

**Growth Curve Cognitive Diagnosis Models for Longitudinal Assessment**

by

**Seung Yeon Lee**

A dissertation submitted in partial satisfaction of the  
requirements for the degree of  
Doctor of Philosophy

in

Education

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Sophia Rabe-Hesketh, Chair  
Assistant Professor Zachary Pardos  
Professor Nicholas Jewell

Summer 2017

# Growth Curve Cognitive Diagnosis Models for Longitudinal Assessment

Copyright 2017  
by  
Seung Yeon Lee

## Abstract

Growth Curve Cognitive Diagnosis Models for Longitudinal Assessment

by

Seung Yeon Lee

Doctor of Philosophy in Education

University of California, Berkeley

Professor Sophia Rabe-Hesketh, Chair

This dissertation proposes longitudinal growth curve cognitive diagnosis models (GC-CDM) to incorporate learning over time into the cognitive assessment framework. The approach was motivated by higher-order latent trait models (de la Torre & Douglas, 2004), which define a higher-order continuous latent trait that affects all the latent skills. The higher-order latent trait can be viewed as the more broadly defined general ability; and the skills can be viewed as the specific knowledge arising from the higher-order latent trait. GC-CDMs trace changes in the higher-order latent traits over time by using latent growth curve model with respondent-specific random intercept and random slope of time, and simultaneously trace students' skill mastery through the CDM measurement model.

GC-CDMs are estimated using marginal maximum likelihood (MML) estimation in Mplus. Relevant issues for estimating GC-CDMs are addressed, e.g., the high-dimensional computation problem, model specification for the relationship between the higher-order latent trait and the multiple skills, and model identification. In simulation studies, we use the DINA measurement model, and examine parameter recovery of the GC-DINA model under differing conditions. Particularly, the effects of the design of the Q-matrix, the number of respondents and the number of time points are discussed. Overall, MML estimation in Mplus shows good parameter recovery; especially, the average growth, which is the parameter of most interest, is well estimated in all conditions. We also illustrate the application of the GC-DINA model to real data using two datasets from multi-wave experiments designed to assess the effects of the Enhanced Anchored Instruction (EAII; Bottge et al., 2003) on mathematics achievement. In addition, the GC-DINA model is compared to the latent transition analysis DINA model (LTA-DINA) (Li et al., 2016; Kaya & Leite, 2016) and a longitudinal item response theory (IRT) model (Andersen, 1985) using a simulated data. The results suggest that the GC-DINA model and the LTA-DINA model are similar in terms of the predicted skill mas-

tery; and the GC-DINA model and the longitudinal IRT model are similar in terms of the predicted higher-order latent trait.

**Keywords:** cognitive diagnosis models, higher-order latent trait models, item response models, latent growth curve models, latent transition models, longitudinal assessment, diagnostic classification models, latent class analysis

# Contents

<b>Contents</b>	i
<b>List of Figures</b>	iii
<b>List of Tables</b>	v
<b>1 Introduction</b>	1
<b>2 Literature Review: Previous Approaches to Longitudinal Modeling</b>	3
2.1 Item Response Theory for Measuring Change in Latent Ability . . . . .	3
2.2 Cognitive Diagnosis Models for Assessing Change in Mastery of Latent Skills	7
2.3 Dynamic Bayesian Networks for Assessing Change in Knowledge States . . . . .	10
<b>3 Growth Curve Cognitive Diagnosis Models</b>	14
3.1 Model . . . . .	14
3.2 Estimation . . . . .	18
3.3 Model when the Number of Occasions is Two . . . . .	24
<b>4 Simulation Study</b>	25
4.1 Simulation Study 1 . . . . .	25
4.2 Simulation Study 2 . . . . .	42
<b>5 Empirical Study</b>	45
5.1 Fraction of the Cost (FOC) test data . . . . .	45
5.2 Problem Solving Test (PST) data . . . . .	51
<b>6 Comparison of Longitudinal Psychometric Models</b>	56
6.1 Introduction . . . . .	56
6.2 Method . . . . .	57
6.3 Results . . . . .	57

<b>7 Conclusion</b>	<b>64</b>
<b>A R and Mplus Codes for Chapter 3</b>	<b>66</b>
A.1 R code generating data from the HO-DINA with the Rasch model . . . . .	66
A.2 R code generating data from the HO-DINA with the 2PL model . . . . .	68
A.3 Mplus code of the HO-DINA with the Rasch model . . . . .	69
A.4 MODEL command syntax of the Mplus code of the HO-DINA with the 2PL model	82
<b>B R and Mplus Codes for Chapter 4</b>	<b>83</b>
B.1 R code for the simulation study for comparison between the simple and complex Q-matrix for the simple DINA model . . . . .	83
B.2 R code generating six data sets for Model 1 to Model 6 . . . . .	87
B.3 Mplus code for fitting the GC-DINA model for Model 1 to Model 6 . . . . .	90
B.4 R code generating data for simulation study 2 . . . . .	116
B.5 Mplus code for fitting the GC-DINA in simulation study 2 . . . . .	118
<b>C R and Mplus Codes for Chapter 6</b>	<b>141</b>
C.1 Mplus code for the LTA-DINA model . . . . .	141
C.2 Mplus code for the longitudinal Rasch model . . . . .	141
<b>References</b>	<b>144</b>

# List of Figures

2.1	Knowledge tracing models . . . . .	12
2.2	Markov decision process framework (Almond, 2007a) . . . . .	13
2.3	Simple dynamic Bayesian network with two time points (Culbertson, 2016) . . . . .	13
3.1	Growth curve cognitive diagnosis model (subscript $j$ not shown) . . . . .	17
3.2	Scatter plots of item parameter estimates : (1) Mplus Rasch/1PL estimates vs. generating values and (2) Mplus Rasch/1PL vs. GDINA 1PL. . . . .	22
3.3	Scatter plots of item parameter estimates : (1) Mplus 2PL estimates vs. generating values and (2) Mplus 2PL vs. GDINA 2PL. . . . .	23
4.1	Scatter plots of GC-DINA item parameter estimates vs. generating values for Model 1 and Model 2. . . . .	39
4.2	Scatter plots of GC-DINA item parameter estimates vs. generating values for Model 3 and Model 4. . . . .	40
4.3	Scatter plots of GC-DINA item parameter estimates vs. generating values for Model 5 and Model 6. . . . .	41
4.4	Scatter plots of GC-DINA item parameter estimates vs. generating values for simulation study 2. . . . .	43
4.5	Scatter plots of predicted higher-order latent traits from the GC-DINA model vs. generating values for simulation study 2. The estimated correlation, $r$ , is in parenthesis. . . . .	44
5.1	Predicted growth trajectories in the higher-order latent trait across four time points for the FOC data. . . . .	48
5.2	Scatter plots of GC-DINA item parameter estimates vs. LTA-DINA item parameter estimates reported in Li et al. (2016) for the FOC data. . . . .	49
5.3	Scatter plots of GC-DINA item parameter estimates vs. generating values for the simulated data using parameters recovered from the FOC data. . . . .	51

5.4	Scatter plots of GC-DINA item parameter estimates vs. generating values for the simulated data using parameters recovered from the PST data. . . . .	55
6.1	Scatter plots of predicted higher-order latent traits vs. generating values: (1) GC-DINA estimates vs. generating values and (2) longitudinal Rasch estimates vs. generating values. The estimated correlation, $r$ , is in parenthesis. . . . .	62
6.2	Scatter plots of predicted higher-order latent traits: GC-DINA estimates vs. longitudinal Rasch estimates. The estimated correlation, $r$ , is in parenthesis. . . . .	63

# List of Tables

3.1	Q-matrix of 20 items and 4 skills for the HO-DINA model simulation study. . . . .	20
3.2	Estimated parameter values from Mplus and GDINA R package. . . . .	21
4.1	Simple Q-matrix of 20 items and 4 skills for the simulation study. . . . .	27
4.2	Complex Q-matrix of 20 items and 4 skills for the simulation study. . . . .	27
4.3	Classification accuracy with the simple and complex Q-matrices for the simple DINA model. . . . .	27
4.4	Parameter estimates and standard errors (in parentheses) for Model 1 and Model 2	31
4.5	Parameter estimates and standard errors (in parentheses) for Model 3 and Model 4	32
4.6	Parameter estimates and standard errors (in parentheses) for Model 5 and Model 6	33
4.7	Square root of relative efficiencies for $\beta$ : the simple Q-matrix vs. the complex Q-matrix. . . . .	34
4.8	Parameter estimates and standard errors (in parentheses) for Model 1 and Model 3	36
4.9	Parameter estimates and standard errors (in parentheses) for Model 3 and Model 5	37
4.10	Generating values and GC-DINA estimates for simulation study 2. . . . .	43
5.1	Q-matrix for Fraction of Cost (FOC) test. . . . .	47
5.2	Estimated GC-DINA model parameters for the FOC data. . . . .	48
5.3	Proportion of students predicted to have mastered each skill at each occasion for the FOC data (LTA results from Li et al. (2016)). . . . .	49
5.4	GC-DINA parameter estimates and standard errors for the simulated data using parameters recovered from the FOC data. . . . .	50
5.5	Q-matrix for Problem Solving Test (PST). . . . .	52
5.6	GC-DINA parameter estimates and standard errors for the PST data. . . . .	53
5.7	GC-DINA item parameter estimates and standard errors for the PST data. . . . .	54
5.8	Proportion of students predicted to have mastered each skill at each occasion for the PST data. . . . .	54
5.9	GC-DINA parameter estimates and standard errors for the simulated data using parameters recovered from the PST data. . . . .	55

6.1	Parameter estimates for the GC-DINA model and the longitudinal Rasch model.	58
6.2	Higher-order structural parameter estimates for the GC-DINA model. . . . .	59
6.3	Parameter estimates for the LTA-DINA model. . . . .	59
6.4	Item parameter estimates for the GC-DINA model, the LTA-DINA model, and the longitudinal Rasch model. . . . .	59
6.5	Model fit information of the GC-DINA model, the LTA-DINA model, and the longitudinal Rasch model. . . . .	60
6.6	Proportion of correctly classified skills (PCS) and last element of the proportion of correctly classified skill profile vectors (PCV) for the GC-DINA model and the LTA-DINA model. . . . .	60
6.7	Model-implied probability of having each skill at each time point for the GC-DINA model and the LTA-DINA model. . . . .	61

## Acknowledgments

First and foremost, I would like to express the deepest gratitude to my advisor Professor Sophia Rabe-Hesketh. This work would not have been possible without her unwavering support and intellectual guidance that she has provided to me over the last five years. She has not only shown me the route to scholarly curiosity, but also served as a mentor, a guide, and a true friend as I was about to succumb, when faced with numerous hurdles throughout the process. I also would like to thank the professors who served as members of my dissertation committee. I thank Professor Zachary Pardos, because without him, I would not have had a chance to get introduced to the exciting world of educational data mining. I thank Professor Nicholas Jewell, for his insightful and indispensable comments on my work from the very first draft to the very last one. I thank Professor Mark Wilson, as he was the one who brought me in the realm of educational measurement.

I was truly fortunate to meet great mentors, colleagues and friends along the way. I sincerely appreciate the learning experience UCSF Department of Psychiatry provided, including Professor Kaja LeWinn, Katrina Roundfield and Ellen Kersten. I am deeply indebted to Professors Taeyoung Park and Hakbae Lee at Yonsei University for their support throughout my academic journey. I am grateful to all my friends I met in Berkeley, and special thanks to QME folks.

Finally, my sincere and greatest appreciation goes to my family who provided unconditional love and support throughout my life. This dissertation is dedicated to them.

# Chapter 1

## Introduction

In educational measurement, psychometric models have been used to measure students' latent characteristics such as knowledge or aptitude. One well-known class of psychometric models are item response theory (IRT) models. IRT models define students' ability as continuous latent variables and have been used to order students on a continuum. Recently, cognitive diagnosis models (CDM), alternatively called diagnostic classification models (DCM; e.g., Rupp et al., 2010), have received increasing attention. CDMs define the target proficiency as a set of multiple skills specified at a fine grain size. By examining students' mastery of these skills, the model provides diagnostic information for instruction and learning, i.e., students' strengths and weaknesses in the domain.

Most CDMs treat responses as if they were from a single time point and do not account for change in knowledge proficiency across time. Students' skill knowledge, however, changes over time as they learn and it is important to know their learning trajectories. For example, students take periodic tests to prepare for high stakes exams in schools or students interact with intelligent tutors on a daily basis to prepare for an end of unit exam. In addition, pre-post tests can be administered to evaluate educational treatments. By understanding students' learning over time, educators can monitor students' progress toward learning goals and decide what to adjust to achieve better learning; and students themselves can also be informed on what they can and cannot and what they need to focus on next to reach their goals. However, while there is an extensive literature on IRT-based longitudinal models, little work has been done on extending CDMs for longitudinal settings.

In this dissertation, I propose a longitudinal CDM, "growth curve cognitive diagnosis model" (GC-CDM). GC-CDMs were motivated by higher-order latent trait models (de la Torre & Douglas, 2004), which define a higher-order continuous latent trait that affects all the latent skills. The higher-order latent trait can be viewed as the more broadly defined general ability associated with the fine grained skills. GC-CDMs trace changes in the higher-order latent traits over time by using latent growth modeling for the higher order traits and

simultaneously trace students' skill mastery through the CDM measurement model.

The outline of this dissertation is as follows. Chapter 2 reviews commonly used psychometric models (i.e., item response theory models, cognitive diagnosis models and dynamic Bayesian networks) and their extension for analysis of longitudinal data. Chapter 3 introduces our proposed method, GC-CDMs. The details of the modeling framework and the estimation method are discussed. In Chapter 4, two simulation studies are discussed. We particularly examine parameter recovery for the GC-CDM under various conditions. Chapter 5 discusses application of GC-CDMs to two datasets from the Enhanced Anchored Instruction (EAI; Bottge et al., 2003) project. In Chapter 6, different longitudinal psychometric models are compared using a simulated data set from the GC-CDM. We end with a conclusion in Chapter 7.

