.3}$$

Where $N_{\vec{\alpha}_l} = \sum_{i=1}^{N} \mathbf{1}(\vec{\alpha}_{il} = 1)$.

Note that in fact it is not possible to get the true value of vector of $\hat{p}$-vector because it is a latent class size. But the value of y-vector of response pattern proportions can be obtained from the real item response data.

After all the definitions are issued, the estimation model for Q-matrix can be addressed. A good way to restrict the matrix elements q_{jk} between 0 and 1 is to use a transformation function: a logit function is a good option.

This approach is based on relationships between $T(Q)$ and response data. If function (3.1) is strictly respected, then

$$T(Q)\hat{\mathbf{p}} = \vec{y} \tag{3.4}$$

The idea is to build a non-linear transformation function T to get $T(Q)$ from Q, then use the relationship between $T(Q)$ and $\vec{y}$ in (3.1) to create a criteria, minimize it and optimize the elements of Q. Pseudocode for the algorithm is givien below:

```
$\mathrm{Obj} = \mathrm{function}(\Gamma)\{$
$\Gamma = (\gamma_{jk})_{J\times K}$
$q_{jk} = \frac{e^{\gamma_{jk}}}{1+e^{\gamma_{jk}}}$
$\mathbf{Q} = (q_{jk})_{J\times K}$
$\mathbf{T} = \boldsymbol{T}(\mathbf{Q})$
$\hat{\mathbf{p}} = (\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\vec{y}$
$\mathrm{penalty} = \lambda(\sum q_{jk}^a(1-q_{jk})^a)$
$\mathrm{SSE} = (\mathbf{T}\hat{\mathbf{p}} - \vec{y})'(\mathbf{T}\hat{\mathbf{p}} - \vec{y}) + \mathrm{penalty}$
$\}$
$\mathbf{Q} = \arg\min\{\mathrm{Obj}(\mathbf{Q})\}$
```

The method shrinks estimations by imposing a penalty on q_{jk} and minimize a penalized residual sum of squares,

$$\hat{\mathbf{Q}} = \arg\min_q\{\min_p ||\mathbf{T}(\mathbf{Q})\hat{\mathbf{p}} - \vec{y}||_2 + \text{penalty}_q\} \tag{3.5}$$

The penalty function penalty $= \lambda(\sum q_{ij}^a(1-q_{ij})^a)$ constrains q_{ij} between 0 and 1. Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ the greater the amount of shrinkage. The function of $\sum q_{ij}^a(1-q_{ij})^a$ will force the estimations of q_{ij} as close as to 0 or 1 as possible to the extent when parameter a goes down. The contours of the penalty function are shown in Figure 3.1, for the care of different parameters.

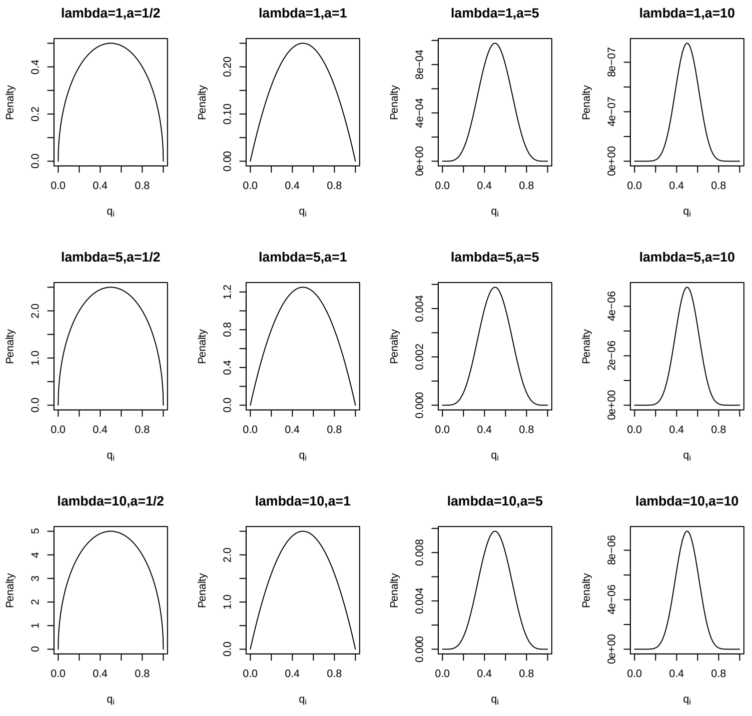

Figure 3.1: The penalty function of T-matrix approach

Regarding the estimation theory, we have multiple ways to optimize the penalized SSE and get estimated elements of Q. For example, Non-Linear Minimization carries out a minimization of the function f using a Newton-Raphson type algorithm.

General-purpose optimization based on Nelder-Mead, quasi-Newton and conjugate-gradient algorithms is also a good alternative [Dennis and Schnabel, 1987]. It includes an option for box-constrained optimization and simulated annealing (R documents). These methods are positive to optimize the function and get final results of Q.

However, in real situation, the nonlinear transformation of T from Q to $T(Q)$ is not easy to build, and the algorithm is hard to program, which is a key barrier in this method. So far the T-matrix transformation method is only theoretically stated here, the application is remained to be further discussed in the future.

3.2.2 Model 2: probabilistic modeling

The other approach that can possibly get the estimation of Q is to create the density function of item response and based on maximal likelihood estimation method. The idea is shown below:

- Let $(\vec{\alpha_l})_{1\times K}$ be the binary attribute vector for an individual i. We have N subjects and 2^K attribute patterns in total, $i = 1, 2, \ldots, N$, $l = 1, 2, \ldots, 2^K$.

- The Q-matrix $Q = (q_{jk})$ has elements $q_{jk} \in (0, 1)$ that indicates the probability for item j requiring attribute k. We have J items and K attributes in total, $j = 1, 2, \ldots, J$ $k = 1, 2, \ldots, K$.

- y_m is the proportion of the sample that has item response pattern $\vec{\mathbf{R}}_m$ (the same as y-vector defined above), where $m = 1, 2, \ldots, 2^J$.

- The response of the ith subject on the jth item is defined as

$$R_{ij} = \begin{cases} 1 & \text{if subject } i \text{ gets the } j\text{th item right} \\ 0 & \text{if subject } i \text{ gets the } j\text{th item wrong} \end{cases}$$

Thus $\Pr\{R_{ij} = 1|\vec{\alpha}_l\}$ indicates the probability of the correct response on the jth item by the ith subject under the condition of the attribute pattern vector $\vec{\alpha}_l$. We can also take it as the probability that the attribute pattern $\vec{\alpha}_l$ contains all attributes required by the jth item ($\vec{\alpha}_l$ corresponds to one possible attribute pattern vector within the total 2^K attribute patterns where K attributes exist). Equivalently the probability equals the one that the jth item does not require any of the skills individual i does not have ($\vec{\alpha}_l$ does not contain). Let $\vec{\alpha}_l = (\alpha_{ik})_{1\times K}$, α_{ik} =1 or 0 indicates whether the ith person has the kth attribute or not, we will have

$$\Pr\{R_{ij} = 1|\vec{\alpha}_l\} = \prod_{k=1}^{K} (1 - q_{jk})^{(1-\alpha_{ik})} \tag{3.6}$$

In this model q can be considered as the probability that the item requires the skill, or the proportion of persons who need the skill to get the item right. The combined item response function would be

$$\Pr\{R_{ij} = r_{ij}, \forall i|\vec{\alpha}_l, q_{jk}\} = [1 - \prod_{k=1}^{K} (1 - q_{jk})^{(1-\alpha_{ik})}]^{(1-r_{ij})}[\prod_{k=1}^{K} (1 - q_{jk})^{(1-\alpha_{ik})}]^{r_{ij}} \tag{3.7}$$

Where $r_{ij} = 1$ or 0. So the probability for individual i to get a response pattern $\mathbf{R}_i$ is

$$\Pr\{\mathbf{R}_i = \mathbf{r}_i, \forall i|\vec{\alpha}_l, q_{jk}\} = \prod_{j=1}^{J}[1-\prod_{k=1}^{K} (1 - q_{jk})^{(1-\alpha_{ik})}]^{(1-r_{ij})}[\prod_{k=1}^{K} (1 - q_{jk})^{(1-\alpha_{ik})}]^{r_{ij}} \tag{3.8}$$

In the estimation process, in order to make sure that the Q elements are restricted within 0 and 1, we can assign a logit function to q_{jk}. Let $q_{jk} = \frac{e^{\gamma_{jk}}}{1+e^{\gamma_{jk}}}$ or $\text{logit}(q_{jk}) = \log(\frac{q_{jk}}{1-q_{jk}}) = \log(q_{jk}) - \log(1 - q_{jk}) = \gamma_{jk}$, the item response function could become

$$\Pr\{R_{ij} = r_{ij}, \forall i, j|\vec{\alpha}_l, \gamma_{jk}\} = [1-\prod_{k=1}^{K} (1 - \frac{e^{\gamma_{jk}}}{1 + e^{\gamma_{jk}}})^{(1-\alpha_{ik})}]^{(1-r_{ij})}[\prod_{k=1}^{K} (1 - \frac{e^{\gamma_{jk}}}{1 + e^{\gamma_{jk}}})^{(1-\alpha_{ik})}]^{r_{ij}} \tag{3.9}$$

The next step is to use an EM-algorithm to find the maximum likelihood estimates of q_{jk}. The likelihood function of the item response function (3.8) is

$$\mathcal{L} = \prod_{i=1}^{N} (\sum_{\vec{\alpha}_l} \Pr\{R_{ij} = r_{ij}, \forall i, j | \vec{\alpha}_l, q_{jk}\} \Pr\{\vec{\alpha}_l\}) \tag{3.10}$$

Let $\vec{\theta} = \{q_{jk}\}_{J \times K}$, $\vec{\pi} = Pr\{\vec{\alpha}_l\}$,$l = 1, 2, \ldots, 2^K$, then $\vec{\psi} = (\vec{\theta}, \vec{\pi})$ will be the parameter space (total number of $J \times K + 2^K$ parameters). The parameters are all probability based which are continuous variables. Let $\mathbf{A} = \{\vec{\alpha}_l\}_{2^K}$ be the latent class which are discrete classification patterns, drawn from a fixed number of 2^K values. The parameter $\vec{\pi} = Pr\{\vec{\alpha}_l\}$ are actually latent class sizes.

Again a penalty function is implemented into the likelihood function to push estimated matrix elements to either 1 or 0 so that the results may be more accurate and robust. Based on visual inspection of several function plots, a beta distribution is selected here due to our purpose and its feasibility to push the elements of Q-matrix to {0,1}. The penalty function is constructed as below:

$$\text{Penalty} = -\lambda \sum_{j=1}^{J} \sum_{k=1}^{K} [\log q_{jk} + \log(1 - q_{jk})] \tag{3.11}$$

which is equivalent to

$$\text{Penalty'} = \sum_{j=1}^{J} \sum_{k=1}^{K} [q_{jk} * (1 - q_{jk})]^{-\lambda} \tag{3.12}$$

If we use a logit function to transfer q_{jk} to γ_{jk} then the penalty function will be

$$\text{Penalty} = -\lambda \sum_{j=1}^{J} \sum_{k=1}^{K} [\gamma_{jk} - 2\log(1 + e^{\gamma_{jk}})] \tag{3.13}$$

Here $\lambda \geq 0$ is a complexity parameter of the penalty function that controls the amount of shrinkage. Normally in LASSO regression the larger the value of λ the greater the amount of shrinkage.

As a result the Penalized Log-Likelihood function will be

$$\begin{aligned}
&\log(\mathcal{L}_{\text{penalized}}) = \log \mathcal{L} + \text{Penalty} \\
&= \sum_{i=1}^{N} \log(\sum_{\vec{\alpha}_l} \Pr\{R_{ij} = r_{ij}, \forall i, j | \vec{\alpha}_l, q_{jk}\} \Pr\{\vec{\alpha}_l\}) \\
&-\lambda \sum_{j=1}^{J} \sum_{k=1}^{K} [\log q_{jk} + \log(1 - q_{jk})] \\
&= \sum_{i=1}^{N} \log(\sum_{\mathbf{A}} \Pr\{\mathbf{R} = \mathbf{r} | \mathbf{A}, \vec{\theta}\} \Pr\{\mathbf{A}\}) + \text{Penalty}(\vec{\theta}) \\
&= \Delta(\mathbf{A}, \vec{\theta})
\end{aligned}$$

Let $\Delta = \Delta(\mathbf{A}, \vec{\theta}) = \log(\mathcal{L}_{\text{penalized}})$ which is a function of the parameters $\vec{\theta}$ (Q-matrix elements) and latent class $\mathbf{A}$, the maximum likelihood estimators (MLE) of the parameters are obtained by maximizing Δ, or minimizing -2Δ, or $-2\log(\mathcal{L}_{\text{penalized}})$, which is more commonly used. The Expectation-Maximization (EM) algorithm seeks to find the MLE of the marginal likelihood with latent class by iteratively applying the following two steps:

Expectation step (E step): Calculate the expected value of the log likelihood function, with respect to the conditional distribution of $\mathbf{A}$ given $\mathbf{R} = \mathbf{r}$ under the current estimate of the parameters $\vec{\theta}^{(t)}$

$$\mathcal{Q}(\vec{\theta} | \vec{\theta}^{(t)}) = E_{\mathbf{A}|\mathbf{R}, \vec{\theta}^{(t)}}(\Delta) \tag{3.14}$$

Maximization step (M step): Find the parameter that maximizes this quantity:

$$\vec{\theta}^{(t+1)} = \arg\max_{\vec{\theta}} \mathcal{Q}(\vec{\theta}|\vec{\theta}^{(t)}) \tag{3.15}$$

The model strictly follows the assumptions of EM algorithm and the penalty function does not include any latent variable. The EM algorithm is set with a maximum $100,000$ iterations in this research. The latent variables $\mathbf{A}$ are attribute patterns which are 2^K discrete classifications, and there is one latent variable per observed data point. The parameter $\vec{\theta}$ is continuous probabilities, and it is associated with data points whose corresponding latent variable has a particular value. The iterative algorithm will calculate us estimates for the parameter of Q-matrix elements and the latent class sizes of attribute pattern probabilities by the following algorithm:

1. First, initialize the parameters $\vec{\theta}$ to some random values.

2. Compute the best value for $\mathbf{A}$ given these parameter values.

3. Then, use the just-computed values of $\mathbf{A}$ to compute a better estimate for the parameters $\vec{\theta}$. Parameters associated with a particular value of $\mathbf{A}$ will use only those data points whose associated latent variable has that value.

4. Iterate steps 2 and 3 until convergence.

Typically the convergence criteria for EM is that the values of $\ln f(y|\theta^{(i)})$ converge. For the moment, the stopping criteria is set when $|\theta^{(i+1)} - \theta^{(i)}| < \epsilon$, where ϵ is any arbitrarily small positive number. Evaluations of stochastic models are normally based on comparing the equivalent AIC (Akaike information criterion) or BIC (Bayesian information criterion) among multiple models to measure the relative goodness of fit.

$$\text{AIC} = 2 \times \text{number of parameters} - 2 \times \Delta_{\max} \tag{3.16}$$

$$\text{BIC} = -2 \times \Delta_{\text{max}} + \text{number of parameters} \times \ln(\text{sample size}) \qquad (3.17)$$

Where Δ_{max} is the the logarithm of the maximized value of the penalized likelihood function $\log(\mathcal{L}_{\text{penalized}})$. Besides the value of log-likelihood function, AIC and BIC have also considered the number of parameters (BIC takes sample size into account as well).

AIC can be said to describe the tradeoff between bias and variance in model construction, or loosely speaking between accuracy and complexity of the model. Similarly the BIC resolves overfitting by introducing a penalty term for the number of parameters in the model. The penalty term is larger in BIC than in AIC due to the additional factor of sample size. However, AIC and BIC values provide a means for model selection without a test of a model in the sense of testing a null hypothesis; i.e. they can tell nothing about how well a model fits the data in an absolute sense.

