2013

Estimation and Q-matrix validation for diagnostic classification models

YULING FENG
University of South Carolina - Columbia

ESTIMATION AND Q-MATRIX VALIDATION FOR DIAGNOSTIC CLASSIFICATION MODELS

by

Yuling Feng

Bachelor of Science
Sun Yat-sen University, 2006

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Statistics

College of Arts and Sciences

University of South Carolina

2012

Accepted by:

Brian Habing, Major Professor

John Grego, Committee Member

Xiaoyan Lin, Committee Member

Tammiee Dickenson, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

Dedicated

to

my parents,

Runan Feng and Yiqing Li,

my husband,

Peixi Zhu

ACKNOWLEDGEMENTS

First of all, I would like to give my sincere and earnest thankfulness to my advisor Dr. Brian Habing. He gave me a lot of guidance, encouragement and inspiration which led me to the completion of this dissertation. I am grateful for his time spent on discussing my research, and always being patient in clarifying my confusions. I am also thankful for his support and friendship during my PhD study.

My gratitude also goes to Dr. John Grego, Dr. Xianyan Lin, and Dr. Tammiee Dickenson who served as my committee members. I appreciate their time and constructive suggestions on the revisions of my dissertation. My gratitude also goes to Dr. Edsel Pena, Dr. David Hitchcock, Dr. Ou Zhao, Dr. Joshua Tebbs, and Dr. Tim Hanson at the department of Statistics. I gained a lot of benefits from taking their classes.

I would also like to thank Ms. Georgie Baker, Ms. Suzanne Rizzo, Ms. Tina Ormenisan and Ms. Anita Wood for helping me whenever I needed assistance. I owe my gratitude to the department of Statistics, University of South Carolina for providing me the graduate assistantship. I thank Ms. Maureen Petkewich for her guidance on teaching. I am also thankful to Dr. Lee van Horn at the department of Psychology at the University of South Carolina. It has been an enjoyable experience working with him.

I would sincerely thank my friends and all my colleagues in the department of Statistics. Last but not the least, none of these would be possible without the love and support from my family. I would like to thank my husband, Peixi Zhu for always

being supportive. His love and patience got me through the difficulty times during this journey. I would also like to thank my parents, Runan Feng and Yiqing Li, for their countless love and encouragement.

ABSTRACT

Diagnostic classification models (DCMs) are structured latent class models widely discussed in the field of psychometrics. They model subjects' underlying attribute patterns and classify subjects into unobservable groups based on their mastery of attributes required to answer the items correctly. The effective implementation of DCMs depends on correct specification of a Q-matrix which is a binary matrix linking attribute patterns to items. Current literature on assessing the appropriateness of Q-matrix specifications has focused on validation methods for the deterministic-input, noisy-and-gate (DINA) model. The goal of the study is to develop general Q-matrix validation methods that can be applied to a wider class of DCMs. The study proposes a two-stage validation method which incorporates the idea of sequential searching based on the posterior distribution of attribute patterns and Bayesian model selection techniques. Simulation studies show that the proposed methods successfully detect and correct misspecifications in a Q-matrix for a complicated non-compensatory DCM, the reduced reparameterized unified model (RUM), and a compensatory DCM, the deterministic input, noisy-or-gate (DINO) model.

Model estimation is the first step in validating a Q-matrix. The EM algorithm is shown to provide accurate estimates for the reduced RUM, with the advantage of significant computational time savings compared to estimation by Markov chain Monte Carlo (MCMC). In addition, factors affecting the performance of the validation methods

are discussed. Suggestions on implementation of the methods under the case when items are from a combination of DCMs are given.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Cognitive diagnosis models or diagnostic classification models (DCMs, e.g., diBello, Roussos, & Stout, 2007; Rupp, Templin, & Henson, 2010) are multidimensional latent class models that provide detailed feedback on students' learning and progress. In contrast to the traditional multidimensional item response theory (IRT) under which latent variables are on continuous scales, DCMs model multiple discrete latent variables, or attributes, and lead to classification of respondents' attribute patterns. These attribute patterns specify membership in various latent classes. Each respondent's attribute pattern is a binary vector with 1 indicating mastery on an attribute and 0 otherwise. Such mastery profiles often aim to help teachers design targeted remedial instruction.

A wide array of DCMs has been proposed over the past decade (see Rupp, Templin and Henson, 2010, for a recent taxonomy). DCMs can be classified into two categories, non-compensatory and compensatory, depending on the nature of the models. The commonly used non-compensatory DCMs are the deterministic-input, noisy-and (DINA) model, the NIDA model (e.g. de la Torre, 2009; Junker and Sijtsma, 2001), and the reparameterized unified model (RUM, Hartz, 2002, Roussos et al., 2007). Some well-known compensatory analogues are the deterministic input, noisy-or-gate (DINO)

model (see Templin & Henson, 2006), the NIDO model (see Junker and Sijtsma, 2001), and the compensatory RUM (see Templin, 2006). Many of the individual DCMs can be organized and estimated in more general model families, such as the generalized DINA models (G-DINA, de la Torre, 2008b), the generalized diagnostic models (GDM, von Davier, 2005), and the log-linear cognitive diagnostic models (LCDM, Henson, Templin, & Willse, 2009).

One critical step when implementing DCMs is the specification of which attributes are required to successfully answer each item on the diagnostic assessment. This matrix of specification is often called a Q-matrix (e.g., Tatsuoka, 1983). Consider an assessment consisting of I items measuring on a domain of K attributes or skills. It is then an I by K matrix with elements q_{ik} , $i=1,2,\ldots,I$, $k=1,2,\ldots,K$, taking on 0/1 value with 1 indicating that attribute k is required by item i and 0 otherwise. The construction of the Q-matrix is usually conceptual. After the attributes are well-defined, multiple subject matter experts who may be item developers from testing companies or school teachers are asked to carefully inspect items and determine the required attributes for each items based on their professional experiences. Their opinions are then collected and aggregated to form a Q-matrix. For non-compensatory models, a Q-matrix is properly defined if the attributes specified as 1s in the Q-matrix are all needed for giving the maximum probability of correctly answering each item and only those attributes are required. In a compensatory model, at least one of the attributes must be a 1 to give the maximum probability of correct response (de la Torre, 2008). Most implementations of DCMs assume that the Q-matrix is properly defined.

However, a Q-matrix might be subjective in reflecting the true relationship between items and attributes, since it is constructed based on human beings' opinions. When a Q-matrix is not properly defined, we said that it is misspecified. There are three types of misspecifications: underspecified, overspecified, or combination of both. In an underspecified q-vector (i.e., Q-matrix row vector), entries of '1' are recoded as '0' so that fewer model parameters are estimated for the item under consideration. In an overspecified q-vector entries of '0' are recoded as '1' so that parameters that represent pure noise are unduly estimated. The misspecification of a Q-matrix would lead to undesirable consequences, e.g., poor model fit, inaccurate model parameter estimation (e.g. Henson and Templin, 2009; Rupp and Templin 2008), and incorrect interpretations of the set of user-specified attributes. Therefore, the development of validation methods to access the specification accuracy of an existing Q-matrix by learning it from empirical data is important for the successful implementation of DCMs.

An intuitive method would be to compare model fit indices among models with possible Q-matrices. However, this method involves intense computation. For an assessment with I items and K attributes, there are 2^{K*I} possible Q-matrices and the model fit indices for 2^{K*I} models need to be compared. As the K and I get large, the number of possible Q-matrices increases exponentially, and so is the computation involved.

Exiting literature focuses on the implications of Q-matrix misspecifications in the area of DCMs. Rupp and Templin (2009) investigated the effect of Q-matrix misspecification on item parameter estimation for the DINA model. DeCarlo (2011) examined the impact of Q-matrix misspecifications on latent class sizes under the DINA

model. Kunina-Habenicht, Rupp, and Wilhel (2011) examineed the effects of model misspecification due to Q-matrix misspecifications on item parameter estimation and respondent classification within a broader DCM framework. There are only a few studies have been done for validating the Q-matrix. de la Torre (2008a) proposed a empirically based sequential search method to validate a Q-matrix. The search algorithm is based on the comparison of correct response probabilities between two specific groups of people. With reasonable computation time, the method is able to correct a misspecified Q-matrix under two conditions: 1) the response data is modeled by a DINA model; 2) the number of misspecified q-vectors is small compared to the number of items in the assessment. Liu, Xu and Ying (2011) stated that under the DINA or DINO model, if a Q-matrix is correctly specified, the Euclidean distance between expected proportions of positive responses to all items and a model-based combination of items and the corresponding observed proportions converges to zero in probability. They then suggested that a procedure can be form to validate of an existing Q-matrix by checking the closeness of the Euclidean distance between the above two vectors to zero. Close (2012) investigated the application of principle component analysis in the construction of Q-matrix with data that satisfies the DINA model. This method is effective in building a Q-matrix only when there are multiple items for each of the skill sets.

Note that the existing literature is all for the DINA model which assumes only two possible correct response probabilities for each item. In many cases, more flexible models, such as NIDA and reduced RUM, are needed to fit response data. In other cases when not all required attributes for an item have to be mastered for a successful response, compensatory models, such as the DINO model, are needed. Thus, the development of

Q-matrix validation methods for a broader class of DCMs is important. This study focuses on developing validation procedures for two specific DCMs, the reduced RUM and the DINO model. The NIDA model is a special case of the reduced RUM, with the item parameters being constrained to be the same across attributes, so validation procedure for the reduced RUM can be easily extended to the NIDA model which will not be considered in the study.

The rest of Chapter 1 will be structured as following. Section 1.1 describes in detail two noncompensatory DCMs, the DINA model and the reduced RUM. Section 1.2 describes in detail a compensatory DCM, the DINO model. Section 1.3 discusses in detail the sequential search validation method for the DINA model (de la Torre, 2008).

1.1 The Noncompensatory DCMs

In non-compensatory DCMs, the absence of one attribute cannot be compensated by the presence of another attribute. Consider the item "1+2*3" which requires two elementary math skills: adding and multiplication. The noncompensatory assumption is reasonable for this item, because respondent can only answer the item correctly with having mastered both of the skills if there is no guessing effect present. In this section, two popular noncompensatory DCMs, the deterministic-input, noisy-and-gate (DINA) model and the reduced reparameterized unified model (reduced RUM) will be introduced.

1.1.1 The DINA Model

The deterministic-input, noisy-and-gate (DINA) model (e.g., Haertel, 1989; Junker & Sijtsma, 2001; de la torre & Douglas, 2004) is a noncompensatory DCM. Under

this model, respondents need to have mastered all the required attributes to get a correct answer for an item. Thus, the model divides the respondents into two mastery groups for each item: respondents having mastered all the required attributes, and those lacking at least one of them. There is no further differentiation between respondents who lack different attributes. The DINA model takes into account the possibility that a respondent with all the required skills misses an item and the possibility through careless errors (i.e., a slip), and for the possibility that a respondent who lack at least one of the required skills gives a correct response by guessing.

Consider an assessment consisting of *I* items measuring a domain of *K* attributes or skills. Let Y_{ij} , *i=1,2,…, I, j=1,2,…,J,* be a binary 0/1 response for item *i* by respondent *j* with 1 representing the respondent providing a correct response to the item and 0 otherwise. The attribute pattern for respondent *j*, α_j is a vector of length *K* with binary 0/1 elements with 1 meaning the respondent has mastered the attribute and 0 otherwise. For a test requiring *K* attributes, respondents can be classified into one of the 2^K possible attribute patterns. The slipping and guessing parameters for an item are defined based on the two scenarios:

$$s_i^{DINA} = P(Y_{ij} = 0 \mid \xi_{ij} = 1) \tag{1.1}$$

$$g_i^{DINA} = P(Y_{ij} = 1 \mid \xi_{ij} = 0) \tag{1.2}$$

where,

$$\xi_{ij} = \prod_{k=1}^{K} a_{jk}^{q_{ik}} \tag{1.3}$$

It is easily seen that ξ_{ij} takes on two values, 1 or 0 and is the indicator of whether respondent *j* has mastered all the required skills by item *i*. When ξ_{ij} =1, the respondent *j*

has mastered all the required skills and 0 otherwise. Therefore, the probability of a correct response by respondent *j* to item *i* under the DINA model is given by:

$$P(Y_{ij}=1|\alpha_j, s_i^{DINA}, g_i^{DINA}) = (1-s_i^{DINA})^{\xi_{ij}} g_i^{DINA^{1-\xi_{ij}}} \quad (1.4)$$

Based on formula (1.4), a respondent gets item *i* correct with two possible probabilities. If the respondent has mastered all the required attributes, his/her probability of providing a correct answer to item *i* is $1-s_i^{DINA}$, and it is g_i^{DINA} if the respondent lacks at least one of the required attributes. The guessing and slipping are defined at item level, thus the DINA model cannot differentiate between the respondents who lack different attributes. There are several ways to estimate the DINA models. It can be estimated in a form of a constrained log-linear model (Henson, Templin, & Willse, 2009). It can also be estimated using an EM algorithm and Markov chain Monte Carlo (MCMC) methods (de la Torre, 2009).

1.1.2 The reduced RUM model

Unlike the DINA model which has only two parameters for each item, the number of parameters varies across items under the reduced RUM, where there is one parameter per attribute for each item. Thus, this model allows for a more flexible impact of attribute mastery on item response probabilities (Rupp, Templin & Henson, 2010). Under the reduced RUM, the probability of a correct answer to item *i* given that a respondent has attribute pattern α_j is

$$P_i(\alpha_j) = P(Y_{ij}=1|\alpha_j) = \pi_i \prod_{k=1}^{K} r_{ik}^{(1-\alpha_{jk})q_{ik}} \quad (1.5)$$

The baseline parameters π_i is the probability of a correct response to item *i* given a respondent has mastered all the required attributes for the item. An item with a large π_i parameter indicates the specified attributes can explain examinee responses to the item well. The penalty parameter r_{ik} is the reduction to the probability of a correct response to item *i* resulting from not having mastered attribute *k*. A small value of penalty parameter for an attribute implies that the probability of a correct response is greatly reduced when the attribute is not mastered. Table 1.1 shows an example of a hypothetical item with known item parameters in a four-attribute domain. The item loads on the 1st and 4th attributes as indicated by its q-vector is (1,0,0,1). A respondent has an 80% chance of responding correctly if both of attributes are mastered, a 48% (0.8*0.6) chance of responding correctly if only the 4th attribute is mastered, and a 64% (0.8*0.8) chance of replying correctly if only the 1st attribute is mastered. If the examinee lacks both of the required attributes, the examinee can still get the item correct by guessing, and the probability of guessing successfully is 38.4% (0.8*0.8*0.6). Whether or not the examinee has mastered the 2nd and 3rd attributes does not affect the probability of a correct response, since these attributes are not required by the Q matrix. In practice, the baseline and penalty parameters are unknown, and need to be estimated. Thus, the number of parameters to estimate for item *i* is equal to the number of required attributes for the item plus 1, e.g., three item parameters are estimated in the example in table 1.1.

Table 1.1 Hypothetical Item under the reduced RUM in a Four-Attribute Domain

Hypothetical Item	Baselines	Penalty Probabilities Attributes			
		1	2	3	4
(1,0,0,1)	0.8	0.60	1.00	1.00	0.80

One option for estimating DCMs is to regard them as constrained log-linear models within M-plus (Henson, Templin & Willse, 2009), but the reduced RUM cannot be estimated with the current version of M-plus due to a large number of constraints placed on the item parameters (Templin, personal communication, May 2010). Like the DINA, the reduced RUM may be estimated using MCMC methods (Henson, & Templin, 2007), but the computation time may be prohibitively lengthy when the numbers of respondents and items are large. In Chapter 2, the reduced RUM will be estimated as a structured latent class models using the EM algorithm.

The reduced RUM is a reduced form of the RUM which is also known as the fusion model (e.g., DiBello et al., 1995; Hartz, 2002). In addition to the response function of the recuded RUM, the fusion model also contains a logistic function of continuous latent variable measuring respondents' ability, which is used to account for information that cannot be explained by attributes. Due to this complexity, the fusion model is not discussed as often as the reduced RUM in recent literatures.

1.2 The DINO Model

In compensatory DCMs, the absence of a particular measured attribute can be compensated by the presence of another measured attribute. Such models are usually used for modeling the responses to a psychological scale instead of an achievement test. For example, in a real study to access the prevalence of pathological gambling (Templin and Henson 2006), an item "I have gotten into trouble over things I have done to finance my gambling." which may require the presence of two attributes as follows:

- Attribute 1: The respondent has broken the law to finance his or her gambling.

- Attribute 2: The respondent has lost relationships because of his or her gambling.

A respondent is likely to provide a positive response to this item if he/she has either broken the law to finance his or her gambling, or lost relationship because of his or her gambling, or done both of them. In this case, we only require that one attribute is present for a respondent to have a high probability of a positive response to the item. In this section, the most widely discussed compensatory DCM, the deterministic input, noisy-or-gate (DINO) model will be introduced. This model is analogous to the DINA model.

The deterministic input, noisy-or-gate (DINO) model (e.g., Templin & Henson, 2006, Templin, 2006) is a simple compensatory DCM. Similar to the DINA model, it has two parameters at the item level, the slip and guess parameters. Unlike the DINA model, respondents have high probability of providing a correct answer with at least one of the required skills instead of all of the required skills. Under the DINO model, the slipping and guessing parameters for an item are defined based on the two scenarios:

$$s_i^{DINO} = P(Y_{ij} = 0 \mid \varpi_{ij} = 1) \tag{1.6}$$

$$g_i^{DINO} = P(Y_{ij} = 1 \mid \varpi_{ij} = 0) \tag{1.7}$$

where,

$$\varpi_{ij} = 1 - \prod_{k=1}^{K} (1 - \alpha_{jk})^{q_{jk}} \tag{1.8}$$

Here, ϖ_{ij} takes on value 0 and 1. ϖ_{ij} =1 indicates that respondent *j* has mastered at least one of the required skills, and ϖ_{ij} =0 indicates that respondent *j* has mastered none of the required skills. The probability of a correct answer is given by

$$P(Y_{ij} = 1 \mid \alpha_j, s_i^{DINO}, g_i^{DINO}) = (1 - s_i^{DINO})^{\varpi_{ij}} g_i^{DINO\ 1-\varpi_{ij}} \tag{1.9}$$

Based on the formula above, a respondent gets item *i* correct with two possible probabilities. If a respondent has mastered none of the required attributes, his/her is still likely to provide a correct answer via guessing, so the correct response probability in this case is g_i^{DINO}. When a respondent has mastered at least one of the required attributes, the correct response probability is $1-s_i^{DINO}$. The DINO model can be estimated using MCMC (Templin & Henson, 2006), or as a constrained log-linear model with latent classes.

1.3 Sequential search Q-matrix validation method for the DINA model

de la Torre (2008) developed a sequential method to validate a Q-matrix based only on information from responses from a DINA model. As mentioned above, respondents could be classified into 2^K possible attribute patterns using an assessment measuring on *K* attributes. For item *i*, its correct q-vector, $q_{i,}$ could be one of the 2^K-1 possible attribute patterns. The q-vector could not be a vector with all 0 elements. Let α_l be a binary vector of length *K, l=1,2,…., 2^K-1,* and let δ_{il} be the difference in probabilities of correct responses for item *i* between respondents who have mastered the required attributes when item *i*'s q-vectors is specified as α_l and respondents who are lack of at least one of the required attributes, that is

$$\delta_{il} = P(Y_{ij} = 1 \mid \xi_{lj} = 1) - P(Y_{ij} = 1 \mid \xi_{lj} = 0) \tag{1.10}$$

The correct q-vector for item *i* is then defined (de la Torre, 2008) as the binary vector that maximizes δ_{il},

$$q_i = \arg\max_{\alpha_l}(\delta_{il}) \tag{1.11}$$

Table 1.3 shows how to identify the correct q-vector using this definition for a hypothetical item in a four-attribute domain as shown in Table 1.2. This hypothetical item requires the first and the second attribute, and its guessing and slipping parameter under the DINA model is 0.2 and 0.2, respectively.