# Chapter 2

## Literature Review: Previous Approaches to Longitudinal Modeling

In educational assessment, different types of psychometric models exist and there have been increasing efforts to extend them for the analysis of longitudinal data. This chapter reviews commonly used psychometric models and discusses existing longitudinal modeling approaches in each framework. In particular, item response theory models, cognitive diagnosis models and Bayesian networks will be discussed.

### 2.1 Item Response Theory for Measuring Change in Latent Ability

#### Item Response Theory (IRT) Models

Item response theory (IRT) is used to develop, score and analyze assessments that measure respondents' latent characteristics. In education, IRT has been widely used in large-scale assessments measuring students' ability. The main idea of IRT is the item response function (IRF), which specifies the probability of a given response as a function of the student's true ability.

The simplest IRT model for binary responses ( $Y = 0$  if the question or item has been answered incorrectly and  $Y = 1$  if it has been answered correctly) is the one-parameter logistic (1PL) model with an item difficulty parameter for each item, most commonly known as the Rasch (Rasch, 1961) model. Under the Rasch model, the probability that person  $j$  with latent ability  $\theta_j$  gives a correct response ( $Y_{ij} = 1$ ) to item  $i$  with difficulty  $\beta_i$  is

$$P(Y_{ij} = 1|\theta_j) = \frac{\exp(\theta_j - \beta_i)}{1 + \exp(\theta_j - \beta_i)}. \quad (2.1)$$

The model with two item parameters, item difficulty and discrimination, is called the two-parameter logistic (2PL) model (Birnbaum, 1968) and is defined as

$$P(Y_{ij} = 1|\theta_j) = \frac{\exp\{\alpha_i(\theta_j - \beta_i)\}}{1 + \exp\{\alpha_i(\theta_j - \beta_i)\}}, \quad (2.2)$$

where  $\alpha_i$  is the discrimination and  $\beta_i$  is the difficulty of item  $i$ . The discrimination parameter characterizes how well the item differentiates among persons who are at different ability levels.

It is assumed that the responses to an item are independent of the responses to any other item conditional on the person's ability, which is referred to as *local independence*. The joint probability of a response vector  $\mathbf{y}_j$ , given latent ability  $\theta_j$ , can be expressed as  $P(\mathbf{y}_j|\theta_j) = \prod_{i=1}^I p_{ij}^{y_{ij}} \{1 - p_{ij}\}^{1-y_{ij}}$ , where  $p_{ij} = P(Y_{ij} = 1|\theta_j)$  is defined in (2.1) and (2.2). In maximum likelihood estimation (MLE) approaches, the item parameters,  $\boldsymbol{\xi}$ , are assumed to be fixed effects. The ability parameters,  $\boldsymbol{\theta}$ , can be considered as either fixed effects or random effects, and different MLE methods have been used depending on the assumption on  $\boldsymbol{\theta}$ . The joint maximum likelihood (JML) method assumes that the ability parameters are fixed effects and the likelihood is jointly maximized with respect to item and ability parameters. The conditional maximum likelihood (CML) method maximizes the conditional likelihood only with respect to item parameters after conditioning on the sum score of respondents which is a sufficient statistic for  $\theta_j$  (Andersen, 1970). This approach works for the Rasch model without the discrimination parameter. The ability parameters are then estimated based on the estimated item parameters.

When the ability parameters are considered as random effects, assuming  $\theta_j \sim N(0, \sigma^2)$ , the model can be viewed as a mixed effects logistic regression model (Rijmen et al., 2003) and the marginal maximum likelihood (MML) approach is used. In this case, the ability parameters are marginalized out of the likelihood function, and the marginal likelihood is maximized with respect to item parameters and the variance parameter  $\sigma^2$ . Then, the person abilities are predicted using empirical Bayes techniques (Carlin & Louis, 2000). Unlike the MLE approaches, Bayesian estimation methods place prior distributions on the parameters of interest,  $\boldsymbol{\beta}$  and  $\sigma$ . The prior distributions are updated by the observed data using Bayes' Theorem and the updated distribution is referred to as posterior. The posterior distribution is approximated by Markov chain Monte Carlo (MCMC). We can simultaneously estimate both item and ability parameters based on the joint posterior distribution. More details about estimation methods for IRT models are discussed in Baker & Kim (2004).

The basic IRT models (i.e., Rasch and 2PL) have been extended for more flexible modeling in various situations. The three-parameter logistic (3PL) model (Harris, 1989) introduces the guessing parameter for each item to incorporate chance success on an item which may occur in assessments with the multiple-choice or true/false response format. When test items

are associated with more than one latent trait such as in personality assessments, multidimensional IRT models (Van Der Linden & Hambleton, 1997) have been applied. Explanatory IRT incorporates explanatory variables (e.g., item covariates, person covariates) to take into account different characteristics among items and persons, such as why some items are more difficult than others and why some students have higher abilities than others (De Boeck & Wilson, 2004). For the situation in which the sample of respondents consists of several latent sub-populations that are qualitatively different but an IRT measurement model holds within each subgroup, mixture IRT models (Rost, 1990)—a combination of IRT and latent class analysis (LCA)—has been developed.

## IRT-based Longitudinal Models

Several studies have suggested IRT-based longitudinal models to measure individual differences in growth over time. Andersen (1985) proposed a multidimensional Rasch model for the situation in which individuals are tested with the same set of items at two different time points. Andersen's model is

$$P(Y_{ijt} = 1 | \theta_{jt}) = \frac{\exp(\theta_{jt} - \beta_i)}{1 + \exp(\theta_{jt} - \beta_i)},$$

where  $Y_{ijt}$  is the response of person  $j$  to item  $i$  at time  $t$ ;  $\theta_{jt}$  is the ability of person  $j$  at time  $t$ ; and  $\beta_i$  is the difficulty of item  $i$ . The model assumes that the item difficulties are constant across time. The item responses at each time-point are modeled with a unidimensional Rasch model and the abilities are assumed to have a bivariate normal distribution with an unstructured covariance matrix, i.e., the abilities are allowed to be correlated over time. The individual growth can be estimated by calculating differences between time-specific predicted abilities.

Similarly to Andersen's model, Andrade & Tavares (2005) suggested a longitudinal IRT model to measure individual abilities at  $T$  consecutive occasions. The model considers the case where the different tests administered over time either have common items or different sets of items. Assuming that the item parameters are known (e.g., the tests consist of calibrated items on the same metric), the model can be expressed as

$$P(Y_{ijt} = 1 | \theta_{jt}) = c_i + (1 - c_i) \frac{\exp\{\alpha_i(\theta_{jt} - \beta_i)\}}{1 + \exp\{\alpha_i(\theta_{jt} - \beta_i)\}},$$

$$\boldsymbol{\theta}_j = (\theta_{j1}, \theta_{j2}, \dots, \theta_{jT})^T \sim MVN_T(\boldsymbol{\mu}, \Sigma),$$

where  $Y_{ijt}$ ,  $\theta_{jt}$  and  $\beta_i$  are defined as in Andersen's model;  $\alpha_i$  and  $c_i$  are discrimination and guessing parameters of item  $i$ , respectively, of the 3PL model; and  $MVN_T(\boldsymbol{\mu}, \Sigma)$  is

the  $T$ -dimensional multivariate normal distribution with mean vector  $\mu$  and unstructured covariance matrix  $\Sigma$ .

Embretson (1991) proposed a multidimensional Rasch model for learning and change (MRMLC). Embretson's model reparameterizes Andersen's model to provide a direct estimate of change between two adjacent occasions. The model can be written as

$$P(Y_{ijt} = 1 | \theta_{j1}^*, \dots, \theta_{jt}^*) = \frac{\exp(\sum_{k=1}^t \theta_{jk}^* - \beta_i)}{1 + \exp(\sum_{k=1}^t \theta_{jk}^* - \beta_i)},$$

where  $Y_{ijt}$  and  $\beta_i$  are defined as in Andersen's model;  $\theta_{j1}^* = \theta_{j1}$  is the initial ability for person  $j$ ; and  $\theta_{jt}^* = \theta_{jt} - \theta_{j(t-1)}$  is the change between abilities at time  $t$  and  $t - 1$  for person  $j$ .

Several studies have proposed longitudinal IRT models that incorporate a growth curve. With a growth curve, the ability can be modeled as a smooth function of time, and the number of occasions and their timing do not need to be the same across persons. For example, Pastor & Beretvas (2006) suggested a linear growth model,

$$\theta_{jt} = \gamma_{j1} + \gamma_{j2} \text{time}_{jt} + \epsilon_{jt},$$

where  $\gamma_{j1}$  and  $\gamma_{j2}$  are the random intercept and random slope which correspond to initial ability (at time  $jt = 0$ ) and growth rate respectively;  $(\gamma_{j1}, \gamma_{j2})^T \sim N(0, \Sigma)$ ; and there are time-specific random effects  $\epsilon_{jt} \sim N(0, \sigma_\epsilon^2)$ .

Cho et al. (2010) introduced a longitudinal extension of mixture IRT models. They combined a latent transition model (Graham et al., 1991) with a mixture Rasch model to incorporate change in latent class membership over time. In the model, the latent class patterns up to time  $T$ ,  $h_T$ , are modeled by a Markov chain that is stationary over time points. The probability of  $h_T = (g_1, g_2, \dots, g_T)$ , where  $g_t$  denotes the class membership at time  $t$  ( $t = 1, \dots, T$ ), can be written as

$$P[h_T = (g_1, g_2, \dots, g_T)] = P(g_1) \prod_{t=2}^T P(g_t | g_{t-1}) = \pi_{g_1} \prod_{t=2}^T \tau_{g_t | g_{t-1}}^{(t-1)},$$

where  $\pi_{g_1}$  is the proportion of the population in latent class  $g_1$  at time 1 and  $\tau_{g_t | g_{t-1}}^{(t-1)}$  is the transition probability from latent class  $g_{t-1}$  at time  $t-1$  to latent class  $g_t$  at time  $t$ . The model assumes that the class membership at time  $t$  is only influenced by the class membership at the previous time-point  $t-1$ . Within the latent class pattern  $h_T$ , the probability that person  $j$  responds correctly to item  $i$  of the instrument administered at time  $t$  can be written as

$$P(Y_{ijt} = 1 | h_T, \theta_{jth_T}) = \frac{\exp(\theta_{jth_T} - \beta_{ith_T})}{1 + \exp(\theta_{jth_T} - \beta_{ith_T})},$$

where  $\theta_{jth_T}$  is the ability of person  $j$  within the pattern  $h_T$  and  $\theta_{jth_T} \sim MVN(\mu_{h_T}, \Sigma_{h_T})$  with mean vector  $\mu_{h_T}$  and covariance matrix  $\Sigma_{h_T}$ ;  $\beta_{ith_T}$  is the difficulty of item  $i$  at time  $t$  for the pattern  $h_T$  and item parameter invariance across times is assumed.

## 2.2 Cognitive Diagnosis Models for Assessing Change in Mastery of Latent Skills

### Cognitive Diagnosis Models

Cognitive diagnosis models (CDM) have been commonly used in formative assessments for diagnostic purposes. In contrast to IRT where person ability is treated as continuous, CDMs assume the presence or absence of multiple fine-grained skills (or attributes). The presence and absence of skills are referred to as “mastery” and “non-mastery” respectively. A respondent’s knowledge is represented by a binary vector, referred to as “skill profile”, to indicate which skills have been mastered or have not.

CDMs can be compensatory or non-compensatory in terms of how the multiple skills required by an item interact with each other. In compensatory models, mastery of one skill can compensate for non-mastery of other skills. In non-compensatory models, however, one skill cannot compensate for the lack of others. Non-compensatory models are sometimes referred to as conjunctive models in that all skills associated with an item should be mastered in order to have the required knowledge for the correct response.

The deterministic inputs, noisy “and” gate (DINA) model (Haertel, 1990; Junker & Sijtsma, 2001) is a popular non-compensatory and conjunctive CDM. In the DINA model, the probability of getting the item correct depends on whether or not the person possesses all required skills. Specifying the model requires information about which skills are required by each item. For this, we use a Q-matrix (Tatsuoka, 1985) which is an  $I \times K$  matrix where  $q_{ik} = 1$  if item  $i$  requires skill  $k$  and 0 if not.  $I$  is the number of items and  $K$  is the number of skills in the assessment.

We also define a latent mastery indicator  $\alpha_{jk}$  for person  $j$ ’s knowledge of skill  $k$ , where  $\alpha_{jk} = 1$  if person  $j$  has mastered skill  $k$  and 0 if he or she has not. The skill profile  $\alpha_j$  of person  $j$  is a binary vector of length  $K$  that indicates whether or not the person has mastered each of the  $K$  skills. Combining information on  $\alpha_j$  and the Q-matrix, a latent variable  $\xi_{ij}$  indicating whether person  $j$  has mastered all required skills for item  $i$  is defined as follows

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

However,  $\xi_{ij}$  is not directly observed but only indirectly via the observed response  $Y_{ij}$  which is subject to noise or misclassification. Specifically, the model allows for the probability of slipping and guessing defined by

$$s_i = P(Y_{ij} = 0 | \xi_{ij} = 1),$$

$$g_i = P(Y_{ij} = 1 | \xi_{ij} = 0).$$

The slipping parameter ( $s_i$ ) is the probability that person  $j$  responds incorrectly to item  $i$  even if he or she has mastered all required skills ( $\xi_{ij} = 1$ ). The guessing parameter ( $g_i$ ) is the probability that person  $j$  responds correctly to item  $i$  even if he or she has not mastered all the required skills ( $\xi_{ij} = 0$ ).