In the dissertation, AIC or BIC can be used to compare models when different number of attributes K is applied to the model so that we are able to select the best K with the smallest value of AIC or BIC.

3.3 Determination of λ and attribute dimension

The estimation process comes across two problems that need to be solved. The first one is the determination of the penalty function, which is equivalent to the selection of the parameter λ (and possibly a in T-matrix Approach). The second problem is how to determine the number of attributes K. The estimation of parameters (Q-matrix elements) are drawn from the values which can maximize the penalized maximum likelihood function $-2(\log \mathcal{L} + \text{Penalty}) = -2 \sum \log \mathcal{L} + \lambda \sum (\log Q + \log (1 - Q))$, in which the penalty function we select is a beta distribution and it is constructed as below:

$$\text{Penalty} = -\lambda \sum_{j=1}^{J} \sum_{k=1}^{K} [\log q_{jk} + \log(1 - q_{jk})] \tag{3.18}$$

The idea of the penalty function in the both models is to shrink the value of $q \in (0, 1)$ and push q's to either 1 or 0 as close as possible. However, when a penalty function is added, the estimations of our parameter q's are biased estimator, and as the $\lambda \to \infty$ then $q \to 0$, due to the nature of the constraint. If we treat λ as an estimated parameter together with q's in the model, there is a great possibility that λ has to be equal to zero in order to get the optimized results. Thus this is not a good approach to determine the value of λ. In fact, in Lasso regression, a shrinkage condition is added to do a kind of continuous subset selection by causing some of the coefficients to be exactly zero. Lasso translates each coefficient by a constant factor λ, truncating at zero. This is called "soft thresholding", and is used in the context of wavelet-based smoothing. The idea is very similar to our approaches that we would like to keep some of the q's 0 and some approaching 1. Figure 3.2 is an example of the profiles of Lasso coefficients, as the tuning parameter t is varied, where $\sum \beta^2 \leq t$.

In Ridge and Lasso regression, the parameters of the penalty function are adaptively chosen to minimize an estimate of expected prediction error. The idea of prediction can be used in our case to determine the parameter in the penalty function. Choosing the penalty function according to prediction error is out-of-sample evaluation. It normally split data into training and test sets and focus on how well the model predicts things. Prediction error is all that matters, and the parameter λ in the penalty function is determined from a set of values by the one with the least prediction error. If model is overfit, will not perform well on out-of-sample data. As a result, it reduced the chance that $\lambda = 0$ in this situation and avoid overfitting of

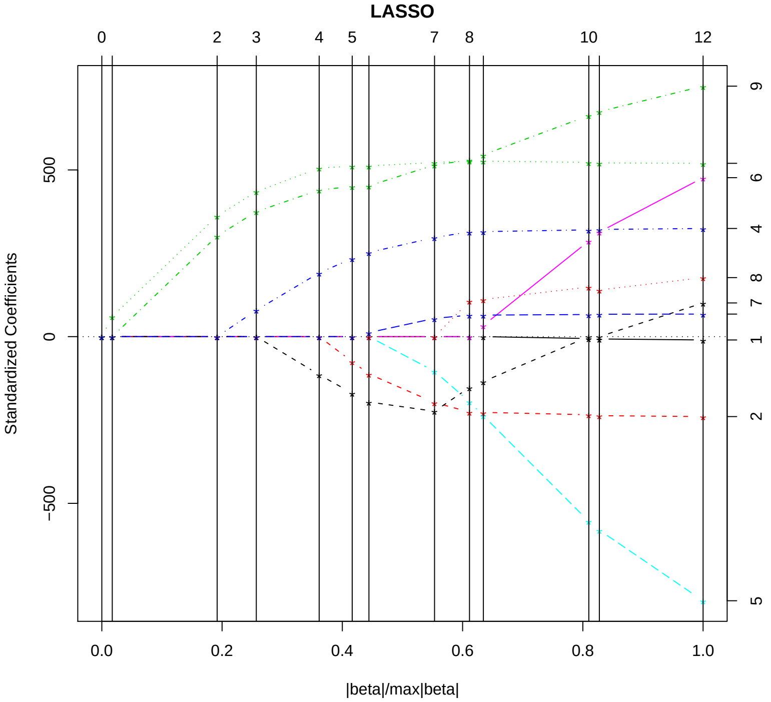

Figure 3.2: The Lasso regression

the model.

In order to evaluate the prediction error, we have to randomly partition data into training and test set first. In training set, data is used to train/build the model and estimate parameters. The test set is a set of examples not used for model induction but used for performance evaluation. The evaluation criteria is the prediction (generalization) error: the model error on the test data. Cross-validation is an estimate of the expected generalization error for each λ and λ can sensibly be chosen as the minimizer of this estimate.

The cross-validation method is the most popular and effective type of repeated holdout methods. Repeated holdout is repeating the process with different subsamples. In each iteration, a certain proportion is randomly selected for training (possibly with stratification) and the error rates on the different iterations are averaged to yield an overall error rate. K-fold cross-validation avoids overlapping test sets by spliting the data in k subsets of equal size, and using each subset in turn for testing and the remainder for training. Often the subsets are stratified before the cross-validation is performed because stratification reduces the estimate's variance. Extensive experiments have shown that stratified ten-fold cross-validation is the best choice to get an accurate estimate. It is even better if ten-fold cross-validation is repeated ten times and results are averaged which could reduce the sampling variance. This is called repeated stratified cross-validation. Error estimate is the mean across all repetitions.

There are two kinds of predictions that can be conducted here which depends on what we are interested in predicting for. We could look at how well the the model could be used to predict future students' performance on the test, and split the sample by N students. However, in cognitive diagnostic models we are more interested in how effective the model is to predict the students' performance on a specific item. Therefore it is preferred here to split the total responses ($N \times J$, total N subjects and J items) into ten folds instead of the students.

In the T-matrix approach, each error estimate in a repetition is the penalized SSE (function 3.5) calculated by estimated Q and the item responses in the test set. The final error estimate is the mean across all repetitions (all penalized SSEs). The best choice of λ and a relies on which could come out with the smallest average SSE.

The probabilistic model can apply a similar idea as the one in the T-matrix approach to construct an estimate error. One way is to implement the deviance distances as the error estimate for the test set data. Because the total responses are split into training and test sets, we can regard the training set as observed data and the test set as missing data. The judgement is based on the prediction performance on the test set by using the training set data, and the prediction performance is assessed by the deviance distance. The deviance distance is defined as:

$$D_\lambda = -2 \sum_{i,j} \log \Pr \left(R_{ij,\text{missing}} | R_{ij,\text{observed}} \right) \tag{3.19}$$

Therefore, D_λ will be a function of the Q and include the information about the conditional probability of the missing responses in the test set given the observed responses in the training set. Estimated Q is obtained through model fit using the observed responses in training set. Due to the repetitions in the cross-validation method, final error estimate is the mean value across all single error estimates in repetitions. For example, if we use ten-fold cross-validation, the final deviance distance (error estimate) will be the mean across all the ten deviance distances from the ten times of model fittings and performance evaluations. The parameter λ can be chosen when the smallest value of final deviance distance is achieved. Mathematically, $\lambda = \arg\min \frac{1}{10} \sum D_\lambda$.

Figure 3.3 is an example of the cross-validation curve in LASSO regression, and upper and lower standard deviation curves, as a function of the λ values used. Dependent and independent variables were generated through a standard normal distribution, fitted by a gaussian model. X-axis represents logarithm values of λ and Y-axis is the mean-square prediction error of each LASSO regression model fitted with corresponding λ. According to Figure 3.3 when $\log(\lambda) = -1.7$, or $\lambda = 0.183$, the regression would have the least mean-square error based on the cross-validation. As a result, $\lambda = 0.183$ would be the best penalty selection for the LASSO regression.

Although we intended to do the traditional cross-validation analysis to find out the optimal λ value of our model, the algorithm to estimate Q elements took much longer time than we expected, and it was impossible to run and select from a very large grid

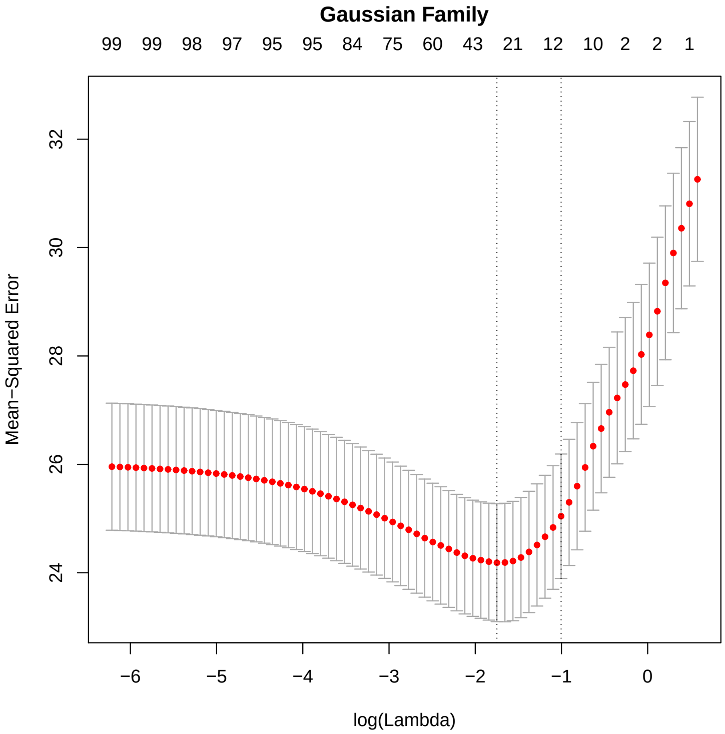

Figure 3.3: The penalty function with different lambda values in Gaussian regression

of λ values in the penalty function. For an exploratory purpose we first select a set of 7 possible λ values, for example, $\lambda \in (0, 0.05, 1, 2, 6, 9, 11)$ according to their penalized effects. When $\lambda = 0$ there is no penalized effect. Figure 3.4 presents the penalized function plots of different λ values that can shrinkage estimated elements to 0 and 1, and we can have an idea on the penalized effects based on visual inspection. For instance, $\lambda = 0.05$ the model would have a larger effect to penalize values to 0 and 1 than the rest of λ values, but it might not be the best fit to the data. With these λ values, we run the models and summarized the deviance distances on the predictive responses of each model. The ten-fold cross-validation average deviance is defined as $\text{Dev} = \frac{1}{10}\sum D_\lambda$ where $D_\lambda = -2\sum_{i,j}\log\Pr\left(R_{ij,\text{missing}}|R_{ij,\text{observed}}\right)$. λ is choose when the minimum deviance is achieved.

Meanwhile the attribute dimension has not been decided yet. If no information is

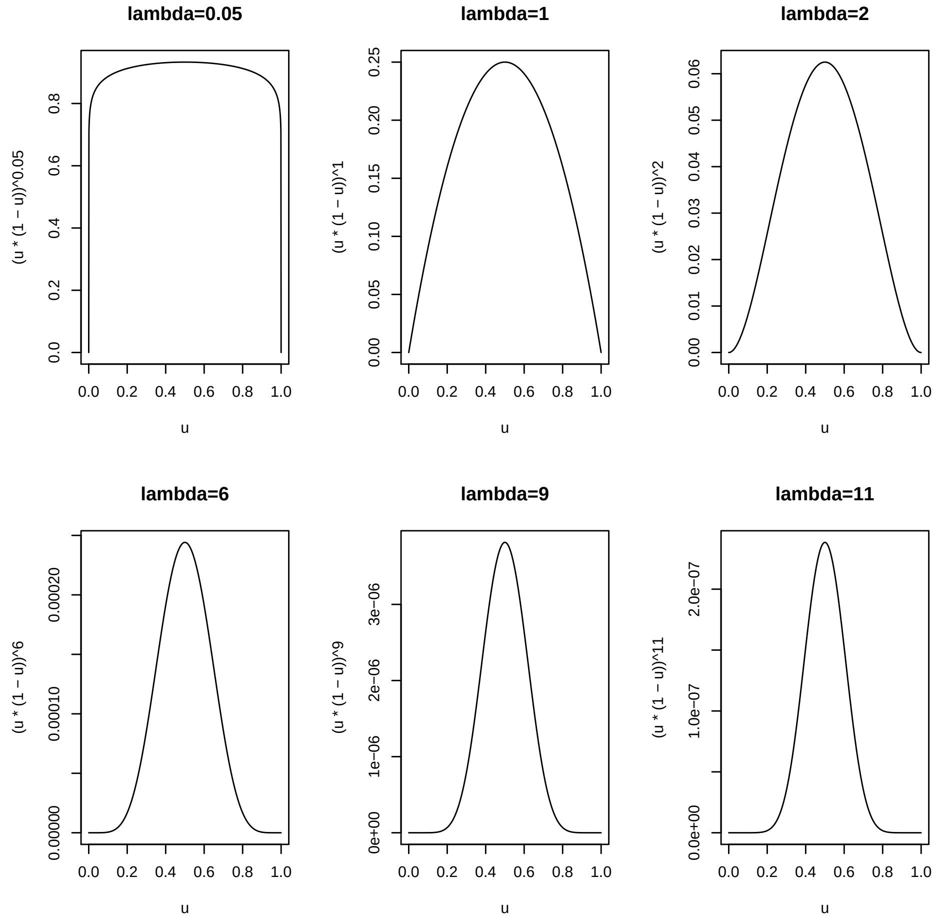

Figure 3.4: The penalty function with different lambda values in the model

given about the number of the domain of skills, this Q-matrix estimation procedure is like finding out the factors and factor loadings in exploratory factor analysis (EFA). EFA is used to uncover the underlying structure of a relatively large set of variables, and describe variability among observed, correlated variables in terms of a potentially lower number of unobserved, uncorrelated variables called factors. In this case we are trying to find out the factor loadings that can connect items with unobserved latent factor attributes.

Determination on attribute dimension is easier than the selection of parameters in penalty function because we only care about the model goodness of fit rather than prediction errors. The number of attributes is chosen when the minimum AIC or BIC is obtained, which means we select the model with best fit to the current data. Another possible way is to treat the number of attributes K as an estimated parameter together with q's in the model. However, as K changes, the number of q's will change too. It will be more complicated if we do not determine K in advance.

3.4 Finalizing Q-matrix and identification of attributes

The next step is finalizing the Q-matrix estimated back to the binary Q-matrix because the cognitive diagnostic models are relying on the discrete Q-matrix rather than the continuous probabilities. This is the main reason that the penalty function is applied into the model to push the element values to 0 or 1. When the estimated Q-matrix is calculated, we are going to use cutoffs of .9/.1 or .8/.2 or .7/.3 etc. to recode the estimated elements back to 1 and 0. The selection of cutoffs depends on the Q-matrix element estimation results. For example, if we use .8/.2 cutoff, those elements which are greater than .8 will be recorded into 1 and those smaller than .2 will be recorded into 0 in the Q-matrix. The rest values not recorded are those item-attribute relationships we are not sure about. For these uncertain elements, one possible solution is to apply a Bayesian extension of the DINA model developed by DeCarlo [2012] to recognize possible values in the estimated Q-matrix. The present study will report results based on .3/.7, .4/.6, and .5/.5 the three cutoff points, where .5/.5 is able to transfer all continuous estimates to binary values. In addition, .5/.5 cutoff point has been empirically proved to be the best choice when all ratio needs to be converted to binary data [Durongwatana, 2011; Fall, 2009]. As a result the estimated Q after .5/.5 cutoff will be selected as the final Q-matrix from the model.

The last problem is how to identify the attributes corresponding to the columns of the Q-matrix when the number of attributes have been determined. However the identification of attribute columns of Q-matrix is very difficult to solve. One possible solution is to look for experts to help determine the identifications of attributes (each column of the estimated Q-matrix) subjectively. Another way is to compare the estimated Q-matrix to the true Q in simulation study or expert-designed Q in real data study. The approach is more like finding the meaning of factor loading in confirmatory factor analysis (CFA). CFA seeks to determine if the number of factors and the loadings of measured variables on them conform to what is expected on the basis of pre-established theory. Indicator variables are selected on the basis of prior theory and factor analysis is used to see if they load as predicted on the expected number of factors. The researcher's a priori assumption is that each factor (the number and labels of which may be specified a priori) is associated with a specified subset of indicator variables.