Table 1.2 A Hypothetical Item under the DINA Model in a Four-Attribute Domain

Attribute					
1	2	3	4	Guessing	Slipping
1	1	0	0	0.2	0.2

Table 1.3 lists the δ for all the 16 attribute patterns in a four-attribute domain. The second and third column list the membership to the two groups ξ=1 and ξ=0, and probability of providing a correct answer to the hypothesized item for examinees with the corresponding attribute patterns. Here, a flat distribution is assumed for the attribute pattern distribution, i.e. every respondent is equally likely to be classified into one of the 16 possible attribute patterns. The last three columns provide the probabilities of correct answers in two group ξ=1 and ξ=0, and their difference δ. These probabilities are found as the mean probabilities of correct response for the two groups given the item q-vector is the corresponding q-vector. Take pattern 6 for instance, examinees with 12 patterns (pattern 1- 5, 7-11, 13, 14) fall in group ξ=0 when the item q-vector is the attribute pattern (1, 1, 0, 0). The mean probability of correct response in ξ=0 is thus found by adding the true correct response probabilities for these 12 patterns and dividing the sum by 12 (2.4/12 = 0.2). Examinees with pattern 6, 12, 15, 16 fall in group ξ=1 and the mean probability of correct response is given by (0.8+0.8+0.8+0.8)/4=0.8. Based on the table, the largest δ is obtained at the true item q-vector.

Table 1.3 Probabilities of Correct Response for a Hypothetical Item under the DINA Model in a Four-Attribute Domain

	Under True Q vector		Attributes				Probability of Correct Response		
Pattern	ξ	P(X=1|ξ)	1	2	3	4	ξ=1	ξ=0	δ
1	0	0.2	0	0	0	0	---	---	---
2	0	0.2	1	0	0	0	0.5	0.2	0.3
3	0	0.2	0	1	0	0	0.5	0.2	0.3
4	0	0.2	0	0	1	0	0.35	0.35	0
5	0	0.2	0	0	0	1	0.35	0.35	0
6	**1**	**0.8**	**1**	**1**	**0**	**0**	**0.8**	**0.2**	**0.6**
7	0	0.2	1	0	1	0	0.5	0.3	0.2
8	0	0.2	1	0	0	1	0.5	0.3	0.2
9	0	0.2	0	1	0	1	0.5	0.3	0.2
10	0	0.2	0	1	1	0	0.5	0.3	0.2
11	0	0.2	0	0	1	1	0.35	0.35	0
12	1	0.8	1	1	1	0	0.8	0.29	0.51
13	0	0.2	0	1	1	1	0.5	0.33	0.17
14	0	0.2	1	0	1	1	0.5	0.33	0.17
15	1	0.8	1	1	0	1	0.8	0.29	0.51
16	1	0.8	1	1	1	1	0.8	0.32	0.48

Searching for the correct q-vector by definition is straightforward but computationally intensive. As K increases, the number of attribute patterns increases exponentially. A more efficient algorithm, the sequential search algorithm (de la Torre, 2008), was proposed for searching for correct q-vectors. This algorithm starts by comparing the δ for all single-attribute patterns. The attribute resulting in the largest δ is selected as one of the attributes in the q-vector. Then, all two-attribute patterns with the first selected attribute are compared by their δs. The second attribute is chosen based on two criteria: (1) its corresponding two-attribute pattern has the largest δ, say $\delta^{(2)}$; and (2) $\delta^{(2)} > \delta^{(1)}$. If $\delta^{(2)} < \delta^{(1)}$, it is unnecessary to include a second attribute. The process stops here and the correct q-vector is a single-attribute pattern. The process continues in the

same way to choose the rest of the attributes. Table 1.4 demonstrates how this sequential method works for the hypothesis item in Table 1.2.

Table 1.4 Selecting Attribute based on δ

	Attributes			
Number of Attributes	1	2	3	4
One	**0.3**	0.3	0	0
Two (including α_1)	---	**0.6**	0.2	0.2
Three(including α_1,α_2)	---	---	0.51	0.51

As shown in table 1.4, δs are compared for all single-attribute patterns at the first step. Both the pattern with the first attribute only and that with the second attribute only have the highest difference of 0.3, and either one can be included in the correct q-vector. Suppose we choose the first attribute for this step. In the second step, δs are compared for those two-attribute patterns which include the first attribute. At this step, the second attribute is picked because it has the highest difference 0.6, and it is larger than the highest difference in the first step (0.3). A third attribute is not needed because the largest difference for q-vectors with three attributes is 0.51, which is less than 0.6. Thus, the correct q-vector is (1,1,0,0). The correct q-vector found using the sequential method agrees with that from the definition.

In the previous cases, the δs are computed using known item parameter values and known distribution of attribute patterns. However, item parameters values and distribution of attribute patterns are unknown and need to be estimated. Let $P(\alpha_l | Y_j)$ be the posterior probability that examinee *j* has the attribute pattern α_l. For item *i* with q-

vector $\alpha_{l'}$, the MLE of the guessing and slipping parameters under the DINA model are given by (de la Torre, 2009),

$$\hat{g}_{il'}^{DINA} = \frac{\sum_{j=1}^{J}\sum_{\{l:\xi_{il}=0\}}^{L} P(\alpha_l \mid Y_j) Y_{ij}}{\sum_{j=1}^{J}\sum_{\{l:\xi_{il}=0\}}^{L} P(\alpha_l \mid Y_j)} \tag{1.12}$$

$$\hat{s}_{il'}^{DINA} = 1 - \frac{\sum_{j=1}^{J}\sum_{\{l:\xi_{il}=1\}}^{L} P(\alpha_l \mid Y_j) Y_{ij}}{\sum_{j=1}^{J}\sum_{\{l:\xi_{il}=1\}}^{L} P(\alpha_l \mid Y_j)} \tag{1.13}$$

The estimated guessing parameter is the proportion of people who are expected to be lack of at least one of the required attributes and provide correct responses to item *i* out of people who are expected to be lack of at least one of the required attributes. The estimated slipping parameter is the proportion of people who are expected to have mastered all required attributes but provide incorrect responses to item *i* out of people who are expected to have mastered all the required attributes. Under the DINA model, the $\delta_{il'}$ is then estimated as

$$\hat{\delta}_{il'}^{DINA} = (1 - \hat{s}_{il'}^{DINA}) - \hat{g}_{il'}^{DINA} = 1 - (\hat{s}_{il'}^{DINA} + \hat{g}_{il'}^{DINA}) \tag{1.14}$$

Note that a misspecified Q-matrix has an impact on the accuracy of item parameter estimation, which affects the validation accuracy of a Q matrix. de la Torre (2008) demonstrated that the sequential search method works well for the DINA model despite of the negative effect the Q-matrix misspecifications have on the item parameter estimation. The goal of the study is to extend the validation methods for other DCMs. The first model to be considered is the reduced RUM, which is also a noncompensatory DCM like the DINA model. However, the affect of misspecifications in a Q-matrix on

the model estimation is more severe for the reduced RUM than for the DINA model. In this case, the sequential search method might not work well to validate a Q-matrix, and validation methods have to be developed. The implementation of Q-matrix involves model estimation, which is challenging for the reduced RUM for the reasons that the model includes unobservable latent classes, and that the number of parameters that need to be estimated vary across items. So estimation of the reduced RUM must be discussed. Chapter 2 describes the parameter estimation of the reduced RUM using the EM algorithm and Chapter 3 describes the implementation of the sequential method on this model. A two-stage validation method will be developed for the reduced RUM.

The second model we consider is the DINO model, which is a compensatory DCM and is different from the DINA model in nature. Chapter 4 will extend Q-matrix validation methods for the DINO model. In Chapter 5, answers to two questions concerning the implementation of the validation methods developed in the previous chapters will be explored: 1) How does the validation method work when the items are from a combination of DCMs? 2) How are factors such as number of misspecifications, type of misspecifications in a Q-matrix affecting the performance of the validation methods?

CHAPTER 2

ESTIMATION OF THE REDUCED RUM USING THE EM ALGORITHM

Chapters 2 and 3 demonstrate our trial of the sequential method on the reduced RUM, beginning with estimation of the model in this chapter. There are two main reasons why the reduced RUM is chosen. First, it is one of the most commonly used DCMs (e.g., Henson & Templin, 2006). Compared to the DINA model, it has greater flexibility in modeling the probability of correct item response for different attribute patterns. Secondly, similar to the DINA model, it is a noncompensatory DCM. Intuitively, the definition of a correct q-vector for the DINA model should work fairly well for the reduced RUM, which will be tested in Chapter 3. The first step in implementing the sequential method for the reduced RUM is to estimate the item parameters and respondents' attribute patterns. However, due to its complexity, the reduced RUM is not estimated as readily as the DINA model. One option for estimating DCMs is to regard them as constrained log-linear models within M-plus (Henson, Templin & Willse, 2009), but the estimation of the reduced RUM with the current version of M-plus could be very lengthy due to the large number of constraints placed on the item parameters. Like the DINA model, the reduced RUM may be estimated using MCMC methods (Henson, & Templin, 2007), but the computation time may be prohibitively lengthy when the numbers of respondents and items are large.

The EM algorithm by de la Torre (2009) for estimating the DINA model took advantage of its simple form, i.e., for each item, the attribute pattern space can be

partitioned into two parts. One partition consists of attribute patterns with all required attributes, and the other consists of patterns lacking at least one of the required attributes. Thus, the complete likelihood function under the DINA model could be written as a sum of two components with the first component as a function of only the slip parameter and the second component as a function of only the guess parameter. In this way, closed forms for the slipping and guessing maximum likelihood estimators could be obtained, i.e. the slipping estimator is the proportion of the examinees who are expected to miss the item out of those who are expected to have mastered all the required attributes, and the guessing estimator is the proportion of the examinees who are expected to respond correctly out of those who are expected to lack at least one of the required attributes. This method cannot be extended to the reduced RUM because the number of partitions of the attribute pattern space varies across items, and no closed forms for item parameter estimator can be obtained. Thus, the EM algorithm proposed in this study seeks to remedy these shortcomings.

Section 2.1 reviews two examinee classification methods, i.e. the maximum a posterior (MAP) and expected a posterior (EAP). Then, a detailed description of the application of the EM algorithm to the reduced RUM is given. Section 2.2 assesses the estimation accuracy of the proposed algorithm by fitting the reduced RUM to simulated data with known true item parameters. Then, a real data set consisting of responses for the Examination for the Certificate for Proficiency in English is modeled by the reduced RUM, and the parameter estimates obtained using the EM algorithm are compared to results obtained via the MCMC by Henson & Templin (2007). The connections between the DINA model and the reduced RUM in terms of item parameters are reviewed, and the

parameter estimates obtained using the EM algorithm are compared to those from the DINA model.

2.1 Item Parameter Estimation using EM Algorithm

2.1.1 Examinee Classification Methods

Using the same notations as in the previous chapter, let Y_{ij} be the observed binary 1/0 response of the examinee *j* to item *I* with 1 representing the occurrence that examinee *j* provides a correct response to item *i* and 0 otherwise. Let $\alpha_{l,}\ l = 1,2,...,L=2^K$ be a possible attribute pattern which examinee *j* may possess. Under the reduced RUM, the likelihood of this examinee's responses on the assessment given that respondent *j* has attribute pattern α_l is

$$f(Y_j \mid \alpha_l) = \prod_{i=1}^{I} p_{il}^{Y_{ij}} (1 - p_{il})^{1-Y_{ij}} \tag{2.1}$$

where $Y_j=(Y_{1j}, \ldots Y_{ij}, \ldots Y_{Ij})$, and p_{il} is the probability that a respondent with attribute pattern α_l provides a correct response to item *i*, especially under the reduced RUM,

$$p_{il} = P(Y_{ij} = 1 \mid \alpha_l) = \pi_i^* \prod_{k=1}^{K} r_{ik}^{*(1-\alpha_{lk})q_{ik}} \tag{2.2}$$

Equation 2.1 is based on the conditional independence assumption that respondent *j* provides independent responses to the *I* items given his/her attribute pattern. Let λ_l be the probability that a randomly chosen examinee has attribute pattern α_l, and the sum of λ_l over all possible attribute pattern is equal to 1. The likelihood of examinee *j*'s response vector on the assessment, Y_j, is given by

$$f(Y_j) = \sum_{l=1}^{L} \lambda_l \prod_{i=1}^{I} p_{il}^{Y_{ij}} (1 - p_{il})^{1-Y_{ij}} \tag{2.3}$$

The marginal likelihood (Equation 2.3) is a weighted sum of the likelihoods of the respondent's answers on the assessment given all possible attribute patterns he/she might have mastered. Let $P(\alpha_l \mid Y_j)$ be the posterior probability that examinee *j* has the attribute pattern α_l. By Bayes' theorem,

$$P(\alpha_l \mid Y_j) = \frac{\lambda_l f(Y_j \mid \alpha_l)}{f(Y_j)} = \frac{\lambda_l \prod_{i=1}^{I} p_{il}^{Y_{ij}} (1 - p_{il})^{1-Y_{ij}}}{\sum_{m=1}^{L} \lambda_m \prod_{i=1}^{I} p_{im}^{Y_{ij}} (1 - p_{im})^{1-Y_{ij}}} \tag{2.4}$$

An examinee's attribute pattern is an unobservable latent variable and can be estimated using maximum a posterior (MAP) or expected a posterior (EAP) estimation. Using MAP, the estimated attribute pattern for examinee *j* is the attribute pattern that maximizes the posterior probability of attribute patterns given Y_j,

$$\hat{\alpha}_{j(MAP)} = \arg\max_{\alpha_l} P(\alpha_l \mid Y_j) \tag{2.5}$$

Using EAP, the expected attribute pattern for examinee *j* is calculated by

$$\hat{\alpha}_{j(EAP)} = \sum_{l=1}^{L} \alpha_l P(\alpha_l \mid Y_j) \tag{2.6}$$

and each element in $\hat{\alpha}_j$ is then rounded at 0.5 to obtain a binary skill pattern classification. It was shown that the application of the MAP method had higher numbers of examinees classified correctly on all *K* attributes, while the EAP method resulted in higher total attributes classified correctly and fewer severe misclassifications (Huebner & Wang, 2011). The choice of usage thus depends on the purpose of the diagnostic assessment.

2.1.2 Item Parameter Estimation using EM Algorithm

Taking into account the fact that the attribute pattern for respondent *j*, α_j, is an unobservable latent variable, Equation 2.2 is rewritten as

$$p_{ij} = P(Y_{ij} = 1; \alpha_j) = \pi_i^* \prod_{k=1}^{K} r_{ik}^{*(1-\alpha_{jk})q_{ik}} \tag{2.7}$$

Note that the probability function p_{ij} is now a function of the attribute pattern α_j and the item parameters. For a random sample of *J* respondents, the log-likelihood function for item *i* is

$$L(Y, \alpha; \beta_i) = \sum_{j=1}^{J} Y_{ij} \ln(p_{ij}) + \sum_{j=1}^{J} (1 - Y_{ij}) \ln(1 - p_{ij}) \tag{2.8}$$

where $\beta_i = (\pi_i^*, r_{ik}^*)$ are the item parameters for item *i*, which are unknown in practice and need to be estimated, and α is a *J* by *K* matrix with each row being the attribute pattern for one of *J* respondents. For a given set of observations of item responses, $L(Y, \alpha; \beta_i)$ is a function of both unobservable attribute patterns and unknown item parameters. Our goal is to estimate the item parameters β_i using the respondents' responses to an assessment. However, it is not easy to find the MLE for β_i directly with the presence of unobservable attribute patterns in the log-likelihood function (Equation 2.8).

Instead of maximizing Equation 2.8 directly, the EM algorithm (expectation-maximization; Dempster, Laird & Rubin, 1977) solves an easier maximization problems for a sequence of functions that contain only the β_i as variables. The resulting sequence of MLEs, $\hat{\beta}_i$ s, converge to the β_i that maximizes that original log-likelihood function as in Equation 2.8. In particular, the expectation (E-) step finds the expectation of the log-likelihood function (Equation 2.8) with respect to the conditional distribution of attribute

patterns, given the most recent estimates of both the item parameters and the probability of any given attribute pattern. In the maximization (M-) step, updated item parameter estimates are obtained by maximizing the expected log-likelihood resulted from the E-step. The implementation is described in detail as follows.

Let $\beta_i^{(m-1)}$ and $\lambda_l^{(m-1)}$ be item parameter estimates and the probability that a randomly selected examinee has attribute pattern α_l at the end of the $(m\text{-}1)^{\text{th}}$ iteration. At the m^{th} iteration, the posterior probability that examinee j has the attribute pattern α_l which is denoted as $P^{(m)}(\alpha_l \mid Y_j, \beta_i^{(m-1)}, \lambda_l^{(m-1)})$, could be obtained by plugging $\beta_i^{(m-1)}$ and $\lambda_l^{(m-1)}$ in the position of β_i and λ_l in Equation 2.4. The probability that a randomly selected examinee has attribute pattern α_l could then be updated as (Bartholomew and Knott, 1999, p.138)

$$\lambda_l^{(m)} = \frac{\sum_{j=1}^{J} P^{(m)}(\alpha_l \mid Y_j, \beta_i^{(m-1)}, \lambda_l^{(m-1)})}{J} \tag{2.9}$$

In the E-step of the EM algorithm, the expectation of the log-likelihood function is taken with respect to the conditional distribution of attribute patterns given item responses, under the current item parameter estimate and the current probabilities of any given attribute patterns. This results in:

$$Q(\beta_i \mid \beta_i^{(m-1)}, Y) = \sum_{j=1}^{J} Y_{ij} E_{\alpha_l \mid Y_j, \beta_i^{(m-1)}}(\ln(p_{ij})) + \sum_{j=1}^{J} (1 - Y_{ij}) E_{\alpha_l \mid Y_j, \beta_i^{(m-1)}}(\ln(1 - p_{ij})) \tag{2.10}$$

where, Y is a matrix consisting of J examinees' responses on the whole assessment, and

$$E_{\alpha_l \mid Y_j, \beta_i^{(m-1)}}(\ln(p_{ij})) = \sum_{l=1}^{L} \ln(p_{il}) P^{(m)}(\alpha_l \mid Y_j, \beta_i^{(m-1)}, \lambda_l^{(m-1)}) \text{ , and} \tag{2.11}$$

$$E_{\alpha_l \mid Y_j, \beta_i^{(m)}}(\ln(1 - p_{ij})) = \sum_{l=1}^{L} \ln(1 - p_{il}) P^{(m)}(\alpha_l \mid Y_j, \beta_i^{(m-1)}, \lambda_l^{(m-1)}) \text{ .} \tag{2.12}$$

Note that $Q(\beta_i \mid \beta_i^{(m-1)}, Y)$ no longer contains the unobservable attribute patterns and it is now a function of only the item parameters.

In the M-step, the item parameter estimates can be updated by maximizing the expected log-likelihood function $Q(\beta_i \mid \beta_i^{(m-1)}, Y)$ with respect to β_i,

$$\beta_i^{(m)} = \arg\max_{\beta_i} Q(\beta_i \mid \beta^{(m-1)}, Y) \tag{2.13}$$

The E-step and M-step repeat until a convergence criterion is met. The convergence criterion used for estimating the DINA model was that the maximum difference between the previous and the current parameter estimate was smaller than 0.0001 (de la Torre, 2008). Other commonly used criteria include the absolute log-likelihood convergence criterion and the relative log-likelihood convergence criterion (Muthén, & Muthén).