The probability of a correct response of person  $j$  for item  $i$ ,  $\pi_{ij}$ , is represented by the DINA model as follows:

$$\pi_{ij} = P(Y_{ij} = 1 | \boldsymbol{\alpha}_j) = (1 - s_i)^{\xi_{ij}} g_i^{1 - \xi_{ij}}. \quad (2.3)$$

The deterministic inputs, noisy “or” gate (DINO) model (Templin & Henson, 2006) is the compensatory analog to the DINA model. DINO assumes that a respondent has enough knowledge required by an item if he or she has mastered at least one of the associated skills. The latent variable  $\xi_{ij}$  in DINO is defined by

$$\xi_{ij} = 1 - \prod_{k=1}^K (1 - \alpha_{jk})^{q_{ik}}.$$

Just as in the DINA model, slipping and guessing processes are modeled at the item level and the probability of a correct response to an item is modeled as in (2.3).

The DINA or DINO models determine the probability of item responses,  $\mathbf{Y}$ , given the skill mastery status,  $\boldsymbol{\alpha}$ , which corresponds to the measurement part of CDMs. After specifying the measurement part of the model, we need to consider the probability distribution of  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_K)$ , which is the structural part of the model. Several different types of structural models have been discussed for the joint probability distribution of skill mastery across different skills (Maris, 1999; Rupp et al., 2010). The simplest structural model is the independence model, which assumes the skill mastery indicators are independent of each other:  $P(\alpha_1, \dots, \alpha_K) = \prod_{k=1}^K P(\alpha_k)$ . However, the independence assumption among skills are not realistic in practice. Other structural models are unstructured, log-linear (Maris, 1999; Xu & Davier, 2008), unstructured tetrachoric (Hartz, 2002), and structured tetrachoric models. One type of the structured tetrachoric model is the higher-order latent trait model by de la Torre (2009), which assumes that the skill mastery indicators are conditionally independent given the higher-order latent trait. More details of the higher-order latent trait model will be discussed in Section 3.1.

In likelihood-based approaches, the item parameters (i.e., guessing and slipping) are often estimated by marginal maximum likelihood. In the latent class modeling framework, each skill profile can be viewed as a latent class and the marginalized likelihood of the data can be expressed as  $L(\mathbf{y}) = \prod_{j=1}^J \sum_{c=1}^C \nu_c \prod_{i=1}^I P(Y_{ij} = y_{ij} | \boldsymbol{\alpha}_c)$ , where  $\boldsymbol{\alpha}_c$  is the attribute pattern of latent class  $c$  and  $\nu_c$  is the probability of membership in latent class  $c$ . The likelihood can be maximized using an EM algorithm. Details of the algorithm for estimating parameters

of the DINA model are discussed in de la Torre (2009). After the item parameters are estimated, posterior skill mastery probabilities for each person and skill can be obtained by empirical Bayes, and the skill profiles are then predicted by rounding these probabilities to 0/1 (i.e., choosing the mastery status with the greater posterior probability). In Bayesian approaches, we assign prior distributions to the guessing and slipping parameters and both item parameters and skill profiles are simultaneously estimated via MCMC.

A number of alternative types of CDMs have been proposed. The noisy input, deterministic “And” gate (NIDA) model (Maris, 1999) is an extension of the DINA model but specifies the slipping and guessing parameters at the skill level. The reparameterized unified model (RUM) (Hartz, 2002; Roussos et al., 2007) is a generalization of the NIDA and DINA models, allowing different slipping and guessing parameters at both the item and skill level. Also, different general modeling frameworks have been suggested, e.g., general diagnostic model (GDM) (Davier, 2005), log-linear CDM (LCDM) (Henson et al., 2009) and generalized DINA (G-DINA) (de la Torre, 2011). Rupp et al. (2010) provides a comprehensive discussion of the theory and application of CDMs.

## Latent Transition Analysis Cognitive Diagnosis Models

Although CDMs have recently received increasing attention in educational measurement, most CDMs are static models and there has been little previous work on longitudinal cognitive diagnosis modeling.

In recent papers on CDM-based longitudinal models, latent transition analysis (LTA; Collins & Wugalter, 1992) has been combined with a CDM to account for change in attribute mastery over time (Li et al., 2016; Kaya & Leite, 2016). The LTA approaches specify four growth transition probabilities: the probability from nonmastery to mastery,  $p_{m|n}$ , from nonmastery to nonmastery,  $p_{n|n}(= 1 - p_{m|n})$ , from mastery to nonmastery,  $p_{n|m}$ , and from mastery to mastery,  $p_{m|m}(= 1 - p_{n|m})$ , where  $m$  and  $n$  stand for mastery and nonmastery respectively.

Assuming that the latent skills are independent of each other at each occasion, the model can be expressed as:

$$\begin{aligned} & P(\mathbf{Y}_j = \mathbf{y}_j) \\ &= \sum_{c=1}^C \prod_{k=1}^K \left\{ P(A_{k1} = \alpha_{k1}) \prod_{t=2}^T P(A_{kt} = \alpha_{kt} | A_{k(t-1)} = \alpha_{k(t-1)}) \right\} \prod_{t=1}^T P(\mathbf{Y}_{jt} = \mathbf{y}_{jt} | \mathbf{A}_t = \boldsymbol{\alpha}_t) \end{aligned}$$

where  $c$  is the index for the skill profile across  $T$  occasions,  $c = 1, \dots, C$  with  $C = 2^{TK}$ ;  $\mathbf{Y}_{jt}$  is the response of person  $j$  observed at time  $t$ ;  $\mathbf{Y}_j$  is the set of responses of person  $j$  observed over  $T$  time points;  $A_{kt}$  is skill  $k$  measured at time  $t$ ;  $P(A_{k1} = \alpha_{k1})$  is the probability of mastery status  $\alpha_{k1}$  of skill  $k$  at the first time;  $P(A_{k1} = \alpha_{k1}) \prod_{t=2}^T P(A_{kt} = \alpha_{kt} | A_{k(t-1)} = \alpha_{k(t-1)})$

$\alpha_{k(t-1)}$ ) indicates the probability of a sequence of skill mastery indicators for skill  $k$  across occasions; and the measurement model  $P(\mathbf{Y}_{jt} = \mathbf{y}_{jt} | \mathbf{A}_t = \boldsymbol{\alpha}_t)$  is either the DINA or DINO model. The latent transition analysis DINA (LTA-DINA) model assumes that the item parameters (guessing and slipping) are constant over time, while the skill knowledge state is allowed to change over time. In addition, transition probabilities  $P(A_{kt} = \alpha_{kt} | A_{k(t-1)} = \alpha_{k(t-1)})$  are allowed to differ for each skill and are sometimes constrained to be constant over time.

## 2.3 Dynamic Bayesian Networks for Assessing Change in Knowledge States

### Bayesian Networks

Bayesian networks (BNs) have been extensively used as student models for intelligent tutoring systems (ITS; for examples, Bunt & Conati, 2002; Zapata-Rivera & Greer, 2004; Koedinger & Aleven, 2007), and they also have been successfully applied for cognitively diagnostic assessments (R. J. Mislevy, 1995; R. Mislevy et al., 1999; Levy & Mislevy, 2004; Sinharay & Almond, 2007). A BN consists of a *directed acyclic graph* (DAG) and a probability distribution. In the graph, nodes represent random variables and edges between nodes represent probabilistic dependencies among them. For the directed edge from node  $A$  to node  $B$ , we say that  $A$  is a *parent* of  $B$  and  $B$  is a *child* of  $A$ . The set of all parents of node  $C$ , as well as the parents of these parents, etc., are referred to as *ancestors* of  $C$ , and the set of all children of node  $C$  including the children of children are called *descendants* of  $C$ . Each node in a BN is conditionally independent of all its non-descendants given the state of that node's parents. By the general multiplication rule, the joint probability distribution over all the variables in a BN factorizes into a series of conditional probability distributions. This property provides a convenient way to define the joint distribution compactly by only specifying the conditional probability distribution for every node (Heckerman, 1998; Pearl, 1988).

One of the powerful features of BNs is that they can be used for probabilistic inference about unobserved variables through the complete structure of the variables and their relationships. For example, when the *leaves* (nodes with no children) are observed, the prior belief on their *roots* (nodes with no parents) can be updated based on the observed evidence by applying Bayes Theorem. There are two kinds of inference algorithms: exact and approximate. The most common inference methods are variable elimination and a junction tree algorithm for exact inference, and importance sampling and MCMC simulation for approximate inference.

When BNs are applied to educational measurement, hidden nodes usually represent discrete proficiency levels, and leaves generally represent observed responses to items. Specifically, for an assessment designed to measure latent knowledge on multiple skills, BNs can be used as a CDM approach. Conjunctive or disjunctive CDMs are available in the BN framework using Noisy-OR and Noisy-AND models. The edges from hidden nodes to observed nodes correspond to the elements of the Q-matrix. More details about application of BNs for CDMs are discussed in Almond et al. (2015). One of BNs' benefits over traditional CDMs is that BNs can easily incorporate associations between skills into the model by edges connecting hidden nodes, which can improve measurement precision.

## Dynamic Bayesian Networks in Assessment for Learning Systems

Dynamic Bayesian networks (DBNs) (Dean & Kanazawa, 1989) have been widely used to model student learning in ITS. Changes over time can be modeled by two pieces: (1) a single time-slice model for the initial state of the system and (2) a two-time-slice model for the transition from one-time slice to the next. By a Markov property, it is assumed that the knowledge state in any time slice is independent of the past history given the previous time slice. Almond et al. (2015) discusses two ways to model the transition from one state to the next in learning systems: mathematical learning models and Markov decision process (MDP).

The basic idea of mathematical learning models is that a student has mastered or not mastered a certain knowledge at time  $t$ , responds to a practice question, and the mastery state at time  $t + 1$  depends on the previous mastery state. A popular example is the knowledge tracing (KT) model introduced by Corbett & Anderson (1994). The KT model monitors a student's proficiency on a single knowledge component during practice, and the system provides proper guidance needed for the student and determines when the student has reached the mastery. Figure 2.1 presents the basic form of KT models.  $A_t$  is a student's latent knowledge for a particular skill at time  $t$  and  $Y_t$  is the student's response to a question at time  $t$ . A circle represents a latent variable and a rectangle represents an observable variable.

A KT model uses four parameters: *Prior*, *Learn*, *Guess* and *Slip*. The *Prior* is  $P(A_1 = 1)$ , which is the initial probability that students know the skill a priori. *Learn* is the probability that students' knowledge state will transition from unlearned (non-mastery) to learned (mastery),  $P(A_t = 1|A_{t-1} = 0)$ , after interacting with each question. It is assumed that it is impossible to lose a skill,  $P(A_t = 0|A_{t-1} = 1) = 0$ . *Guess* is the probability that students get a correct answer by guessing when they do not know the skill, and *Slip* is the probability that students respond incorrectly to an item by slipping although they know the skill.

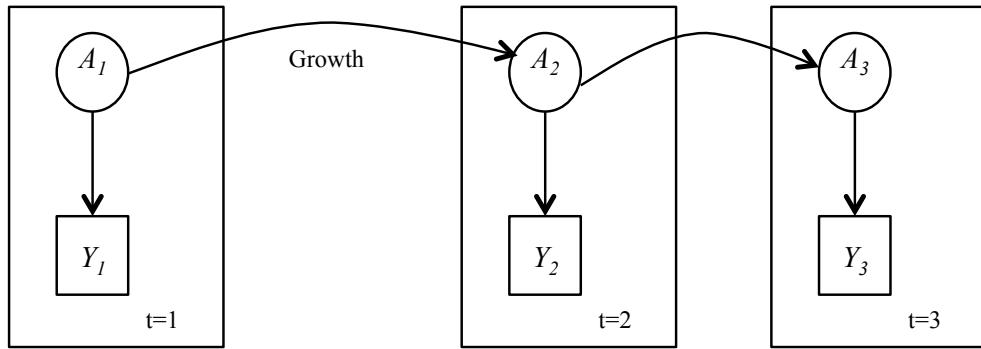


Figure 2.1: Knowledge tracing models

The prediction of student  $j$ 's knowledge state following the  $(t-1)$ th question is as follows:

$$\begin{aligned} P(A_{jt} = 1 | Y_{j(t-1)} = y_{j(t-1)}) &= P(A_{j(t-1)} = 1 | Y_{j(t-1)} = y_{j(t-1)}) \\ &\quad + P(A_{j(t-1)} = 0 | Y_{j(t-1)} = y_{j(t-1)}) \times Learn \end{aligned}$$

This is the sum of (1) the posterior probability that the skill has been mastered given the response to the  $(t-1)$ th question and (2) the posterior probability that the skill has not been mastered given the response but transitioned from unlearned to learned state.  $P(A_{j(t-1)} = \alpha_{j(t-1)} | Y_{j(t-1)} = y_{j(t-1)})$  is calculated by Bayes' theorem under the BN framework, so the model is also called a Bayesian knowledge tracing model.

MDPs (Boutilier et al., 1999) have been often used to inform instructional decision making by incorporating nodes that influence the transition between the time slices into the model. A MDP model contains (1) a set of possible states  $S$ , (2) a set of possible actions  $A$ , (3) the transition model  $T$  for each action's effects in each state, (4) a reward structure of the actions and (5) a discount rate indicating the relative value of future compared to immediate rewards. LaMar (2017) discussed the use of a MDP to model students' decision making process for complex assessment tasks. Matsuda & VanLehn (2000) applied a MDP to make decisions about selecting hints or new problems. Almond (2007a,b) employed a MDP to understand the effectiveness of an instruction and student learning. Figure 2.2 shows the general framework of MDP presented in Almond (2007a). The transition from time  $t$  to time  $t+1$  depends on activity variables (e.g., the choice of instruction). The student's knowledge states,  $A$ , are not observable so this model is called a partially observable Markov decision process (POMDP).