Attribute identification can also draw from the ideas of rotation in factor analysis and label switching in latent class analysis. Rotation serves to make the output from factor analysis more understandable, by seeking a pattern of loadings where items load most strongly on one factor, and much more weakly on the other factors. The label switching methods deal with the unidentifiability of the permutation of clusters or more generally latent variables, which makes interpretation of results computed with MCMC sampling difficult.

However, sometimes it is still impossible to estimate the Q-matrix precisely and identify the attributes. For example, the model is hard to distinguish the following two Q-matrices because the second attribute always comes with the first one in the second Q-matrix, thus making the first attribute meaningless when an item requires the second one. Both of the two Q-matrices would possibly produce the same probabilities of success under the same attribute pattern through our model. This problem is called "Rotation" problem in cognitive diagnostic models [Johnson, 2009]. A possible way to solve the problem is to look at the ideal response patterns drawn from each Q-matrix. If two Q-matrices come to the same ideal response pattern we can regard the two matrices are equivalent.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \text{and} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Despite of the difficulties, the results from our model can provide a general idea on how these attributes are distributed among items, and it is still worth trying to identify the attributes as possible as we can. For pure exploratory analysis, the columns of estimated Q-matrix should be carefully examined by researchers together with domain experts to interpret the latent meanings, which is similar to mapping loadings in factor analysis. In this research, the model applies the discrepancy distance to match the attributes to the Q-matrix columns.

$$\text{Discrepancy} = -(\sum_{j,k} q_{jk} \log \hat{q_{jk}} + \sum_{j,k} (1 - q_{jk}) \log (1 - \hat{q_{jk}})) \tag{3.20}$$

Where q_{jk} is from true Q-matrix or expert-designed Q-matrix, and $\hat{q_{jk}}$ is from the estimated Q-matrix. By switching the matrix columns, we are expecting to get a match between Q and $\hat{Q}$ with the minimum discrepancy distance, and this column match will be considered as the results of identification.

The next steps will be model evaluation and application. The primary research and secondary questions are recorded here:

Are the methods designed able to estimate the Q-matrix? If they are, how accurate they perform on estimation of Q-matrix? How well they perform on real response data and what are the differences between estimated Q-matrix and the designed Q-matrix? Are those Q-matrices are identical? How about their latent class size distributions?

The questions will be answered through both simulation studies and empirical study of real data. The next section contains designs of both simulation studies and empirical study, together with their evaluation criteria. Evaluation methods on comparison of Q-matrices include discrepancy distance judgment; counts and proportions of correct identified elements in simulation studies, or consistent elements in empirical study. Item response patterns will be constructed to evaluate if those Q-matrices are identical.

3.5 Study designs

This research is going to perform two simulation studies in which the response data is generated through the DINA model. The first one is simulated from a made-up Q-matrix with a optimized attribute combination property. The second one uses a real Q-matrix designed in the fraction subtraction test to simulate a new response data set. The idea of simulation study is to use the simulated response data to estimate the Q-matrix through our methods, and then compare the estimated Q with the true Q to evaluate the estimation performance. Thus the simulation studies are able to check the model feasibility and evaluate the model performance. Note that in simulation studies, the attribute dimensions are pre-determined because we have already known the true Q-matrices. Besides, it also helps us to reduce computation complexity, and makes results easier to compare if Q-matrices have the same number of columns.

We are also going to conduct an empirical study based on real response data. The purpose of empirical study is to find out the differences between estimated Q-matrix from our model and the original Q-matrix designed by domain experts. Again, the total numbers of attributes are assumed to be the same as the number of columns in designed Q-matrix, due to computation and comparison feasibility. In other words, we do not consider the procedure to determine the total number of attributes, which is similar to exploratory factor analysis, but set the number fixed as in confirmatory factor analysis. However, we have to identify the attributes to the columns of our estimated matrices.

3.5.1 Simulation study 1: optimized Q-matrix

The first simulation adopts the optimized Q-matrix designed from de la Torre's research in 2009 on DINA model [de la Torre, 2009b]. This Q-matrix has ordered and well organized attribute combinations in items. 30 items and 5 attributes are included in the Q-matrix. The first ten items include only one attribute; the next ten items require two and the last ten items need three skills. None of the items contains exactly the same attributes as any other. This Q-matrix has been used for response data simulation before [de la Torre, 2009b]. The simulated data employs $2,000$ examinees. Because our model does not have slipping or guessing parameters, the simulation program is set with small slipping and guessing parameters equal to 0.1 in the DINA model.

$$P(X_{ij} = 1|\xi_{ij}) = 0.9^{\xi_{ij}} 0.1^{1-\xi_{ij}} \tag{3.21}$$

where

$$\xi_{ij} = \prod_{k:q_{jk}=1} \alpha_{ik} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}} \tag{3.22}$$

We use A1 to A5 to denote the five attributes. The q_{jk}'s in the Q-matrix for this data are given in Table 3.1.

3.5.2 Simulation study 2: fraction subtraction Q-matrix

The second simulation applies a real expert-designed Q-matrix without such regular arrangement. The purpose is to see if the method still works well for estimating a complex and unorganized Q-matrix. We adopt The Q-matrix of fraction substraction data in de la Torre's research in 2009 on DINA model [de la Torre, 2009b], which is a simplified version of the whole Q-matrix developed by Tatsuoka [1990], and treat it as the true Q-matrix for the data simulation. Assigning detailed contents to any attribute or item is meaningless in simulation study. We use A1 to A5 to indicate the five attributes associated with the the 15 items (Table 3.2). The Q-matrix contains 15 items and 5 attributes. The same designed DINA model simulates $2,000$ examinees, with the slip and guessing parameters equal to 0.1. The Q-matrix for this data is given in Table 3.2.

An example of simulation program based on DINA model is shown in the Appendix A.1. First of all the program generates 2000 examinees' attribute patterns using binomial distribution with a 0.5 probability that an examinee either has a skill or not. Then the program calculates the deterministic parameter η by attribute patterns and the true Q-matrix, and applies it into the DINA model with pre-determined guessing and slipping parameters (0.1) to get the probabilities of correct item responses for each examine. The last step is to generate responses by binomial distribution with these probabilities from DINA model. Because the people are randomly assigned

Table 3.1: Optimized Q-matrix

Item	A1	A2	A3	A4	A5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1
6	1	0	0	0	0
7	0	1	0	0	0
8	0	0	1	0	0
9	0	0	0	1	0
10	0	0	0	0	1
11	1	1	0	0	0
12	1	0	1	0	0
13	1	0	0	1	0
14	1	0	0	0	1
15	0	1	1	0	0
16	0	1	0	1	0
17	0	1	0	0	1
18	0	0	1	1	0
19	0	0	1	0	1
20	0	0	0	1	1
21	1	1	1	0	0
22	1	1	0	1	0
23	1	1	0	0	1
24	1	0	1	1	0
25	1	0	1	0	1
26	1	0	0	1	1
27	0	1	1	1	0
28	0	1	1	0	1
29	0	1	0	1	1
30	0	0	1	1	1

Table 3.2: Fraction subtraction Q-matrix

Item	A1	A2	A3	A4	A5
1	1	0	0	0	0
2	1	1	1	1	0
3	1	0	0	0	0
4	1	1	1	1	1
5	0	0	1	0	0
6	1	1	1	1	0
7	1	1	1	1	0
8	1	1	0	0	0
9	1	0	1	0	0
10	1	0	1	1	1
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	1	1	0
14	1	1	1	1	1
15	1	1	1	1	0

50% chance of occupying an attribute, the theoretical latent class sizes should be the same across all possible patterns. For example in our situation we have 5 attributes which lead to $2^5 = 32$ possible attribute patterns. Therefore the 32 latent class sizes should be equally distributed with a proportion of $1/32 = 3.125\%$ in each latent class. Meanwhile, the marginal latent class size for each attribute should be 50% as people are designed to have only a half chance to occupy each skill.

The simulated data was then used in the model for backward estimation of the Q. After the Q-matrix is estimated, the columns have to be corresponded to A1 to A5 in the true Q-matrix. According to the discrepancy distance function $-(\sum_{j,k} q_{jk} \log \hat{q_{jk}} + \sum_{j,k} (1 - q_{jk}) \log (1 - \hat{q_{jk}}))$, where q_{jk} is from true Q-matrix and $\hat{q_{jk}}$ is from the estimated Q-matrix, the matched $\hat{Q}$ is the one with the minimum discrepancy distance, and the matched columns will be considered as A1 to A5 in estimated Q-matrix. The final model with the best λ value is selected according to the prediction criteria (deviance distances) or the highest counts of correct estimated elements in simulation studies. The final estimation efficacy is evaluated by counting the numbers of q's correctly estimated, which is similar to sensitivity and specificity (false positive or negative) in categorical data analysis.

3.5.3 Empirical study: fraction subtraction responses

It is also important to apply the methods to real data situation and see how it works for real response data. We can compare the estimated Q-matrix with the expert-designed Q-matrix to evaluate the estimation effects and find out if there is any inappropriate designed Q-matrix element in the test. We are again very interested in what the differences are between the estimated Q-matrix and expert-designed Q-matrix, and we expect to see a small discrepancy. However, if the difference is huge, it could be a problem of the estimation method that does not fit the data well, or the Q-matrix was poorly designed by the experts and the items did not measure correctly what they were supposed to measure for. The item response data used in analysis is from fraction subtraction test. To simplify the computation, a less complicated version of the fraction subtraction data is adopted which has been used by de la Torre [2009b]. The data contains responses of 536 middle school students to 15 fraction subtraction items measuring the five skills listed by experts. The five attributes are: A1: subtract basic fractions; A2: reduce and simplify; A3: separate whole from fraction; A4: borrow from whole; and A5: convert whole to fraction. The data were originally described and used by Tatsuoka [1990] and more recently de la Torre [2008, 2009b], and DeCarlo [2011]. The expert-designed Q-matrix is shown in Table 3.3. The identification of attributes is done by the minimum discrepancy distance, and the best model with λ value is selected by the minimum deviance

Table 3.3: Designed fraction subtraction Q-matrix

No.	Item	A1	A2	A3	A4	A5	No. of total attributes needed
1	$\frac{3}{4} - \frac{3}{8}$	1	0	0	0	0	1
2	$3\frac{1}{2} - 2\frac{3}{2}$	1	1	1	1	0	4
3	$\frac{6}{7} - \frac{4}{7}$	1	0	0	0	0	1
4	$3 - 2\frac{1}{5}$	1	1	1	1	1	5
5	$3\frac{7}{8} - 2$	0	0	1	0	0	1
6	$4\frac{4}{12} - 2\frac{7}{12}$	1	1	1	1	0	4
7	$4\frac{1}{3} - 2\frac{4}{3}$	1	1	1	1	0	4
8	$\frac{11}{8} - \frac{1}{8}$	1	1	0	0	0	2
9	$3\frac{4}{5} - 3\frac{2}{5}$	1	0	1	0	0	2
10	$2 - \frac{1}{3}$	1	0	1	1	1	4
11	$4\frac{5}{7} - 1\frac{4}{7}$	1	0	1	0	0	2
12	$7\frac{3}{5} - \frac{4}{5}$	1	0	1	1	0	3
13	$4\frac{1}{10} - 2\frac{8}{10}$	1	1	1	1	0	4
14	$4 - 1\frac{4}{3}$	1	1	1	1	1	5
15	$4\frac{1}{3} - 1\frac{5}{3}$	1	1	1	1	0	4

distance of response prediction, or the direct results from simulations studies. The comparison between our final estimated Q and expert-designed Q is evaluated by counting the numbers of consistent q elements.

Chapter 4

Results

In general, the Q-matrix elements are virtually regarded as parameters and estimated with latent variables using EM algorithm in the probabilistic model. The total number of parameters is the total number of Q-matrix elements. For example, in simulation study 1 the number of parameters estimated is $30 \times 5 = 150$ for a Q-matrix with 30 rows and 5 columns; the total number of parameters in simulation study 2 and empirical study is $15 \times 5 = 75$ because the Q-matrix has 15 rows and 5 columns. Both data simulation and model estimation processes are programmed in R language and EM algorithm is set with a maximum $100,000$ iterations. The starting points are randomly generated by the program.

All studies report the information of selected λ values in penalty function, the penalized likelihood values from each model fit, the minimum discrepancy distance, counts and proportions of identical elements after different cutoff points. Although we are suppose to determine our best final model based on the response prediction criteria of minimum deviance distances, the cross-validation analysis is not taken into account at this time. One reason is that in simulation studies the information given by deviance distances is redundant because we can choose the best model according to the highest correct identification of true Q-matrices. However, in empirical study it is not the case because the designed Q-matrix might be wrong and should not be the criteria to select the best model. One option to select the best model in empirical study could be based on the results from simulation studies that we select the same λ values which provide the highest estimation accuracy in simulation studies. For example, if simulation study 1 and 2 indicate that the best model has a penalty function with $\lambda = 10$, then we should use $\lambda = 10$ in empirical study regardless of how many consistent elements between the estimated Q-matrix and designed Q-matrix in this model. Besides we can also look at the estimated Q-matrix which is most close to the expert-designed Q-matrix, if we trust the original Q-matrix is well designed. Moreover, so far the researcher has not found a practical way to predict the corresponding missing responses from the estimated Q-matrices in the cross-validation analysis of deviance distances, thus the selection method of λ based on deviance distances is not discussed temporarily at this time but possibly in the future studies.

The purpose of discrepancy distance is to identify estimated Q-matrix columns to the corresponding attributes; cutoff points are used to transfer the continuous elements to binary values (0 or 1) in estimated Q-matrix; the correct counts and proportions show the performance to evaluate our estimated Q-matrix compared with the true Q's or expert-designed Q.

4.1 Simulation study 1

Table 4.1 demonstrates the results from simulation study 1 on the optimized Q-matrix $Q_{30\times 5}$. A set of ten different λ values are deliberately pre-determined: 0, 0.001, 0.05, 1, 2, 3, 6, 9, 11, 13 ($\lambda \leq 0$). When $\lambda = 0$ the penalty function is zero and no penalized effect exists in fact; as the λ values increase from 0.001 to 13, the penalized effects vary differently. When λ is small there is a large penalty and when λ gets bigger the penalized effects are decreasing (Figure 3.4). In other words, for smaller λ values such as 0.001 or 0.05, estimated elements are more likely to be pushed to either 0 or 1; for larger λ values such as 11 or 13 it is more possibly to get estimated values around 0.5. Initially we believe that large penalty may work well because the correct Q-matrix elements should be either 1 or 0, therefore we select some comparatively small λ values such as 0.001, 0.05, 1, 2, and 3. But we also choose some larger numbers (6, 9, 11, and 13) because they might be better fit the data. In addition, we try to have these λ values uniformly distributed and that is how the ten values are selected.

According to Table 4.1, the probabilistic models with different λ values are able to identify 64% to 91.3% correct Q-matrix elements in the first simulation study, after we transfer all results into binary values using 0.5 cutoff point. The model with $\lambda = 11$ is selected as our final model because it has the highest probability of correctly identified Q-matrix elements among all models, regardless of which cutoff pint we choose (0.3/0.7, 0.4/0.6, or 0.5). But 0.5 cutoff is able to transfer all continuous Q-matrix elements to binary values. When we use 0.5, the estimated binary Q-matrix has a very high correct identification (91.3%) of the true Q elements. The continuous estimated Q-matrix and the final one after 0.5 cutoff are shown in Table 4.2 (red numbers indicate the incorrectly identified elements).