2.2 Examples

2.2.1 Simulated Data

The simulated data consists of replicating responses from 3000 examinees to 30 items. Four attributes were measured, and examinees were classified into $2^K = 16$ attribute patterns. The baseline parameters π_j were generated from a uniform distribution (0.6, 1.0). The penalty parameters r_j were generated from a uniform distribution (0.05, 0.4). Both ranges are set to result in behavior comparable to the ranges of slipping and guessing in the DINA model estimation (de la Torre, 2009). Initial item parameters values to begin the estimation were chosen as the one set out of 10 sets of values generated in a similar way, which had the largest likelihood. A script was written for the R statistical software environment (R Development Core Team, 2010) to implement the

EM algorithm. The algorithm is considered to be convergent when both the absolute log-likelihood convergence criterion (the absolute differences between the observed log-likelihood value from the previous iteration and that from the current iteration is less than 0.0001), and the relative log-likelihood convergence criterion (the proportion of the absolute differences between the observed log-likelihood value from the previous iteration and that from the current iteration out of the absolute observed log-likelihood value is less than 0.0001) are met. Preliminary studies to the one reported here showed that the criteria result in similar levels of accuracy for both item parameter estimation and the estimation of examinees' attribute pattern. The amount of time required for convergence for the different criteria was similar. 100 replications were implemented.

Examinees' attribute patterns

When diagnosing a set of attributes in practice, the mastery of one attribute most likely affects the mastery of another attribute, so it is reasonable to assume that the attributes are correlated. Examinees' attribute patterns were generated from a multivariate normal distribution with $\mu = (0, 0, 0, 0)$ and correlation matrix

$$\Sigma = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix},$$

i.e. the four attributes are positively correlated and the correlation coefficient for any pair of attributes is 0.5. The generated values were then converted to 0 or 1 using the cut-off value of 0.5. With $\mu = (0, 0, 0, 0)$, the marginal proportion of each attribute is 0.5, which is similar to the setting in studies such as Henson and Templin (2009) in which marginal attribute difficulty was held constant. Correlations between attributes of this magnitude

are widely reported in educational studies (e.g. Kunina-Habenicht, Rupp & Wilhelm, 2011).

Q-matrix Specification

The Q-matrix for this data set is given in Table 2.1. There are 17 items loading on the 1st attribute, 17 items on the 2nd attribute, 16 items on the 3rd attribute, and 17 items on the 4th attribute. 6 items tap one attribute, 13 items tap two attributes, 9 items tap three attribute and 4 items tap all the four attributes. The items measure two attributes on average.

Table 2.1 Q-Matrix for K=4 and J=30

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

Table 2.2 gives the mean item parameter estimates over the 100 replications. For comparison, the true parameter values are listed in parentheses. The estimates for the baseline parameters are close to the true values: the differences between the estimated

baseline values and true baseline values are within 0.0021. The estimates for the penalty parameters are close to their true values as well. The mean differences between the estimated penalty values and the true penalty values are within 0.0049 at the 1st attribute, 0.0056 at the 2nd attribute, 0.0031 at the 3rd attribute, and 0.0074 at the 4th attribute. The magnitudes of the difference are similar to those reported for the DINA model (de la Torre, 2009), demonstrating that the EM algorithm provides accurate parameter estimates.

The root mean square error (RMSE) for the baseline parameter estimate for item *i*, $\hat{\pi}_i$, over the 100 replications is calculated as

$$RMSE.\pi_i = \sqrt{\frac{1}{100}\sum_{r=1}^{100}(\hat{\pi}_i - \pi_i)^2} \tag{2.14}$$

The RMSE for the penalty parameter estimate for item *i* and attribute *k*, $\hat{r}_{ik}$, is calculated similarly. Table 2.3 summarizes the baseline and penalty parameter estimates. The RMSEs for the baseline parameter estimates are within the interval from 0.0052 to 0.0269. The RMSEs for the penalty parameter estimates range from 0.0163 to 0.0196 for items requiring one attribute, from 0.0099 to 0.0287 for items measuring on two attributes, from 0.0116 to 0.0341 for items measuring on three attributes, and from 0.0171 to 0.0420 for items measuring on all of the four attributes. The EM algorithm does best in estimating the item parameters for single-attribute items, and the estimation accuracy decreases as the number of attribute required by an item increases.

Table 2.2 Item Parameter Estimates for the reduced RUM with K=4 and J=30 using the EM Algorithm.

			Penalty Probabilities							
Item	Baseline		*k*=1		*k*=2		*k*=3		*k*=4	
1	0.6462	-0.6448	0.3557	-0.3529	----	----	0.3159	-0.3136	0.3455	-0.3475
2	0.778	-0.7812	----	----	0.3388	-0.3372	0.2608	-0.2621	----	----
3	0.7717	-0.771	0.2586	-0.2576	----	----	----	----	----	----
4	0.6883	-0.69	0.2481	-0.2436	0.2607	-0.258	----	----	0.328	-0.3304
5	0.8577	-0.8567	----	----	----	----	0.1698	-0.1702	0.08	-0.0824
6	0.9258	-0.9261	0.0781	-0.0781	----	----	0.2998	-0.3019	0.0652	-0.0631
7	0.9058	-0.9039	0.1391	-0.1416	0.3919	-0.3863	0.3178	-0.3192	0.1175	-0.118
8	0.9588	-0.9587	0.199	-0.1949	----	----	0.0986	-0.0985	0.3423	-0.3422
9	0.8524	-0.8508	----	----	0.1357	-0.1372	----	----	0.1639	-0.1616
10	0.6577	-0.6583	----	----	0.3545	-0.3567	----	----	----	----
11	0.9629	-0.9618	0.1811	-0.1815	----	----	0.0732	-0.0727	----	----
12	0.7407	-0.7405	0.0766	-0.0774	----	----	----	----	0.2081	-0.205
13	0.9063	-0.9065	----	----	0.0518	-0.0518	----	----	0.0842	-0.0831
14	0.6135	-0.612	----	----	----	----	----	----	0.1758	-0.1767
15	0.6733	-0.6739	----	----	0.3514	-0.3478	----	----	----	----
16	0.9824	-0.9828	0.2602	-0.2606	----	----	----	----	0.1084	-0.1069
17	0.8245	-0.823	0.3185	-0.3172	0.2512	-0.2496	0.3162	-0.3169	0.3749	-0.3675
18	0.8634	-0.8624	----	----	----	----	0.2296	-0.2294	----	----
19	0.6455	-0.6438	0.3644	-0.3596	----	----	----	----	0.0693	-0.0696
20	0.7198	-0.7177	----	----	0.1164	-0.116	0.3377	-0.3377	----	----
21	0.7842	-0.7849	0.0993	-0.1027	0.18	-0.1777	----	----	0.2499	-0.2481
22	0.7149	-0.7136	0.3967	-0.3967	0.2842	-0.2836	----	----	----	----
23	0.76	-0.759	----	----	0.3695	-0.3694	----	----	----	----
24	0.6019	-0.6041	0.1354	-0.1345	----	----	0.2665	-0.2663	0.0733	-0.074
25	0.6842	-0.6852	----	----	0.1672	-0.1691	0.0838	-0.083	----	----
26	0.6475	-0.6492	0.1255	-0.1254	----	----	----	----	0.2269	-0.2253
27	0.9878	-0.9881	0.1819	-0.1832	0.3291	-0.3282	0.3104	-0.3078	----	----
28	0.6284	-0.6286	0.3465	-0.3421	0.1611	-0.1651	0.2549	-0.2551	----	----
29	0.9385	-0.9369	----	----	0.3295	-0.329	0.3611	-0.3635	0.3508	-0.3502
30	0.7969	-0.7975	----	----	0.1657	-0.1672	0.2467	-0.2436	----	----

Table 2.3 Summary of RMSEs for Item Parameter Estimates

		Number of Attributes Required			
	Baseline	1	2	3	4
Min.	0.0052	0.0163	0.0099	0.0116	0.0171
Max.	0.0269	0.0196	0.0287	0.0341	0.0420

The correct classification rate (CCR) is calculated by dividing the number of examinees whose attribute patterns are correctly identified by the total number of examinees. Two classification methods EAP and MAP using the estimated parameters were used to classify examinees' attribute patterns. The CCR from EAP is 0.7991, and the CCR from MAP is 0.8080.

2.2.2 ECPE Data

The Examination for the Certificate for Proficiency in English (ECPE) is a test developed by the English Language Institute of the University of Michigan with the aim of measuring English skills of examinees whose native language is not English. This ECPE data was described and analyzed by Henson & Templin (2007) and Liu, Douglas, and Henson (2009). It consists of the responses of 2922 examinees to 28 multiple-choice questions in the grammar section of the ECPE test. An example item is shown below:

I have always _____ snow.

a. to enjoy

b. enjoyed

c. enjoying

d. to enjoyed

The items were designed to measure three grammar attributes: knowledge of (1) morphosyntactic rules, (2) cohensive rules, and (3) lexical rules (Henson & Templin, 2007). Table 2.4 shows the Q-matrix for the 28 items. 19 items tap one attribute and 9 items tap two attributes; there are no items tapping all the three attributes. There are 14 items measuring on the attribute "morphosyntactic rules", 6 items on the attribute "cohensive rules", and 17 items on the attribute "lexical rules".

Table 2.4 Q-Matrix for the 28 ECPE Items

	Attributes				Attributes		
Item	Mor	Coh	Lex	Item	Mor	Coh	Lex
G1	1	1	0	G17	0	0	1
G3	0	1	0	G18	1	0	1
G4	1	0	1	G19	0	1	1
G5	0	0	1	G20	0	0	1
G6	0	0	1	G21	0	0	1
G8	0	0	1	G22	1	0	1
G9	1	0	1	G23	1	0	1
G10	0	1	0	G24	0	0	1
G11	0	0	1	G25	0	1	0
G12	1	0	0	G26	0	1	0
G13	1	0	1	G27	1	0	0
G14	1	0	1	G28	1	0	0
G15	1	0	0	G29	1	0	0
G16	1	0	0	G30	0	0	1

The reduced RUM was fit the ECPE data using the EM algorithm described in the methods section. Potential sets of initial values for the baseline parameters were generated from a uniform distribution (0.6, 1) and the initial values for the penalty parameters were generated from a uniform distribution (0.05, 0.4). The initial values were chosen from among 10 sets of random values by selecting the set with the largest likelihood value. The stopping criterion was that the difference between item parameter

estimates from two consecutive iterations was smaller than 0.0001, and the algorithm took about 15 minutes to converge. Table 2.5 shows the item parameter estimates using the EM algorithm. The estimated values were compared to those using MCMC from Henson & Templin (2007). Both baseline and penalties estimates from the two estimation methods agreed within 0.01on all parameters for the 28 items.

Table 2.5 Item Parameter Estimates of the reduced RUM using EM and MCMC

			Penalties					
	Baseline		Mor		Coh		Lex	
Item	EM	MCMC	EM	MCMC	EM	MCMC	EM	MCMC
G1	0.93	0.93	0.88	0.89	0.85	0.84	--	--
G3	0.91	0.90	--	--	0.81	0.81	--	--
G4	0.78	0.78	0.64	0.63	--	--	0.82	0.83
G5	0.83	0.82	--	--	--	--	0.56	0.56
G6	0.96	0.96	--	--	--	--	0.78	0.78
G8	0.93	0.92	--	--	--	--	0.76	0.76
G9	0.94	0.94	0.74	0.73	--	--	0.70	0.70
G10	0.97	0.97	--	--	0.84	0.84	--	--
G11	0.79	0.79	--	--	--	--	0.67	0.67
G12	0.89	0.89	0.58	0.58	--	--	--	--
G13	0.93	0.92	0.77	0.77	--	--	0.69	0.69
G14	0.73	0.73	0.53	0.51	--	--	0.36	0.38
G15	0.91	0.90	0.73	0.73	--	--	--	--
G16	0.83	0.82	0.66	0.66	--	--	--	--
G17	0.96	0.96	--	--	--	--	0.76	0.76
G18	0.91	0.91	0.75	0.75	--	--	0.71	0.72
G19	0.94	0.94	--	--	0.92	0.93	0.92	0.91
G20	0.91	0.91	--	--	--	--	0.79	0.78
G21	0.84	0.84	--	--	--	--	0.54	0.53
G22	0.76	0.76	0.50	0.49	--	--	0.51	0.52
G23	0.92	0.92	0.85	0.85	--	--	0.70	0.69
G24	0.80	0.79	--	--	--	--	0.37	0.37
G25	0.94	0.94	--	--	0.70	0.70	--	--
G26	0.70	0.70	--	--	0.48	0.47	--	--
G27	0.78	0.77	0.67	0.68	--	--	--	--
G28	0.78	0.78	--	--	--	--	0.69	0.69
G29	0.70	0.69	0.42	0.43	--	--	--	--
G30	0.91	0.91	--	--	--	--	0.70	0.69

Table 2.6 displays the counts and percentages of examinees classified into each attribute pattern using EAP and MAP. The two classification methods had similar results. Most examinees were classified into two attribute patterns: mastering all three attributes or mastering none of the attributes, with MAP placing more examinees into these extreme categories than EAP. Based on the results from the EAP classifications, about 37% of examinees have mastered the 1st attribute "morphosyntactic rules", 57% of examinees have mastered "cohensive rules" and 67% of examinees have mastered "lexical rules". Based on the results from MAP classifications, about 40% of examinees have mastered the 1st attribute "morphosyntactic rules", 57% of examinees have mastered "cohensive rules" and 66% of examinees have mastered "lexical rules". This implies that knowledge of "morphosyntactic rules" is the most difficult attribute among the three.

Table 2.6 Proportions of examinees from each attribute pattern using EAP and MAP

	Attributes			EAP		MAP	
Patterns	Mor	Coh	Lex	# of examinees	%	# of examinees	%
1	0	0	0	908	31.07	979	33.50
2	1	0	0	7	0.24	1	0.03
3	0	1	0	13	0.44	0	0.00
4	0	0	1	343	11.74	280	9.58
5	1	1	0	5	0.17	3	0.10
6	1	0	1	12	0.41	3	0.10
7	0	1	1	573	19.61	507	17.35
8	1	1	1	1061	36.31	1149	39.32

Another way to verify that the estimates for the reduced RUM using the EM algorithm are reasonable is to compare the results to those from the DINA model. For items tapping only one attribute, the slip and guess as defined in the DINA model could be found using the reduced RUM item parameters:

$$s_i = 1 - \pi_i \tag{2.15}$$

$$g_i = \pi_i \cdot r_i \tag{2.16}$$

Table 2.7 shows the slip and guess estimates (Liu, & Douglas, & Henson, 2009) from both the DINA model and the reduced RUM for the 19 one-attribute items. The parameter estimates agree on most of the items.

Table 2.7 Slip and Guess from the DINA and the reduced RUM

				Reduced RUM		DINA	
Item	Mor	Coh	Lex	Slip	Guess	Slip	Guess
G3	0	1	0	0.09	0.74	0.1	0.74
G5	0	0	1	0.17	0.46	0.16	0.48
G6	0	0	1	0.04	0.75	0.04	0.76
G8	0	0	1	0.07	0.70	0.07	0.72
G10	0	1	0	0.03	0.81	0.04	0.81
G11	0	0	1	0.21	0.53	0.2	0.53
G12	1	0	0	0.11	0.51	0.16	0.48
G15	1	0	0	0.09	0.66	0.12	0.63
G16	1	0	0	0.17	0.54	0.21	0.52
G17	0	0	1	0.04	0.73	0.01	0.75
G20	0	0	1	0.09	0.71	0.09	0.73
G21	0	0	1	0.16	0.45	0.15	0.47
G24	0	0	1	0.20	0.30	0.19	0.32
G25	0	1	0	0.06	0.66	0.07	0.66
G26	0	1	0	0.30	0.33	0.31	0.33
G27	1	0	0	0.22	0.52	0.27	0.51
G28	1	0	0	0.22	0.54	0.21	0.55
G29	1	0	0	0.30	0.29	0.37	0.27
G30	0	0	1	0.09	0.64	0.09	0.66

CHAPTER 3

Q-MATRIX VALIDATION METHOD FOR THE REDUCED RUM

As described in section 1.4, the sequential search method (de la Torre, 2008) was demonstrated to successfully correct a true Q-matrix with misspecifications when response data was modeled by the DINA model. In this chapter, the Q-matrix validation method for the reduced RUM will be developed based on the idea of sequential searching based on δ. Note that δ defined in section 1.4 is for the DINA model, especially it is defined, for each item, on the partition of respondents by whether a respondent has mastered all required attributes to answer an item correctly. Under the reduced RUM, the partitions of respondents vary across items. However, the definition of δ could still apply to the reduced RUM, because the reduced RUM is a compensatory model. Under this model, the probabilities of correct responses for respondents who have mastered all required attributes are always larger than the probabilities for those who don't, and δ always takes positive values.

The chapter is structured as the following. In section 3.1, sequential search method for the reduced RUM will be developed. The estimates for δ under the reduced RUM will be derived in section 3.1.1. The performance of three variations of the sequential search methods for the reduced RUM will be investigated using simulation studies. In section 3.2, a two-stage Q-matrix validation method will be proposed. It will be applied on both simulated data and a real data set.

3.1 Sequential Search Method for the reduced RUM

3.1.1 Estimation of δ under the reduced RUM

Assuming that a randomly selected respondent cannot be classified into two attribute patterns at the same time, (1.23) for item *i* whose q-vector is $\alpha_{l'}$ can be written as

$$\delta_{il'} = \sum_{\{\xi_{ll'}=1\}} \lambda_l \cdot p_{il} \Big/ \sum_{\{\xi_{ll'}=1\}} \lambda_l - \sum_{\{\xi_{ll'}=0\}} \lambda_l \cdot p_{il} \Big/ \sum_{\{\xi_{ll'}=0\}} \lambda_l \tag{3.1}$$

Under the reduced RUM, p_{il} is given by (2.2), i.e., the probability that a respondent with attribute pattern α_l provides a correct respondent to item *i*. Unlike the DINA model, it is impossible to derive the closed form expression for the MLEs of baseline and penalty parameters for the reduced RUM. Note that the reduced RUM is a latent class model with binary variables. Under the framework of the latent class model, the MLEs of λ_l and p_{il} are given by (Bartholomew and Knott, 1999, p.138-p.139)

$$\hat{\lambda}_l = \sum_{j=1}^{J} P(\alpha_l \mid Y_j) \Big/ J \tag{3.2}$$

$$\hat{p}_{il} = \sum_{j=1}^{J} Y_{ij} \cdot P(\alpha_l \mid Y_j) \Big/ \sum_{j=1}^{J} P(\alpha_l \mid Y_j) \tag{3.3}$$

Plugging (3.2), (3.3) into (3.1), the $\delta_{il'}$ under the reduced RUM can be estimated as

$$\hat{\delta}_{il'}^{rRUM} = \frac{\sum_{\{\xi_{ll'}=1\}} \sum_{j=1}^{J} Y_{ij} \cdot P(\alpha_l \mid Y_j)}{\sum_{\{\xi_{ll'}=1\}} \sum_{j=1}^{J} P(\alpha_l \mid Y_j)} - \frac{\sum_{\{\xi_{ll'}=0\}} \sum_{j=1}^{J} Y_{ij} \cdot P(\alpha_l \mid Y_j)}{\sum_{\{\xi_{ll'}=0\}} \sum_{j=1}^{J} P(\alpha_l \mid Y_j)} \tag{3.4}$$

The estimate in (3.4) is essentially the same as that in (1.27). It is computed based on posterior distribution of attribute patterns $\{ P(\alpha_l | Y_j) \}$.

Let $\hat{\alpha}_j$ be the estimated attribute pattern for respondent *j* using either the MAP or EAP classification methods. $\hat{\xi}_{jl'} = \prod_{k=1}^{K} \hat{\alpha}_{jk}^{\alpha_{l'k}}$ indicates whether respondent *j* has all required attributes based the respondent's estimated pattern. A natural estimate of $\delta_{il'}$, which is based on the observed proportions of respondents in each group who provide correct responses, is given by

$$\tilde{\delta}_{il'}^{rRUM} = \sum_{j=1}^{J} Y_{ij} \cdot \hat{\xi}_{jl'} \Big/ \sum_{j=1}^{J} \hat{\xi}_{jl'} - \sum_{j=1}^{J} Y_{ij} \cdot (1 - \hat{\xi}_{jl'}) \Big/ \sum_{j=1}^{J} (1 - \hat{\xi}_{jl'}) \qquad (3.5)$$

The remaining of the sequential search method is the same as in described in section1.4. The search for true attributes begins with all single-attribute patterns. One additional attribute is added at each time until the stopping criteria is met. A simulation study was conducted in Section 3.1.2 to examine the performance of the following sequential search methods for validating misspecified Q-matrices when response data is modeled by the reduced RUM: 1) sequential search method based on the estimate $\hat{\delta}_{il'}^{rRUM}$; 2) sequential search method based on the estimate $\tilde{\delta}_{il'}^{rRUM}$ when respondents are classified using MAP; 3) sequential search method based on the estimate $\tilde{\delta}_{il'}^{rRUM}$ when respondents are classified using EAP.