Culbertson (2016) discussed the possibility of more general DBNs with multiple latent skills. For example, Figure 2.3 represents a DBN with three skills ( $A_{at}$ ,  $A_{bt}$ ,  $A_{ct}$  for  $t = 1, 2$ ) measured at two time points. More complex temporal relationships can also be modeled such as the effect of change in one latent variable on other related latent variables (Schäfer

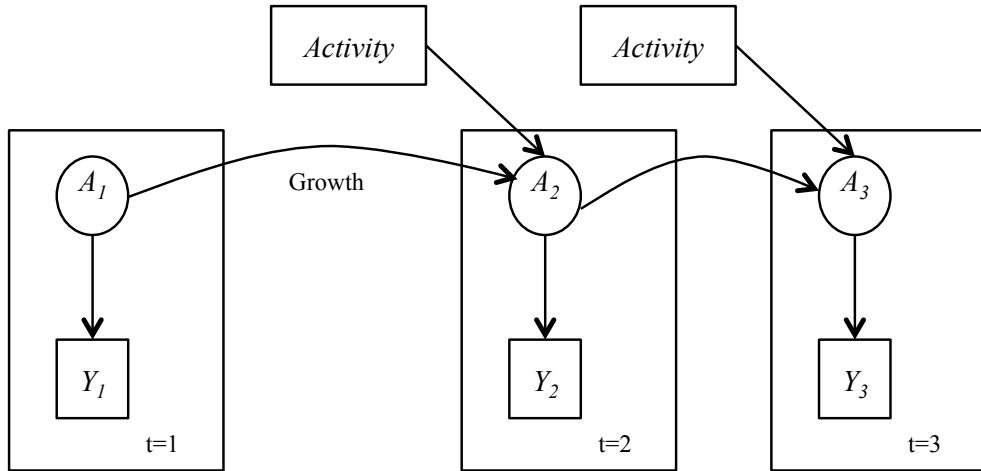


Figure 2.2: Markov decision process framework (Almond, 2007a)

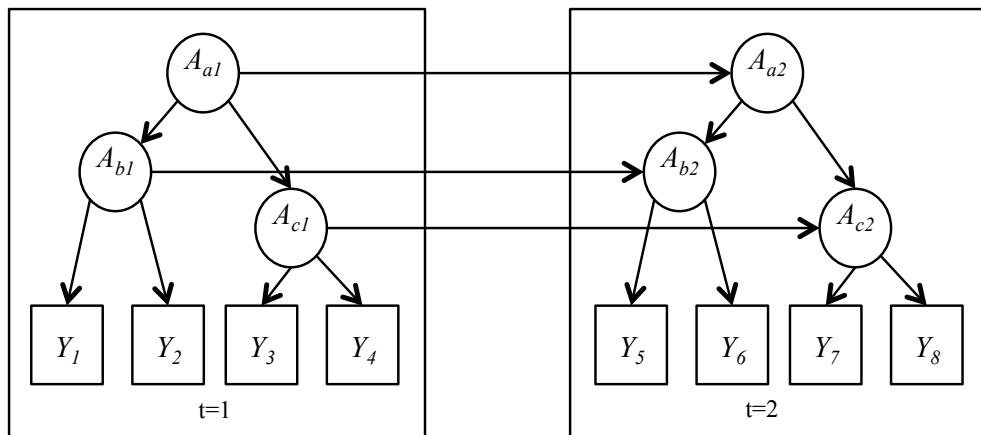


Figure 2.3: Simple dynamic Bayesian network with two time points (Culbertson, 2016)

& Weyrath, 1997). But the application of such DBNs to real data has been understudied for various reasons; the model can become overly complex, for example, when the model allows different parameters for each person (e.g., individualized learn rate). Also, the fact that little is known about the calibration of BNs serves as another hindrance to their application.

# Chapter 3

## Growth Curve Cognitive Diagnosis Models

In this chapter, we describe our proposed longitudinal cognitive diagnosis model—‘growth curve cognitive diagnosis model (GC-CDM)’. This model can be best understood as an extension of the higher-order latent trait model by de la Torre & Douglas (2004). The higher-order latent trait can be viewed as a continuous aptitude defined more broadly in the higher level on top of latent skills. Our model traces change in the higher-order latent trait over time by using latent growth modeling, so it also traces the skill mastery profiles associated with the trait. We first discuss the higher-order latent trait model for a single time point and introduce the GC-CDM for multiple time points. We describe the estimation method for our model—in particular, we apply a nested integration method to deal with the high-dimensional computation problem in the evaluation of the marginal likelihood.

### 3.1 Model

#### Higher-order Latent Trait Models for Cognitive Diagnosis

Cognitive diagnosis models can be thought of as latent class models (LCM) in that item responses and skill mastery status (latent variables) are both categorical. For a test developed to measure  $K$  skills, for example, we can consider  $2^K$  mastery patterns (latent classes) and would like to assign each student to one of the skill patterns. Under the LCM framework, the responses of person  $j$  across  $I$  items,  $\mathbf{Y}_j = (Y_{1j}, Y_{2j}, \dots, Y_{Ij})'$ , which are associated with mastery of  $K$  skills,  $\boldsymbol{\alpha}_j = (\alpha_{j1}, \alpha_{j2}, \dots, \alpha_{jK})'$ , can be modeled as follows:

$$P(\mathbf{Y}_j = \mathbf{y}_j) = \sum_{c=1}^C P(\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_c) P(\mathbf{Y}_j = \mathbf{y}_j | \boldsymbol{\alpha}_j),$$

where  $\boldsymbol{\alpha}_c$  indicates the skill mastery pattern of latent class  $c$  and the sum is over all  $C = 2^K$  classes. Following the structural equation modeling convention, the first part of the model,  $P(\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_c)$ , is often called the *structural* component and the second part,  $P(\mathbf{Y}_j = \mathbf{y}_j | \boldsymbol{\alpha}_j)$ , is called the *measurement* component.

The measurement component can be rewritten by the local independence assumption:  $P(\mathbf{Y}_j = \mathbf{y}_j | \boldsymbol{\alpha}_j) = \prod_{i=1}^I P(Y_{ij} = y_{ij} | \boldsymbol{\alpha}_j)$ . The measurement part of the DINA and DINO models is described in Section 2.2. For the structural part, a structure for the  $K$  skills should be defined. The simplest assumption is the independence model which assumes that the skills are independent of each other:

$$P(\boldsymbol{\alpha}_j) = P(\alpha_{j1}, \alpha_{j2}, \dots, \alpha_{jK}) = \prod_{k=1}^K P(\alpha_{jk}).$$

The independence model has been considered in many studies due to its simplicity but this assumption is usually implausible in practice. In some situations, it is likely that there exists a higher order structure among the skills. For such cases, de la Torre & Douglas (2004) proposed a ‘higher-order latent trait model’. The model assumes that the skills are specific knowledge related to one or more broadly defined constructs of general intelligence or aptitude. This idea is reflected by introducing an item response model at a higher order in which the latent skills  $(\alpha_{j1}, \alpha_{j2}, \dots, \alpha_{jK})$  play the role of items and are locally independent given the general aptitude, which is represented by  $\theta_j$ .

Assuming that mastery of a set of skills of respondent  $j$  is related to an unidimensional trait  $\theta_j$ , the probability model for  $\boldsymbol{\alpha}_j$  conditional on  $\theta_j$  is

$$P(\boldsymbol{\alpha}_j | \theta_j) = \prod_{k=1}^K P(\alpha_{jk} | \theta_j),$$

where

$$P(\alpha_{jk} = 1 | \theta_j) = \frac{\exp(\lambda_{0k} + \lambda_{1k}\theta_j)}{1 + \exp(\lambda_{0k} + \lambda_{1k}\theta_j)}. \quad (3.1)$$

Here,  $\lambda_{0k}$  and  $\lambda_{1k}$  are higher order structural parameters resembling item “easiness” and discrimination parameters in IRT. Then, the marginal probability of the observed responses of respondent  $j$  can be represented as:

$$P(\mathbf{Y}_j = \mathbf{y}_j) = \int_{\theta_j} \left\{ \prod_{k=1}^K P(\alpha_{jk} | \theta_j) \prod_{i=1}^I P(Y_{ij} = y_{ij} | \boldsymbol{\alpha}_j) \right\} P(\theta_j) d\theta_j.$$

In many applications,  $\theta_j$  is assumed to be normally distributed with mean 0 and variance 1. Sometimes two-dimensional higher-order latent traits are used, but rarely higher-dimensional

traits because, in general, the number of skills is much less than the number of items and needs to be much greater than the dimension of  $\boldsymbol{\theta}_j$  for identifiability. When the measurement part is defined as DINA, the model is called the higher-order DINA (HO-DINA) model.

## Growth Curve Cognitive Diagnosis Models

Our proposed ‘growth curve cognitive diagnosis model (GC-CDM)’ extends the higher-order latent trait model to incorporate learning over time into cognitive diagnosis models. Assuming a respondent-specific linear relationships between time and the higher-order latent trait, respondent  $j$ ’s changing latent trait over time can be modeled as latent growth curve model with respondent-specific random intercept and random slope of time. A similar approach to extend the HO-DINA model for longitudinal data was suggested by Ayers & Rabe-Hesketh (2011).

We consider a unidimensional latent trait for person  $j$  at occasion  $t$ ,  $\theta_{jt}$ , which is modeled as:

$$\theta_{jt} = \zeta_{1j} + (\beta + \zeta_{2j}) \times \text{time}_{jt} + \epsilon_{jt}, \quad (3.2)$$

where  $\text{time}_{jt}$  is the time associated with occasion  $t$  for respondent  $j$ ,  $\beta$  is the mean slope of time,  $\zeta_{1j}$  and  $\zeta_{2j}$  are the random intercept and random slope of time for person  $j$  respectively, and  $\epsilon_{jt}$  is the time-specific error. It is assumed that

$$\begin{pmatrix} \zeta_{1j} \\ \zeta_{2j} \end{pmatrix} \sim N \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \psi_{11} & \psi_{12} \\ \psi_{21} & \psi_{22} \end{pmatrix} \right], \quad \epsilon_{jt} \sim N(0, \sigma_t^2) \quad (3.3)$$

We can assume constant variance over time for  $\epsilon_{jt}$ , i.e.,  $\sigma_t^2 = \sigma^2$ . Given  $\theta_{jt}$ , the probability of skill profile  $\boldsymbol{\alpha}_{jt}$  of person  $j$  at occasion  $t$ , is

$$P(\boldsymbol{\alpha}_{jt} | \theta_{jt}) = \prod_{k=1}^K P(\alpha_{jkt} | \theta_{jt}),$$

where

$$P(\alpha_{jkt} = 1 | \theta_{jt}) = \frac{\exp(\lambda_{0k} + \lambda_{1k}\theta_{jt})}{1 + \exp(\lambda_{0k} + \lambda_{1k}\theta_{jt})}. \quad (3.4)$$

The higher-order structure parameters,  $\lambda_{0k}$  and  $\lambda_{1k}$ , are constant over time.

For the measurement part, respondent  $j$ ’s response  $Y_{ijt}$  to item  $i$  at time  $t$ , given his or her skill profile and the higher-order trait at time  $t$ , can be modeled with the DINA model:

$$\pi_{ijt} = P(Y_{ijt} = 1 | \boldsymbol{\alpha}_{jt}, \theta_{jt}) = (1 - s_{it})^{\xi_{ijt}} g_{it}^{1 - \xi_{ijt}}, \quad (3.5)$$

where  $\xi_{ijt}$  is the indicator whether respondent  $j$  possesses all skills required by item  $i$  of the assessment administered at occasion  $t$ , and  $s_{it}$  and  $g_{it}$  are slipping and guessing parameter of

item  $i$  at occasion  $t$ , respectively. Different sets of items can be tested at different occasions or the same items can be repeated at each occasion. For the latter, the number of items,  $I_t$ , are constant across occasions,  $I_t = I$ , and the guessing and slipping parameters of items are constant over time.

Figure 3.1 presents a GC-CDM for four skills and four time points. A set of items ( $I_1$  items at occasion 1,  $I_2$  items at occasion 2, ...) are administered at each time point. In the diagram, ovals indicate latent variables and rectangles are observed variables. The higher-order latent trait  $\theta_{jt}$  and skill profile  $(\alpha_{j1t}, \alpha_{j2t}, \dots, \alpha_{jKt})$  are measured at each of the four time points. The edges from  $\alpha_{jkt}$  to  $Y_{ijt}$  represent elements of the Q matrix that are equal to 1. For example, the edge between  $\alpha_{j11}$  and  $Y_{1j1}$  means Item 1 at occasion 1 requires mastery of skill 1. The directed edges connecting latent variables indicate regression relationships; logit models from  $\theta_{jt}$  to  $\alpha_{jkt}$  and linear models for the other edges. The curved edge between  $\zeta_{1j}$  and  $\zeta_{2j}$  represents their covariance. The values associated with the edges between  $\zeta_{1j}$ ,  $\zeta_{2j}$  and  $\theta_{1j}, \dots, \theta_{4j}$  indicate factor loadings or regression coefficients. The short directed edges to  $\theta_{jt}$  indicate the time and respondent specific residuals  $\epsilon_{jt}$ .