To answer the primary research questions, according to the final estimated Q-matrix in Table 4.2, 13 out of 150 elements are not correctly identified (8.7%). It is very interesting to see that for items which require only one skill (Item 1 to 10), the model is able to identify all the required attributes perfectly. However as the true Q-matrix goes more complex when the items need more than one skills, the model starts to make incorrect identification of these Q-matrix elements. For the

Table 4.1: Simulation study 1 results

				Identified elements of the true Q-matrix					
				0.3/0.7 Cutoff		0.4/0.6 Cutoff		0.5/0.5 Cutoff	
No.	λ	-2LL	Discrepancy	Counts	Prob	Counts	Prob	Counts	Prob
1	0	61017.75	99.80	98/150	65.3%	117/150	78%	128/150	85.3%
2	0.001	66103.89	121.22	60/150	40%	76/150	50.7%	105/150	70%
3	0.05	62268.25	104.30	89/150	59.3%	105/150	70%	121/150	80.7%
4	1	59772.38	85.71	98 / 150	65.3%	109/150	72.7%	116/150	77.3%
5	2	60703.65	72.09	105/150	70%	125/150	83.3%	134/150	89.3%
6	3	60093.59	215.11	78/150	52%	89/150	59.3%	101/150	67.3%
7	6	65367.08	90.78	46 / 150	30.7%	76/150	50.7%	96/150	64%
8	9	60966.08	67.50	72/150	48%	103/150	68.7%	124/150	82.7%
9	11	54467.29	46.69	108/150	72%	126/150	84%	137/150	91.3%
10	13	49649.16	162.63	106/150	70.7%	119/150	79.3%	124/150	82.7%

items which require two attributes (Item 11 to 20), 3 out of 50 elements (6%) are incorrectly identified, occurring in item 15 and 19. For the items which require three attributes (Item 21 to 30), 10 out of 50 elements (20%) are not identified correctly. Item 21, 22, 24, 25, 27, and 28 have misspecified elements. In sum, all 13 incorrectly identified elements happen in the last 15 items which require at least two skills. For example, item 15 does not require the 4th attribute according to the true Q-matrix but happens to require the skill in our estimated Q. The same situation occurs again in item 28. The other incorrectly identifications appear to reverse a required skill to non-required and another unnecessary attribute into required group within one item. For instance, item 21, 22, 25, and 27 need total three attributes both in true Q-matrix and estimated Q-matrix, but the specific attributes are different. For example, item 21 is designed to require A1, A2, and A3 in true Q-matrix, but it is estimated to need A1, A2, and A4 instead from the model. Overall our model performs well on the estimation of Q-matrix from the simulated response data in the first study. For the rest 9.7% discrepancies in estimated Q-matrix that are different from the true Q-matrix, we regard them as the elements that the model is not able to recognize correctly. They are indicators of mistakes performed by the model and

Table 4.2: Estimated Q-matrix in simulation study 1

	Continuous Q-matrix					Binary Q after 0.5 cutoff					True Q-matrix				
Item	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5
1	0.827	0.092	0.232	0.217	0.112	1	0	0	0	0	1	0	0	0	0
2	0.186	0.831	0.054	0.242	0.183	0	1	0	0	0	0	1	0	0	0
3	0.073	0.171	0.678	0.417	0.019	0	0	1	0	0	0	0	1	0	0
4	0.032	0.288	0.031	0.891	0.291	0	0	0	1	0	0	0	0	1	0
5	0.194	0.277	0.238	0.367	0.952	0	0	0	0	1	0	0	0	0	1
6	0.792	0.029	0.021	0.214	0.378	1	0	0	0	0	1	0	0	0	0
7	0.145	0.844	0.098	0.063	0.174	0	1	0	0	0	0	1	0	0	0
8	0.053	0.031	0.763	0.440	0.145	0	0	1	0	0	0	0	1	0	0
9	0.211	0.200	0.143	0.921	0.194	0	0	0	1	0	0	0	0	1	0
10	0.057	0.191	0.196	0.348	1.000	0	0	0	0	1	0	0	0	0	1
11	0.785	0.842	0.160	0.172	0.118	1	1	0	0	0	1	1	0	0	0
12	0.924	0.118	0.602	0.302	0.260	1	0	1	0	0	1	0	1	0	0
13	0.907	0.177	0.006	0.763	0.390	1	0	0	1	0	1	0	0	1	0
14	0.904	0.387	0.000	0.034	0.992	1	0	0	0	1	1	0	0	0	1
15	0.152	0.785	0.779	0.566	0.237	0	1	1	1	0	0	1	1	0	0
16	0.251	0.808	0.027	0.908	0.263	0	1	0	1	0	0	1	0	1	0
17	0.327	0.819	0.275	0.425	0.746	0	1	0	0	1	0	1	0	0	1
18	0.250	0.082	0.701	0.913	0.463	0	0	1	1	0	0	0	1	1	0
19	0.652	0.359	0.498	0.290	0.990	1	0	0	0	1	0	0	1	0	1
20	0.295	0.420	0.172	0.683	0.969	0	0	0	1	1	0	0	0	1	1
21	0.844	0.912	0.395	0.521	0.384	1	1	0	1	0	1	1	1	0	0
22	0.795	0.489	0.387	0.709	0.659	1	0	0	1	1	1	1	0	1	0
23	0.777	0.735	0.276	0.439	0.974	1	1	0	0	1	1	1	0	0	1
24	0.726	0.174	0.676	0.836	0.664	1	0	1	1	1	1	0	1	1	0
25	0.865	0.326	0.399	0.700	0.814	1	0	0	1	1	1	0	1	0	1
26	0.709	0.459	0.196	0.750	0.919	1	0	0	1	1	1	0	0	1	1
27	0.432	0.828	0.086	0.910	0.633	0	1	0	1	1	0	1	1	1	0
28	0.471	0.623	0.604	0.549	0.778	0	1	1	1	1	0	1	1	0	1
29	0.176	0.899	0.171	0.848	0.883	0	1	0	1	1	0	1	0	1	1
30	0.335	0.458	0.556	0.819	0.966	0	0	1	1	1	0	0	1	1	1

we cannot do much about them at this time. These discrepancies could be due to the fundamental difference between simulation model and estimation model, or a single random data simulation, or the estimation procedures based on EM algorithm (no model is guaranteed to perform perfectly).

To answer the secondary research questions, we look at the latent class sizes from our model and compare the ideal item response patterns between the estimated Q-matrix and the true Q-matrix (Table 4.3). Ideal item response patterns indicate the theoretical item responses for each latent class (attribute pattern) under the Q-matrix of the relationship between items and attributes. It is entirely possible two different Q-matrices can generate the same item response pattern, and in this case we regard the two Q-matrices are identical. We have mentioned this issue called "rotation" problem in the method section and pointed out that looking at item response patterns could be one solution to this problem. By comparing item response patterns between true Q-matrix and estimated Q-matrix, we are able to confirm whether the two Q-matrices are in fact identical or not. If they generate the same item response patterns we will not care about the differences among the elements any more. But if they do not generate the same item response patterns, we will say that true differences exist and the estimated Q-matrix is indeed different from the true one. If discrepancies really exist, they may be caused by different models we use for simulation and estimation, or the simulation process considered additional fixed guessing and slipping parameters.

According to Table 4.3, each item response pattern generated from all attribute patterns is unique, both for estimated Q and true Q. In other words, students within every 32 attribute pattern will have a total of 32 unique ideal item responses on the 30 items. It is not possible for two persons who have different attributes to get the same item responses. As a result, if one of the item response patterns from estimated Q-matrix cannot be found in the item response pattern list generated in true Q-matrix, we are not able to make the conclusion that the two Q-matrices are identical. Clearly there are some differences between the two lists. For example, the item response pattern from attribute combination A3 and A5 (00101) from estimated Q-matrix does not exist in the list of item response patterns from true Q-matrix in Table 4.4. Only 14 out of 32 (43.75%) latent classes come up with the same item response patterns in Table 4.5, and the forms of attribute patterns are simper than different item response patterns in Table 4.4. For example, latent classes in Table 4.5

Table 4.3: Item response patterns and latent class sizes in simulation study 1

Attribute 1 to 5		Ideal response patterns Item 1 to 30		Latent class sizes	
No.	Latent class	True Q	Estimated Q	True Q	Estimated Q
1	00000	000000000000000000000000000000	000000000000000000000000000000	3.13%	0.30%
2	00001	000010000100000000000000000000	000010000100000000000000000000	3.13%	0.10%
3	00010	000100001000000000000000000000	000100001000000000000000000000	3.13%	1.10%
4	00011	000110001100000000010000000000	000110001100000000010000000000	3.13%	0.30%
5	00100	001000010000000000000000000000	001000010000000000000000000000	3.13%	2.20%
6	00101	001010010100000000000000000000	001010010100000000100000000000	3.13%	10.30%
7	00110	001100011000000000100000000000	001100011000000000100000000000	3.13%	0.10%
8	00111	001110011100000000101000000001	001110011100000000111000000001	3.13%	0.00%
9	01000	010000100000000000000000000000	010000100000000000000000000000	3.13%	1.70%
10	01001	010010100100000010000000000000	010010100100000010000000000000	3.13%	0.10%
11	01010	010100101000000100000000000000	010100101000000100000000000000	3.13%	0.10%
12	01011	010110101100000110010000001010	010110101100000110010000000010	3.13%	0.00%
13	01100	011000110000000000000000000000	011000110000001000000000000000	3.13%	0.40%
14	01101	011010110100000010000000000000	011010110100001010100000000100	3.13%	0.00%
15	01110	011100111000001101000000000000	011100111000001101000000001000	3.13%	15.50%
16	01111	011110111100001111010000001111	011110111100001111110000001111	3.13%	5.30%
17	10000	100001000000000000000000000000	100001000000000000000000000000	3.13%	0.00%
18	10001	100011000100010000100000000000	100011000100010000000000000000	3.13%	0.00%
19	10010	100101001000100000000000000000	100101001000100000000000000000	3.13%	0.30%
20	10011	100111001100110000110100110000	100111001100110000010000010000	3.13%	3.00%
21	10100	101001010001000000000000000000	101001010001000000000000000000	3.13%	12.70%
22	10101	101011010101010000100000000000	101011010101010000100000100000	3.13%	8.50%
23	10110	101101011001100001000000000000	101101011001100001000001000000	3.13%	7.30%
24	10111	101111011101110001110101110001	101111011101110001110001110001	3.13%	4.60%
25	11000	110001100010000000000000000000	110001100010000000000000000000	3.13%	0.20%
26	11001	110011100110010010100010000000	110011100110010010000010000000	3.13%	5.70%
27	11010	110101101010100100001000000000	110101101010100100000100000000	3.13%	4.70%
28	11011	110111101110110110111110111010	110111101110110110010110010010	3.13%	4.00%
29	11100	111001110011000000000000000000	111001110011001000001000000000	3.13%	0.00%
30	11101	111011110111010010100010000000	111011110111011010101010100100	3.13%	6.00%
31	11110	111101111011101101001000000000	111101111011101101001101001000	3.13%	4.50%
32	11111	111111111111111111111111111111	111111111111111111111111111111	3.13%	1.00%

contain equal or less than 2 attribute combinations or all total 5 attributes; patterns in Table 4.4 are mostly combined by 3 or 4 attributes. Therefore we have to conclude that the two matrices are different from each other.

The model will also generate the probabilities within each of the 32 latent attribute patterns from EM algorithm, and these estimates are important indicators of estimation on latent variables in the model. When we simulated the data from DINA model, we assumed an equal distribution of students in all 32 latent class so that the latent class size in any group is $1/32 = 3.125\%$. Of course, a single simulation might not generate exactly the same percentage of subjects in each latent class, but if we simulate thousands of times the average latent class sizes will be close to 3.125% in each attribute pattern. Replications could be one possible solution but this research is based on a single simulated dataset due to computation complexity. Moreover, latent class sizes are not the key purpose of the model. It is entirely possible that the discrepancies on class sizes are caused by the difference between simulation DINA model and the probabilistic model in this research. As a result, we will still use 3.125% as the theoretical latent class distributions to compare with the estimated latent class sizes from our model, for an exploratory attempt. If our estimated latent class sizes are close to 3.125% for all attribute patterns, the estimated Q-matrix will be considered identical to the true Q-matrix.

Latent class sizes in either Table 4.4 or 4.5 are very different from the theoretical ones thus we are not able to conclude that the estimated Q-matrix is identical to the true one. According to Table 4.3, we can sum up the marginal latent class sizes for each attribute: 62.5% of examinees have A1, 49.2% have A2, 78.4% have A3, 51.8% have A4, and 48.9% have A5. Although the proportions of people who have A2, A4, or A5 are close to 50%, the marginal latent class sizes of A1 and A3 are much higher than 50%. Again we are not able to conclude the latent class sizes are similar to what we designed in the simulation. This supports the conclusion that the estimated Q-matrix is different from the true Q in the first simulation study. It could also be possible that the simulated response data has different true latent class sizes than the theoretical probabilities because of randomly simulation with guessing and slipping parameters. Fortunately no extremely large estimate of latent class sizes come from our model (Table 4.3) so that we do not see obvious misspecification problems [DeCarlo, 2011].

In sum our probabilistic model is able to identify 64% to 91% Q-matrix elements in simulation study 1. When $\lambda = 11$ in penalty function, the model can achieve a high accuracy up to 91.3%. The estimated Q-matrix is not identical to the true Q-matrix according to their differences on item response patterns or latent class sizes.

Table 4.4: Different item response patterns between estimated-Q and true-Q in simulation study 1

Attribute 1 to 5		Ideal response patterns		Latent class sizes	
No.	Latent class	True Q	Estimated Q	True Q	Estimated Q
6	00101	001010010100000000000000000000	001010010100000000100000000000	3.13%	10.30%
8	00111	001110011100000001010000000001	001110011100000001110000000001	3.13%	0.00%
12	01011	010110101100000110010000001010	010110101100000110010000000010	3.13%	0.00%
13	01100	011000110000000000000000000000	011000110000001000000000000000	3.13%	0.40%
14	01101	011010110100000010000000000000	011010110100001010100000000100	3.13%	0.00%
15	01110	011100111000001101000000000000	011100111000001101000000001000	3.13%	15.50%
16	01111	011110111100001111010000001111	011110111100001111110000001111	3.13%	5.30%
18	10001	100011000100010000100000000000	100011000100010000000000000000	3.13%	0.00%
20	10011	100111001100110000110100110000	100111001100110000010000010000	3.13%	3.00%
22	10101	101011010101010000100000000000	101011010101010000100000100000	3.13%	8.50%
23	10110	101101011001100001000000000000	101101011001100001000001000000	3.13%	7.30%
24	10111	101111011101110001110101110001	101111011101110001110001110001	3.13%	4.60%
26	11001	110011100110010010100010000000	110011100110010010000010000000	3.13%	5.70%
27	11010	110101101010100100001000000000	110101101010100100000100000000	3.13%	4.70%
28	11011	110111101110110110111110111010	110111101110110110010110010010	3.13%	4.00%
29	11100	111001110011000000000000000000	111001110011001000001000000000	3.13%	0.00%
30	11101	111011110111010010100010000000	111011110111011010101010100100	3.13%	6.00%
31	11110	111101111011101101001000000000	111101111011101101001101001000	3.13%	4.50%

Table 4.5: Identical item response patterns between estimated-Q and true-Q in simulation study 1

Attribute 1 to 5		Ideal response patterns		Latent class sizes	
No.	Latent class	True Q	Estimated Q	True Q	Estimated Q
1	00000	000000000000000000000000000000	000000000000000000000000000000	3.13%	0.30%
2	00001	000010000100000000000000000000	000010000100000000000000000000	3.13%	0.10%
3	00010	000100001000000000000000000000	000100001000000000000000000000	3.13%	1.10%
4	00011	000110001100000000010000000000	000110001100000000010000000000	3.13%	0.30%
5	00100	001000010000000000000000000000	001000010000000000000000000000	3.13%	2.20%
7	00110	001100011000000001000000000000	001100011000000001000000000000	3.13%	0.10%
9	01000	010000100000000000000000000000	010000100000000000000000000000	3.13%	1.70%
10	01001	010010100100000010000000000000	010010100100000010000000000000	3.13%	0.10%
11	01010	010100101000000100000000000000	010100101000000100000000000000	3.13%	0.10%
17	10000	100001000000000000000000000000	100001000000000000000000000000	3.13%	0.00%
19	10010	100101001000100000000000000000	100101001000100000000000000000	3.13%	0.30%
21	10100	101001010001000000000000000000	101001010001000000000000000000	3.13%	12.70%
25	11000	110001100010000000000000000000	110001100010000000000000000000	3.13%	0.20%
32	11111	111111111111111111111111111111	111111111111111111111111111111	3.13%	1.00%

The model works well for simple items with single attribute but performs worse on more complicated items which require combination of at least two skills. Similarly, for skill patterns with simple attribute combinations (either one or two attributes, or all attributes), the model can come up with the same item response patterns. But for the rest their item response patterns are different. Last but not least, either estimated latent class sizes or marginal latent class sizes are different from theoretical ones designed in simulation.