3.1.2 Simulation Study

The simulated data sets consisted of responses from three sample sizes (250/500/1000) of examinees on 19 items with each loading on one of the combinations of four fine-grained attributes. The baseline parameters π_j were generated from a uniform distribution (0.8, 1.0). The penalty parameters r_j were generated from a uniform distribution (0.05, 0.3). An item with large baseline values and small penalty values is more capable in discriminating its loaded attributes. Table 3.1 shows the true Q-matrix and item parameters used to generate the simulated data sets. The true Q-matrix is complete (Liu, Xu & Ying, 2011), i.e., for each attribute, there exits an item only requiring that attribute. The 1s for penalty parameters indicated the corresponding attributes are not required for an item. Examinees' attribute patterns were generated from a flat distribution, i.e. examinees are equally likely to be classified into each of the 16 possible attribute patterns.

In addition to the true Q-matrix, 10 misspecified Q-matrices were used to estimate analyze the simulated data sets, especially to estimate item parameters and respondents' attribute patterns. These 10 matrices shown in Table 3.2 were defined in a similarly as those in de la Torre's (2008). The first 9 Q matrices each have a single misspecified q-vector and the last one has three misspecified q-vectors. The item parameters were estimated using an EM algorithm written in R statistical software environment (Core development team, 2011) with convergence criterion of 0.0001. Examinees were then classified into $2^K = 16$ attribute patterns. 100 replications were implemented.

Table 3.1 Correct Q-matrix and Item Parameters used to Generate Simulated Data

	Attribute				Penalty	Attribute			
Item	1	2	3	4	Baseline	1	2	3	4
1	1	0	0	0	0.81	0.09	1.00	1.00	1.00
2	0	1	0	0	0.95	1.00	0.21	1.00	1.00
3	0	0	1	0	0.93	1.00	1.00	0.13	1.00
4	0	0	0	1	0.80	1.00	1.00	1.00	0.28
5	1	0	0	0	0.99	0.12	1.00	1.00	1.00
6	0	1	0	0	0.97	1.00	0.29	1.00	1.00
7	0	0	1	0	0.96	1.00	1.00	0.23	1.00
8	0	0	0	1	0.91	1.00	1.00	1.00	0.27
9	1	1	0	0	0.82	0.28	0.16	1.00	1.00
10	1	0	1	0	0.83	0.11	1.00	0.23	1.00
11	1	0	0	1	0.89	0.07	1.00	1.00	0.10
12	0	1	1	0	0.90	1.00	0.16	0.24	1.00
13	0	1	0	1	0.83	1.00	0.28	1.00	0.16
14	0	0	1	1	0.94	1.00	1.00	0.25	0.28
15	1	1	1	0	0.92	0.09	0.28	0.11	1.00
16	1	1	0	1	0.93	0.13	0.09	1.00	0.12
17	1	0	1	1	0.93	0.14	1.00	0.29	0.27
18	0	1	1	1	0.93	1.00	0.24	0.24	0.12
19	1	1	1	1	0.91	0.20	0.30	0.11	0.25

Table 3.2 Summary of Q-vector Misspecifications

		Q-vector before Altered				Q-vector after Altered				
	Item	Attribute				Attribute				Number of
Conditions	Altered	1	2	3	4	1	2	3	4	Alterations
0	0	---	---	---	---	---	---	---	---	---
1	1	1	0	0	0	0	1	0	0	2
2	1	1	0	0	0	1	1	0	0	1
3	9	1	1	0	0	1	0	0	0	1
4	9	1	1	0	0	0	1	1	0	2
5	9	1	1	0	0	1	1	1	0	1
6	15	1	1	1	0	0	1	1	0	1
7	15	1	1	1	0	0	0	1	0	2
8	15	1	1	1	0	0	1	1	1	2
9	15	1	1	1	0	0	0	1	1	3
10	1	1	0	0	0	1	1	0	0	5
	9	1	1	0	0	1	0	1	0	
	15	1	1	1	0	0	1	1	1	

The performance of a method is evaluated by the correct recovery rate (CRR), i.e. the percentage of replications when the resulting Q-matrix from a method is identical to the true Q-matrix. High CRR values indicate strong capability of a method in recovering the true Q-matrix from a misspecified one. Figure 3.1 shows CRRs of three variations of the sequential search method under the 10 simulation conditions: 1) sequential search method based on $\hat{\delta}^{rRUM}$; 2) sequential search method based on $\tilde{\delta}^{rRUM}$ when respondents are classified using MAP; 3) sequential search method based on $\tilde{\delta}^{rRUM}$ when respondents are classified using EAP. From Figure 3.1, we have the following observations:

1. High CRR values are associated with large sample size. The average CRR is about 87% for all three methods for sample size of 1000. However, for sample size of 500, the average CRR drops to 65.7% for the sequential search method based on $\tilde{\delta}^{rRUM}$ with classification method EAP, and 68% for the other two methods for sample size of 500. With sample size of 250, the average CRR drops dramatically to 37.2% for the sequential search method based on $\tilde{\delta}^{rRUM}$ with classification method EAP, and 42% for the other two methods. The performance of the sequential search method for the reduced RUM in correcting a Q-matrix with misspecifications is not satisfactory when sample size is small. The reason is that model estimation of the reduced RUM using a Q-matrix with misspecifications is less accurate for small sample sizes than large sample sizes.
2. The three methods have similar level of capability in recovering a true Q-matrix from a misspecified one, because the three methods have similar

average CRRs at different values of sample size. For sample sizes of 1000 and 500, the sequential search method based on $\hat{\delta}^{rRUM}$ has larger variations in CRRs over simulation conditions than the other two methods, indicating that the sequential search method based on $\hat{\delta}^{rRUM}$ is more sensitive to the types of misspecifications and the number of misspecifications. The reason is that, the extent to which the estimate $\hat{\delta}^{rRUM}$ is close to the true δ^{rRUM} depends on the accuracy of model estimation, which is affected by the type of misspecifications and the number of misspecifications. On the other hand, the estimate $\tilde{\delta}^{rRUM}$ which is computed based on observed proportions is more resistant to the effect of model estimation. The p-value from Kruskal-Wallis rank sum test with null hypothesis that there is significant difference in the correct recovery rates between the three sequential search methods is 0.8697, which shows evidence that there is no significant difference in the performance of validating a Q-matrix between the three sequential search methods.

3. Two factors, type of misspecifications, i.e., whether it is underspecified, overspecified, or the combination the two, and numbers of misspecifications, i.e., how many 0s in a q-vector are wrongly coded as 1, and how many 1s are wrongly coded as 0, have impact on the performance of the sequential search methods. The sequential search methods have lower CRRs at the Q-matrices with two or more misspecified elements than at those Q-matrices with only one misspecification on the same items. When the number of

misspecifications is constant and the misspecified items are fixed, the methods have higher CRRs at the Q-matrices with misspecification type of adding an unrequired attribute than the type of deleting a required attribute. The effect of the two factors on the validation performance will be further explored in Chapter 5.

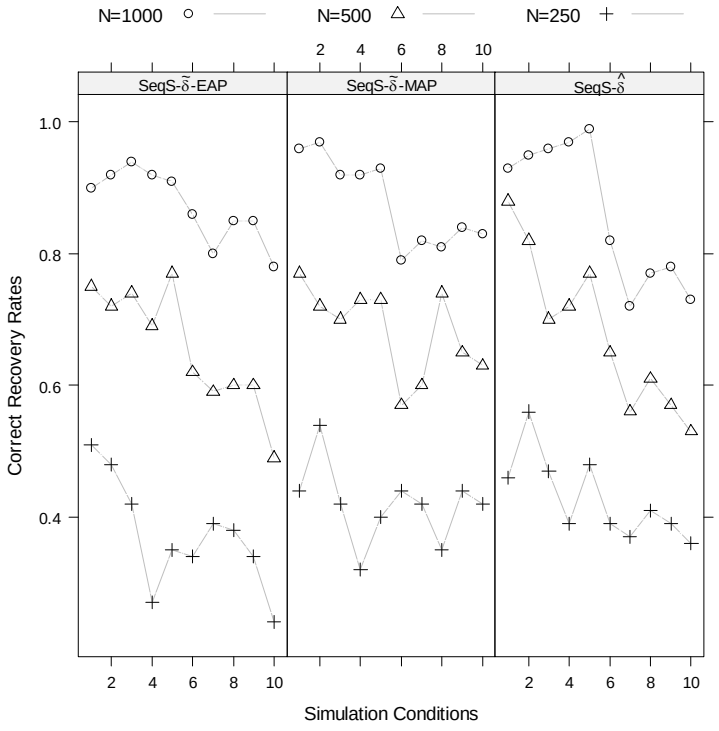

Figure 3.1 Correct Recovery Rates using the Sequential Search Methods for the Reduced RUM

Figure 3.2 to Figure 3.5 show fours cases when the three sequential search methods for the reduced RUM didn't make correct changes to a misspecified Q-matrix. Figure 3.2 shows the proportions of replications when the methods didn't made change to the misspecified Q-matrices. Ideally, the proportions are expected to be all 0s, because none of the misspecified Q-matrices are identical to the true one, and at least one change has to be made. When sample size is 250, the method based on $\tilde{\delta}^{rRUM}$ with MAP, and the

method based on $\hat{\delta}^{rRUM}$ didn't make changes to with misspecifications 4% of times, while the method based on $\tilde{\delta}^{rRUM}$ with EAP failed to make changes to a misspecified Q-matrix 2% of times. When sample size is 500, the method based on $\tilde{\delta}^{rRUM}$ with EAP made changes under all simulation conditions. The proportions of failing to make changes drops to 2% and 1% for the method based on $\tilde{\delta}^{rRUM}$ with MAP, and the method based on $\hat{\delta}^{rRUM}$, respectively. When sample size is 1000, the three sequential search methods made changes to the Q-matrices with misspecifications under all simulation conditions.

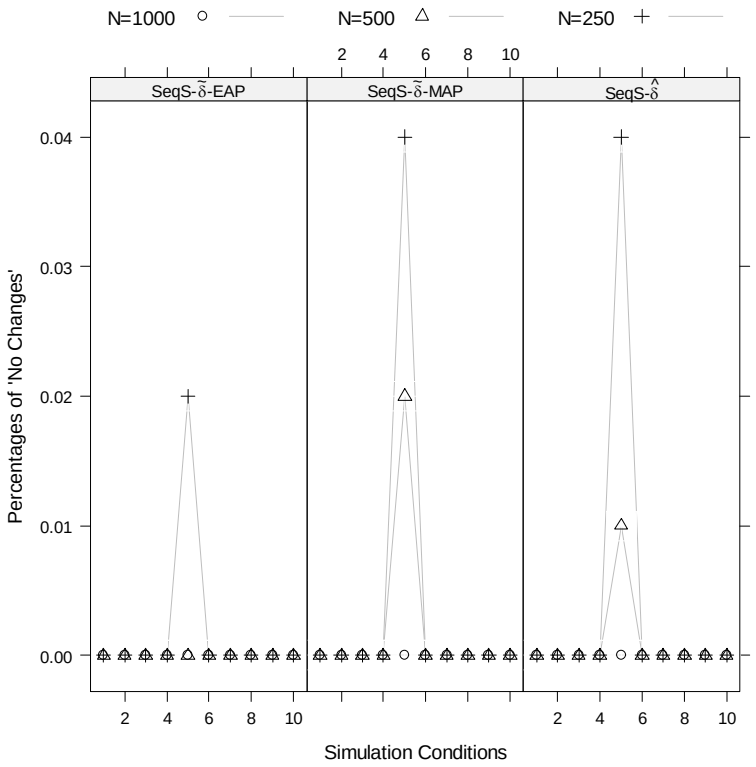

Figure 3.2 Proportions of Replications when the Sequential Search Methods for the reduced RUM Made No Changes to the Q-Matrices with Misspecifications

Figure 3.3 shows the proportions of replications when all misspecifications were identified by the sequential search methods, but some of them were changed incorrectly. None of other items were identified as misspecified in this case. High values are expected if a method works well in validating a Q-matrix with misspecifications. The sequential search method based on $\hat{\delta}^{rRUM}$ has higher proportions than the other two methods for all sample sizes. The average proportions across all simulation conditions with sample size of 1000 are 0.97, 0.965, 0.94, for the method based on $\hat{\delta}^{rRUM}$, the method based on $\tilde{\delta}^{rRUM}$ with MAP, and the method based on $\tilde{\delta}^{rRUM}$ with EAP, respectively. The average proportions for the three methods decrease to 0.831, 0.819, 0.764 when sample size is 500, and 0.566, 0.543, 0.486 when sample size is 250.

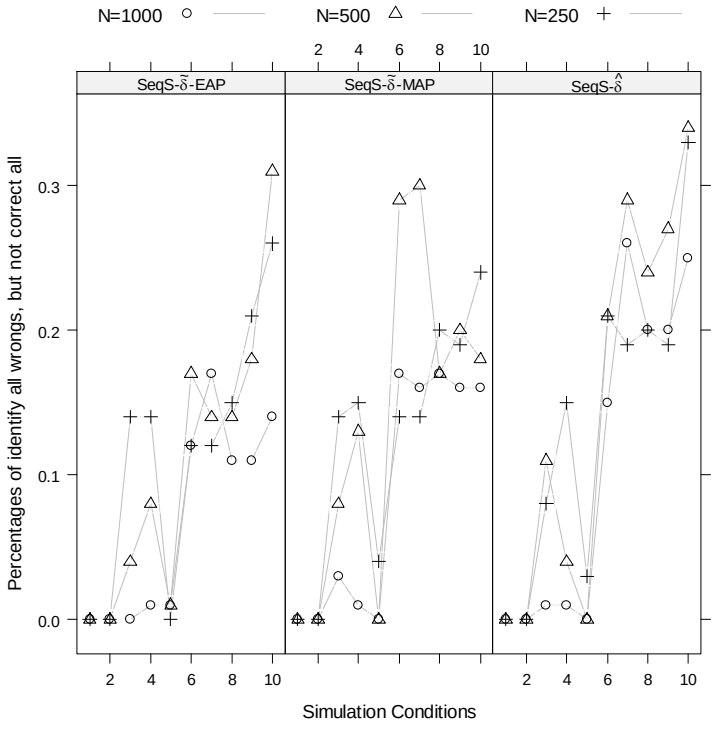

Figure 3.3 Proportions of Replications when the Sequential Search Methods for the reduced RUM Made Changes at the Exact Items with Misspecifications, but the changes were at least partially incorrect.

Figure 3.4 shows the proportions of times when changes made to a Q-matrix with misspecifications are not only at the misspecified items but also at items with correct q-vectors and should not be changed. As sample size gets smaller, these proportions get larger, indicating that the sequential search methods tend to make changes to correct items as the sample size decreases. The average proportions across all simulation conditions with sample size of 1000 are 0.03, 0.035, 0.06, for the method based on $\hat{\delta}^{rRUM}$, the method based on $\tilde{\delta}^{rRUM}$ with MAP, and the method based on $\tilde{\delta}^{rRUM}$ with EAP, respectively. The average proportions for the three methods increase to 0.167, 0.179, 0.236 when sample size is 500, and 0.509, 0.449, 0.425 when sample size is 250.

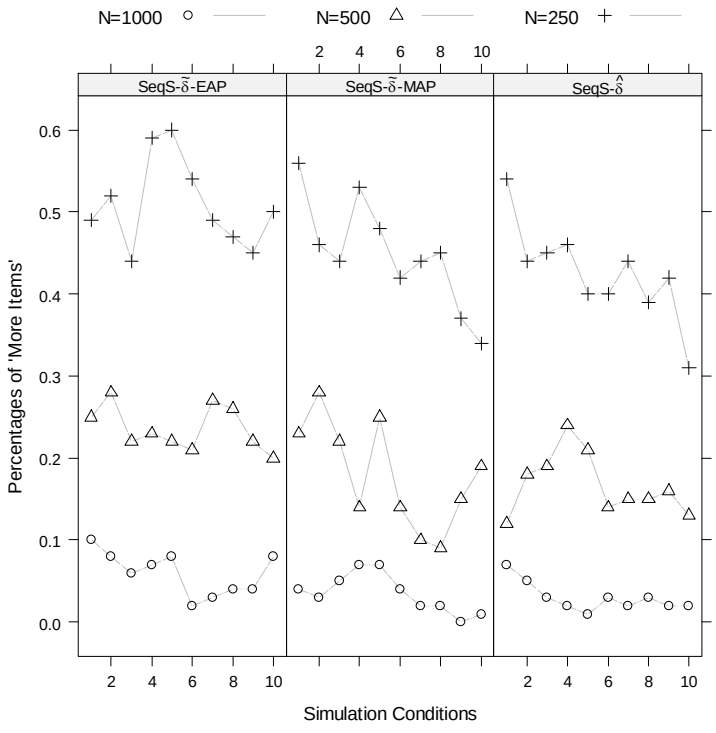

Figure 3.4 Proportions of Replications when the Sequential Search Methods for the reduced RUM Made Changes to Items Other than Those with Misspecifications

Figure 3.5 shows the proportions of replications when the methods made change to a subset of misspecified items, or to none of them. The proportions are expected to be all 0s if the method works well. When sample size is 250, these proportions are 2%, 3% and 5% for the method based on $\tilde{\delta}^{rRUM}$ with EAP, the method based on $\tilde{\delta}^{rRUM}$ with MAP and the method based on $\hat{\delta}^{rRUM}$, respectively. The methods based on $\tilde{\delta}^{rRUM}$ didn't make any of this type of validation errors for sample size of 1000 and 500.

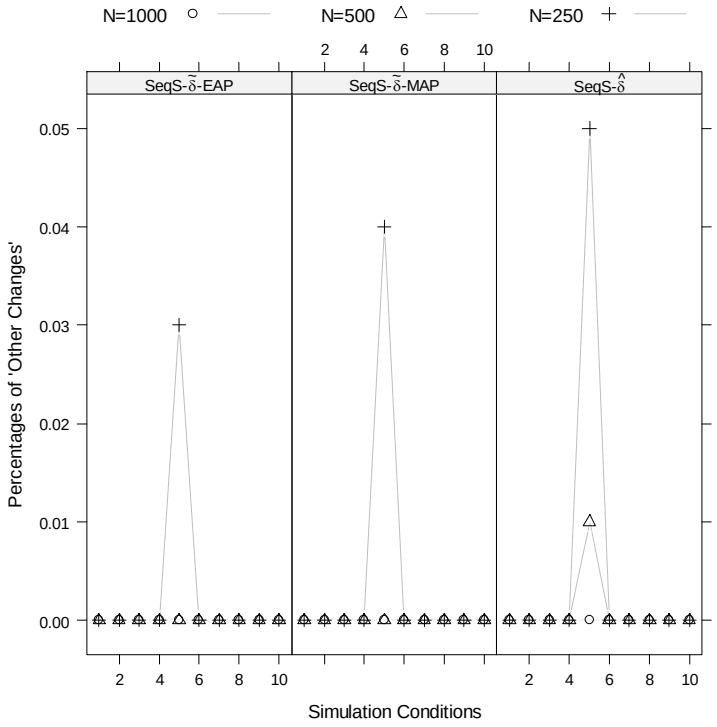

Figure 3.5 Proportions of Replications when the Sequential Search Methods for the reduced RUM Made Changes to More Items than Those with Misspecifications.