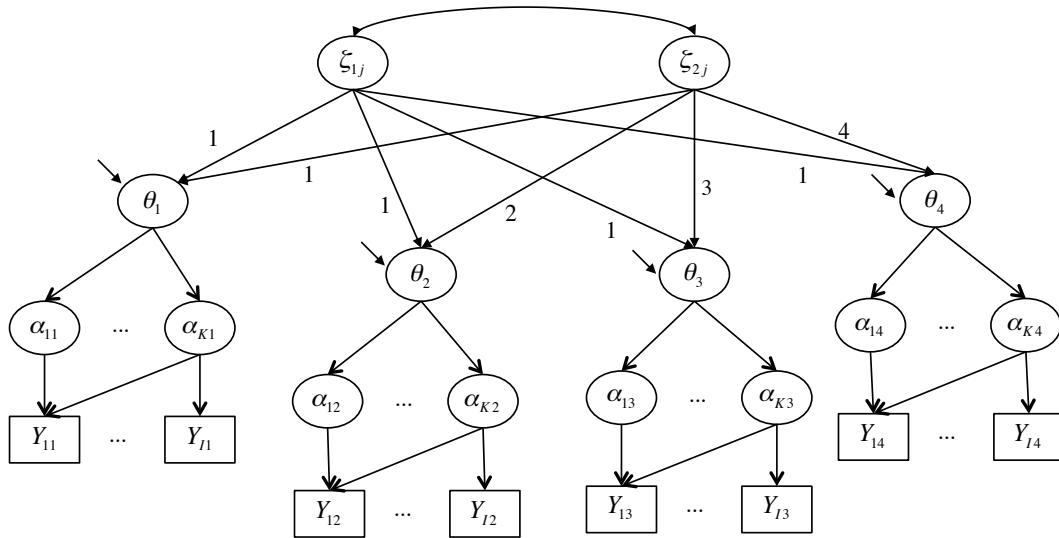


Figure 3.1: Growth curve cognitive diagnosis model (subscript  $j$  not shown)

## Possible Extensions

GC-CDMs can be extended in several ways to be more flexible. Based on (3.2) and (3.4) and assuming  $\lambda_{ik} = 1$  (the reason will be addressed in Section 3.2), the log-odds of person  $j$

having skill  $k$  at occasion  $t$  can be written as follows:

$$\begin{aligned}\text{logit}(P(\alpha_{jkt} = 1 | \theta_{jt})) &= \lambda_{0k} + \theta_{jt} \\ &= \lambda_{0k} + \zeta_{1j} + (\beta + \zeta_{2j}) \times \text{time}_{jt} + \epsilon_{jt}.\end{aligned}$$

Here, the model assumes that the relationship between  $\theta_{jt}$  and  $\alpha_{jkt}$  is constant over time and that changes in the log-odds of mastering skill  $k$  per unit of time, represented  $\beta + \zeta_{2j}$ , are also constant over time. If the higher-order latent trait changes over time, the log-odds of having each skill  $k$  changes by the same amount. However, the log-odds may change differently for different skills. For example, skill 1 may be learned faster than skill 2. GC-CDMs can be extended to allow different amounts of learning for each skill and each occasion by defining a skill and time specific intercept  $\lambda_{0kt}$  instead of  $\lambda_{0k}$ .  $\lambda_{0kt}$  can be reparametrized as  $\lambda_{0kt} = \lambda_{0k} + \delta_{kt}$ , where  $\delta_{kt}$  indicates time-specific deviation from the overall knowledge of skill  $k$ . Some constraints would be necessary for the model to be identified. A natural approach would be to test for the necessity of including  $\delta_{kt}$  for a given skill one skill at a time - similar to testing for differential item functioning in IRT. Alternatively, we can define skill-specific linear growth with coefficients  $\beta_k$  and remove  $\beta$  from the model.

In addition, GC-CDMs can be extended to incorporate respondents' covariate information to account for the effect of covariates on skill mastery. Ayers et al. (2013) proposed a method to incorporate covariates into the DINA model, and the same approach can be applied to GC-CDMs. In GC-CDMs, the log-odds of person  $j$  having skill  $k$  at time  $t$  can be defined as follows:  $\text{logit}(P(\alpha_{jkt} = 1 | \theta_{jt})) = \lambda_{0k} + \theta_{jt} + \mathbf{x}'_j \boldsymbol{\gamma}$ , where  $\mathbf{x}_j$  is a vector of covariates for respondent  $j$  and  $\boldsymbol{\gamma}$  is a vector of coefficients. The term  $\mathbf{x}'_j \boldsymbol{\gamma}$  can alternatively be thought of as part of  $\theta_{jt}$  by adding this term to (3.2). The interpretation becomes that the mean higher-order latent trait depends on covariates. By including interactions between covariates and time, the mean growth in the higher-order trait can depend on covariates. The model can be further extended by including interactions between skills and covariates, representing "differential skill functioning".

Although several extended GC-CDMs are available, the dissertation focuses on basic GC-CDMs.

## 3.2 Estimation

### Maximum Marginal Likelihood Estimation

The marginal likelihood of the GC-CDM can be expressed as:

$$L(\mathbf{y}) = \prod_{j=1}^J \int_{\boldsymbol{\theta}_j} \left\{ \prod_{t=1}^T \left( P(\boldsymbol{\alpha}_{jt} | \theta_{jt}) \prod_{i=1}^I P(y_{ijt} | \boldsymbol{\alpha}_{jt}) \right) \right\} P(\boldsymbol{\theta}_j) d\boldsymbol{\theta}_j, \quad (3.6)$$

where  $\boldsymbol{\theta}_j = (\zeta_{1j}, \zeta_{2j}, \epsilon_{j1}, \dots, \epsilon_{jT})^T$  and  $P(\boldsymbol{\theta}_j) = P(\zeta_{1j}, \zeta_{2j}) \prod_{t=1}^T P(\epsilon_{jt})$ , a product of one bivariate and  $T$  univariate densities as defined in (3.2). Evaluation of this likelihood requires  $(T + 2)$ -dimensional integration. The integral can be evaluated by numerical integration techniques (e.g., Gaussian quadrature), but an issue arises when the number of time point increases; the computational complexity increases exponentially.

As an alternative, we consider a dimension reduction technique using nested integration based on the fact that the  $\epsilon_{jt}$  are independent across occasions. Then, the marginal likelihood can be reexpressed as:

$$L(\mathbf{y}) = \prod_{j=1}^J \int_{\zeta_{1j}, \zeta_{2j}} \left\{ \prod_{t=1}^T \int_{\epsilon_{jt}} \left( P(\boldsymbol{\alpha}_{jt} | \zeta_{1j}, \zeta_{2j}, \epsilon_{jt}) \prod_{i=1}^I P(y_{ijt} | \boldsymbol{\alpha}_{jt}) \right) P(\epsilon_{jt}) d\epsilon_{jt} \right\} P(\zeta_{1j}, \zeta_{2j}) d\zeta_{1j} d\zeta_{2j}, \quad (3.7)$$

This approach provides significant computational savings as the MLE requires only 3-dimensional integration regardless of the number of time points. Such nested integration was discussed in (Gibbons & Hedeker, 1992; Rijmen, 2009; Cai, 2010) for bifactor-type item response models and (Jeon & Rabe-Hesketh, 2016) for an autoregressive growth IRT model. This likelihood has the structure of the likelihood for a multilevel model in which occasions are nested within persons (occasions are units and persons are clusters). In contrast, the likelihood in (3.6) treats persons and the  $T$  occasions as being on the same level (multivariate perspective).

The marginal likelihood is maximized using the Expectation Maximization (EM) algorithm (Dempster et al., 1977). Details about the EM algorithm for DINA model estimation are discussed in de la Torre (2009). The Mplus software (version 8) (Muthén & Muthén, 2017) was used to implement estimation of GC-CDMs.

## Estimation of the Higher-order Structural Parameters

Prior to implementation of the GC-CDM, we developed Mplus codes for the HO-DINA model for a single occasion and applied it to a simulated data to confirm its feasibility. For the simulation, responses of 1,000 respondents to 20 items were generated. The slipping and guessing parameter values were randomly generated from a uniform distribution from 0.1 to 0.3. The Q-matrix in Table 3.1 was used.

The IRT model for the higher-order joint skill distribution defined in (3.1) can be specified as a two-parameter logistic (2PL) or a Rasch model. For the 2PL model, the following values were used for the higher-order structural parameters:  $\boldsymbol{\lambda}_0 = (\lambda_{01}, \lambda_{02}, \lambda_{03}, \lambda_{04})' = (1.51, -1.42, -0.66, 0.50)'$  and  $\boldsymbol{\lambda}_1 = (\lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14})' = (1.34, 0.65, 1.11, 0.97)'$ . For the Rasch model, the slopes  $\boldsymbol{\lambda}_1$  were fixed to 1. The higher-order latent traits were assumed to be normally distributed with mean 0 and variance 1. The R codes for generating data are in Appendix.

Table 3.1: Q-matrix of 20 items and 4 skills for the HO-DINA model simulation study.

Skill	Item																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
2	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
3	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1

Mplus codes for HO-DINA models with different higher-order IRT models were developed and are presented in Appendix. The HO-DINA model is also available using the `GDINA()` function in ‘GDINA’ R package (Ma & de la Torre, 2017). The GDINA function includes an additional structural model called “1PL” which estimates a common slope parameter. Thus, the three models (Rasch, 1PL and 2PL) were considered for the simulation study and the parameter estimates from Mplus and GDINA were compared.

The Rasch model can be defined as

$$P(\alpha_{jk} = 1 | \theta_j) = \frac{\exp(\lambda_{0k} + \theta_j)}{1 + \exp(\lambda_{0k} + \theta_j)}, \quad \theta_j \sim N(0, \sigma_\theta^2). \quad (3.8)$$

Note that the GDINA function with the Rasch model does not freely estimate the variance of the latent trait,  $\sigma_\theta^2$ , while we estimate the variance in Mplus. Thus, the Rasch models from GDINA and Mplus cannot be compared directly. The Rasch model from Mplus can rather be compared with the 1PL model from GDINA because the 1PL model can be written as

$$P(\alpha_{jk} = 1 | \theta_j) = \frac{\exp(\lambda_{0k} + \lambda_1 \theta_j)}{1 + \exp(\lambda_{0k} + \lambda_1 \theta_j)}, \quad \theta_j \sim N(0, 1). \quad (3.9)$$

The models in (3.8) and (3.9) are equivalent and merely use different parameterizations. Let  $\eta_j = \lambda_1 \theta_j \sim N(0, \lambda_1^2)$  in (3.9) then  $\eta_j$  corresponds to  $\theta_j$  in (3.8) so  $\lambda_1^2$  should be identical to  $\sigma_\theta^2$ .

The parameters were estimated using MLE with 20 integration points for both Mplus and GDINA, and the convergence criterion of  $10^{-7}$  was used for the absolute change in log-likelihood in Mplus and for the absolute change in deviance in GDINA. The estimates are reported in Table 3.2. The two Rasch models from Mplus and GDINA are different models and are therefore not compared. The Rasch model is equivalent to the 1PL model so the results show the same log-likelihood of -11307.61 from the Mplus Rasch, the Mplus 1PL and the GDINA 1PL models. Focusing on the Mplus results first, the common slope,  $\lambda_1$ ,

Table 3.2: Estimated parameter values from Mplus and GDINA R package.

True	Mplus			GDINA		
	Rasch	1PL	2PL	Rasch	1PL	2PL
$\lambda_{01}$	1.51	1.44 (0.12)	1.44 (0.12)	1.28 (0.12)	1.49 (0.10)	1.43 (N/A) <sup>b</sup>
$\lambda_{02}$	-1.42	-1.40 (0.10)	-1.40 (0.10)	-1.41 (0.10)	-1.45 (0.09)	-1.40 (N/A)
$\lambda_{03}$	-0.66	-0.63 (0.10)	-0.63 (0.10)	-0.65 (0.13)	-0.65 (0.09)	-0.63 (N/A)
$\lambda_{04}$	0.50	0.56 (0.10)	0.56 (0.10)	0.62 (0.14)	0.58 (0.09)	0.56 (N/A)
$\lambda_1$	1	-	0.86 (0.09)	-	-	0.87 (N/A)
$\lambda_{11}$	1.34	-	-	0.62 (0.19)	-	-
$\lambda_{12}$	0.65	-	-	0.33 (0.17)	-	-
$\lambda_{13}$	1.11	-	-	1.21 (0.41)	-	-
$\lambda_{14}$	0.97	-	-	1.21 (0.39)	-	-
$\sigma_\theta^2$	1	0.74 (0.15)	-	-	-	-
LL <sup>a</sup>		-11307.61	-11307.61	-11289.25	-11308.81	-11307.61
						-11289.25

Note: Standard errors in parentheses

<sup>a</sup> LL indicates Log-likelihood

<sup>b</sup> Standard errors of 1PL model parameters from GDINA were not available.

was estimated as 0.86 (0.863 before rounding) and this is identical to the square root of the estimated variance of the latent trait from the Rasch model,  $\sqrt{0.74} = 0.86$ . The estimated structural parameters are also identical between the Rasch and the 1PL models and the estimates are sufficiently close to the generating values, taking into account the estimated standard errors. The 1PL model estimates are similar between Mplus and GDINA. But the GDINA could not produce the estimated standard errors due to singularity of the information matrix. For the 2PL model, Mplus and GDINA show the same log-likelihood of -11289.25 and similar estimates although the estimates are less similar for  $\lambda_{13}$  and  $\lambda_{14}$ . Unfortunately, the slope parameter estimates are a bit off from the true values—the estimate of  $\lambda_{11}$  differs from its generating value by more than three standard errors and the estimate of  $\lambda_{12}$  differs from its generating value by nearly two standard errors.

The guessing and slipping parameter estimates were also compared. Figure 3.2 and 3.3 compare the estimates in the Rasch/1PL model and the 2PL model respectively. The two graphs in the first row show good parameter recovery of the Mplus estimates—all 95% confidence intervals contain the generating value. The graphs in the second row show that the Mplus estimates and the GDINA estimates are identical (they differ by less than 0.0005).