4.2 Simulation study 2

In the second simulation study, we simulate response data from a irregular and complex Q-matrix which comes from the fraction subtraction exam. The simulation process is the same as the first simulation study. The same set of ten λ values are pre-determined. Table 4.6 presents the results from simulation study 2. The Q-matrix from fraction subtraction test has 15 items and 5 attributes $Q_{15\times5}$ so that total 75 elements are estimated from the model. According to Table 4.6, when all estimated continuous elements are transformed to binary values 0, 1 by 0.5 cutoff point, the models are able to identify 61.3% to 88% true Q-matrix elements. The highest correct Q-matrix identification occurs when $\lambda = 9$ with a corresponding identified proportion of 88%. Table 4.7 is the final Q-matrix information from the model with $\lambda = 9$ (red numbers are incorrectly identified elements).

Table 4.6: Simulation study 2 results

				Identified elements of the true Q-matrix					
				0.3/0.7 Cutoff		0.4/0.6 Cutoff		0.5/0.5 Cutoff	
No.	λ	-2LL	Discrepancy	Counts	Prob	Counts	Prob	Counts	Prob
1	0	26067.06	50.09	45/75	60%	50/75	66.7%	55/75	73.3%
2	0.001	26140.86	47.40	50/75	66.7%	53/75	70.7%	55/75	73.3%
3	0.05	26114.67	51.77	52/75	69.3%	54/75	72%	57/75	76%
4	1	25074.66	134.55	45/75	60%	47/75	62.7%	47/75	62.7%
5	2	25020.93	142.37	47/75	62.7%	49/75	65.3%	50/75	66.7%
6	3	24803.77	47.70	54/75	72%	56/75	74.7%	61/75	81.3%
7	6	23455.52	132.63	42/75	56%	44/75	58.7%	46/75	61.3%
8	9	22558.35	41.29	52/75	69.3%	61/75	81.3%	66/75	88%
9	11	15635.16	161.57	45/75	60%	48/75	64%	51/75	68%
10	13	11461.72	337.94	40/75	53.3%	43/75	57.3%	47/75	62.7%

To answer the primary research questions, according to the final estimated Q-matrix in Table 4.7, 9 out of 75 elements are not correctly identified (12%). For items which require only one single attribute (Item 1, 3 and 5) the model is able to identify all correct elements; for items which require two attributes (Item 8, 9, and 11) the model again has a perfect estimation accuracy; for items which require three attributes (Item 12) the model identifies 3 out of 5 (60%) attributes correctly; for items which require four attributes (Item 2, 6, 7, 10, 13, 15), 3 out of 30 (10%) elements are not correctly identified by the model; for items which require five attributes (Item 4 and 14) 4 out of 10 (40%) elements have not been recognized. In sum, identification mistakes come from items that require at least three attributes (Item 2, 4, 6, 10, 12, and 14), which supports the conclusion from simulation study 1 that our model performs worse on items which need more attributes. It is also very interesting to see that 8 out of 9 (88.9%) of the incorrectly identified elements do not believe the corresponding items require a necessary attribute in true Q-matrix (estimates are 0 while true elements are 1). In other words, our model indicates that the items need less required attributes than they should do. For example, in true Q-matrix item 4 and item 14 are supposed to require all five attributes but in our estimated Q-matrix these two items only require three attributes (A2, A3 and A4). Overall the model performs well to estimate the true Q-matrix elements in the second simulation study with the highest accuracy up to 88%.

The secondary research questions can be answered by item response patterns and latent class sizes of this simulation study shown in Table 4.8. The 32 latent classes with 5 attributes generate 10 unique item response patterns under the true Q-matrix and 12 item response patterns under the estimated Q-matrix (Table 4.9). In Table 4.9 the first six item response patterns in blue are identical under both Q-matrices while the rest ones in dark are different in the two lists. For example, the true Q-matrix is able to generate the item response pattern (101010001011000) which cannot be found in the list generated by estimated Q. Similarly the item response pattern

Table 4.7: Estimated Q-matrix in simulation study 2

	Continuous Q-matrix					Binary Q after 0.5 cutoff					True Q-matrix				
Item	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5
1	0.783	0.013	0.038	0.019	0.018	1	0	0	0	0	1	0	0	0	0
2	0.000	0.999	0.892	0.853	0.108	0	1	1	1	0	1	1	1	1	0
3	0.895	0.040	0.022	0.006	0.034	1	0	0	0	0	1	0	0	0	0
4	0.008	0.588	0.968	0.921	0.064	0	1	1	1	0	1	1	1	1	1
5	0.106	0.129	0.948	0.305	0.155	0	0	1	0	0	0	0	1	0	0
6	0.754	0.940	0.649	0.165	0.490	1	1	1	0	0	1	1	1	1	0
7	0.656	1.000	0.988	0.766	0.002	1	1	1	1	0	1	1	1	1	0
8	0.842	0.660	0.025	0.006	0.399	1	1	0	0	0	1	1	0	0	0
9	0.889	0.094	0.999	0.044	0.071	1	0	1	0	0	1	0	1	0	0
10	0.123	0.436	0.998	0.503	0.621	0	0	1	1	1	1	0	1	1	1
11	0.859	0.001	0.963	0.275	0.006	1	0	1	0	0	1	0	1	0	0
12	0.031	0.565	0.778	0.914	0.023	0	1	1	1	0	1	0	1	1	0
13	0.679	0.994	0.858	0.670	0.057	1	1	1	1	0	1	1	1	1	0
14	0.038	0.971	1.000	0.874	0.130	0	1	1	1	0	1	1	1	1	1
15	0.557	0.902	0.896	0.698	0.139	1	1	1	1	0	1	1	1	1	0

(000010000100000) generated by estimated Q-matrix is not in the list of true Q-matrix. Their corresponding pattern sizes are also different in Table 4.9. Therefore the two Q-matrices do not generate the same list of item response patterns and we have to conclude that they are not identical Q-matrices.

In Table 4.8 the theoretical class sizes in simulation studies should be $1/32 = 3.125\%$ in each of the 32 latent classes. However, the latent class sizes from our estimated results are again different from the theoretical ones. The marginal latent class sizes for each attributes are A1: 25.2%, A2: 83.45%, A3: 7.72%, A4: 83.52%, and A5: 43.36%. None of them is close to 50% (the marginal sizes designed in the simulation) except the fifth attribute. It is interesting that most people are considered to have the second and fourth attributes (above 80%) while only a few occupy the third skill (less than 10%) from our estimation. Similarly, there is an extremely high latent class size (52.8%) of attribute combination A2 and A4 (01010). DeCarlo [2011] mentioned in his paper that obtaining large estimates of the latent class sizes could indicate misspecification of the Q-matrix, such as the inclusion of an irrelevant skill. According to our estimated Q-matrix, there is only one item (Item 10) that requires the fifth attribute. Thus the fifth skill might not be necessary according to our model based on the response data. In the true Q-matrix 3 out of 15 (20%) items need the fifth attribute, which is not a frequently required attributes. Therefore the fifth attribute might be considered irrelevant because the data is simulated from a DINA model with added in guessing and slipping parameters which we do not consider in our probabilistic model. In sum we are not able to conclude that either latent class sizes or marginal latent class sizes are similar to what we have designed in the simulation. This supports the conclusion drawn from item response patterns that the estimated Q-matrix is different from the true Q in the second simulation study.

In general, for complex Q-matrices such as the one from fraction subtraction test, the model can still perform very well with 61% to 88% accuracy on estimation of Q-matrix elements in the second simulation study after binary transformation by 0.5 cutoff point. The highest accuracy of 88% is achieved when $\lambda = 9$ in the penalty function. The estimated Q-matrix and true Q-matrix are not identical according to their differences on unique item response patterns or latent class sizes. For simple items with a single or two attributes the model is able to identify all correct elements. But it performs not well when it comes to items with three or more skills. Large estimates of latent class sizes might indicate misspecification of the Q-matrix. The fifth attribute might not be required according to the model but that is not the case in the true Q-matrix. These discrepancies could be explained by fundamental difference between simulation DINA and our estimation model, or random generation of simulated response data which might not be consistent with what we have designed for the data properties, or true estimation mistakes.

Table 4.8: Item response patterns and latent class sizes in simulation study 2

Attribute 1 to 5		Ideal response patterns		Latent class sizes	
No.	Latent class	True Q	Estimated Q	True Q	Estimated Q
1	00000	000000000000000	000000000000000	3.13%	0.05%
2	00001	000000000000000	000000000000000	3.13%	0.54%
3	00010	000000000000000	000000000000000	3.13%	0.07%
4	00011	000000000000000	000000000000000	3.13%	0.00%
5	00100	000010000000000	000010000000000	3.13%	0.45%
6	00101	000010000000000	000010000000000	3.13%	0.07%
7	00110	000010000000000	000010000000000	3.13%	0.71%
8	00111	000010000000000	000010000100000	3.13%	4.69%
9	01000	000000000000000	000000000000000	3.13%	1.07%
10	01001	000000000000000	000000000000000	3.13%	0.01%
11	01010	000000000000000	000000000000000	3.13%	52.82%
12	01011	000000000000000	000000000000000	3.13%	13.79%
13	01100	000010000000000	000010000000000	3.13%	0.01%
14	01101	000010000000000	000010000000000	3.13%	0.19%
15	01110	000010000000000	010110000001010	3.13%	0.00%
16	01111	000010000000000	010110000101010	3.13%	0.34%
17	10000	101000000000000	101000000000000	3.13%	0.00%
18	10001	101000000000000	101000000000000	3.13%	1.82%
19	10010	101000000000000	101000000000000	3.13%	0.00%
20	10011	101000000000000	101000000000000	3.13%	7.03%
21	10100	101010001010000	101010001010000	3.13%	0.89%
22	10101	101010001010000	101010001010000	3.13%	0.01%
23	10110	101010001011000	101010001010000	3.13%	0.00%
24	10111	101010001111000	101010001110000	3.13%	0.23%
25	11000	101000010000000	101000010000000	3.13%	0.00%
26	11001	101000010000000	101000010000000	3.13%	11.26%
27	11010	101000010000000	101000010000000	3.13%	0.48%
28	11011	101000010000000	101000010000000	3.13%	3.35%
29	11100	101010011010000	101011011010000	3.13%	0.09%
30	11101	101010011010000	101011011010000	3.13%	0.03%
31	11110	111011111011101	111111111011111	3.13%	0.01%
32	11111	111111111111111	111111111111111	3.13%	0.00%

4.3 Empirical study

Empirical study applies the model with a real response data set from the fraction subtraction test, including 536 responses of middle school students and their responses to 15 items. The expert-designed Q-matrix is the same as the one we used in simulation study 2. However, in empirical study the expert-designed Q-matrix could be wrong and should not be the criteria for our judgment. As discussed in the earlier

Table 4.9: Unique item response patterns in simulation study 2

No.	Item response patterns		Pattern sizes	
	True Q	Estimated Q	True Q	Estimated Q
1	000000000000000	000000000000000	25%	68.35%
2	000010000000000	000010000000000	25%	1.43%
3	101000000000000	101000000000000	12.5%	8.85%
4	101000010000000	101000010000000	12.5%	15.09%
5	101010001010000	101010001010000	6.25%	0.9%
6	111111111111111	111111111111111	3.13%	0.00%
7	101010001011000	000010000100000	3.13%	4.69%
8	101010001111000	010110000001010	3.13%	0.00%
9	101010011010000	010110000101010	6.25%	0.34%
10	111011111011101	101010001110000	3.13%	0.23%
11		101011011010000		0.12%
12		111111111011111		0.01%

section, one option to select the best λ values is based on the results from simulation studies. In previous two simulations studies, when $\lambda = 9$ or 11 (small penalty effects) the model has the highest correct identification of true Q-matrix elements. As a result we will choose the models with λ equal to 9 and 11 as final models to compare the estimated Q to the designed Q. There is no better options at this time as the calculation of deviance distances is not practical. In addition, it is also interesting to look at the estimated Q-matrix which is most close to the designed Q-matrix, if we assume that the designed Q is similar to the true one and take the experts' opinions into account. Thus total three estimated Q-matrices will be discussed here in the empirical study.

Table 4.10 demonstrates the estimated results from our model. The estimated Q-matrix has 75 elements $Q_{15\times5}$ with a set of the same ten λ values as simulation study 1 and 2. The proportions of consistency of elements between estimated and expert-designed Q-matrices are from 57.3% to 72% among all models, if we use 0.5 as the cutoff point. The highest consistency comes with the $\lambda = 0.001$, which indicates a very strong penalty effect. Thus the estimated Q-matrix is most close to the designed

Q-matrix (72% identical elements) when $\lambda = 0.001$. However based on the best λ's which are either 9 or 11 from the simulation studies, the estimated Q-matrices have 52 out of 75 (69.3%) consistent elements with the expert-designed Q-matrix. The results indicate that about 30% of the elements might be possibly misspecificated in the expert-designed Q-matrix. Table 4.11 to 4.13 demonstrate the estimated Q-matrices from the three models with $\lambda = 9$, 11, or 0.001, compared with the expert-designed Q (red numbers indicate the incorrectly identified elements).