In this section, the sequential search method was extended to apply for the reduced RUM, which shares the same definition of correct q-vectors as the DINA model. Two estimators of δ were derived for the reduced RUM. One is computed based on the posterior distribution of attribute patterns, and the other is computed based on the

observed proportions of correct responses for two exclusive groups of respondents. Though these two estimators were derived for the reduced RUM, they can also be used to estimate δ for other non-compensatory DCMs. The simulation studies showed the performance of the sequential search methods based on different estimators are all affected by sample size, i.e., the methods work better for large sample size than small sample size. The sequential search method based on $\hat{\delta}^{rRUM}$ is more likely to make changes to the exact items with misspecifications. The sequential search methods based on $\widetilde{\delta}^{rRUM}$ tend to made changes not only to the wrong ones but also to items with correct q-vectors. The sequential search methods based on $\widetilde{\delta}^{rRUM}$ has smaller chance of missing the misspcified items than the other two. Overall, these methods are able to detect items with misspecified q-vectors, especially the sequential search method based on $\widetilde{\delta}^{rRUM}$ with classification method EAP.

However, unlike for the DINA model, the sequential search method works poorly for the reduced RUM in recovering a true Q-matrix from a misspecified Q-matrix for small sample sizes. The reason is that, the misspecifications in the input Q-matrix have a more severe impact on the estimation accuracy of δ for the reduced RUM than the DINA model. For a DINA model, there is only one probability of positive response for a respondent lacking of at least one of the required attributes. Thus, by (1.23), there might still be good separation in probabilities of positive responses between respondents who have mastered all required attributes and those who haven't, even when the item parameters estimates are off their true values due to the misspecifications of the Q-matrix. However, under the reduced RUM (1.5), the number of possible probabilities

varies for respondents who do not have all required attributes, and the separation of the two groups of respondents might not be clear especially when the parameters are not estimated well enough. The poor performance of the sequential search methods for the reduced RUM calls for effective validation methods that have high CRR values under most misspecification conditions and sample sizes.

3.2 Two-Stage Validation Methods for the Reduced RUM

According to the results in the previous section, the sequential search method for the reduced RUM is able to identify the items whose q-vectors were incorrectly coded under almost all combinations of sample size levels and misspecification conditions. However, it has a major disadvantage: it works well in validating a Q-matrix only when the sample size is large. In this section, two-stage validation methods for the reduced RUM will be developed based on the sequential search method and model selection techniques. The two-stage validation methods aim to improve the validation accuracy and work for small sample sizes.

3.2.1 Model Selection

The Akaike's information criterion (AIC) (Akaike, 1974) and the Bayesian information criterion (BIC) (Schwarz, 1976) are two indices often used in model selection. They are defined as the followings:

$$\text{AIC} = 2p - 2\ln(Lik) \tag{3.6}$$

$$\text{BIC} = -2\ln(Lik) + p*\ln(n) \tag{3.7}$$

Here, p is the number of model parameters, n is the number of respondents, and Lik represents the log-likelihood value for the estimated model. The candidate model with smallest AIC or BIC is considered to have the best model-data fit. Another third opinion is to select the model with the largest log-likehood values. Note that competitive models with difference in BIC values within 2 are considered to have the same level of model-data fit. A difference greater than 10 shows strong evidence that data is in favor of a particular model (Kass and Raftery, 1995).

At the first stage of the two-stage validation methods, the sequential search method for the reduced RUM is applied to a Q-matrix with misspecifications. For each item that is identified by the sequential search method as misspecified, a possible q-vector forms a candidate model for which model selection indices, AIC, BIC, and log-likelihood values are calculated. At the second stage, we search through all possible q-vectors, and the correct q-vector for an item is then selected according to the each of the four criterions: 1) the one with smallest AIC value; 2) the one with smallest BIC value; 3) the one with the largest log-likelihood value; 4) the simplest one with BIC value within a range of 10 of the smallest BIC value. The preliminary simulation results showed that the two criterions, log-likehood and AIC have the problem of overfitting, i.e. unnecessary attributes were added into the q-vectors. So we will focus on the 2^{nd} and 4^{th} criterions, and call the two-stage method based on the 2^{nd} criteria the BIC based sequential search method, and the other the BIC-range based sequential search method.

3.2.2 Simulation Study

A simulation study was conducted to compare the performance of the two two-stage validation methods and the sequential search method on validating 10 misspecified Q-matrices under the reduced RUM. The setting of the study was the same as described in section 3.1.2. For each set of simulated data, the three methods were applied to validate the same Q-matrix with misspecifications, and correction recovery rates (CRR) were then computed.

Figure 3.6 shows correct recovery rates for Q-Matrix validation methods for the reduced RUM based on $\widetilde{\delta}^{rRUM}$ with classification method MAP. From this figure, we see that, when $\widetilde{\delta}^{rRUM}$ and MAP classification method were used, the BIC-range based modified sequential search method worked the best among the three for all sample sizes. It has consistently high correction rates at all combinations of simulation conditions and sample sizes with average recovery rate of about 99.7% when sample sizes were 1000 and 500, 96.2% for sample size of 250. The BIC based modified sequential search method performed not as well as the BIC-range method, but still had overall high rates for all sample sizes. The average recovery rates were 92.9% for sample size of 1000, 92.2% for 500, and 88.9% for 250. The recovery rates are not consistent for all simulation conditions with correction rates at the first condition much lower than the others.

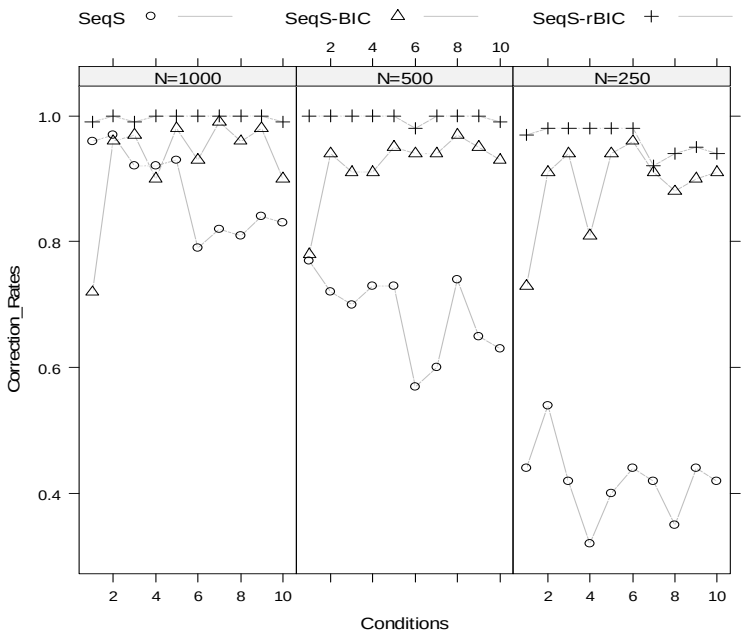

Figure 3.6 Correct Recovery Rates of Q-matrix Validation Methods for the reduced RUM when Sequential Search is based on $\tilde{\delta}^{rRUM}$ and Classification Method MAP

Figure 3.7 shows correct recovery rates for Q-Matrix validation methods for the reduced RUM based on $\tilde{\delta}^{rRUM}$ with classification method EAP. With classification method of EAP, the recovery rates for the BIC based method were greatly improved. The average recovery rates were 99% for sample size of 1000, 97.10% for 500, and 94.8% for 250. The recovery rate for the first misspecified matrix was also improved. The BIC range based method still worked slightly better than the BIC based method.

Figure 3.8 shows correct recovery rates for Q-Matrix validation methods for the reduced RUM based on $\hat{\delta}^{rRUM}$. It has similar results as the method based on $\tilde{\delta}^{rRUM}$ and classification methods EAP. The average recovery rates were 99.5% for sample size of 1000, 98.10% for 500, and 95.3% for 250 for the BIC based method, and 100% for sample size of 1000 and 500, and 96.1% for 250, for the BIC range based method. All

three figures show that the BIC based method and BIC range based methods work much better than the sequential search method only.

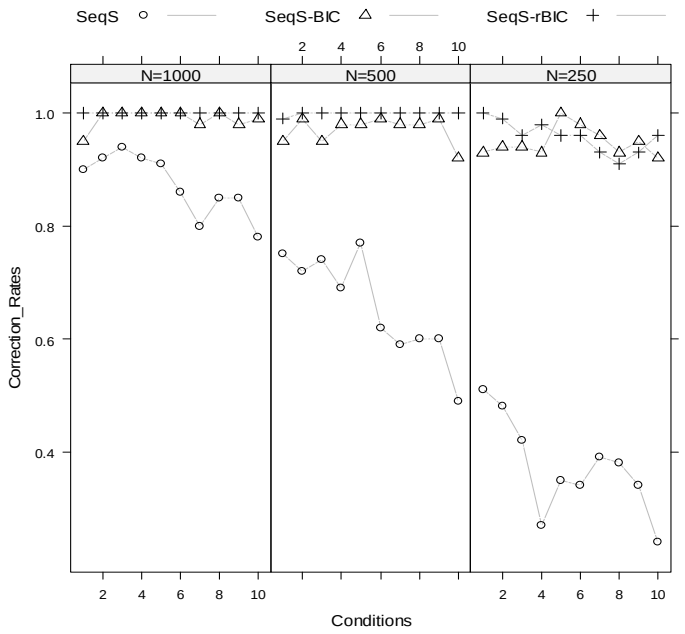

Figure 3.7 Correct Recovery Rates of Q-matrix Validation Methods for the reduced RUM when Sequential Search is based on $\tilde{\delta}^{rRUM}$ and Classification Method EAP

Table 3.3-3.5 show how the two BIC based methods compared to the sequential search method. The first three columns of the tables were sample sizes, simulation conditions, and the items with incorrect q-vectors at each condition. The 4th column lists percentages of replications when validation results from the BIC based method agreed with the sequential search method and were correct. The 5th column lists the percentages of replications when validation results from the BIC based method agreed with the sequential search method and were incorrect. The 6th column shows the percentages of replications when validation results from the BIC based method didn't agree with the sequential search method and were correct. The 7th column lists the percentages of replications when validation results from the BIC based method didn't agree with the

sequential search method and were incorrect. The last four columns were for validation results from the BIC range based sequential search method, and were arranged similarly as the previous ones. From the two tables, the agreement between the BIC based methods and the sequential search method decreased as the sample size decreases.

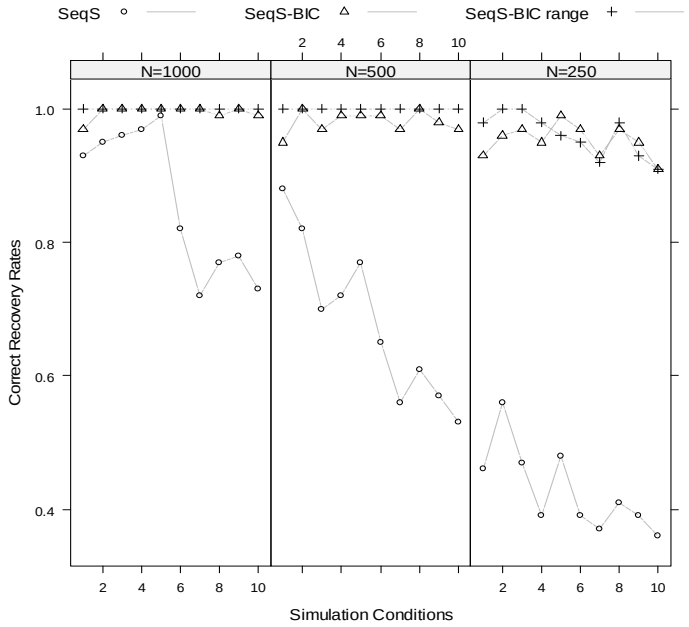

Figure 3.8 Correct Recovery Rates of Q-matrix Validation Methods for the reduced RUM when Sequential Search is based on $\hat{\delta}^{rRUM}$ and Classification Method EAP

Table 3.3 Agreement in Validation Results between Sequential Search Method based on $\tilde{\delta}^{rRUM}$ with MAP, and BIC based / BIC range based Methods. (Results shown in %)

			Select q-vector(s) with the smallest BIC				Select simplest q-vector(s) from a BIC range			
			Results agree with		NOT agree with		Results agree with		NOT agree with	
Sample		Item	Seq.Search		Seq.Search		Seq.Search		Seq.Search	
Size	Conditions	Altered	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
	1	1	70	1	2	27	96	0	3	1
	2	1	94	1	2	3	97	0	3	0
	3	9	92	3	5	0	92	1	7	0
	4	9	85	3	5	7	92	0	8	0
1000	5	9	93	2	5	0	93	0	7	0
	6	15	79	7	14	0	79	0	21	0
	7	15	82	1	17	0	82	0	18	0
	8	15	81	3	15	1	81	0	19	0
	9	15	84	2	14	0	84	0	16	0
	10	1,9,15	79	5	11	5	83	1	16	0
	1	1	65	5	13	17	77	0	23	0
	2	1	68	1	26	5	72	0	28	0
	3	9	68	5	23	4	70	0	30	0
	4	9	71	4	20	5	73	0	27	0
500	5	9	74	4	21	1	75	0	25	0
	6	15	48	1	46	5	48	0	50	2
	7	15	60	4	34	2	60	0	40	0
	8	15	74	3	23	0	74	0	26	0
	9	15	65	4	30	1	65	0	35	0
	10	1,9,15	62	2	31	5	63	1	36	0
	1	1	41	7	32	20	44	0	53	3
	2	1	53	2	38	7	54	0	44	2
	3	9	42	3	52	3	42	1	56	1
	4	9	32	11	49	8	32	0	66	2
250	5	9	48	1	46	5	48	0	50	2
	6	15	47	1	49	3	46	0	52	2
	7	15	42	5	49	4	39	0	53	8
	8	15	35	2	53	10	33	0	61	6
	9	15	44	7	46	3	40	0	55	5
	10	1,9,15	41	4	50	5	41	0	53	6

Table 3.4 Agreement in Validation Results between Sequential Search Method based on $\tilde{\delta}^{rRUM}$ with EAP, and BIC based / BIC range based Methods. (Results shown in %)

			Select q-vector(s) with the smallest BIC				Select simplest q-vector(s) from a BIC range			
			Results agree with		NOT agree with		Results agree with		NOT agree with	
Sample		Item	Seq.Search		Seq.Search		Seq.Search		Seq.Search	
Size	Conditions	Altered	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
	1	1	85	0	10	5	90	0	10	0
	2	1	92	0	8	0	92	0	8	0
	3	9	94	0	6	0	94	0	6	0
	4	9	92	0	8	0	92	0	8	0
1000	5	9	91	0	9	0	91	0	9	0
	6	15	86	0	14	0	86	0	14	0
	7	15	80	2	18	0	80	0	20	0
	8	15	85	0	15	0	85	0	15	0
	9	15	85	1	13	1	85	0	15	0
	10	1,9,15	78	1	21	0	78	0	22	0
	1	1	71	1	24	4	74	0	25	1
	2	1	72	0	27	1	72	0	28	0
	3	9	74	4	21	1	74	0	26	0
	4	9	69	2	29	0	69	0	31	0
500	5	9	77	1	21	1	77	0	23	0
	6	15	62	1	37	0	62	0	38	0
	7	15	59	1	39	1	59	0	41	0
	8	15	60	2	38	0	60	0	40	0
	9	15	60	1	39	0	60	0	40	0
	10	1,9,15	48	3	44	5	49	0	51	0
	1	1	49	1	44	6	51	0	49	0
	2	1	48	1	46	5	48	0	51	1
	3	9	42	1	52	5	42	0	54	4
	4	9	27	3	66	4	27	0	71	2
250	5	9	37	0	63	0	37	0	59	4
	6	15	34	0	64	2	34	0	62	4
	7	15	39	1	57	3	37	0	56	7
	8	15	38	3	55	4	35	0	56	9
	9	15	34	2	61	3	34	0	59	7
	10	1,9,15	23	1	69	7	24	0	72	4

Table 3.5 Agreement in Validation Results between Sequential Search Method based on with EAP and BIC based / BIC range based Methods. (Results shown in %)

			Select q-vector(s) with the smallest BIC				Select simplest q-vector(s) from a BIC range			
			Results agree with		Results NOT agree with		Results agree with		Results NOT agree with	
Sample		Item	Seq.Search		Seq.Search		Seq.Search		Seq.Search	
Size	Conditions	Altered	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
1000	1	1	90	0	7	3	93	0	7	0
	2	1	95	0	5	0	95	0	5	0
	3	9	96	0	4	0	96	0	4	0
	4	9	97	0	3	0	97	0	3	0
	5	9	99	0	1	0	99	0	1	0
	6	15	82	0	18	0	82	0	18	0
	7	15	72	0	28	0	72	0	28	0
	8	15	77	1	22	0	77	0	23	0
	9	15	78	0	22	0	78	0	22	0
	10	1,9,15	73	1	26	0	73	0	27	0
500	1	1	83	0	12	5	88	0	12	0
	2	1	82	0	18	0	82	0	18	0
	3	9	70	3	27	0	70	0	30	0
	4	9	72	1	27	0	72	0	28	0
	5	9	78	1	21	0	78	0	22	0
	6	15	65	1	34	0	65	0	35	0
	7	15	56	2	41	1	56	0	44	0
	8	15	61	0	39	0	61	0	39	0
	9	15	57	2	41	0	57	0	43	0
	10	1,9,15	53	1	44	2	53	0	47	0
250	1	1	45	1	48	6	46	0	52	2
	2	1	55	2	41	2	56	0	44	0
	3	9	47	1	50	2	47	0	53	0
	4	9	39	2	56	3	39	0	59	2
	5	9	51	0	48	1	52	0	44	4
	6	15	39	2	58	1	37	0	58	5
	7	15	37	3	56	4	35	0	57	8
	8	15	41	1	56	2	41	0	57	2
	9	15	39	1	56	4	37	0	56	7
	10	1,9,15	36	3	55	6	33	0	58	9

3.2.3 EPCE Data

In the previous session, it was shown that the validation methods based on BIC were able to correct a misspecified Q-matrix when response data was simulated from the reduced RUM. In this session, the validation methods will be applied on the ECPE data as described in session 2.2.2. The Q-matrix used to estimate the item parameters under the reduced RUM was introduced in Table 2.4. Henson and Templin (2007) showed that at least 3 items had nonsignificant interactions and small main effect on one attribute, and suggested that Q-matrix may not be correct.

Table 3.6 shows the Q-matrix validation results by the three methods with sequential search method based on $\widetilde{\delta}^{rRUM}$ and classification method EAP. Items with validated q-vectors from the sequential search method different from its originally specified q-vectors are considered to be potentially misspecified. 7 items as listed in the 1st column were identified as having misspecified q-vectors. The rest columns in the table are their corresponding q-vectors in the original Q-matrix, from the sequential search method, from the BIC based sequential search method, and from BIC range based sequential search method. From the table, the validated q-vectors from the last two methods were identical for all 7 items, and they are also identical to the original q-vectors for item G1, G13, G18, G22 and G23, indicating that the original q-vectors for the 5 items are correctly specified. The validation results from the last two methods didn't agree with the original Q-matrix at only 2 of the 7 items, item G4 and G19. This disagreement shows evidence from response data not in favor of the original specifications in the q-vectors of these two items. Originally, item G4 was designed to measure two attributes: knowledge of the morphosyntactic rules and the lexical rules.

Test takers had to master both of the two attributes in order to provide correct answers to this item. However, according to results from the validation methods, the item only measured the attribute "morphosyntactic rules", and test-taker with only knowledge of the morphosyntactic rules had high chance of providing correct responses to the item. Similarly, item G19 measures only the attribute "cohensive rules" instead of the two attributes, "morphosyntactic rules" and "the lexical rules" as it was originally designed to measure.