From the simulation study, we confirmed that the Mplus code was correctly developed by comparing estimates with those from the existing GDINA R package. Also, we found that the 2PL model was not successfully estimated by either Mplus or GDINA. Thus, we decided to use the Rasch model for the GC-CDM, and this model will be used for the simulation

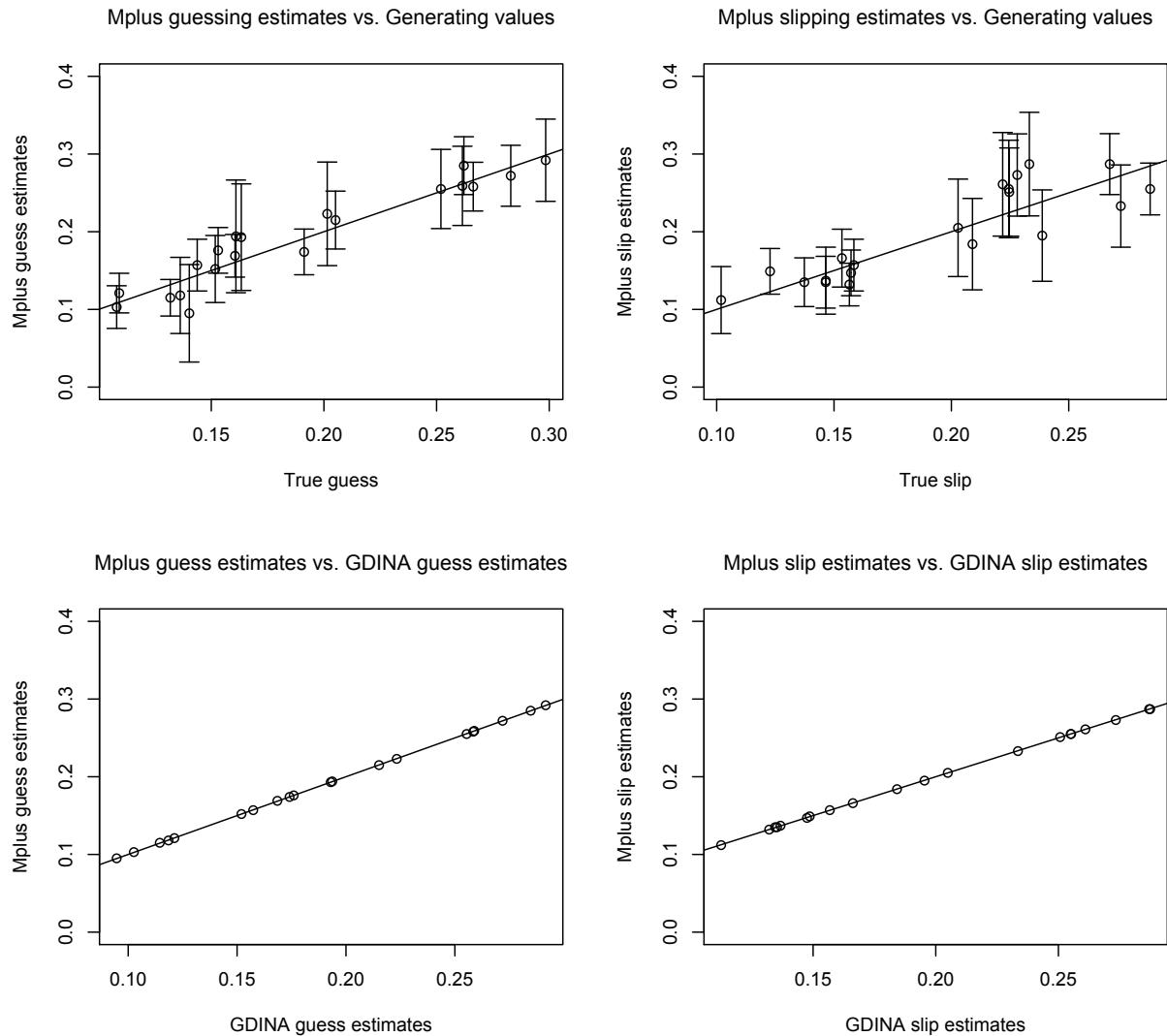


Figure 3.2: Scatter plots of item parameter estimates : (1) Mplus Rasch/1PL estimates vs. generating values and (2) Mplus Rasch/1PL vs. GDINA 1PL.

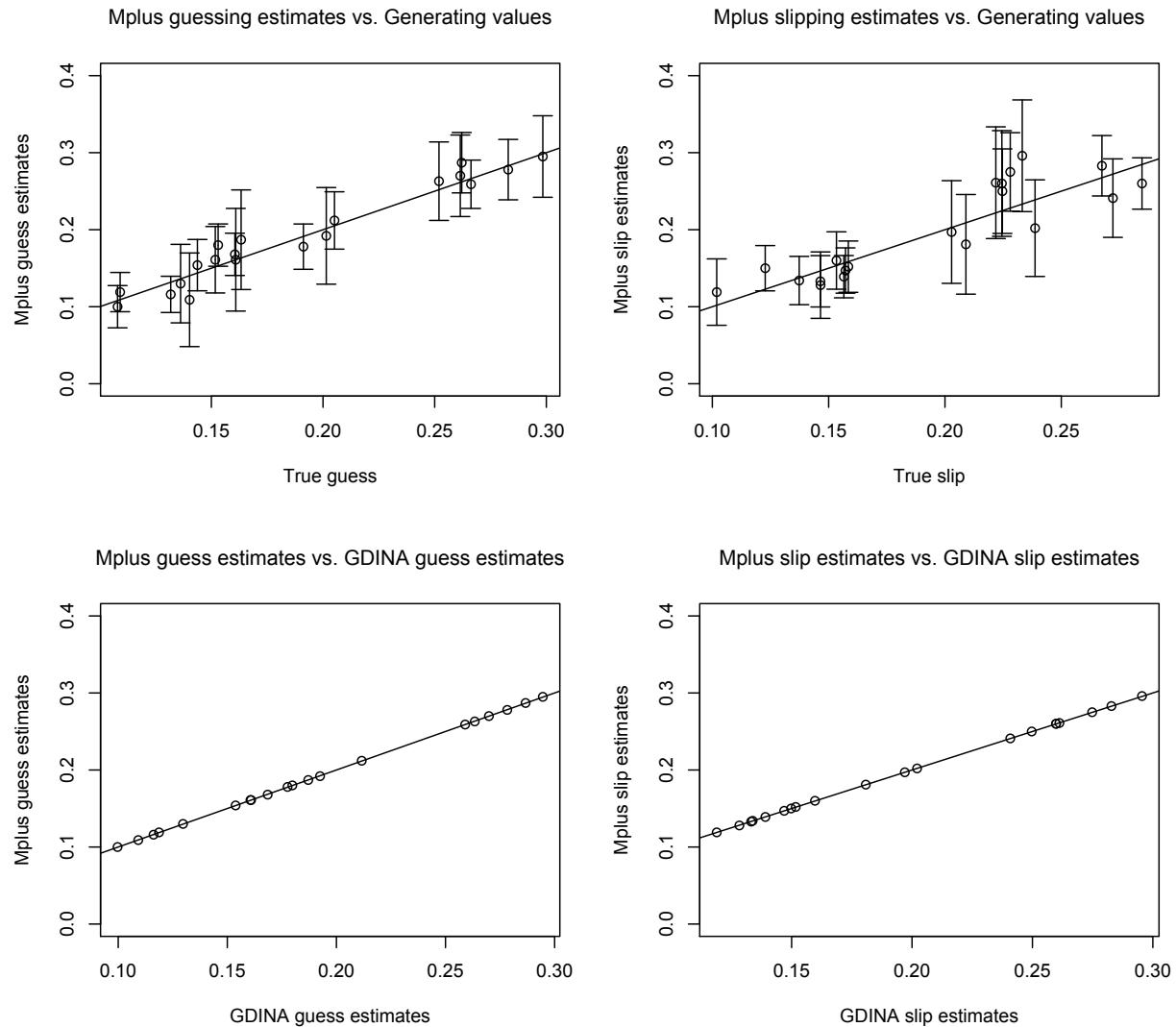


Figure 3.3: Scatter plots of item parameter estimates : (1) Mplus 2PL estimates vs. generating values and (2) Mplus 2PL vs. GDINA 2PL.

studies and empirical studies in the next chapters.

### 3.3 Model when the Number of Occasions is Two

When only two time points are available ( $T = 2$ ) and the timing is identical across subjects,  $\text{time}_{jt} = \text{time}_t$ , the growth curve model in (3.2) is not identified. For such case, we let the higher-order latent traits at occasion 1 and occasion 2 be correlated with each other rather than specifying a linear growth curve model. The correlation between  $\theta_{j1}$  and  $\theta_{j2}$  is modeled by using a bivariate normal distribution:

$$\begin{pmatrix} \theta_{j1} \\ \theta_{j2} \end{pmatrix} \sim N \left[ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix} \right],$$

where  $(\mu_1, \mu_2)'$  is the mean vector of the higher-order latent traits;  $\sigma_1^2$  and  $\sigma_2^2$  are the variances of  $\theta_{j1}$  and  $\theta_{j2}$  respectively; and  $\sigma_{12}$  ( $=\sigma_{21}$ ) is the covariance between  $\theta_{j1}$  and  $\theta_{j2}$ .

For the higher-order IRT model, we used the Rasch model:

$$P(\alpha_{jkt} = 1 | \theta_{jt}) = \frac{\exp(\lambda_{0k} + \theta_{jt})}{1 + \exp(\lambda_{0k} + \theta_{jt})},$$

where  $\lambda_{0k}$  is constant over time.

This approach is similar to Andersen's longitudinal Rasch model described in Section 2.1 in that both models use a bivariate normal distribution to allow correlation between latent traits over time. To measure individual growth between two occasions, we can calculate the difference between predicted latent traits at each time point.

# Chapter 4

## Simulation Study

Two simulation studies were conducted to investigate how well the parameters of the growth curve DINA (GC-DINA) model can be recovered by the estimation method described in Chapter 3. The first simulation study focuses on parameter recovery under differing conditions—three factors were manipulated: the number of respondents, the design of the Q-matrix and the number of time points. The second simulation study focuses on the analysis of pre-post assessment data when only two time points are available.

### 4.1 Simulation Study 1

#### Simulation Conditions

In this section, we examine parameter recovery of the GC-DINA using maximum marginal likelihood estimation with the nested integration described in Section 3.2. Parameter recovery was investigated under various conditions. In particular, we manipulated three factors: the number of respondents, the design of the Q-matrix and the number of time points. For CDMs, in general, it has been recognized that the design of the Q-matrix is an important factor in model estimation. We especially considered the complexity of the Q-matrix which depends on the number of items measuring each skill, the number of skills each item measures and the number of skills that are measured jointly with other skills. Madison & Bradshaw (2015) demonstrated the effect of the Q-matrix design on classification accuracy for the log-linear cognitive diagnosis model. Their study showed that classification accuracy for a given skill increases as the number of items measuring that skill only increases, whereas the accuracy deteriorates when the item measuring that skill tend to also measure other skills in conjunction.

Estimation time for the GC-DINA model is considerable, so it was not feasible to conduct

a full factorial design with multiple replicates per combination of conditions. By varying the number of respondents, the design of the Q-matrix and the number of time points, we considered the following six models:

- Model 1: 1,000 respondents & Simple Q-matrix & Three time points
- Model 2: 1,000 respondents & Complex Q-matrix & Three time points
- Model 3: 500 respondents & Simple Q-matrix & Three time points
- Model 4: 500 respondents & Complex Q-matrix & Three time points
- Model 5: 500 respondents & Simple Q-matrix & Four time points
- Model 6: 500 respondents & Complex Q-matrix & Four time points

We chose these models so that we can investigate the effect of changing one factor at a time. We first considered Model 1 and Model 2 to examine the effect of the design of the Q-matrix. Next, we decreased the number of respondents to examine the effect of sample size (Model 3 and Model 4). We then increased the number of time points to investigate the effect of number of time points (Model 5 and Model 6).

## Simple Q-matrix vs. Complex Q-matrix

Table 4.1 and 4.2 present the simple and complex Q-matrix used for the simulation, respectively. In the simple Q-matrix, each item measures only one skill. The number of items that measure each skill is as follows: 4 items for Skill 1, 6 items for Skill 2, 5 items for Skill 3, and 5 items for Skill 4. In the complex Q-matrix, 7 items out of 20 measure a single skill: item 1, 2, 17, 18 and 19 measure Skill 1 in isolation; item 9 measures only Skill 2; and item 13 measures only Skill 3. There is no item that measures only Skill 4. 10 items measure two skills jointly, and 3 items measure all four skills. In terms of the number of items that measure each skill, the skills (Skill 1 - Skill 4) are measured by 16 items, 12 items, 5 items and 6 items respectively. Both the simple Q-matrix and complex Q-matrix were derived from the real application of Enhanced Anchored Instruction (EAII; Bottge et al., 2003). Thus, these Q-matrices reflect practical cognitive diagnostic assessment. For the simple Q-matrix, we used the Q-matrix developed for the Problem Solving Test (PST) (Bottge et al., 2014, 2015). The complex Q-matrix was derived from the Q-matrix developed for the Fraction of the Cost (FOC) test (Bottge et al., 2007; Li et al., 2016). The FOC test has 23 items and 4 skills. We deleted three items that measure all of the four skills from the original Q-matrix, so the Q-matrix has 20 items (to be comparable to the simple Q-matrix). By taking three items out, estimation becomes more challenging because there is less information about the latent

skills. Real data with the original versions of the Q-matrices will be used in the empirical study and more details about the Q-matrices will be discussed in Chapter 5.

Table 4.1: Simple Q-matrix of 20 items and 4 skills for the simulation study.

Skill	Item																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
2	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
3	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1

Table 4.2: Complex Q-matrix of 20 items and 4 skills for the simulation study.

Skill	Item																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1	1	1	1
2	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0

Table 4.3: Classification accuracy with the simple and complex Q-matrices for the simple DINA model.