Table 4.10: Empirical study results

				Consistent elements with the expert-designed Q-matrix					
				0.3/0.7 Cutoff		0.4/0.6 Cutoff		0.5/0.5 Cutoff	
No.	λ	-2LL	Discrepancy	Counts	Prob	Counts	Prob	Counts	Prob
1	0	6556.15	71.53	46/75	61.3%	48/75	64%	52/75	69.3%
2	0.001	6574.38	61.84	44/75	58.7%	50/75	66.7%	54/75	72%
3	0.05	6491.15	68.92	44/75	58.7%	50/75	66.7%	51/75	68%
4	1	5460.46	164.77	46/75	61.3%	48/75	64%	50/75	66.7%
5	2	4928.05	203.92	44/75	58.7%	48/75	64%	48/75	64%
6	3	2516.49	344.47	33/75	44%	36/75	48%	43/75	57.3%
7	6	2208.01	137.31	47/75	62.7%	50/75	66.7%	52/75	69.3%
8	9	4467.84	49.35	39/75	52%	46/75	61.3%	52/75	69.3%
9	11	4605.37	46.93	32/75	42.7%	45/75	60%	52/75	69.3%
10	13	2305.89	79.64	36/75	48%	40/75	53.3%	43/75	57.3%

The primary research questions are answered by the three models in Table 4.11 to 4.13. Models with $\lambda = 9$ and 11 have 23 out of 75 (30.7%) inconsistent elements between estimated Q-matrix and designed Q-matrix, while the model with $\lambda = 0.001$ has 21 out of 75 (28%) inconsistent elements. According to the estimated Q-matrix when $\lambda = 9$ in Table 4.11, item 1, 2, 4, 5, 7 to 15 (13 out of 15) have inconsistent elements with the designed Q-matrix and that accounts to 86.7% of total 15 items. In most cases the estimated Q-matrix does not require the same amount of skills designed in the original Q-matrix, which could be reasonable if we believe that people may have different strategies to solve these items and sometimes some strategies need less required skills. For example if we look at the 7th item $4\frac{1}{3} - 2\frac{4}{3}$ which requires

Table 4.11: Estimated Q-matrix when $\lambda = 9$ in empirical study

		Continuous Q-matrix					Binary Q after 0.5 cutoff					Expert-designed Q				
No.	Item	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5
1	$\frac{3}{4}-\frac{3}{8}$	0.631	0.157	0.202	0.660	0.137	1	0	0	1	0	1	0	0	0	0
2	$3\frac{1}{2}-2\frac{3}{2}$	0.121	0.131	0.959	0.222	0.226	0	0	1	0	0	1	1	1	1	0
3	$\frac{6}{7}-\frac{4}{7}$	1.000	0.117	0.390	0.043	0.127	1	0	0	0	0	1	0	0	0	0
4	$3-2\frac{1}{5}$	0.153	0.745	0.548	0.723	0.178	0	1	1	1	0	1	1	1	1	1
5	$3\frac{7}{8}-2$	0.111	0.309	0.238	0.041	0.077	0	0	0	0	0	0	0	1	0	0
6	$4\frac{4}{12}-2\frac{7}{12}$	0.959	0.572	0.904	0.703	0.111	1	1	1	1	0	1	1	1	1	0
7	$4\frac{1}{3}-2\frac{4}{3}$	0.731	0.363	0.826	0.780	0.128	1	0	1	1	0	1	1	1	1	0
8	$\frac{11}{8}-\frac{1}{8}$	0.216	0.090	0.213	0.118	0.052	0	0	0	0	0	1	1	0	0	0
9	$3\frac{4}{5}-3\frac{2}{5}$	0.572	0.603	0.726	0.711	0.382	1	1	1	1	0	1	0	1	0	0
10	$2-\frac{1}{3}$	0.471	0.039	0.125	0.089	0.175	0	0	0	0	0	1	0	1	1	1
11	$4\frac{5}{7}-1\frac{4}{7}$	0.437	0.072	0.261	0.172	0.337	0	0	0	0	0	1	0	1	0	0
12	$7\frac{3}{5}-\frac{4}{5}$	0.852	0.654	0.736	0.576	0.142	1	1	1	1	0	1	0	1	1	0
13	$4\frac{1}{10}-2\frac{8}{10}$	0.534	0.901	0.438	0.863	0.102	1	1	0	1	0	1	1	1	1	0
14	$4-1\frac{4}{3}$	0.019	0.331	0.649	0.868	0.653	0	0	1	1	1	1	1	1	1	1
15	$4\frac{1}{3}-1\frac{5}{3}$	0.820	0.734	0.215	0.962	0.461	1	1	0	1	0	1	1	1	1	0

Table 4.12: Estimated Q-matrix when $\lambda = 11$ in empirical study

		Continuous Q-matrix					Binary Q after 0.5 cutoff					Expert-designed Q				
No.	Item	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5
1	$\frac{3}{4}-\frac{3}{8}$	0.891	0.038	0.155	0.587	0.276	1	0	0	1	0	1	0	0	0	0
2	$3\frac{1}{2}-2\frac{3}{2}$	0.149	0.222	0.054	0.911	0.308	0	0	0	1	0	1	1	1	1	0
3	$\frac{6}{7}-\frac{4}{7}$	0.177	0.162	0.188	0.301	0.399	0	0	0	0	0	1	0	0	0	0
4	$3-2\frac{1}{5}$	0.519	0.513	0.360	0.710	0.664	1	1	0	1	1	1	1	1	1	1
5	$3\frac{7}{8}-2$	0.138	0.638	0.336	0.225	0.091	0	1	0	0	0	0	0	1	0	0
6	$4\frac{4}{12}-2\frac{7}{12}$	0.314	0.447	0.603	0.360	0.435	0	0	1	0	0	1	1	1	1	0
7	$4\frac{1}{3}-2\frac{4}{3}$	0.581	0.491	0.261	0.956	0.289	1	0	0	1	0	1	1	1	1	0
8	$\frac{11}{8}-\frac{1}{8}$	0.170	0.422	0.190	0.269	0.071	0	0	0	0	0	1	1	0	0	0
9	$3\frac{4}{5}-3\frac{2}{5}$	0.887	0.609	0.922	0.214	0.408	1	1	1	0	0	1	0	1	0	0
10	$2-\frac{1}{3}$	0.116	0.055	0.033	0.882	0.227	0	0	0	1	0	1	0	1	1	1
11	$4\frac{5}{7}-1\frac{4}{7}$	1.000	0.207	0.688	0.370	0.357	1	0	1	0	0	1	0	1	0	0
12	$7\frac{3}{5}-\frac{4}{5}$	0.977	0.379	0.567	0.906	0.216	1	0	1	1	0	1	0	1	1	0
13	$4\frac{1}{10}-2\frac{8}{10}$	0.893	0.540	0.204	0.761	0.085	1	1	0	1	0	1	1	1	1	0
14	$4-1\frac{4}{3}$	0.937	0.753	0.470	0.700	0.696	1	1	0	1	1	1	1	1	1	1
15	$4\frac{1}{3}-1\frac{5}{3}$	0.658	0.783	0.419	0.671	0.682	1	1	0	1	1	1	1	1	1	0

Table 4.13: Estimated Q-matrix when $\lambda = 0.001$ in empirical study

		Continuous Q-matrix					Binary Q after 0.5 cutoff					Expert-designed Q				
No.	Item	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5	A1	A2	A3	A4	A5
1	$\frac{3}{4} - \frac{3}{8}$	0.998	0.108	0.440	0.309	0.648	1	0	0	0	1	1	0	0	0	0
2	$3\frac{1}{2} - 2\frac{3}{2}$	0.407	0.143	0.815	0.002	0.006	0	0	1	0	0	1	1	1	1	0
3	$\frac{6}{7} - \frac{4}{7}$	0.907	0.023	0.065	0.023	0.137	1	0	0	0	0	1	0	0	0	0
4	$3 - 2\frac{1}{5}$	1.000	0.276	0.725	0.764	0.854	1	0	1	1	1	1	1	1	1	1
5	$3\frac{7}{8} - 2$	0.010	0.164	0.251	0.000	0.993	0	0	0	0	1	0	0	1	0	0
6	$4\frac{4}{12} - 2\frac{7}{12}$	0.862	0.256	0.944	0.727	0.013	1	0	1	1	0	1	1	1	1	0
7	$4\frac{1}{3} - 2\frac{4}{3}$	0.737	0.071	0.908	0.610	0.040	1	0	1	1	0	1	1	1	1	0
8	$\frac{11}{8} - \frac{1}{8}$	0.900	0.059	0.135	0.359	0.011	1	0	0	0	0	1	1	0	0	0
9	$3\frac{4}{5} - 3\frac{2}{5}$	0.601	0.392	0.964	0.985	0.911	1	0	1	1	1	1	0	1	0	0
10	$2 - \frac{1}{3}$	0.995	0.037	0.063	0.562	0.212	1	0	0	1	0	1	0	1	1	1
11	$4\frac{5}{7} - 1\frac{4}{7}$	0.941	0.055	0.132	0.464	0.154	1	0	0	0	0	1	0	1	0	0
12	$7\frac{3}{5} - \frac{4}{5}$	0.927	0.111	0.926	0.112	0.457	1	0	1	0	0	1	0	1	1	0
13	$4\frac{1}{10} - 2\frac{8}{10}$	0.824	0.162	0.697	0.981	0.202	1	0	1	1	0	1	1	1	1	0
14	$4 - 1\frac{4}{3}$	0.828	0.350	0.961	0.048	0.981	1	0	1	0	1	1	1	1	1	1
15	$4\frac{1}{3} - 1\frac{5}{3}$	0.004	0.207	0.984	0.990	0.007	0	0	1	1	0	1	1	1	1	0

4 attributes in the expert-designed Q but only needs 3 attributes in the estimated Q, the second attribute is considered unnecessary in the model. Record that the five attributes are A1: subtract basic fractions; A2: reduce and simplify; A3: separate whole from fraction; A4: borrow from whole; and A5: convert whole to fraction. It is possible that students can solve the item without using the second attribute (reduce and simplify). The strategy could be transfer $4\frac{1}{3}$ to $3\frac{4}{3}$ and then calculate $3\frac{4}{3} - 2\frac{4}{3} = 1$. The whole process does not have to use the skill of reduce and simplify. However, it is interesting that items 5, 8, 10, and 11 do not require any attribute in the estimated Q-matrix. It is reasonable that the 5th item $3\frac{7}{8} - 2$ might not need the third attribute (separate whole from fraction) to get it right. Item 8 and 11 might require at least the first attribute but it is hard to define what is subtracting basic fractions at the first place. However item 10 may need some skills to the researcher's perspective and estimation could be wrong. Thus it is not appropriate to rely only on the estimation results from model solely without considering experts' opinions.

Similarly, according to the estimated Q-matrix in Table 4.12 when $\lambda = 11$, only two items 11 and 12 (2 out of 15 is 13.3%)have all elements consistent to the designed Q-matrix. Similar to the model with $\lambda = 9$ the estimated Q-matrix does not require as many attributes as needed in the designed Q-matrix, especially for A1 to A3. However, unlike the first model, in this model the third item $\frac{6}{7} - \frac{4}{7}$ and the eighth item $\frac{11}{8} - \frac{1}{8}$ are considered to require no attribute at all. Besides, item 5 is estimated to require the second attribute (reduce and simplify) instead of A3 (separate whole from fraction). Item 10 needs only the fourth attribute (borrow from whole) in estimated Q, which might be understandable to some extent based on possible different strategies. However the model consider the first item: $\frac{3}{4} - \frac{3}{8}$ to require the fourth skill (borrow from whole) but in fact there is no whole number associated with the fractions in the item. It is surprising to see the same issue happens in the model with $\lambda = 9$ too. It could be an estimation mistake the model performs, or the different meanings of attributes generated in the estimated Q-matrix. Moreover, the last item: $4\frac{1}{3} - 1\frac{5}{3}$ in the estimated Q-matrix requires attribute 1, 2, 4 and 5 while in the designed Q-matrix it requires 1, 2, 3, and 4. Again the difference on the 3rd and 4th attributes may account to different strategies. It is entirely possible here that either combinations of four attributes can lead to the correct answer of the last item.

In sum, estimated Q-matrices generated from the models with $\lambda = 9$ or 11 can have up to 70% consistent elements with the designed Q-matrix. Some of the discrepancies are reasonable that we may have different strategies to solve the problems. For example, both models indicate that students without any required attribute are still able to get item 8: $\frac{11}{8} - \frac{1}{8}$ right as long as they have a simple subtraction skill of calculating $11 - 1$ while keeping the denominator constant. However some of the discrepancies make no sense; for example, item 10: $2 - \frac{1}{3}$ requires no skill in the estimated Q-matrix with $\lambda = 9$ and the first item $\frac{3}{4} - \frac{3}{8}$ requires the fourth attribute (borrow from whole) in the estimated Q with $\lambda = 11$. In addition, in simulation studies the model is not able to estimate all true Q-matrix. It is entirely possible that the models could make estimation mistakes. These discrepancies could also be caused by different meanings of attributes generated by the model. It is too arbitrary if we only depend on the model without considering the experts' opinions. As a result it is necessary to investigate the estimated Q-matrix that is most close to the expert-designed Q in Table 4.13, with $\lambda = 0.001$.

According to the model with $\lambda = 0.001$ in Table 4.13, 54 out of 75 elements (72%) are consistent between estimated Q and expert-designed Q, and the rest of 21 (28%) elements are different. In general, no item is estimated to require no attribute in this model, but the model concludes that the second attribute (reduce and simplify) is not required by all 15 items. On the other hand, in experts' opinion, half of the items need the second attribute. This result strongly supports the findings by DeCarlo [2011, 2012] that an irrelevant attribute might have been included in the designed fraction subtraction Q-matrix. One possible reason is that the second attribute always comes with the first attribute, but that is the same case for the 4th and 5th attributes. Another possible explanation is based on the different strategies used to achieve the right answers. It is entirely possible that students can get the answers right without the unclear defined skill: reduce and simplify, such as the item 7 we discussed above. In addition to this finding, some items require less attributes in estimated Q-matrix than we expected in designed Q-matrix, which is similar to the previous two models. For instance, experts considered that student should have four attribute A1 to A4 to solve the second item: $3\frac{1}{2} - 2\frac{3}{2}$. However our model tells that as long as students had the third attribute: separate whole from fraction, they would be able to get the answer right. Likewise, item 10, 11, 12, 14, and 15 also require less attributes than what experts believed. To our surprise, Item 9: $3\frac{4}{5} - 3\frac{2}{5}$ needs four attributes to get it right in estimated Q, but in experts' perspectives this item only requires A1 and

A3.

It is really hard to judge which Q-matrix assignment is correct. One possible solution is to fit the DINA model to the response data with estimated Q's and designed Q separately, and then compare the goodness of fit. The log-likelihood of DINA model with the designed Q is: -3555.116, while the log-likelihoods of estimated Q's of the model with $\lambda = 9$ is: -3821.356; $\lambda = 11$ is: -3722.744; and $\lambda = 0.001$ is: -3405.737. It is interesting that the estimated Q from the model with $\lambda = 0.001$ fits better than the others due to its largest log-likelihood value. Therefore, if we apply the DINA model to classify students into different attribute groups in the fraction subtraction test, the modified Q-matrix by the model with $\lambda = 0.001$ is the best to be used.

All in all, the three models come out with up to 70% consistency with the designed Q-matrix, and they find out that the items require less attributes in the corresponding estimated Q than what they need in the expert-designed Q. Misspecification of designed Q-matrix is supported with evidence of these discrepancies. However, these discrepancies might be caused by some other reasons too, and they should be carefully discussed not only based on the model results, but also with professional suggestions of test makers.