Table 3.7 shows the Q-matrix validation results by the three methods with sequential search method based on $\widetilde{\delta}^{rRUM}$ and classification method MAP. As compared to results using EAP, more items were selected out as potentially misspecifed by the sequential search method. The two BIC based methods showed only G4 and G19 had incorrect q-vectors, which agreed with results using classification method EAP.

Table 3.8 shows the Q-matrix validation results by the three methods with sequential search method based on $\hat{\delta}^{rRUM}$ and classification method EAP. Only three items were selected as potentially misspecified. G19 were diagnosed as misspecified by the two BIC based methods.

Table 3.6 Q-matrix Validation Result on ECPE Data with Methods based on $\widetilde{\delta}^{rRUM}$ and Classification Method EAP

	Q			Q from Seq.S			Q from BIC			Q from BIC-range		
Item	Mor	Coh	Lex	Mor	Coh	Lex	Mor	Coh	Lex	Mor	Coh	Lex
G1	1	1	0	0	1	0	1	1	0	1	1	0
G4	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**
G13	1	0	1	0	0	1	1	0	1	1	0	1
G18	1	0	1	1	0	0	1	0	1	1	0	1
G19	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**
G22	1	0	1	1	0	0	1	0	1	1	0	1
G23	1	0	1	0	0	1	1	0	1	1	0	1

Table 3.7 Q-matrix Validation Result on ECPE Data with Methods based on $\widetilde{\delta}^{rRUM}$ and Classification Method MAP

	Q			Q from Seq.S			Q from BIC			Q from BIC-range		
Item	Mor	Coh	Lex	Mor	Coh	Lex	Mor	Coh	Lex	Mor	Coh	Lex
G4	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
G13	1	0	1	0	0	1	1	0	1	1	0	1
G16	1	0	0	1	0	1	1	0	0	1	0	0
G18	1	0	1	1	0	0	1	0	1	1	0	1
G19	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**
G22	1	0	1	1	0	0	1	0	1	1	0	1
G23	1	0	1	0	0	1	1	0	1	1	0	1
G27	1	0	0	1	1	0	1	0	0	1	0	0

Table 3.8 Q-matrix Validation Result on ECPE Data with Methods based on $\hat{\delta}^{rRUM}$ and Classification Method EAP

	Q			Q from Seq.S			Q from BIC			Q from BIC-range		
Item	Mor	Coh	Lex	Mor	Coh	Lex	Mor	Coh	Lex	Mor	Coh	Lex
G13	1	0	1	0	0	1	1	0	1	1	0	1
G19	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**
G23	1	0	1	0	0	1	1	0	1	1	0	1

CHAPTER 4

Q-MATRIX VALIDATION FOR THE DINO MODEL

In this previous chapter, methods were developed to validate a Q-matrix for a non-compensatory DCM, the reduced RUM. In this chapter, these validation methods will be extended to a compensatory DCM, especially the DINO model. Note that previously a correct q-vector has been defined as the attribute pattern which maximizes the difference in the probabilities of correct responses between two exclusive groups of respondents, those who have mastered all the required skills and those who have not. The definition is based on the assumption that probability of a correct answer is a monotone increasing function of the number of required attributes that have been mastered, i.e. mastering a required attribute should increase the probability of a correct answer. If an attribute doesn't affect the probability of a correct response, it is probably unnecessary. However, in a DINO model, the probabilities of a correct response are the same for mastering one required attribute and mastering more than one attributes. In this case, the definition previously defined would not work. Modifications have to be made such that the validation methods can be applied. This chapter is structured as follows. The sequential search method for the DINO model will be introduced in Section 4.1, in which the estimation of the DINO model using the EM algorithm will also be discussed. The δ for the DINO model based on which the sequential search method developed for the DINO model will be defined and estimated. A simulation study will be implemented to evaluate the performance of the sequential search method for the DINO model. In Section 4.2, a

two-stage Q-matrix validation method for the DINO model will be discussed, and its performance for the DINO model will be evaluated using a simulation study.

4.1 Sequential Search Method for the DINO Model

For item *i* with q-vector α_l, let δ_{il}^{DINO} be the difference in probabilities of correct responses between respondents who have mastered at least one of the required attributes for this item and those who have none of the required attributes for the item,

$$\delta_{il}^{DINO} = P(Y_{ij} = 1 \mid \varpi_{lj} = 1) - P(Y_{ij} = 1 \mid \varpi_{lj} = 0) \tag{4.1}$$

where,

$$\varpi_{lj} = 1 - \prod_{k=1}^{K} (1 - \alpha_{jk})^{\alpha_{lk}} \tag{4.2}$$

The correct q-vector for item *i* under the DINO model is then defined as the attribute pattern that maximized this difference,

$$q_i^{DINO} = \arg\max_{\alpha_l} \{\delta_{il}^{DINO}\} \tag{4.3}$$

Similarly to the reduced RUM, the sequential search method for the DINO model aims to detect and correct any specifications in a Q-matrix. It is developed based on δ_{il}^{DINO}. The steps of this method are the same as that for the reduced RUM. For each item, it begins with searching through all single-attribute patterns, and selects the one with largest δ_{il}^{DINO}. Then, it searches through two-attribute patterns which contains the selected attribute from the first step, and choose the one with largest δ_{il}^{DINO}. The process repeats until the stopping criterions are met.

One advantage is that, the method is computationally feasible when *K* is large. An exhaustive search algorithm would require computing the δ_{il}^{DINO} for 2^K-*1* times for each item, which is computationally expensive when *K* is large. The computation required by the sequential search method depends on the number of attributes required, and is usually less than that taken by the exhaustive search algorithm.

In the implementation of the sequential search method for the DINO model, δ_{il}^{DINO} is unknown and has to be estimated. Its estimates are computed based on the estimates of model parameters for the DINO model. Section 4.1.1 will discuss the parameter estimation of the DINO model using the EM algorithm. Section 4.1.2 will developed two estimates for δ_{il}^{DINO}. The performance of the sequential search method for the DINO will be examined in Section 4.1.3.

4.1.1 Parameter Estimation for the DINO Model using EM Algorithm

Similar to the DINA model, the DINO model can be estimated using the EM algorithm. From (2.3) the log-likelihood for a random sample of *J* examinees is

$$L = \sum_{j-1}^{J} \ln(\sum_{l=1}^{L} \lambda_l \prod_{i=1}^{I} p_{il}^{Y_{ij}} (1 - p_{il})^{1-Y_{ij}}) \tag{4.4}$$

Under the DINO model, the probability that a respondent with attribute pattern α_l provides a correct response to item *i* is given by

$$p_{il} = (1 - s_i)^{\varpi_{il}} g_i^{1-\varpi_{il}} \tag{4.5}$$

(4.4) has to be maximized subject to $\sum_{l=1}^{L} \lambda_l = 1$, so the MLEs of g_i and s_i are found as the maximums of the unrestrained function

$$\phi = L + \theta * \sum_{l=1}^{L} \lambda_l$$

where θ is an undetermined multiplier. Taking partial derivatives and a few steps of algebra, we have

$$\frac{\partial \phi}{\partial g_i} = \sum_{j=1}^{J} \sum_{l=1}^{L} P(\alpha_l \mid Y_j) \frac{\partial p_{il}}{\partial g_i} * \left(\frac{Y_{ij} - p_{il}}{p_{il}(1 - p_{il})} \right)$$

$$= \sum_{j=1}^{J} \left\{ \sum_{\{l:\varpi_{il}=1\}}^{L} P(\alpha_l \mid Y_j) \frac{\partial p_{il}}{\partial g_i} \left(\frac{Y_{ij} - p_{il}}{p_{il}(1 - p_{il})} \right) + \sum_{\{l:\varpi_{il}=0\}}^{L} P(\alpha_l \mid Y_j) \frac{\partial p_{il}}{\partial g_i} \left(\frac{Y_{ij} - p_{il}}{p_{il}(1 - p_{il})} \right) \right\}$$

where $P(\alpha_j = \alpha_l \mid Y_j)$ is the posterior probability of α_l for respondent *j*, as given in (2.4).

Note that $\frac{\partial p_{il}}{\partial g_i}$=0 when $\varpi_{il} = 0$, $\frac{\partial p_{il}}{\partial g_i}$=1 when $\varpi_{il} = 1$. Therefore,

$$\frac{\partial \phi}{\partial g_i} = \sum_{j=1}^{J} \sum_{\{l:\varpi_{il}=0\}}^{L} P(\alpha_l \mid Y_j) \frac{\partial p_{il}}{\partial g_i} \left(\frac{Y_{ij} - p_{il}}{p_{il}(1 - p_{il})} \right)$$

Let $\frac{\partial \phi}{\partial g_i} = 0$, we get the MLE for g_i

$$\hat{g}_i = \frac{\sum_{j=1}^{J} \sum_{\{l:\varpi_{il}=0\}}^{L} P(\alpha_l \mid Y_j) Y_{ij}}{\sum_{j=1}^{J} \sum_{\{l:\varpi_{il}=0\}}^{L} P(\alpha_l \mid Y_j)} \tag{4.6}$$

Similarly, the MLE for s_i can be obtained as

$$\hat{s}_i = 1 - \frac{\sum_{j=1}^{J} \sum_{\{l:\varpi_{il}=1\}}^{L} P(\alpha_l \mid Y_j) Y_{ij}}{\sum_{j=1}^{J} \sum_{\{l:\varpi_{il}=1\}}^{L} P(\alpha_l \mid Y_j)} \tag{4.7}$$

The MLE for λ_l is given by (2.9).

The EM algorithm is implemented as follows:

1) Choose the initial values for guessing and slipping parameters $\{\hat{g}_i\}$, $\{\hat{s}_i\}$, and the probabilities that a randomly selected respondent has one particular pattern $\{\lambda_l\}$.
2) Use (2.3) and (2.4) to obtain the estimates for posterior probabilities $P(\alpha_j = \alpha_l \mid Y_j)$
3) Substitute these estimates from 2) into (4.6) and (4.7) to obtain the updates of item parameter estimates.
4) Use (2.9) to update estimates of $\{\lambda_l\}$. Repeat the four steps until convergence criterions are met.

4.1.2 Estimation of δ^{DINO}

One way to estimate the quantity is to substitute the true probabilities of correct responses by the estimated probabilities computed from the DINO model,

$$\hat{\delta}_{il'}^{DINO} = (1 - \hat{s}_{il'}^{DINO}) - \hat{g}_{il'}^{DINO} \tag{4.6}$$

with $\hat{g}_{il'}^{DINO}$ and $\hat{s}_{il'}^{DINO}$ given by (4.6),(4.7). The estimate is computed based on the posterior distribution of attribute patterns.

Another estimate is computed based on the observed proportions of respondents in each group who provide correct responses. The group memberships for respondents are calculated based on their estimated attribute patterns from either MAP or EAP. Let $\hat{\alpha}_j$ be the estimated attribute pattern for respondent *j* using either the MAP or EAP classification methods. $\hat{\varpi}_{jl'} = \prod_{k=1}^{K} \hat{\alpha}_{jk}^{\alpha_{l'k}}$ indicates whether respondent *j* have mastered at

least one of the required attributes. A natural estimate of $\delta_{il'}$, which is based on the, is given by

$$\tilde{\delta}_{il'}^{DINO} = \sum_{j=1}^{J} Y_{ij} \cdot \hat{\varpi}_{jl'} \Big/ \sum_{j=1}^{J} \hat{\varpi}_{jl'} - \sum_{j=1}^{J} Y_{ij} \cdot (1 - \hat{\varpi}_{jl'}) \Big/ \sum_{j=1}^{J} (1 - \hat{\varpi}_{jl'}) \tag{4.7}$$

4.1.3 Simulation Study

A simulation study was implemented to explore the performance of the three variations of the sequential search method for the DINO model: the sequential search method for the DINO model based on $\hat{\delta}_{il'}^{DINO}$, the sequential search method for the DINO model based on $\tilde{\delta}_{il'}^{DINO}$ with respondents classified using method MAP, and the sequential search method for the DINO model based on $\tilde{\delta}_{il'}^{DINO}$ with respondents classified using method EAP.

The setting of the simulation study is similar to that of the reduced RUM. Three samples sizes were used (N=1000/500/250). 19 items were designed to measure on four attributes. The Q-matrix used to generate responses is shown in Table 3.1. The guessing and slipping parameters were generated from a uniform distribution (0, 0.2). The correct response probabilities were generated using the DINO model, and the responses were generated using a binomial distribution with those probabilities. . Examinees' attribute patterns were generated from a flat distribution, i.e. examinees are equally likely to be classified into each of the 16 possible attribute patterns.

In addition to the true Q-matrix, 10 misspecified Q-matrices were used to estimate analyze the simulated data sets, especially to estimate item parameters and

respondents' attribute patterns. These 10 matrices were shown in Table 3.2. The item parameters were estimated using an EM algorithm written in R statistical software environment (Core development team, 2011) with convergence criterion of 0.0001. Examinees were then classified into $2^K = 16$ attribute patterns .100 replications were implemented.

The performance of a method is evaluated by the correct recovery rate (CRR), i.e. the percentage of replications when the resulting Q-matrix from a method is identical to the true Q-matrix. High correction rates indicate strong capability of this method in recovering the true Q-matrix. Figure 4.1 shows CRRs for the three sequential search methods for the DINO model on the 10 misspecified Q- matrices. From Figure 4.1, we have the following observations:

1. The high values of CRRs are associated with large sample sizes. With sample size of 1000, the mean CRRs across all simulation conditions are 74%, 73%, 71% for the sequential search method based on $\tilde{\delta}^{DINO}$ with EAP, sequential search method based on $\tilde{\delta}^{DINO}$ with MAP, and the sequential search method based on $\hat{\delta}^{DINO}$, respectively. The mean CRRs drop to 52%, 54.8%, and 55.7% for the three methods respectively with sample size of 500, and 34.7%, 37.2% and 36.2% for sample size of 250. Compared to CRRs of the sequential search methods for the reduced RUM, the sequential search methods do not perform as well as those for the reduced RUM at all sample size levels.

2. Similar to the reduced RUM, the sequential search method based on $\hat{\delta}^{DINO}$ has larger variations across simulation conditions for sample sizes of

1000 and 500. The standard deviations are 4.52%, 7.82% and 8.32% for the three methods, respectively with sample size of 1000, 4.1%, 5.3%, 6.3% for the three methods, respectively with sample size of 500. The sequential search method based on $\tilde{\delta}^{DINO}$ with EAP has the smallest variations among the three methods for all levels of sample size.

3. Similar to the reduced RUM, the CRRs decrease as the number of misspecifications increases. CRRs are higher for the case of overspecifications than other types of misspecifications present in a Q-matrix.

4. Similar to the reduced RUM, the three methods do not differ significantly in their performance of recovering a Q-matrix from its form with misspecificaitons. None of them have satisfactory results in correcting Q-matrices with misspecifications.

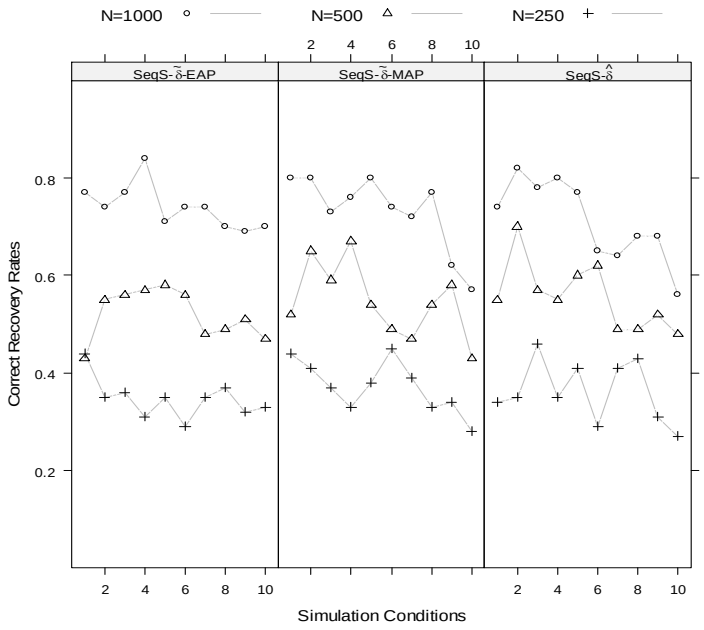

Figure 4.1 Correct Recovery Rates using the Sequential Search Methods for the DINO Model

Figure 4.2 to Figure 4.5 show how the three sequential search methods for the DINO model make change to a Q-matrix with misspecificaitons. Figure 4.2 shows the percentages of replications when the methods didn't made change to the misspecified Q-matrices. Ideally, percentages are expected to be all 0s, because none of the misspecified Q-matrices are identical to the true one, and at least one change has to be made. We can see from this graph, that the sequential search method based on $\hat{\delta}^{DINO}$ made changes to all Q-matrices with misspecifications at all levels of sample size. The other two methods made changes to all Q-matrices with misspecifications with sample size of 1000 and 500, and failed to do so at 1% of replications when sample size is 250.

Figure 4.3 shows the percentages of replications when changes made to a Q-matrix with misspecifications are at the exact items whose q-vectors are misspecified. High values are expected if a method works well in validating a Q-matrix with misspecifications. From Figure 4.3, the sequential search method based on $\hat{\delta}^{DINO}$ has higher percentages than the other two sequential search methods based on $\tilde{\delta}^{DINO}$ for sample sizes of 1000 and 500, indicating estimate $\hat{\delta}^{DINO}$ works better than $\tilde{\delta}^{DINO}$ in detecting the items with wrong q-vectors for large sample sizes. Large percentages are also associated with large sample size and large number of misspecifications. The reason is that, items with many misspecifications are more likely to be detected than those with few misspecifiications.

Figure 4.4 shows the proportions of times when changes made to a Q-matrix with misspecifications are not only at the misspecified items but also at items with correct q-

vectors and should not be changed. As sample size gets smaller, these proportions get larger.

Figure 4.5 shows the percentages of replications when the methods made change to a subset of misspecified items, or to none of them. The percentages are expected to be all 0s if the method works well. The percentages are all 0s except for the sequential search method based on $\tilde{\delta}^{DINO}$ and MAP and the sequential search method based on $\hat{\delta}^{DINO}$ for sample size of 250 (1%). The above figures show that all three sequential search methods do well in identifying items with misspecifications. The sequential search method based on $\hat{\delta}^{DINO}$ is slightly better than the methods based on $\tilde{\delta}^{DINO}$.