Q-matrix structure	ARI	PCS	PCV			
			(1st,	2st,	3st,	4st)
Simple	0.65	0.95	(1,	1,	0.99,	0.80)
Complex	0.32	0.85	(1,	0.99,	0.90,	0.53)

To confirm the effect of the complexity of the Q-matrix on classification accuracy, we conducted another simulation to examine how the classification accuracy changes with the simple Q-matrix and complex Q-matrix. For each of the two Q-matrices, we generated a dataset for 1,000 persons from the simple DINA model (for a single time point) assuming that the four skills are independent of each other. The guessing and slipping parameters were generated from a uniform distribution from 0.1 to 0.3 and were kept the same for both Q matrices. For each model, we calculated the classification accuracy by using the following measures: (1) Adjusted Rand Index (ARI; Hubert & Arabie, 1985), (2) the proportion of correctly classified skills (PCS) and (3) the proportions of correctly classified skill vectors

(PCV). The ARI has been used as a common measure of agreement between two partitions, and it is defined as the number of agreement between two partitions divided by the total number of pairs of objects. The ARI lies between 0 and 1, where 1 indicates perfect agreement. The PCS is the skill level correct classification accuracy rate, while the PCV is the vector level correct classification rate. The PCV is a vector with four elements—the first element is the proportion of at least one skill in the vector being correctly classified; the second element is the proportion of at least two skills being correctly classified; the third element is the proportion of at least three skills being correctly classified; and the last element is the proportion of all elements in the vector being correctly classified. The calculation of the ARI is available using `adjustedRandIndex()` function in the ‘mclust’ R package. The PCS and PCV can be obtained from the `ClassRate()` function in the ‘GDINA’ R package. The R code for generating two simulated data sets from the DINA model and calculating the accuracy measures are in Appendix.

Table 4.3 shows the calculated classification accuracy rates. The AIRs were 0.65 for the simple Q-matrix and 0.32 for the complex Q-matrix. The PCSs were 0.95 for the simple Q-matrix and 0.85 for the complex Q-matrix. In the PCV elements, a large difference was observed in the 4th element which indicates the proportion of correctly identified skill mastery profiles across all the four skills: 0.80 for the simple Q-matrix and 0.53 for the complex matrix. This suggests that the prediction of skill mastery profiles suffers as the complexity of the Q-matrix increases in the independence DINA model.

## Data Generation & Analysis

For all of the six models, we generated response data for 20 items and 4 skills. The guessing and slipping parameters,  $g_i$ ,  $s_i$  ( $i = 1, \dots, 20$ ), were randomly generated from a uniform distribution from 0.1 to 0.3, and were constant over time. The same set of guessing and slipping parameters were used across the six models. To simulate the higher-order latent trait, we used the following generating values: the variance of the random intercept  $\psi_{11} = 0.4$ , the variance of the random slope of time  $\psi_{22} = 0.02$ , the covariance between the random intercept and random slope  $\psi_{12} = \psi_{21} = 0.02$ , the average growth  $\beta = 0.3$ , and the variance of the occasion-specific error  $\sigma^2 = 0.6$ . For the higher-order structural parameters, we used  $\lambda_0 = (\lambda_{01}, \lambda_{02}, \lambda_{03}, \lambda_{04}) = (1.51, -1.42, -0.66, 0.50)$ . We used the same generated values for the higher-order latent traits and skill mastery indicators between Model 1 and 2, between Model 3 and 4, and between Model 5 and 6 by using the same seed number. Then, the comparison of models with different Q-matrices can be more legitimate as we rule out the chance error that occurs in generating latent variables. Time was coded as 0,1,2 for three occasions and 0,1,2,3 for four occasions. The R codes generating six data sets are available in Appendix.

The six GC-DINA models were fitted in Mplus. As discussed in Section 3.2, when using the nested integration approach, the likelihood has the structure of the multilevel model—the occasions are nested within respondents. We implemented the GC-DINA model in Mplus using the multilevel approach. Note that the multivariate approach based on (3.6) was not feasible in Mplus due to high dimensions—the evaluation of the likelihood requires  $(T+2)$ -dimensional integration. We used 15 integration points and the convergence criterion of  $10^{-7}$  for the absolute change in the log-likelihood. The Mplus codes for the GC-DINA models are in Appendix.

In addition to comparing GC-DINA estimates with generating values, we compared them with estimates that would be obtained if latent variables in the model could be observed directly. A best-case scenario for estimating the covariance matrix of the random intercept and slope would be that the random intercepts and slopes,  $\zeta_{1j}$  and  $\zeta_{2j}$ , are observed. This was the first comparison. Moving one level down in the model, if the occasion-specific latent traits,  $\theta_{jt}$ , are observed, we could estimate a linear growth curve model; the model can be defined as the same as in (3.2) but is fitted to the observed  $\theta_{jt}$ . If we don't observe the higher-order latent traits but observe the skill mastery indicators, moving down one more level, then we could estimate the mixed-effects logistic model. The model has a three-level structure, where the skill mastery indicators (level 1) are nested within occasions (level 2) and occasions are nested within persons (level 3). The model can be expressed as:

$$\begin{aligned} \text{logit}(P(\alpha_{jkt} = 1 | s_{1k}, s_{2k}, s_{3k}, s_{4k}, \text{time}_t, \zeta_{1j}, \zeta_{2j}, \epsilon_{jt})) \\ = \lambda_{01}s_{1k} + \lambda_{02}s_{2k} + \lambda_{03}s_{3k} + \lambda_{04}s_{4k} + (\beta + \zeta_{2j})\text{time}_t + \zeta_{1j} + \epsilon_{jt}, \end{aligned} \quad (4.1)$$

where  $s_{1k}, s_{2k}, s_{3k}, s_{4k}$  are dummy variables for  $\alpha_{jkt}$  representing mastery of Skill 1, Skill 2, Skill 3 and Skill 4, respectively (i.e.,  $s_{1k} = 1$  when  $k = 1$  and 0 otherwise.);  $\text{time}_t$  is the time associated with occasion  $t$ ;  $\zeta_{1j}$  and  $\zeta_{2j}$  are the random intercept and random slope of time for person  $j$  respectively; and  $\epsilon_{jt}$  is the occasion-specific random intercept. It is assumed that

$$\begin{pmatrix} \zeta_{1j} \\ \zeta_{2j} \end{pmatrix} \sim N \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \psi_{11} & \psi_{12} \\ \psi_{21} & \psi_{22} \end{pmatrix} \right], \quad \epsilon_{jt} \sim N(0, \sigma^2). \quad (4.2)$$

The scenarios with observed latent variables—(1) the covariance matrix calculated from observed random effects,  $\zeta_{1j}, \zeta_{2j}$ , (2) the growth curve model with observed higher-order latent traits,  $\theta_{jt}$ , and (3) the mixed-effects logistic model with observed skill mastery indicators,  $\alpha_{jkt}$ —should be better cases than the GC-DINA model with observed item responses,  $Y_{ijt}$ , thus the GC-DINA model estimates are not expected to be better than the estimates obtained from the scenarios (1) to (3).

We investigated how the GC-DINA estimates deteriorate due to having only the indirect information about the skill and the higher-order latent traits in terms of point estimates

and the standard errors. To compare the standard errors with less rounding (the estimated standard errors will be reported after being rounded to 2 decimal points), we calculated the square root of relative efficiency (= the ratio between standard errors) to examine how much standard errors increase in the GC-DINA model compared to the growth curve model and the mixed effects logistic model.

## Results

### The Effect of Design of the Q-matrix

We first focus on comparison between Model 1 and Model 2 to examine how the estimates change depending on the complexity of the Q-matrix. The estimated parameters for Model 1 and Model 2 are shown in Table 4.4. In Model 1, the estimates for the GC-DINA model successfully recovered the generating values—the difference is less than one standard error except for  $\lambda_{04}$  (the difference for  $\lambda_{04}$  is only slightly greater than one standard error). Also, the point estimates of the GC-DINA model were not much worse than the estimates from the observed random effects ('Observed'), the growth curve model ('Growth Curve') and the mixed-effects logistic model ('Mixed Logit').

In Model 2, the point estimates of the GC-DINA model, however, deteriorated for some parameters. We can consider this deterioration in the point estimates as the effect of the complex Q-matrix since the generated latent variables were the same between Model 1 and Model 2 (accordingly, the estimates for other models were the same between Model 1 and Model 2). In particular, the point estimates of  $\psi_{11}$  and  $\psi_{12}$  became worse as  $\hat{\psi}_{11} = 0.32$  and  $\hat{\psi}_{12} = 0.08$  in Model 2 while the generating values were  $\psi_{11} = 0.40$  and  $\psi_{12} = 0.02$ . But interestingly, this was not reflected in the standard errors—the standard errors for  $\hat{\psi}_{11}$  and  $\hat{\psi}_{12}$  were nearly the same between Model 1 and Model 2. On the other hand, for  $\hat{\sigma}^2$ , the standard error increased from 0.10 (Model 1) to 0.18 (Model 2) while the point estimates remained the same. The inconsistency between the point estimates and the standard errors may have occurred because we used the estimated standard errors, not true sampling variances and there is a risk that this estimated standard errors computed by Mplus might not be good enough. For  $\lambda_{03}$  and  $\lambda_{04}$ , both of the point estimates and the standard errors became worse. In Model 1,  $\hat{\lambda}_{03} = -0.61$  with a standard error of 0.07 and  $\hat{\lambda}_{04} = 0.59$  with a standard error of 0.08. In Model 2, however, they became  $\hat{\lambda}_{03} = -0.78$  with a standard error of 0.15 and  $\hat{\lambda}_{04} = 0.66$  with a standard error of 0.17. This can be partly explained by the fact that the complex Q-matrix has no item that measures only Skill 4 and the smaller number of items that measure each of Skill 3 and Skill 4: Skill 3 is measured by 5 items and Skill 4 is measured by 6 items whereas Skill 1 and Skill 2 are measured by 16 items and 12 items, respectively.

Next, we compared Model 3 with Model 4 to investigate the effect of the Q-matrix

Table 4.4: Parameter estimates and standard errors (in parentheses) for Model 1 and Model 2

Model 1: 1,000 respondents & Simple Q-matrix & Three time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.43	0.42 (0.05)	0.41 (0.09)	0.39 (0.10)	2.14
$\psi_{22}$	0.02	0.02	0.04 (0.02)	0 (0.01)	0.01 (0.01)	0.64
$\psi_{12}$	0.02	0.02	0.01 (0.02)	0.04 (0.03)	0.04 (0.04)	1.55
$\beta$	0.30		0.30 (0.02)	0.29 (0.03)	0.29 (0.04)	2.28
$\sigma^2$	0.60		0.59 (0.03)	0.69 (0.08)	0.61 (0.10)	3.87
$\lambda_{01}$	1.51			1.60 (0.07)	1.58 (0.10)	1.49
$\lambda_{02}$	-1.42			-1.38 (0.06)	-1.40 (0.08)	1.17
$\lambda_{03}$	-0.66			-0.61 (0.06)	-0.61 (0.07)	1.15
$\lambda_{04}$	0.50			0.57 (0.06)	0.59 (0.08)	1.37

Model 2: 1,000 respondents & Complex Q-matrix & Three time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.43	0.42 (0.05)	0.41 (0.09)	0.32 (0.10)	2.06
$\psi_{22}$	0.02	0.02	0.04 (0.02)	0 (0.01)	0.02 (0.02)	0.84
$\psi_{12}$	0.02	0.02	0.01 (0.02)	0.04 (0.03)	0.08 (0.03)	1.21
$\beta$	0.30		0.30 (0.02)	0.29 (0.03)	0.30 (0.04)	2.66
$\sigma^2$	0.60		0.59 (0.03)	0.69 (0.08)	0.61 (0.18)	6.65
$\lambda_{01}$	1.51			1.60 (0.07)	1.54 (0.09)	1.29
$\lambda_{02}$	-1.42			-1.38 (0.06)	-1.61 (0.09)	1.39
$\lambda_{03}$	-0.66			-0.61 (0.06)	-0.78 (0.15)	2.60
$\lambda_{04}$	0.50			0.57 (0.06)	0.66 (0.17)	2.81

with a smaller sample size. The estimates for Model 3 and Model 4 are given in Table 4.5. Compared to Model 3, Model 4 produced somewhat degraded estimates for the higher-order structural parameters. Particularly, the point estimate  $\hat{\lambda}_{04} = 0.39$  in Model 4 while  $\hat{\lambda}_{04} = 0.51$  in Model 3 which is very close to the generating value. In addition, the point estimate of  $\sigma^2$  noticeably increased: 0.68 in Model 3 and 1.16 in Model 4. Note that Mplus could not successfully estimate the standard error of  $\psi_{22}$  in Model 4, Model 5 and Model 6, producing the warning message “*THE STANDARD ERRORS OF THE MODEL PARAMETER ESTIMATES MAY NOT BE TRUSTWORTHY FOR SOME PARAMETERS DUE TO A NON-POSITIVE DEFINITE FIRST-ORDER DERIVATIVE PRODUCT MATRIX.*” Although the first-order derivative product matrix is not the estimation method for the standard errors in Mplus (regular ML standard errors are used), we could not find an obvious reason for this warning message, so decided not to compare the estimated standard

errors for Model 4, Model 5 and Model 6 in detail.