To answer the secondary research questions in empirical study, the item response patterns and latent class sizes for the three models are shown from Table 4.14 to Table 4.16. When $\lambda = 9$ the ideal responses of item 5, 8, 10, and 11 are correct for all attribute patterns in estimated Q-matrix because these items are considered to require none of the five attributes. The same situation happens to item 4 and 8 in the model with $\lambda = 11$ in Table 4.15. In the last model with $\lambda = 0.001$ there is no such issue because every item requires at least one attribute in the estimated Q in Table 4.16. However, a student without only the second attribute is also able to get all items correct and the latent classes (10111) and (11111) will generate the

Table 4.14: Item response patterns and latent class sizes when $\lambda = 9$ in empirical study

Attribute 1 to 5		Item response patterns		Latent class sizes
No.	Latent class	Expert-designed Q	Estimated Q	Estimated Q
1	00000	000000000000000	000010010110000	2.56%
2	00001	000000000000000	000010010110000	0.45%
3	00010	000000000000000	000010010110000	1.74%
4	00011	000000000000000	000010010110000	1.48%
5	00100	000010000000000	010010010110000	0.92%
6	00101	000010000000000	010010010110000	0.68%
7	00110	000010000000000	010010010110000	1.93%
8	00111	000010000000000	010010010110010	2.51%
9	01000	000000000000000	000010010110000	1.77%
10	01001	000000000000000	000010010110000	2.17%
11	01010	000000000000000	000010010110000	0.82%
12	01011	000000000000000	000010010110000	1.02%
13	01100	000010000000000	010010010110000	2.69%
14	01101	000010000000000	010010010110000	1.56%
15	01110	000010000000000	010110010110000	1.87%
16	01111	000010000000000	010110010110010	7.87%
17	10000	101000000000000	001010010110000	0.54%
18	10001	101000000000000	001010010110000	0.40%
19	10010	101000000000000	101010010110000	1.44%
20	10011	101000000000000	101010010110000	3.05%
21	10100	101010001010000	011010010110000	0.48%
22	10101	101010001010000	011010010110000	1.87%
23	10110	101010001011000	111010110110000	1.63%
24	10111	101010001111000	111010110110010	6.34%
25	11000	101000010000000	001010010110000	0.60%
26	11001	101000010000000	001010010110000	2.16%
27	11010	101000010000000	101010010110101	2.73%
28	11011	101000010000000	101010010110101	28.85%
29	11100	101010011010000	011010010110000	3.11%
30	11101	101010011010000	011010010110000	6.05%
31	11110	111011111011101	111111111111101	4.48%
32	11111	111111111111111	111111111111111	4.24%

same item response patterns of all correct answers. Discrepancies of item response patterns generated by estimated Q-matrices and the designed Q-matrix do exist for all three models according to the unique item response patterns Table 4.17 to 4.19. For example, according to Table 4.19 which is for model with $\lambda = 0.001$, 10 item response patterns come from expert-designed Q-matrix and 14 item response patterns come from estimated Q-matrix, which clearly states that our estimated Q is definitely

Table 4.15: Item response patterns and latent class sizes when $\lambda = 11$ in empirical study

Attribute 1 to 5		Item response patterns		Latent class sizes
No.	Latent class	Expert-designed Q	Estimated Q	Estimated Q
1	00000	000000000000000	001000010000000	2.58%
2	00001	000000000000000	001000010000000	1.28%
3	00010	000000000000000	011000010100000	1.54%
4	00011	000000000000000	011000010100000	0.96%
5	00100	000010000000000	001001010000000	0.54%
6	00101	000010000000000	001001010000000	0.64%
7	00110	000010000000000	011001010100000	2.48%
8	00111	000010000000000	011001010100000	5.94%
9	01000	000000000000000	001010010000000	1.04%
10	01001	000000000000000	001010010000000	1.08%
11	01010	000000000000000	011010010100000	0.97%
12	01011	000000000000000	011010010100000	1.18%
13	01100	000010000000000	001011010000000	0.54%
14	01101	000010000000000	001011010000000	0.97%
15	01110	000010000000000	011011010100000	5.58%
16	01111	000010000000000	011011010100000	7.45%
17	10000	101000000000000	001000010000000	0.85%
18	10001	101000000000000	001000010000000	0.77%
19	10010	101000000000000	111000110100000	2.79%
20	10011	101000000000000	111000110100000	1.23%
21	10100	101010001010000	001001010010000	1.78%
22	10101	101010001010000	001001010010000	2.41%
23	10110	101010001011000	111001110111000	3.26%
24	10111	101010001111000	111001110111000	3.02%
25	11000	101000010000000	001010010000000	1.09%
26	11001	101000010000000	001010010000000	0.63%
27	11010	101000010000000	111010110100100	4.54%
28	11011	101000010000000	111110110100111	3.38%
29	11100	101010011010000	001011011010000	3.37%
30	11101	101010011010000	001011011010000	23.60%
31	11110	111011111011101	111011111111100	5.46%
32	11111	111111111111111	111111111111111	7.03%

different from the expert-designed one. Estimated Q-matrix from the model with $\lambda = 9$ generates 13 item response patterns in Table 4.17, while the estimated Q-matrix from the model with $\lambda = 11$ has 16 item response patterns in Table 4.18. Both of the two models have more item response patterns than the designed Q which generates 10 item response patterns. Only one item response pattern of all correct answers happens in both lists in Table 4.17 and 4.18. Thus all three estimated Q-

Table 4.16: Item response patterns and latent class sizes when $\lambda = 0.001$ in empirical study

Attribute 1 to 5		Item response patterns		Latent class sizes
No.	Latent class	Expert-designed Q	Estimated Q	Estimated Q
1	00000	000000000000000	000000000000000	0.00%
2	00001	000000000000000	000010000000000	1.00%
3	00010	000000000000000	000000000000000	1.00%
4	00011	000000000000000	000010000000000	0.00%
5	00100	000010000000000	010000000000000	6.00%
6	00101	000010000000000	010010000000000	0.00%
7	00110	000010000000000	010000000000001	0.00%
8	00111	000010000000000	010010000000001	1.00%
9	01000	000000000000000	000000000000000	8.00%
10	01001	000000000000000	000010000000000	5.00%
11	01010	000000000000000	000000000000000	0.00%
12	01011	000000000000000	000010000000000	6.00%
13	01100	000010000000000	010000000000000	0.00%
14	01101	000010000000000	010010000000000	0.00%
15	01110	000010000000000	010000000000001	0.00%
16	01111	000010000000000	010010000000001	0.00%
17	10000	101000000000000	001000010010000	1.00%
18	10001	101000000000000	101010010010000	1.00%
19	10010	101000000000000	001000010110000	0.00%
20	10011	101000000000000	101010010110000	0.00%
21	10100	101010001010000	011000010011000	0.00%
22	10101	101010001010000	111010010011010	5.00%
23	10110	101010001011000	011001110111101	0.00%
24	10111	101010001111000	111111111111111	0.00%
25	11000	101000010000000	001000010010000	0.00%
26	11001	101000010000000	101010010010000	2.00%
27	11010	101000010000000	001000010110000	0.00%
28	11011	101000010000000	101010010110000	21.00%
29	11100	101010011010000	011000010011000	0.00%
30	11101	101010011010000	111010010011010	37.00%
31	11110	111011111011101	011001110111101	0.00%
32	11111	111111111111111	111111111111111	5.00%

matrices from the model are different from the designed Q-matrix according to their different item response patterns.

The latent class sizes from our estimated results vary differently across the three models. In model with $\lambda = 9$ the proportion in attribute pattern with A1 A2 A4 and A5 (11011) is extremely higher (28.85%) than other latent class sizes. The next model with $\lambda = 11$ also has a large latent class size up to 23.6% of the attribute

Table 4.17: Unique item response patterns when $\lambda = 9$ in empirical study

No.	Item response patterns		Pattern sizes
	Expert-designed Q	Estimated Q	Estimated Q
1	111111111111111	111111111111111	4.24%
2	000000000000000	000010010110000	12.01%
3	000010000000000	001010010110000	3.70%
4	101000000000000	010010010110000	7.78%
5	101010001010000	010010010110010	2.51%
6	101010001011000	010110010110000	1.87%
7	101010001111000	010110010110010	7.87%
8	101000010000000	011010010110000	11.51%
9	101010011010000	101010010110000	4.49%
10	111011111011101	101010010110101	31.58%
11		111010110110000	1.63%
12		111010110110010	6.34%
13		111111111111101	4.48%

Table 4.18: Unique item response patterns when $\lambda = 11$ in empirical study

No.	Item response patterns		Pattern sizes
	Expert-designed Q	Estimated Q	Estimated Q
1	111111111111111	111111111111111	7.03%
2	000000000000000	001000010000000	5.48%
3	000010000000000	001001010000000	1.18%
4	101000000000000	001001010010000	4.19%
5	101000010000000	001010010000000	3.84%
6	101010001010000	001011010000000	1.51%
7	101010001011000	001011011010000	26.97%
8	101010001111000	011000010100000	2.50%
9	101010011010000	011001010100000	8.42%
10	111011111011101	011010010100000	2.15%
11		011011010100000	13.03%
12		111000110100000	4.02%
13		111001110111000	6.28%
14		111010110100100	4.54%
15		111011111111100	5.46%
16		111110110100111	3.38%

pattern (11101) while all the other latent class sizes are less than 8%. The last model with $\lambda = 0.001$ has two comparatively high latent class sizes (11011, 11101) which are 21% and 37% separately. It is interesting to see that the three models generate comparable high latent class sizes in either (11011) or (11101) or both. The marginal

Table 4.19: Unique item response patterns when $\lambda = 0.001$ in empirical study

No.	Item response patterns		Pattern sizes
	Expert-designed Q	Estimated Q	Estimated Q
1	000000000000000	000000000000000	9.00%
2	000010000000000	000010000000000	12.00%
3	111111111111111	111111111111111	5.00%
4	101000000000000	001000010010000	1.00%
5	101000010000000	001000010110000	0.00%
6	101010001010000	010000000000000	6.00%
7	101010001011000	010000000000001	0.00%
8	101010001111000	010010000000000	0.00%
9	101010011010000	010010000000001	1.00%
10	111011111011101	011000010011000	0.00%
11		011001110111101	0.00%
12		101010010010000	3.00%
13		101010010110000	21.00%
14		111010010011010	42.00%

latent class sizes of the three models are listed in Table 4.20.

The comparison of latent class sizes also indicates that there might be some misspecifications of Q-matrix, for example, the second attribute is not necessary in all items according to the last model, while some of the items might not need any of the five attribute based on the results from the first two models. More than half of the students are estimated to occupy most of the attributes in Table 4.20 regardless of which model is used. It is interesting that the first model with $\lambda = 9$ shows a large proportion of students (71.99%) who have occupies the fourth attribute (borrow from whole) while the last model with $\lambda = 0.001$ indicates only 34% of the examinees who are estimated to have the skill. Some discrepancies also exist for the third attribute (separate whole from fraction) but generally the distributions of marginal latent class sizes across the three models are consistent except for the fourth attribute.

In conclusion the empirical study is an exploratory analysis to estimate the true Q-matrix from a real response data of the fraction subtraction test. Although different models have different results, their estimated Q-matrices still have 55% to 75%

Table 4.20: Marginal latent class sizes in empirical study

λ	A1	A2	A3	A4	A5
9	67.97%	71.98%	48.24%	71.99%	70.71%
11	65.21%	67.91%	74.07%	56.81%	61.57%
0.001	72.00%	84.00%	54.00%	34.00%	84.00%
A1	subtract basic fractions				
A2	reduce and simplify				
A3	separate whole from fraction				
A4	borrow from whole				
A5	convert whole to fraction				

consistent elements with the designed Q-matrix. We select three models based on the results from simulation studies and the smallest discrepancy distance from the expert-designed Q-matrix. There is no uniform criterion to judge which model is the best to use. Although the models with $\lambda = 9$ or 11 perform well in simulation studies, their estimated Q-matrices have at least 30 inconsistent elements from the designed Q, and some of these differences are hard to interpret. In addition, we cannot completely ignore experts' opinions so that the estimated Q-matrix which is closest to the designed Q (72% consistent elements) from the model with $\lambda = 0.001$ is also selected here for a supplemental analysis. In addition, all the three estimated Q-matrices are not identical to the expert-designed Q according to their item response patterns. These discrepancies together with the estimated latent class sizes have indicated possible misspecifications of the expert-designed Q-matrix. For example, there might be a unnecessary skill and some items in fact require less attributes than we originally expected. Due to those conflicts it is important to bring domain experts for a further discussion on the design of Q-matrix, based on the results of our models.

Chapter 5

Discussions

The primary purpose of the dissertation is to set up a mathematical framework to estimate Q-matrix based on item response data. Two methods are developed: the first one is based on a non-linear transformation of Q-matrix and a penalized sum of square errors established to estimate the Q-matrix elements; the second one is based on item response function of q's in which a penalized likelihood function is maximized to estimate the elements by EM algorithm. This research focuses on the second method and develops the probabilistic model with penalized biased estimation. The q's can be considered as the probabilities that the items require a skill, or the proportions of persons who need the skill to get the item right. There are three model assumptions: given a value of the latent skill pattern the item responses are independent from each other; no additional guessing or slipping parameters are considered; and the model is conjunctive while lacking any one of the skills would lead to failure of response.

The primary two research questions include how accurate the method performs on estimation of the Q-matrix, and what the differences would be between estimated Q-matrix and designed Q-matrix in a real situation. These questions are answered by the proportion of correctly identified elements of estimated Q-matrices in simulation studies, and the proportion of identical elements between estimated Q-matrix and the expert-designed Q-matrix in the empirical study. The secondary research questions include whether the estimated Q-matrix is identical to the true or designed Q, and how the latent class sizes are distributed. These questions are answered by the comparison of unique ideal item response patterns between Q-matrices and the estimated results of the latent class sizes.

Two simulation studies are conducted to evaluate the feasibility and goodness of the estimation method. Two Q-matrices from de la Torre [2009b] are considered as true Q's to simulate 2000 response data through a DINA model with both guessing and slipping parameters equal to 0.1. Then our model estimates the Q-matrices from the simulated responses and compare with the true Q-matrices on the estimation accuracy. One empirical study applies the real response data from fraction subtraction test used by de la Torre [2009b] to estimate the Q-matrix directly from the model and then compares the results with the Q-matrix designed by experts. Q-matrix elements are considered as parameters in the model with latent variables of attributes, and the parameters are estimated through EM algorithms with $100,000$ iterations in the study.

Challenges of the model include determination of attribute dimensions, identification of attributes in estimated Q-matrices, and selection of penalty functions (λ values). Total number of attributes used in the model might be determined by the model goodness of fit, such as likelihood value, AIC, BIC. However this issue is not discussed in this research, because matrices with different columns are hard to compare. The identification of attributes is solved by the minimum discrepancy distances between the estimated Q-matrices and the true Q's or designed Q. Although the selection of the complexity parameter λ in penalty functions is designed by the prediction accuracy of deviance distances through cross-validation of a large portion of candidate values, it is not computation feasible at this time. Besides, in simulation studies the information given by cross-validation might be redundant because the best model should have the highest correct identification of true Q-matrices and we can find the information from the estimation results. Thus in this research we have pre-determined ten λ values as the candidate pool, and select the best λ according to its corresponding proportion of identified elements of true Q-matrix after a full binary transformation of estimated Q-matrix by 0.5 cutoff point in simulation studies. In empirical study, we use the same set of potential λ values but select the best models based on the results from simulation studies. For example, if simulation studies indicate that the best model has a penalty function with $\lambda = 10$, then we should use $\lambda = 10$ in empirical study as the final model regardless of how many of consistent elements between the estimated Q-matrix and designed Q-matrix.

The results from simulation studies demonstrate that about 64% to 91% Q-matrix elements can be correctly identified by the model in simulation study 1, and 61% to 88% correct elements in simulation study 2. The model with $\lambda = 11$ can achieve the highest accuracy up to 91.3% in the first simulation study and the model with $\lambda = 9$ has 88% correct identification in the second simulation. In empirical study the model has comparatively lower percentages from 55% to 75% of consistent elements between estimated Q-matrices and designed Q-matrices. Based on the results from simulation studies, $\lambda = 9$ and $\lambda = 11$ are selected as final models. Both of them have 52 out of 75 (70%) consistent elements with the designed Q but the discrepancies are different. For example, item 10: $2 - \frac{1}{3}$ requires no skill in the estimated Q-matrix with $\lambda = 9$ but one attribute A4 (borrow from whole) in the one generated by $\lambda = 11$. However in expert-designed Q item 10 need four skills (A1, A3, A4, A5). Because some of the discrepancies are hard to interpret and we cannot completely ignore experts' opinions, we also select another model with $\lambda = 0.001$ which has the highest counts of consistent elements (54 out of 75 which is 72%) in empirical study. This model indicates that the second attribute of the expert-designed Q-matrix is not required by all items. This result of an irrelative attribute is consistent with the findings by DeCarlo [2011, 2012]. In sum, the empirical study is similar to an exploratory research of a confirmatory factor analysis and there is no single criterion to judge which model is the best to use.

All the final estimated Q-matrices generated by either simulation studies or the empirical study are not identical to the true Q's or expert-designed Q, according to their different item response patterns. Their estimated latent class sizes or marginal latent class sizes are also different from the theoretical ones in simulation studies. In simulation studies, the model works well for simple items with one or two attributes but performs worse on more complicated items which require combinations of three or more skills. These discrepancies could be explained by the non-perfect estimation performance, fundamental difference between simulation DINA model and the probabilistic model we designed, or random simulation of response data which might not be consistent with what we have designed for the data properties. The discrepancies of item response patterns together with the estimated latent class sizes in empirical study have indicated possible misspecifications of the expert-designed Q-matrix. For example, there might be a unnecessary skill identified in the Q-matrix and some items in fact require less attributes than we originally expected. However, some of these discrepancies between estimated Q and designed Q might be caused by different strategies the students used to get the right answers. Due to those conflicts it is important to bring domain experts for a further discussion on the design of Q-matrix.