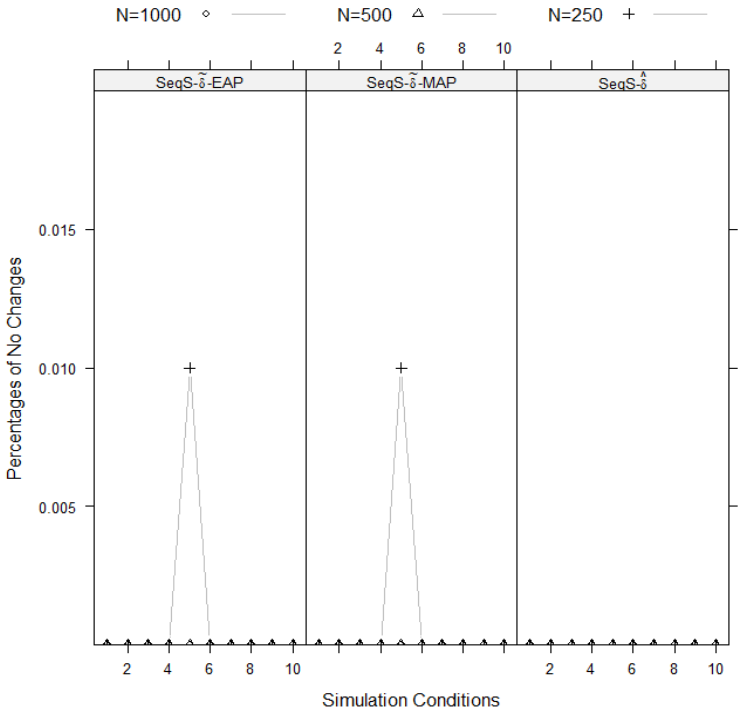

Figure 4.2 Percentages of Replications when the Sequential Search Methods Made No Changes to the Q-Matrices with Misspecifications

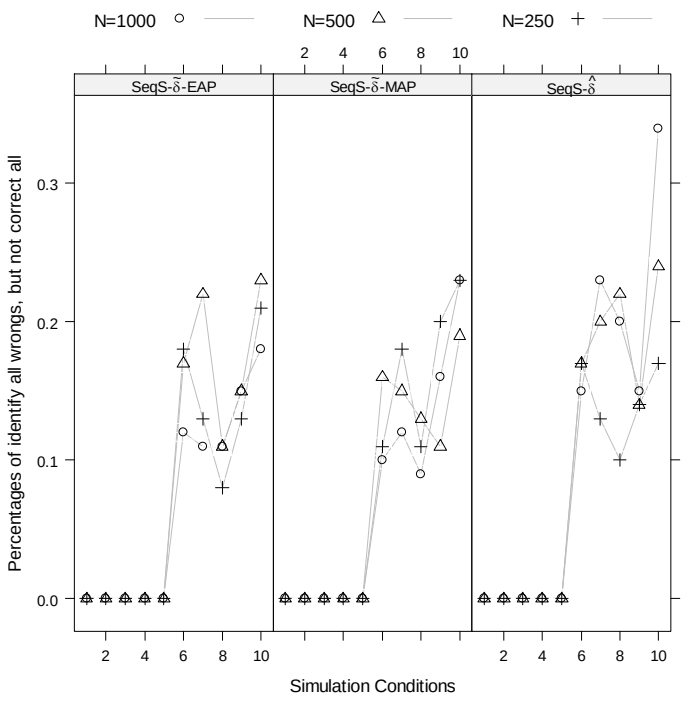

Figure 4.3 Percentages of Replications when the Sequential Search Methods Made Changes at the Exact Items with Misspecifications, but the changes were at least partially incorrect.

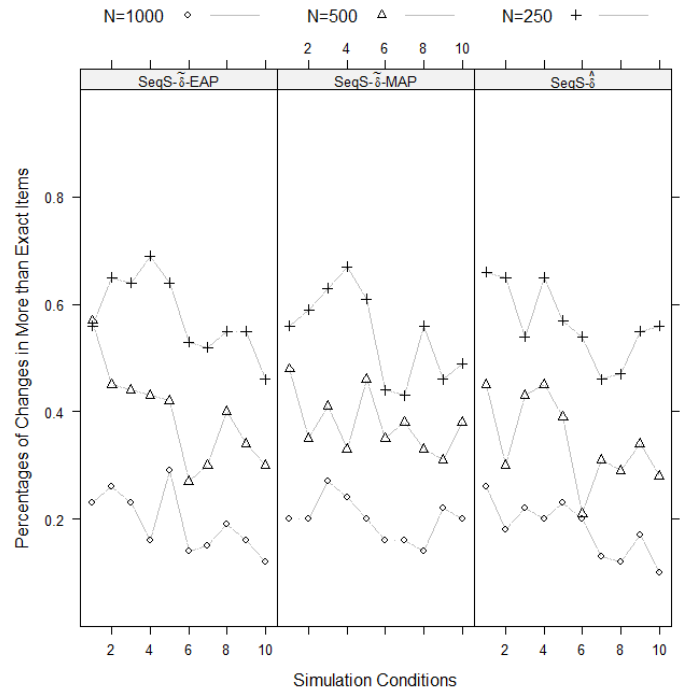

Figure 4.4 Percentages of Replications when the Sequential Search Methods Made Changes to More Items than Those with Misspecifications

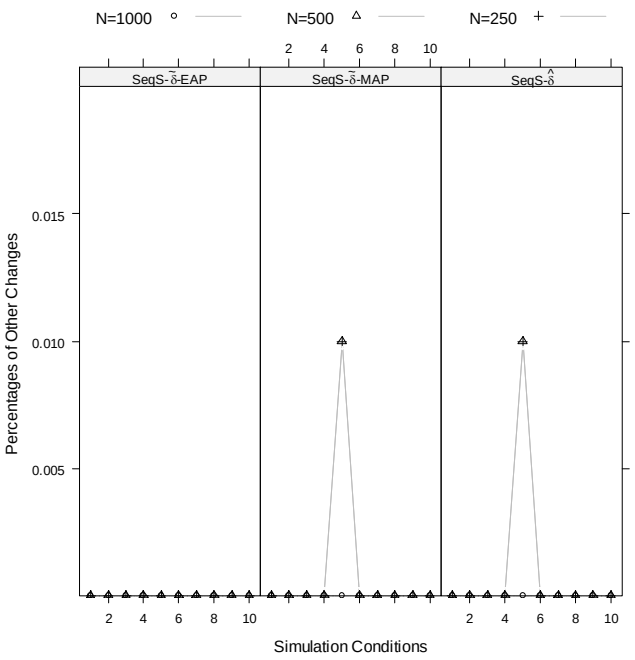

Figure 4.5 Percentages of Replications when the Sequential Search Methods Made Changes to More Items than Those with Misspecifications

4.2. Two-Stage Q-matrix Validation Methods for the DINO Model

The two-stage Q-matrix validation methods are the same for the DINO model as those for the reduced RUM. In the first stage, items with misspecifications are identified using the sequential search methods. The correct q-vectors for those items are then selected from all possible q-vectors using the BIC criterion or the BIC-range criterion. Simulation Study was conduct to compare the performance of three Q-matrix validation methods: the sequential search method, the BIC based sequential search method, and the BIC range based sequential search method. Three variations of the sequential search methods were used. The set up of the simulation study was the same as in Section 4.1.3.

Figure 4.6 shows correct recovery rates for Q-Matrix validation methods for DINO model based on $\tilde{\delta}^{DINO}$ with classification method EAP. Figure 4.7 shows correct

recovery rates for Q-Matrix validation methods for DINO model based on $\tilde{\delta}^{DINO}$ with classification method MAP. Figure 4.8 shows correct recovery rates for Q-Matrix validation methods for DINO model based on with $\hat{\delta}^{DINO}$.The two BIC based methods worked perfectly for the DINO model for sample sizes of 1000 and 500 with 100% CRRs under all simulation conditions. When $\hat{\delta}^{DINO}$ is used, the average CRR is 99.6% for sample size of 250 for the BIC based sequential search method and 97% for the BIC range based sequential search method. Method based on $\tilde{\delta}^{DINO}$ and MAP has similar results. The method based on $\tilde{\delta}^{DINO}$ and EAP also has high average CRR for the BIC based sequential search method (98.9%), but relatively low average CRR (87.1%) for the BIC range based sequential search method. Overall, both BIC based methods recover the true Q-matrix from the matrices with misspecification under all conditions for the DINO model at all sample size levels. The sequential search method failed to recover the true Q-matrix under most conditions and sample sizes.

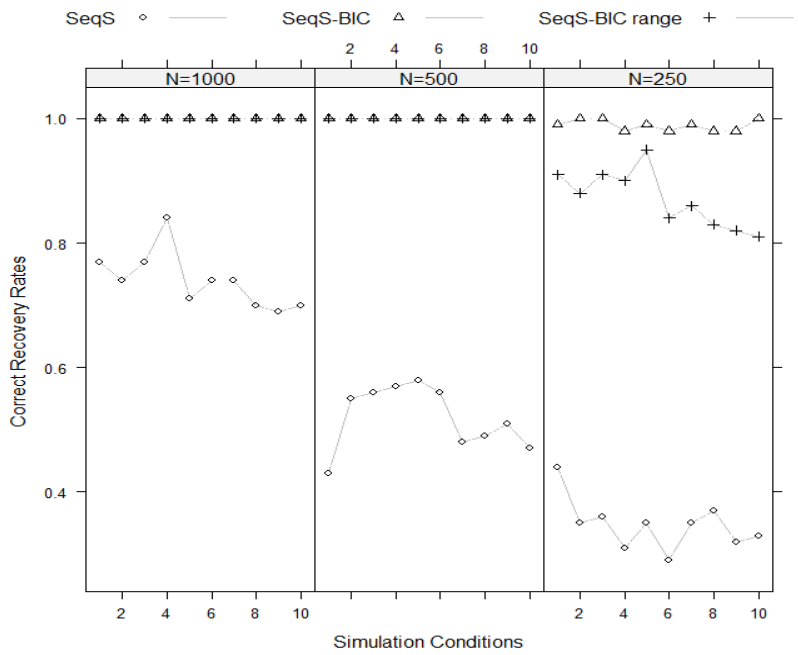

Figure 4.6 Correct Recovery Rates for Q-Matrix Validation Methods for DINO Model based on $\tilde{\delta}^{DINO}$ with Classification Method EAP

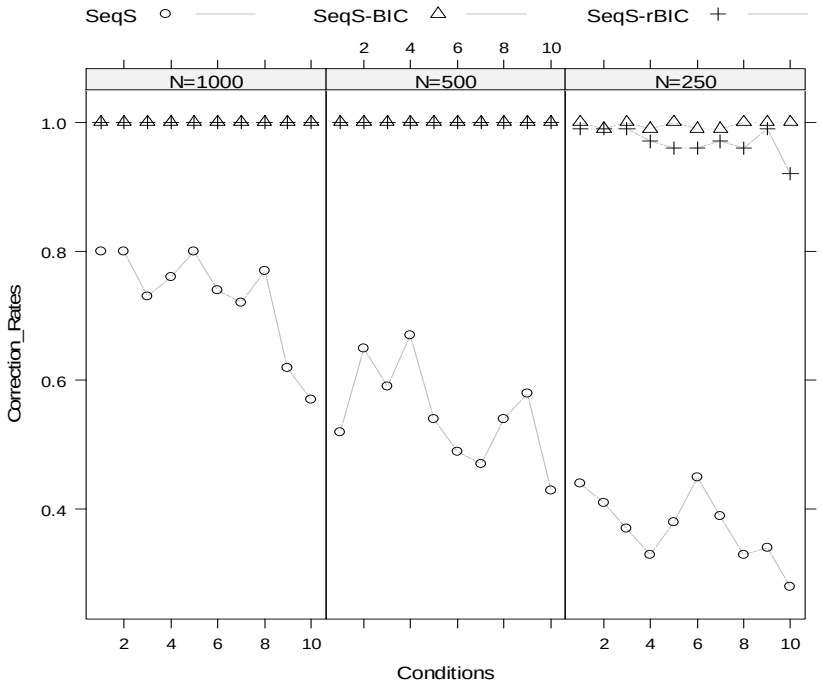

Figure 4.7 Correct Recovery Rates for Q-Matrix Validation Methods for DINO Model based on $\tilde{\delta}^{DINO}$ with Classification Method MAP

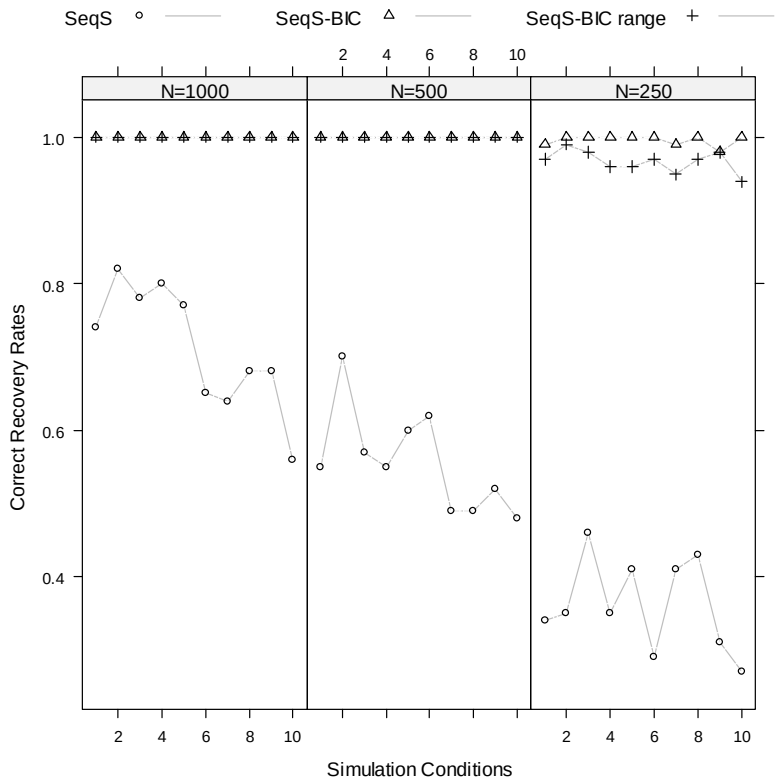

Figure 4.8 Correct Recovery Rates for Q-Matrix Validation Methods for DINO Model based on $\hat{\delta}^{DINO}$

As shown in Table 4.2, validation results from the sequential search method agreed with the two BIC based methods for most of replications when sample sizes are large. There was less agreements between the sequential search method and the two BIC methods as sample size was small and the number of misspecifications got large. The extent to which the three methods agreed was affected by the CRRs of the sequential search method. The two other variations of the sequential search method have similar results on how validation results from the sequential search method agrees with the BIC based method and the BIC range based method.

Table 4.1 Comparisons of Validation Methods for the DINO Model based on $\tilde{\delta}^{DINO}$ with Classification Method MAP

			Select q-vector(s) with the smallest BIC				Select simplest q-vector(s) from a BIC range			
			Results agree with		NOT agree with		Results agree with		NOT agree with	
Sample		Item	Seq.Search		Seq.Search		Seq.Search		Seq.Search	
Size	Conditions	Altered	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
	1	1	80	0	20	0	80	0	20	0
	2	1	80	0	20	0	80	0	20	0
	3	9	73	0	27	0	73	0	27	0
	4	9	76	0	24	0	76	0	24	0
1000	5	9	80	0	20	0	80	0	20	0
	6	15	74	0	26	0	74	0	26	0
	7	15	72	0	28	0	72	0	28	0
	8	15	77	0	23	0	77	0	23	0
	9	15	62	0	38	0	62	0	38	0
	10	1,9,15	57	0	43	0	57	0	43	0
	1	1	52	0	48	0	52	0	48	0
	2	1	65	0	35	0	65	0	35	0
	3	9	59	0	41	0	59	0	41	0
	4	9	67	0	33	0	67	0	33	0
500	5	9	54	0	46	0	54	0	46	0
	6	15	49	0	51	0	49	0	51	0
	7	15	47	0	53	0	47	0	53	0
	8	15	54	0	46	0	54	0	46	0
	9	15	58	0	42	0	58	0	42	0
	10	1,9,15	43	0	57	0	43	0	57	0
	1	1	44	0	56	0	44	0	55	1
	2	1	41	0	58	1	41	0	58	1
	3	9	37	0	63	0	37	0	62	1
	4	9	33	0	66	1	33	0	64	3
250	5	9	38	0	62	0	38	0	58	4
	6	15	45	0	54	1	44	0	52	4
	7	15	39	0	60	1	39	0	58	3
	8	15	33	0	67	0	32	0	64	4
	9	15	34	0	66	0	34	0	65	1
	10	1,9,15	28	0	72	0	28	0	64	8

In this chapter, parallel methods were developed to validate a Q-matrix with misspecifications for the DINO model. Overall, the sequential search methods for the DINO have low correction rates in recovering those q-vectors, especially for small sample and large number of misspecifications, but they are able to detect misspecifications presented in a Q-matrix under all simulation conditions and at all sample size levels. Besides, they tend to diagnosis items with correct q-vectors as wrong items as sample size gets small. Having a higher the percentages of making changes at the exact items with misspecifications, the sequential search method based on $\hat{\delta}^{DINO}$ is better in identifying misspecified items than the other two variations of sequential search methods. The sequential search method can be served as a tool to provide preliminary check of a Q-matrix, but it cannot be used independently to validate a Q-matrix. Its results have to be combined with other two methods in order to validate a misspecified Q-matrix. On the other hand, the two BIC based methods for the DINO model recover a true Q-matrix perfectly for almost all combinations of sample sizes and simulation conditions. Hence, when response data was modeled by the DINO model, either one of the BIC methods can be used to validate a Q-matrix.

CHAPTER 5

OTHER ISSUES CONCERNING Q-MATRIX VALIDATION

As shown in the previous chapters, both the BIC- based sequential search method and the BIC range based sequential search method are able to recover a correct Q-matrix from a misspecified one for the DINO model and the reduced RUM. The two methods work well under the assumption that items in an assessment were modeled by the same class of DCMs, either the DINO model or the reduced RUM. However, there are some cases in which items were from a combination of more than one DCM, and we don't know which item should be modeled by which DCM. In this chapter, the performance of the two Q-matrix validation methods will be investigated for the assessment with items from a combination of several DCMs. Section 5.1 will describe the simulation design for the purpose of the study. Section 5.2 will then discuss the simulation results and their implications.

From the previous results, there are some factors affecting the correction rate of validation results, such as sample size *n*, number of items with misspecifed q-vectors, *nwitem*, and number of misspecified elements, *nwelm*, and the type of misspecificaitons. Studies using large sample sizes tend to have higher correction rates than those with small sample sizes because large response data sets contain more information about the true Q-matrix. The number of items with misspecifed q-vectors, *nwitem*, and the number of misspecified elements, *nwelm*, are measures of the extent to which a Q-matrix is

misspecified. *nwitem* is always smaller than *nwelm*. Previous studies showed that when *n=500,* small values in the two quantities had little impact on the performance of both validation methods. However, high values in these two measures imply a poorly defined Q-matrix, and consequentially result in biased model estimation and poor performance of validation methods. A question rising up from the two measures is: how many misspecifications should a Q-matrix have at most to ensure good performance of the validation methods? To answer this question, a study was designed in section 5.3 to evaluate the impact of these two measures on the performance of validation methods. Results are discussed in section 5.4.

5.1 Simulation Study I

Note that the validation methods were developed on different delta statistics for the compensatory DCMs and noncompensatory DCMs. For simplicity, we only consider the case when items were from a combination of noncompensatory DCMs, especially, the DINA model and the reduced RUM. Responses for 500 respondents were generated for 19 items whose Q-matrix was shown in Table 3.1. The items were a combination of both the DINA model and the reduced RUM. Three values of proportions of items from the DINA model were considered: 20%, 50%, and 80%. The 10^{th} matrix in Table 3.2 was used to estimate item parameters and respondents' attribute patterns. There were three items (1^{st}, 9^{th}, 15^{th}) having incorrect q-vectors in this matrix. Two values of proportions that misspecified items were from the DINA model were considered: 100% and 0%, that is, the misspecified items either came from the DINA model or the reduced RUM. The simulation conditions were shown as in Table 3.3. The simulation condition "DINA4(3)" represented that the case when responses were simulated on 19 items, 4 out of which

were from the DINA model. The number in parenthesis is the number of misspecified items from the DINA model. 100 replications were implemented.

Table 5.1 Simulation Conditions for Q-matrix with Items from a Mixture of DCMs

		Proportion of DINA items		
		20%	50%	80%
Prop. Of	100%	DINA4(3)	DINA10(3)	DINA15(3)
incorrect DINA items	0	DINA4(0)	DINA9(0)	DINA15(0)

The two validation methods, BIC based sequential search method and the BIC range based sequential search method, were used to recover the true Q-matrix. Since we don't know which items are from the DINA model and which are from the reduced RUM, The validation methods for each of the two models were used to validate all items in the matrix. The estimate $\hat{\delta}^{rRUM}$ was used for sequential search method for the reduced RUM.

5.2. Study I Results

The accuracy of validation was evaluated by the correction rates, which was the percentage of replications when the Q-matrices returned by validation methods were identical to the true one. The BIC based method and the BIC range based method were developed on the results that the sequential search method can identify the items with wrong q-vectors for DCMs. Table 5.2 shows that it can also pick out those items when items are from a mixture DCMs. The 2nd column in the table indicates that the sequential search method made changes to the wrong matrix for all replications at all simulation conditions. The last column with all 0s implies that all changes have been made at the

items with q-vectors. The second last column shows that the sequential search method also changed those items with correct q-vectors at quite a few replications.

Table 5.3 shows the correction rates for all six simulation conditions for validations methods were based on the DINA model and the reduced RUM, respectively. When the misspecified items were from the DINA model as in condition DINA4(3), DINA9(3) and DINA15(3), validation methods for the DINA model were able to recover the true Q-matrix with the correction rates all 100%s for assorted values of proportions of DINA items. The correction rates were slightly lower than 100% when validation methods for the reduced RUM were used to validate the DINA misspecified items. The correction rates for reduced RUM validation methods were at the lowest when the BIC based sequential search method was used and the proportion of DINA items in the matrix was large (83% for Seq.s.+BIC and DINA15(3)).

When the misspecified items were from the reduced RUM model as in condition DINA4(0), DINA9(0) and DINA15(0), three validation methods works well in recovering the true Q-matrix with the high correction rates, the BIC based sequential search method for the reduced RUM, the BIC range based sequential search method for the reduced RUM, and the BIC based sequential search method for the DINA. However, the BIC based sequential search method for the DINA didn't work well in validating matrix with misspecified items from the reduced RUM.

Under all conditions, the BIC range based sequential search method for the reduced RUM worked the best, and the BIC based sequential search method for the DINA work almost as well as the BIC range based sequential search method for the reduced RUM. The performance of both methods are invariant to the source of the

misspecified items, i.e., they worked well in validating Q-matrix with misspecified items from both the DINA and the reduced RUM. The performance of the other two methods, the BIC range based sequential search method for the DINA and the BIC based sequential search method for the reduced RUM, were affected by the source of misspecified items. They worked well only when the misspecified items were from the same model. In practice, we don't know what model items with incorrect q-vectors are from. Thus, it is suggested that the two methods, the BIC range based sequential search method for the reduced RUM and the BIC based sequential search method for the DINA, are used to validate a Q-matrix with a combination of item types.

Table 5.2 How Sequential Search Method Made Changes to the Original Matrix

		Sequential Search Method				
		Didn't	Changes made at Wrong items			
		Made	Only wrong items		More than	Other than
	Condition	Changes	Correct	Incorrect	wrong items	wrong Items
DINA	DINA4(3)	0	30	47	23	0
	DINA10(3)	0	27	48	25	0
	DINA15(3)	0	22	41	37	0
	DINA15(0)	0	60	20	20	0
	DINA9(0)	0	29	43	28	0
	DINA4(0)	0	38	53	9	0
reduced RUM	DINA4(3)	0	53	21	26	0
	DINA10(3)	0	40	27	33	0
	DINA15(3)	0	32	18	50	0
	DINA15(0)	0	53	13	34	0
	DINA9(0)	0	48	21	31	0
	DINA4(0)	0	64	24	12	0

Table 5.3 Recovery Rates for Q-matrix with Items from a Mixture of DCMs

	DINA		reduced RUM	
Conditions	Seq.s.+BIC	Seq.s.+rBIC	Seq.s.+BIC	Seq.s.+rBIC
DINA4(3)	1	1	0.91	0.96
DINA10(3)	1	1	0.93	0.97
DINA15(3)	1	1	0.83	0.93
DINA4(0)	0.97	0.83	0.96	1
DINA9(0)	0.9	0.81	0.96	1
DINA15(0)	0.89	0.82	0.98	0.99

5.3 Simulation Study II

To generate a misspecified Q-matrix from the true Q-matrix, we randomly selected *nwitem=3,6,9,12* items to alter, which account for 16%, 32%, 47%, 63% of the total items, respectively. The reason we started with 3 is that, previous results showed that *nwitem* has little impact on performance of the validation methods with values under 3. We consider three types of misspecifications as following: 1) overspecified, i.e., an element that should be coded as 0s are wrongly coded as 1; 2) underspecified, i.e., an element which should be coded as 1s are wrongly coded as 0; 3) a combination of the overspecified and underspecified, i.e. an unnecessary attribute is coded as 1 and a required attribute is coded as 0. To control the number of misspecifications, the same type of misspecification was applied to all items that are randomly selected as wrong items. So *nwelm* is constant within each combination of *nwitem* values and types of misspecifications.

As shown in Table 5.4, we have 4 *nwitem* values * 3 misspecification types =12 simulation conditions. The percentages of misspecified elements in a Q-matrix range from 4% to 32%. The Q-matrix having 3 items with type 1 misspecification has the smallest amount of wrong elements (4%). The Q-matrix having 12 elements with type 3

misspecification has the largest number of wrong elements (32%). Type 1 and type 2 has the same percentages of misspecificaitons for a fixed *nwitem* value. 100 replications were implemented for each condition.

Table 5.4 Simulation Conditions for Q-Matrix with Various Misspecifications

	Misspecification Type		
# of Items Altered	1Add	1Delete	1Add & 1Delete
3	4%	4%	8%
6	8%	8%	16%
9	12%	12%	24%
12	16%	16%	32%

For each replication, response data for 500 respondents were generated Bernoulli distribution with probabilities of positive responses modeled by the reduced RUM using item parameters and the true Q-matrix as shown in Table. From previous chapter, studies of sample size 500 had similar correction rates with those of sample size 1000, but had much higher correction rates than studies of sample size 250, indicating 500 might be a good choice of sample sizes for future studies. Examinees' attribute patterns were generated from flat distribution. A misspecified Q-matrix was generated and used in estimating item parameters. EM algorithm was used in estimating model parameters. Respondents were then classified using classification method EAP. The process of Q-matrix validation began with identifying wrong items using the sequential search method for the reduced RUM. The two validation methods, the BIC- based sequential search method and the BIC range based sequential search method, were then used to find the correct q-vectors for those items identified in the previous step.

5.4. Study II Results

We were not able to generate a misspecified Q-matrix from the true Q-matrix under the condition *nwitem* = 12 with type 2 misspecification, because the true Q-matrix only have 11 items with at least two attributes. So, we didn't include this condition in the study. Further, we found that most replications for the two conditions, *nwitem* = 12 with type 3 misspecification and *nwitem* = 9 with type 3 misspecification, had convergence problem due to the large amount of misspecifications in their Q-matrices. So, the analysis of results was based on the rest 9 simulation conditions.

The performance of the validation methods is affected by the accuracy of respondents' classification. A plot of correct classification rates (CCR) by simulation conditions is presented in Figure5.1.

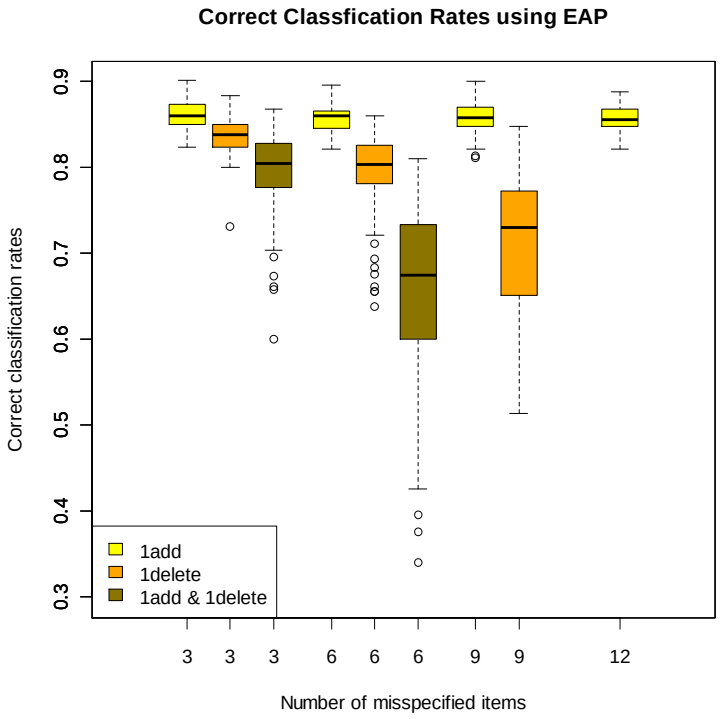

Figure 5.1 Boxplots of Correct Classification Rates under Simulation Conditions

For each value of *nwitem*, type 1 misspecification has the highest mean CCR and the smallest standard deviation in CCRs among all types. We can see that the number of additions to a Q-matrix didn't affect CCR, because the means and standard deviations of CCRs for conditions with type 1 misspecification didn't change much across different values of *nwitem*. The mean CCRs with type 1 misspecifications were 0.8602, 0.8575, 0.8576, 0.8568 for *nwitem=3, 6,9,12* respectively. Comparing to the CCR (0.86) with model estimated using the true Q-matrix, we see that overspecification in a Q-matrix with one extra element in an item didn't have significant impact on the CCR. Though the percentages of wrong elements in a Q-matrix were the same for type 1 and type 2 misspecifications for a fixed number of misspecified items, these two types have differential impact on CCR. For each value of *nwitem*, the mean CCR of type 2 was lower than that of type 1, and its standard deviation was larger. The number of deletions in a Q-matrix also affected CCRs, because the mean CCR of type 2 decreases as *nwitem* increases. The mean CCRs with type 2 misspecifications were 0.8374, 0.7953, 0.71466, for *nwitem=3, 6, 9* respectively. The type 3 misspecification, which is a combination of the first two, randomly added an unnecessary attribute and deleted a required attribute. With this type of misspecification present in a Q-matrix, the mean CCRs were low and standard deviations of CCRs were large.

One question of primary interest is, how well does the sequential search method identify the misspecified items? Table 5.5 contains how the sequential search method made change to the Q-matrix with misspecifications in different settings. The first column lists three types of misspecifications that could happen to q-vectors. The second column lists the number of items with misspecifications for each type. The rest columns

contain the percentages of replications where 1) the sequential search method made no change to the matrix , as shown in the 3rd column; 2) the method change the misspecified q-vectors to correct ones, as shown in the 4rd column; 3) the method made change at the items with mispsecified q-vectors, but didn't correct them all, as shown in the 5th column; 4) the method not only made change at the items with misspecified q-vectors, but also other items with correct q-vectors (as in the 6th column); 5) the method made change to a subset of the items with misspecifications (as in the 7th column); 6)the method made changes at items other than those with misspecifications (as in the 8th column). We had the following observations from this table:

Table 5.5 How Sequential Search Method for the reduced RUM Made Changes to the Original Matrix (Results Shown in %)

			Sequential Search Method				
Misspecification Type	# of Misped Items	Didn't Change	Only at Misped items Correct	Only at Misped items Incorrect	More than Misped Items	Less than Misped Items	Other than Misped Items
1Add	3	0	68	0	24	5	3
	6	0	64	0	17	17	2
	9	0	75	0	8	16	1
	12	0	54	1	8	32	5
1Delete	3	0	63	22	15	0	0
	6	0	56	29	9	1	5
	9	0	36	27	3	27	7
1Add &1Delete	3	0	55	25	20	0	0
	6	0	19	45	26	0	10

1) The method made changes to the original Q-matrices under all simulation conditions.
2) The method identified the misspecified items in an average of 85% of the replications across all conditions. It made changes at the wrong items for all replications when there were 3 misspecified items present in the Q-matrices with

type 2 and type 3 misspecifications. However, under the two conditions, 12 items with type 1 misspecification and 9 items with type 2 misspecifications, the percentages dropped to 63% and 66%, respectively.

3) The sequential search method has a higher average rate in correcting wrong items for type 1 misspecifications (65.25%) than type 2 misspecification (51.67%) and type 3 misspcification (37%).
4) The sequential search method tends to fail to identify all items when type 1 misspecification happened. With type 1 misspecification, the percentages of replications in which the sequential search method were not able to change all misspecified items were 5%, 17%, 16%, 32% for *nwitem=3, 6,9,12* respectively. Failing to change all wrong items didn't happen to conditions having small *nwitem* values with type 2 and 3 misspecifictions.

Table 5.6 shows that correction rates for all simulation conditions. The BIC range based sequential search method has the overall highest correction rates among the three methods when the misspecifications are of type 1 and type 3. When misspecifications are of type 2, the BIC based sequential search method worked the best. The correction rates for both methods dropped under 65% when the number of misspecified items increased to 9 or more.

Table 5.6 Recovery Rates for Q-matrix with Various Misspecifications

Misspecification Type	# of Wrong Items	Correction Rates		
		Seq.Search	Seq.S+BIC	Seq.S+rBIC
1Add	3	68%	99%	100%
	6	64%	98%	100%
	9	75%	100%	100%
	12	54%	96%	100%
1Delete	3	63%	96%	100%
	6	56%	92%	86%
	9	36%	63%	54%
1Add &1Delete	3	55%	80%	96%
	6	19%	44%	61%

CHAPTER 6

DISCUSSION

Q-matrix validation is an important part of model fit for diagnostic classification models. The current literature on this topic focuses on validation methods for the DINA model, which include the sequential search method (de La Torre 2008) searching for correct q-vectors based on statistic δ, and the estimation of the Q-matrix (Liu, Xu, and Ying 2011). The current validation methods were developed based on the unique feature of the DINA model that there are only two possible values of the probability of answering an item correctly. However, other DCMs don't have this feature. Following the sequential search method, we developed a two-stage method to correct misspecified Q-matrices for a wider class of DCMs. In our study, versions of the two-stage method were developed especially for the reduced RUM, and for the compensatory DINO model. It can also be easily implemented for other DCMs. The two-stage method incorporating the idea of sequential searching based on δ and the Bayesian model selection methods were shown to have good performance for validating the Q-matrix for both models using simulated data sets. Its performance for the reduced RUM was also shown on a real data set.

Item parameter estimation is a crucial step in the process of Q-matrix validation. The estimation for the reduced RUM is challenging because of the various number of parameters across items and the more complicated relationship with the underlying latent

classes. Currently estimation for the reduced RUM is commonly implemented through MCMC, which is computationally expensive. In the study, the EM algorithm was shown to have good performance in item parameter estimation for the reduced RUM, with significant time savings. The EM algorithm also provides the posterior distribution of attributes patterns based on which the statistic δ is estimated.

We also explored the performance of the two-stage validation method under two additional cases. In one case, items were generated from a combination of DCMs. Simulation studies in Chapter 5 showed that the BIC range based sequential search method performed well when the items were from a combination of the reduced RUM and the DINA model. We considered the effects of two main factors on the performance of the validation method, the types of misspecifications and the number of misspecifications. A simulation study was conduct to evaluate the effectiveness of the two-stage validation method under various combinations of different levels of the two factors.

Though the statistic δ on which the sequential search method is based works well for the DINA model, it is seemingly not optimal for other DCMs as illustrated by its low correct recovery rates on DCMs besides the DINA model. This is unsurprising, de la Torre (2008) defined it specifically for the DINA model, and we defined a new version of it when we implemented the sequential search method for the noncompensatory DINO model. In future research, we should explore other options for the statistic that has overall high correct recover rates and has uniform definition for a wider class of DCMs.

Note that the misspecifications in the Q-matrix lead to less accurate model estimation than that using a correct Q-matrix. The performance of the two-stage

validation method largely depends on the input Q-matrix which might be severely misspecified and lead to inaccurate model estimation. Another area to explore is in the development of validation methods that are model-resistant, whose performance doesn't depend on model estimation using the Q-matrix with misspecificaitons.

We explored the validation method under the case when the items are from a combination of both the reduced RUM and the DINA model. Note that both models are noncompensatory DCMs. In the future, we might also want to explore validation method under the case when items are a combination of both noncompensatory DCMs and compensatory DCMs. Most of our conclusions are based on results from simulation studies. In future, results should be validated from the theoretical aspect.

In our previous discussion, a Q-matrix is developed after attributes are well defined, that is, attributes are fixed. However, there is also possibility that we cannot decide whether or not we should include a specific attribute into the Q-matrix. The proposed Q-matrix validation method provides a way of treating an undecided attribute. When there is only one undecided attribute, we could construct a new Q-matrix by adding an extra column of zeros to the existing Q-matrix. Item parameters and two-stage validation could then be implemented with the newly constructed Q-matrix. By comparing the proportion of zeros in the column corresponding to the undecided attribute in the validated Q-matrix to the proportion of ones in that column, we could obtain evidence whether the extra attribute should be included. The same procedure can be done with an extra column of ones added to form the new Q-matrix to determine if the data suggests an extra attribute should be added into the existing Q-matrix. For further work, simulation studies could be conducted to see effectiveness of the two-stage

validation methods in determining undecided attribute. The size of respondents and that of item sets at which certain level of accuracy in determined an attribute is achieved are also of interest.

REFERENCES

Bartholomew, D., & Knott, M. (1999). *Latent Variable Models and Factor Analysis*. Second edition. London: Arnold.

Close, N.C. (2012): An Exploratory Technique for Finding the Q-matrix in Cognitive Diagnostic Assessment: Combining Theory with Data. Unpublished doctoral dissertation, University of Minnesota, Minneapolis, MN.

Dempster, A. P., Laird, N.M, & Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society.39,* 1-38.

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

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

de la Torre, J. (2011). The Generalized DINA Model Framework. *Psychometrica,76,*179-199.

de la Torre, J. (2009). DINA Model and Parameter Estimation : A Didactic. *Journal of Educational and Behavioral Statistics, 34,*115-130.

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

Dempster, A. P., Laird, N.M, & Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, 39,* 1-38.

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

Gorin, J. S. (2009). Diagnostic classification model: Are they necessary? Commentary on Rupp and Templin (2008). *Measurement: Interdisciplinary Research and Perspectives 7,* 30–33.

Haertel, E. H. (1989). Using restricted latent class models to map the skill structure of achievement items. Journal of Educational Measurement, *26(4),* 301-323.

Hartz, S.M. (2002). A Bayesian framework *for the unified model for assessing cognitive abilities: Blending theory with practicality.* Unpubilished doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL.

Henson, R. A., & Templin, J. L. (2007, April). *Large-scale language assessment using cognitive diagnosis models.* Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago,IL.

Henson, R.A., & Templin, J.L. (2009). Implications of Q-matrix misspecification in cognitive diagnosis. Manuscript submitted for publication.

Henson, R., Templin, J.,& Willse, J.(2009). Defining a family of cognitive diagnosis models using log linear models with latent variables. *Psychometrika,74,*191-210.

Huebner, A., & Wang, C. (2011). A note on comparing examinee classification methods for cognitive diagnosis models. *Educational and Psychological Measurement, 71(2),* 407-419.

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

Kunina-Habenicht, O., Rupp, A., Wilhelm, O. (2011). Detection of Model Misspecification and its Impact on Parameter Estimation Accuracy in Diagnostic Classification Models. Paper presented at the annual meeting of the National Council on Measurement in Education, New Orleans, LA.

Liu,Y., Douglas, J. A., & Henson,R. (2009). Testing Person Fit in Cognitive Diagnosis. *Applied Psychological Measurement*. *33*,579-598.

Muthén, L. K., & Muthén, B. O. (1998-2011). Mplus User's Guide. Sixth Edition. Los Angeles, CA: Muthén & Muthén.

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

Rupp, A., Templin,J. & Henson,R. (2010). Diagnostic Measurement: Theory, Methods, and Applications. The Guilford Press, New York, London.

Robert, P.C.,& Casella G.(2004). Monte Carlo Statistical Methods (Second Edition). New York: Spinger.

R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for StatisticalComputing, Vienna, Austria. ISBN 3-900051-07-0, URL: www.R-project.org.

Roussos, L., diBello, L. V., Stout, W., Hartz, S., Henson, R. A., & Templin, J. H. (2007). The fusion model skills diagnosis system. In J. P. Leighton, & Gierl, M. J. (Ed.), *Cognitively diagnostic assessment for education: Theory and practice* (pp. 275-318). Thousand Oaks, CA: Sage.

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

Templin, J. (2006). CDM: cognitive diagnosis modelling with mplus user guide. Unpublished manuscript.

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

von Davier, M. (2008). A general diagnostic model applied to language testing data. *British Journal of mathematical and Statistical Psychology*. *61*,287-30.