To our knowledge, this research is the first attempt to explore a statistical approach to purely estimate all Q-matrix elements totally based on item responses without considering experts' opinions. Most of past studies on Q-matrix estimation are built on developments of existing cognitive diagnostic models [Chiu et al., 2009; DeCarlo, 2011, 2012; Liu et al., 2011b,a, 2012; de la Torre, 2008], for example, cluster analysis, generalized DINA model, higher-order DINA model, or reparameterization of DINA model. This research establishes a new probabilistic model with attributes as latent variables and Q-matrix elements as parameters. The focus is on the estimation accuracy of parameters rather than classification of students. On one hand, the model is able to find out backward the estimated Q which is the best fit to the students' item responses, rather than taking the subjective designed Q-matrix for granted; on the other hand, the estimated Q-matrix from the model can be used forward into the existing cognitive diagnostic models, such as DINA or NIDA, for a possibly better classification of students' abilities, according to different assumptions. The final goal is to provide multiple classifications of students from CDMs based on possible Q-matrices, and compare the results to reach a reasonable one. Psychometrician can also use the model to check if there is any misspecification of expert-designed Q-matrix. It is possible that students have different strategies from what the experts' ideas towards how to solve the problems. A different Q-matrix might be reasonable too. If no existing Q-matrix is provided, researchers can generate an initial estimated Q-matrix and then refine it together with experts. However, the model depends on the both the test and examinees, so that it is not appropriate to generate the results outside the population.

However, the model has limitations and there are ways to improve it. First of all, the model assumes the simplest situations, which may not be true in real applications. For example, the correct item responses are totally determined by the occupation of required attributes, and no guessing or slipping factors of the students are considered. These assumptions are even simper than those in DINA model, which have been criticized to be relatively novel in some cases [de la Torre, 2008]. Besides, depending on the construction of parameter q, the model seems to have a built-in "guessing" on the item responses. For example, because q is considered as the probability that the item requires the skill, $1 - q$ will be the probability that the item does not require the skill. For students who have no attribute at all, when q's are not equal to 100%, there are still some possibilities that the item does not require all attributes. As a result, the students come up with a "guessing" opportunity to get the item right. This "guessing" factor depends on the possibilities of an item not requiring a specific skill, rather than the students' personal reasons, so it is not a conflict with the assumptions we made for our model. But the "guessing" will indeed affect the results of students' responses and their classification to attribute patterns.

Besides, the number of skills in estimated Q-matrix is assumed to be known in this research because of comparisons with true and designed Q-matrices. For a pure exploratory analysis the attribute dimensions are not available in real situations, and the meanings of attributes are unclear. Furthermore, in simulation studies, the response data should be replicated multiple times, and an average estimated Q-matrix could be drawn from different models based on those data sets. Replication in computing ensures consistency of estimations and can improve reliability, fault-tolerance, or accessibility. Moreover, results of estimated Q-matrices from real response data through the model should be carefully reviewed and discussed before making any conclusion. One fundamental problem of CDM is that we cannot classify students' abilities without single attribute items [Chiu et al., 2009]. However it is entirely possible that the estimated Q-matrix could include no single attribute item and we are not able to apply it for the further CDM analysis. In addition, discrepancies between the estimated Q-matrix and the designed one cannot be well explained sometimes and could be estimation mistakes. Last but not least, selection of final estimated Q from real response data is still ambiguity. If we run multiple simulation studies and come up with many different λ values, it will be meaningless to use all these values in a real data analysis, because the number of candidate λ values is not reduced significantly.

It is interesting to see that $\lambda = 9$ or 11 works best in simulation studies, but originally we believe small λ values such as 0.001 and 0.05 might perform better because they have much larger penalty effects to push estimates to either 0 or 1. This might be the reason that these values are in fact better fit the response data; or actually the cutoff points we select to transfer continuous elements to binary values affect the results. When λ is large such as 9 or 11, more estimates are around 0.5. However after the transformation by cutoff point 0.5, all these uncertainties around 0.5 are gone. But if we select different cutoff points such as 0.3/0.7, the highest accuracy of identified elements may not be 9 or 11 because they have more uncategorized values between 0.3 and 0.7. This is exactly the case in Table 4.6 of the second simulation study.

As a result, for future improvements, based on the results of the model, one can possibly improve estimation accuracy by selecting cutoff points such as 0.3/0.7 first to transfer part of the Q-matrix elements to binary values, and then apply a Bayesian extension of the DINA model developed by DeCarlo [2012] to find out estimates of the rest Q-matrix elements. Another possible approach could apply different single cutoff points together with ROC curve to look at false positives and false negatives, then decide which cutoff point to use, rather than taking 0.5 only to transfer all estimates to binary values. Analysis of estimated latent class sizes might not be necessary in the future unless we do replications of simulated studies to compare with the theoretical designed latent class sizes. Furthermore, development of the cross-validation approach based on the predication deviance distances could still be a good standard to select the best Q, as long as people can figure out a practical way to predict the missing responses. There are some other penalty functions that might be also able to penalize estimates to 0 or 1. For example, $f(x) = \lambda|x - \frac{1}{2}|$ or $f(x) = \lambda \times x^2$ when $x \leq \frac{1}{2}$ and $f(x) = \lambda \times (1 - x)^2$ when $x \geq \frac{1}{2}$. Better penalty functions could accelerate computation speed and lead to more robust results. The purpose of the penalty in this research is not for dimension reductions because the number of Q-matrix elements is fixed. Moreover, the future studies could also consider on how to determine the total numbers of attributes like an exploratory factor analysis. For example, possibly total four or three attributes in the Q-matrix might fit even better to the response data in the fraction subtraction test. But the identification of attributes in this situation has to be worked with an exam domain expert to understand the meaning of each Q-matrix column. Meanwhile, guessing or slipping factors of students could also be additional parameters in the model and that might generate better results.

In summary, the process of deriving a Q-matrix described in this research begins by building a probabilistic model between item responses and latent attributes as well as continuous Q-matrix elements as parameters. The estimation process adds a penalty in the likelihood function in order to get biased but better estimates more close to 1 or 0. The model is a good example to deal with binary data and nonlinear relationships. It can also be easily extend to other complicated models under different assumptions. However, people have to be aware of the large number of parameters to be estimated in the model, which leads to slow computing and time consuming. This model is solely based on the item responses to estimate all elements of Q-matrix without considering the one designed by experts. However, in the real world it is not appropriate to ignore experts opinions completely because sometimes the model could make mistakes and sometimes we are not able to interpret all discrepancies. In the future the researcher would suggest psychometricians corporate with domain experts to develop an initial Q-matrix together, and improve it according to model estimations combined with experts' opinions.

Bibliography

Cen, H., Koedinger, K., and Junker, B. (2005). Learning factors analysis: A general method for cognitive model evaluation and improvement. *Intelligent Tutoring Systems 8th International Conference*, pages 164–175.

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

de la Torre, J. (2008). An empirically based method of Q-matrix validation for the DINA model: Development and applications. *Journal of Educational Measurement*, 45(4):343–362.

de la Torre, J. (2009a). Common cdms. NCME 2009 Workshop.

de la Torre, J. (2009b). DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130.

de la Torre, J. (2011). The generalized DINA model framework. *Psychometrika*, 76(2):179–199.

de la Torre, J. and Douglas, J. A. (2004). Higher-order latent trait models for cognitive diagnosis. *Psychometrika*, 69(3):333–353.

de la Torre, J. and Douglas, J. A. (2008). Model evaluation and multiple strategies in cognitive diagnosis: An analysis of fraction subtraction data. *Psychometrika*, 73(4):595–624.

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(1):8–26.

DeCarlo, L. T. (2012). Recognizing uncertainty in the Q-matrix via a bayesian extension of the DINA model. *Applied Psychological Measurement*, 36(6):447–468.

Dennis, J. E. and Schnabel, R. B. (1987). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Society for Industrial Mathematics, 3600 University City Science Center, Philadelphia, PA.

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

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

Durongwatana, S. (2011). The optimal cut-off point for predictive classification of ungrounped data with binary logistic regression model. Presentation, Chulalongkorn University.

Fall, E. (2009). Applications of exploratory Q-matrix discovery procedures in diagnostic classification models. Master's thesis, The University of Kansas.

Hartz, S. M. (2002). *A Bayesian framework for the unified model for assessing cog-*

nitive abilities: Blending theory with practicality. PhD thesis, University of Illinois Urbana-Champaign.

Henson, R. (2009). Q-matrix development. NCME 2009 Workshop.

Henson, R. and Douglas, J. (2005). Test construction for cognitive diagnostics. *Applied Psychological Measurement*, 29(4):262–277.

Henson, R. A., Templin, J. L., and 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). Available online.

Im, S. and Corter, J. E. (2011). Statistical consequences of attribute misspecification in the rule space method. *Educational and Psychological Measurement*, 71(4):712–731.

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

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

Lee, Y.-W. and Sawaki, Y. (2009). Cognitive diagnosis and Q-matrices in language assessment. *Language Assessment Quarterly*, 6(3):169–171.

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

Liu, J., Xu, G., and Ying, Z. (2011a). Learning item-attribute relationship in Q-matrix based diagnostic classification models. Available at http://arxiv.org/pdf/1106.0721.pdf.

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

Liu, J., Xu, G., and Ying, Z. (2012). Data-driven learning of Q-matrix. *Applied Psychological Measurement*, 36(7):548–564.

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, 64(2):187–212.

McGlohen, M. and Chang, H.-H. (2008). Combining computer adaptive testing technology with cognitively diagnostic assessment. *Behavior Research Methods*, 40(3):808–821.

Romero, C., Ventura, S., Pechenizkiy, M., and Baker, R., editors (2011). *Handbook of Educational Data Mining*. Chapman and Hall/CRC Press.

Roussos, L. A., DiBello, L. V., Stout, W., Hartz, S. M., Henson, R. A., and Templin, J. H. (2007a). The fusion model skills diagnosis system. *Cambridge University Press*, pages 275–318.

Roussos, L. A., Templin, J. L., and Henson, R. A. (2007b). Skills diagnosis using IRT-based latent class models. *Journal of Educational Measurement*, 44(4):293–311.

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

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

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

Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnosis. *Diagnostic Monitoring of Skill and Knowledge Acquisition*, pages 453–488.

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

Templin, J., He, X., Roussos, L., and Stout, W. (2003). The pseudo-item method: a simple technique for analysis of polytomous data with the fusion model. Technical report, External Diagnostic Research Group.

Templin, J. L. (2006). *CDM: cognitive diagnosis modeling with mplus.*

Templin, J. L. and Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods*, 11(3):287–305.

Appendix A

R programs

Here are the programs used in this dissertation by R languages.

A.1 Response data simulation from fraction subtraction test Q-matrix

```
alpha <- matrix(rbinom(10000, 1, 0.5), nrow=2000)
q <- matrix(c(1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
0,1,0,1,0,1,1,1,0,0,0,0,1,1,1,
0,1,0,1,1,1,1,0,1,1,1,1,1,1,1,
0,1,0,1,0,1,1,0,0,1,0,1,1,1,1,
0,0,0,1,0,0,0,0,0,1,0,0,0,1,0), ncol = 5, nrow = 15)
tmatrix <- alpha%*%t(q)
attsum <- matrix(1, nrow=1, ncol=15)
for (i in 1:15){
attsum[i] <- sum(q[i,])
}
```

```
reattsum <- matrix(rep(attsum, 2000), nrow=2000, byrow=T)
eta <- matrix(1, nrow=2000,ncol=15)
for(i in 1:2000){
for(j in 1:15){
if(tmatrix[i,j]==reattsum[i,j]){
eta[i,j] <- 1
}else{
eta[i,j] <- 0
}
}
}
p <- matrix(1, nrow=2000,ncol=15)
for(i in 1:2000){
for(j in 1:15){
p[i,j] <- (0.9^eta[i,j])*(0.1^(1-eta[i,j]))
}
}
## generating data ##
sim <- matrix(1,nrow=2000,ncol=15)
for(i in 1:2000){
for(j in 1:15){
sim[i,j] <- rbinom(1,1,p[i,j])
}
}
```

A.2 Q-matrix estimation algorithm

```
as.binary <-
    function(x){
        ans <- NULL
        while(any(x!=0)){
            ans <- cbind(x%%2,ans)
            x <- floor(x/2)
        }
        ans
    }


optim.fun <- function(X,Q,pi,a=0.5){
  N = nrow(X)
  J = ncol(X)
  K = ncol(Q)

  all.a = as.binary(0:(2^K-1))   #2^K x K #
  pi = c(1,exp(pi))
  pi = pi/sum(pi)
  Q = exp(Q)
  Q = Q/(1+Q)
                                        #print(pi)
  pR.a = exp(tcrossprod(1-all.a,log(1-Q)))  #2^K x J #
  tmp = tcrossprod(X,log(pR.a)) + tcrossprod(1-X,log(1-pR.a))  # N x 2^K #
  tmp[,2^K] = ifelse(apply(X==1,1,all), 0, -Inf)
  pX = exp(tmp)%*%pi

  result = sum(log(pX)) + (a-1)*sum(log(Q)+log(1-Q))

  -2*result

}




findQ = function(y,ndim=3,a=0.5,trace=0,maxit=500){

  y = as.matrix(y)

  nitem = ncol(y)

  nsubj = nrow(y)


  init = c(rnorm(nitem*ndim),rep(0,2^ndim-1))

  tmp.out = optim(init, function(p){

    optim.fun(y, matrix(p[1:(nitem*ndim)],nitem,ndim), p[-(1:(nitem*ndim))],a=a)},

    control=list(maxit=maxit,trace=trace))

  final = tmp.out$par

  Q = matrix(final[1:(ndim*nitem)], nitem, ndim)

  Q = exp(Q)

  Q = Q/(1+Q)

  pi = c(1,exp(final[-(1:(ndim*nitem))]))

  pi = pi/sum(pi)

  result = list(Q=Q,pi=pi,deviance=tmp.out$value)

  result

}

}
```

A.3 Calculations of minimum discrepancy distance and counts of identical elements for fraction subtraction Q-matrix

```
fn_perm <- function (n, r, v = 1:n)
{
   if (r == 1)
      matrix(v, n, 1)
   else if (n == 1)
      matrix(v, 1, r)
   else {
      X <- NULL
      for (i in 1:n) X <- rbind(X, cbind(v[i], fn_perm(n - 1, r - 1, v[-i])))
      X
   }
}

Q2=q
discrepQ2=function(Q.EST){
-sum(Q2*log(Q.EST)+(1-Q2)*log(1-Q.EST))
}

mindiscrepQ2=function(Q.EST){
all_perm=fn_perm(ncol(Q.EST),ncol(Q.EST))
all_result <- c()
```

```
for(i in 1:nrow(all_perm)){
newQ.EST <- Q.EST[,all_perm[i,]]
one_result <- discrepQ2(newQ.EST)
all_result <- c(all_result, one_result)
}
mindis <- min(all_result)
index_min <- which(all_result == min(all_result), arr.ind = TRUE)
min_perm <- all_perm[index_min,]
FQ.EST=Q.EST[,min_perm]
result=list(mindis=mindis,min_perm=min_perm,FQ.EST=FQ.EST)
result
}


disQ2.EST1=mindiscrepQ2(Q2.EST1)
mindisQ2.EST1=disQ2.EST1$mindis
FQ2.EST1=disQ2.EST1$FQ.EST


count=function(a,b,q,Q){
qnew=matrix(,nrow=nrow(q),ncol=ncol(q))
for (i in 1:nrow(q)){
for (j in 1:ncol(q)){
if(q[i,j]<=a){qnew[i,j]=0}
else {if (q[i,j]>=b){qnew[i,j]=1} else {qnew[i,j]=q[i,j]}}
}
}
dif=Q-qnew
```

```
difnew=matrix(,nrow=nrow(dif),ncol=ncol(dif))

for (i in 1:nrow(dif)){

for (j in 1:ncol(dif)){

if(dif[i,j]==0){difnew[i,j]=1}

else {difnew[i,j]=0}

}

}

rowsum=apply(difnew,1,sum)

counts=sum(rowsum)

total=ncol(Q)*nrow(Q)

prob=counts/total

result=list(cutoff=c(a,b),FinalQ=qnew,Correct_Counts=counts,

Total=total,Correct_Prob=prob)

result

}
```