Table 4.5: Parameter estimates and standard errors (in parentheses) for Model 3 and Model 4

Model 3: 500 respondents & Simple Q-matrix & Three time points							
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$	$\sqrt{\text{R.E. - ML}}$
$\psi_{11}$	0.40	0.38	0.41 (0.06)	0.41 (0.19)	0.28 (0.21)	3.33	1.12
$\psi_{22}$	0.02	0.02	0.06 (0.03)	0.10 (0.10)	0.05 (0.12)	4.18	1.12
$\psi_{12}$	0.02	0.01	-0.01 (0.03)	-0.01 (0.11)	0.07 (0.13)	3.85	1.10
$\beta$	0.30		0.27 (0.02)	0.27 (0.05)	0.32 (0.05)	2.39	1.10
$\sigma^2$	0.60		0.56 (0.04)	0.61 (0.15)	0.68 (0.19)	5.33	1.26
$\lambda_{01}$	1.51			1.59 (0.10)	1.42 (0.13)		1.40
$\lambda_{02}$	-1.42			-1.48 (0.09)	-1.52 (0.10)		1.15
$\lambda_{03}$	-0.66			-0.64 (0.08)	-0.67 (0.09)		1.13
$\lambda_{04}$	0.50			0.49 (0.08)	0.51 (0.12)		1.42

Model 4: 500 respondents & Complex Q-matrix & Three time points							
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$	$\sqrt{\text{R.E. - ML}}$
$\psi_{11}$	0.40	0.38	0.41 (0.06)	0.41 (0.19)	0.23 (0.12)	1.91	0.64
$\psi_{22}$	0.02	0.02	0.06 (0.03)	0.10 (0.10)	0.02 (0.02)	0.86	0.23
$\psi_{12}$	0.02	0.01	-0.01 (0.03)	-0.01 (0.11)	0.07 (0.04)	1.10	0.31
$\beta$	0.30		0.27 (0.02)	0.27 (0.05)	0.32 (0.07)	2.93	1.34
$\sigma^2$	0.60		0.56 (0.04)	0.61 (0.15)	1.16 (0.32)	9.16	2.16
$\lambda_{01}$	1.51			1.59 (0.10)	1.66 (0.13)		1.37
$\lambda_{02}$	-1.42			-1.48 (0.09)	-1.61 (0.13)		1.48
$\lambda_{03}$	-0.66			-0.64 (0.08)	-0.64 (0.25)		2.98
$\lambda_{04}$	0.50			0.49 (0.08)	0.39 (0.21)		2.58

In addition, we examined the effect of Q-matrix when four time points are available by comparing Model 5 and Model 6. The estimates for Model 5 and Model 6 are presented in Table 4.6. Interestingly, when the number of time points increased to four occasions, the effect of the complex Q-matrix was relatively small. The point estimate  $\hat{\psi}_{12} = 0.08$  in Model 6 differs somewhat from the generating value of 0.02, but it is difficult to say that this is due to the complex Q-matrix because the point estimate with the simple Q-matrix is also not close to the generating value,  $\hat{\psi}_{12} = 0.06$  in Model 5. The estimates for  $\lambda_{01}, \dots, \lambda_{04}$  became worse for Model 6 than for Model 5, but overall the estimates are not too bad compared to the generating values. As discussed above, we did not compare the estimated standard errors for Model 5 and Model 6.

Table 4.6: Parameter estimates and standard errors (in parentheses) for Model 5 and Model 6

Model 5: 500 respondents & Simple Q-matrix & Four time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.38	0.38 (0.05)	0.39 (0.15)	0.28 (0.10)	1.86
$\psi_{22}$	0.02	0.02	0.01 (0.01)	0.02 (0.04)	0.01 (0.01)	1.11
$\psi_{12}$	0.02	0.01	0.03 (0.02)	0.03 (0.06)	0.06 (0.02)	1.16
$\beta$	0.30		0.31 (0.01)	0.31 (0.03)	0.30 (0.04)	2.47
$\sigma^2$	0.60		0.61 (0.03)	0.77 (0.12)	0.90 (0.14)	5.03
$\lambda_{01}$	1.51			1.59 (0.09)	1.53 (0.12)	1.39
$\lambda_{02}$	-1.42			-1.46 (0.08)	-1.39 (0.09)	1.13
$\lambda_{03}$	-0.66			-0.68 (0.08)	-0.64 (0.09)	1.14
$\lambda_{04}$	0.50			0.52 (0.08)	0.61 (0.11)	1.40
Model 6: 500 respondents & Complex Q-matrix & Four time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.38	0.38 (0.05)	0.39 (0.15)	0.27 (0.10)	1.88
$\psi_{22}$	0.02	0.02	0.01 (0.01)	0.02 (0.04)	0.02 (0.02)	1.72
$\psi_{12}$	0.02	0.01	0.03 (0.02)	0.03 (0.06)	0.08 (0.02)	1.16
$\beta$	0.30		0.31 (0.01)	0.31 (0.03)	0.28 (0.04)	2.68
$\sigma^2$	0.60		0.61 (0.03)	0.77 (0.12)	0.61 (0.18)	6.63
$\lambda_{01}$	1.51			1.59 (0.09)	1.59 (0.11)	1.19
$\lambda_{02}$	-1.42			-1.46 (0.08)	-1.32 (0.10)	1.22
$\lambda_{03}$	-0.66			-0.68 (0.08)	-0.52 (0.17)	2.21
$\lambda_{04}$	0.50			0.52 (0.08)	0.46 (0.17)	2.13

The main object of the GC-DINA model is to capture learning over time, so the parameter of interest is the average growth,  $\beta$ . In all six models,  $\beta$  was successfully recovered by the GC-DINA model estimates. The generating value for the average growth was 0.3, and the GC-DINA estimates were 0.29, 0.30, 0.32, 0.32, 0.30 and 0.28 for Model 1 through Model 6, respectively.

The relative efficiencies between the GC-DINA model and the growth curve model and between the GC-DINA model and the mixed-effects logistic model indicate how the GC-DINA estimates were degraded compared to the estimates for the other two models due to not observing the latent variables but observing the item responses. Since item responses were generated based on the Q-matrix, we speculated that the relative efficiencies would be similar among the models as long as the Q-matrix is the same; thus, we expected similar relative efficiencies among Model 1, Model 3 and Model 5 (models with the simple Q-matrix)

and among Model 2, Model 4 and Model 6 (models with the complex Q-matrix).

To test our hypothesis, we compared the computed relative efficiencies although the estimated standard errors of the GC-DINA estimates were considered not trustworthy for Model 4, Model 5 and Model 6. We particularly focused on the relative efficiencies for  $\beta$  because it is the parameter of interest of the GC-DINA model, and at the same time, it was sufficiently recovered across models. The square root of relative efficiencies for  $\beta$  are summarized in Table 4.7. Note that the relative efficiencies were calculated using less rounded standard errors. As we expected, the squared root of relative efficiencies were relatively similar among models that use the same Q-matrix. For the models that used the simple Q-matrix, the square roots of relative efficiencies comparing the standard error of  $\hat{\beta}$  between the GC-DINA model and the growth curve model were 2.28 for Model 1, 2.39 for Model 3 and 2.47 for Model 5. When compared to the mixed-effects logistic model, the square root of relative efficiencies were even closer to each other: 1.08 for Model 1, 1.10 for Model 3 and 1.10 for Model 5. Also for the models that used the complex Q-matrix, they had similar relative efficiencies. The square root of relative efficiencies comparing with the growth curve model were 2.66 for Model 2, 2.93 for Model 4 and 2.68 for Model 6; the square root of relative efficiencies comparing to the mixed-effects logistic model were 1.25 for Model 2, 1.34 for Model 4 and 1.19 for Model 6. Also for each condition, the relative efficiencies increased when the complex Q-matrix was used. The change in the relative efficiencies between Model 2 and Model 1, between Model 4 and Model 3, and between Model 6 and Model 5 indicate how much the standard errors of  $\hat{\beta}$  increased by using the complex Q-matrix.

Overall, the GC-DINA model estimates showed good recovery; especially, the average growth  $\beta$  was successfully estimated regardless of the design of the Q-matrix. This result, however, is based on only one replication per model, thus this is a tentative conclusion.

Table 4.7: Square root of relative efficiencies for  $\beta$ : the simple Q-matrix vs. the complex Q-matrix.

Models with the simple Q-matrix		Models with the complex Q-matrix		
	$\sqrt{R.E. - GC}$	$\sqrt{R.E. - ML}$	$\sqrt{R.E. - GC}$	
Model 1	2.28	1.07	Model 2	2.66
Model 3	2.39	1.10	Model 4	2.93
Model 5	2.47	1.10	Model 6	2.68

### The Effect of Sample Size

We examined the effect of sample size by comparing Model 1 with Model 3. Model 1 has 1,000 respondents while Model 3 has 500 respondents. Both models used the simple Q-matrix

and three time points. Table 4.8 presents the estimates for Model 1 and Model 3.

Focusing on the GC-DINA estimates first, the point estimates deteriorated in Model 3, except for  $\beta$ , in terms of both the point estimates and the standard errors. The standard errors increased for all parameters, especially for the covariance matrix of the random intercepts and slopes. This degradation in parameter estimates was found not only for the GC-DINA model but also for the mixed-effects logistic model. To investigate the effect of sample size only applied to the GC-DINA model, excluding the change occurred in the mixed-effects logistic model, we compared the square root of relative efficiencies comparing the GC-DINA estimates and the mixed-effects logistic model estimates between Model 1 and Model 3. Based on the last column in Table 4.8, the relative efficiencies are not very different between Model 1 and Model 3 except for  $\psi_{22}$ .

The smaller sample size had effects on both the GC-DINA model and the mixed-effects logistic models. The deterioration in the estimates was not more severe for the GC-DINA model than for the logistic mixed-effects model. Although the point estimates got worse and the standard errors increased for Model 3, the parameter recovery of the GC-DINA estimates still seems fine as the difference is less than one standard error.

In addition, we also compared Model 2 with Model 4, to investigate the effect of sample size when the complex Q-matrix was used. Similarly to the comparison between Model and Model 3, the point estimates in Model 4 became worse for some parameters;  $\hat{\psi}_{11}$  decreased to 0.23 (Model 4) from 0.32 (Model 2) and  $\hat{\sigma}^2$  increased to 1.16 (Model 4) from 0.61 (Model 2). We did not further compare Model 2 and Model 4 as the standard errors for Model 4 were considered untrustworthy.

### The Effect of Number of Time Points

To investigate the GC-DINA estimates with increasing number of time points, we compared Model 3 and Model 5. Both models considered 500 respondents and the simple Q-matrix, but Model 3 has three time points while Model 5 has four time points. The estimates from Model 3 and Model 5 are given in Table 4.9.

Comparing Model 3 and Model 5, we did not detect huge difference in the point estimates of the GC-DINA model. One thing occurred to all of the growth curve model, the mixed-effects logistic model and the GC-DINA model is that the estimates for the covariance matrix improved in Model 5. Both the point estimates and standard errors improved for the covariance matrix for the models when four time points are available, but again, note that the standard errors of the GC-DINA estimates may not be reliable. In Model 5, unexpectedly, the GC-DINA point estimates of  $\sigma^2$  and  $\lambda_{04}$  became worse.  $\hat{\sigma}^2 = 0.9$  for Model 5 and  $\hat{\sigma}^2 = 0.68$  for Model 3;  $\hat{\lambda}_{04} = 0.61$  for Model 5 and  $\hat{\lambda}_{04} = 0.51$  for Model 3. However, the reason of this degradation is not obvious and even the poorer estimates for Model 5 are still not too far from the generating values.

Table 4.8: Parameter estimates and standard errors (in parentheses) for Model 1 and Model 3

Model 1: 1,000 respondents & Simple Q-matrix & Three time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.43	0.42 (0.05)	0.41 (0.09)	0.39 (0.10)	2.14
$\psi_{22}$	0.02	0.02	0.04 (0.02)	0 (0.01)	0.01 (0.01)	0.64
$\psi_{12}$	0.02	0.02	0.01 (0.02)	0.04 (0.03)	0.04 (0.04)	1.55
$\beta$	0.30		0.30 (0.02)	0.29 (0.03)	0.29 (0.04)	2.28
$\sigma^2$	0.60		0.59 (0.03)	0.69 (0.08)	0.61 (0.10)	3.87
$\lambda_{01}$	1.51			1.60 (0.07)	1.58 (0.10)	1.49
$\lambda_{02}$	-1.42			-1.38 (0.06)	-1.40 (0.08)	1.17
$\lambda_{03}$	-0.66			-0.61 (0.06)	-0.61 (0.07)	1.15
$\lambda_{04}$	0.50			0.57 (0.06)	0.59 (0.08)	1.37

Model 3: 500 respondents & Simple Q-matrix & Three time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.38	0.41 (0.06)	0.41 (0.19)	0.28 (0.21)	3.33
$\psi_{22}$	0.02	0.02	0.06 (0.03)	0.10 (0.10)	0.05 (0.12)	4.18
$\psi_{12}$	0.02	0.01	-0.01 (0.03)	-0.01 (0.11)	0.07 (0.13)	3.85
$\beta$	0.30		0.27 (0.02)	0.27 (0.05)	0.32 (0.05)	2.39
$\sigma^2$	0.60		0.56 (0.04)	0.61 (0.15)	0.68 (0.19)	5.33
$\lambda_{01}$	1.51			1.59 (0.10)	1.42 (0.13)	1.40
$\lambda_{02}$	-1.42			-1.48 (0.09)	-1.52 (0.10)	1.15
$\lambda_{03}$	-0.66			-0.64 (0.08)	-0.67 (0.09)	1.13
$\lambda_{04}$	0.50			0.49 (0.08)	0.51 (0.12)	1.42

To examine the effect of the number of time points for the case with the complex Q-matrix, we also compared Model 4 with Model 6. Overall, the point estimates were improved in Model 6 with the increased number of time points. The parameter estimates of Model 4 and Model 6 can be found in Table 4.5 and Table 4.6 respectively. It is noticeable that  $\hat{\sigma}^2$  improved from 1.16 (Model 4) to 0.61 (Model 6); and the point estimates for  $\boldsymbol{\lambda}_0$  improved in Model 6. The changes in standard errors between Model 4 and Model 6 are not discussed since the estimated standard errors are not reliable for these models.

### Estimated Guessing and Slipping Parameters for the GC-DINA Model

We evaluated how the estimated guessing and slipping parameters differ among different models. Figure 4.1 to Figure 4.3 show scatter plots comparing the estimated item parameters

Table 4.9: Parameter estimates and standard errors (in parentheses) for Model 3 and Model 5

Model 3: 500 respondents & Simple Q-matrix & Three time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.38	0.41 (0.06)	0.41 (0.19)	0.28 (0.21)	3.33
$\psi_{22}$	0.02	0.02	0.06 (0.03)	0.10 (0.10)	0.05 (0.12)	4.18
$\psi_{12}$	0.02	0.01	-0.01 (0.03)	-0.01 (0.11)	0.07 (0.13)	3.85
$\beta$	0.30		0.27 (0.02)	0.27 (0.05)	0.32 (0.05)	2.39
$\sigma^2$	0.60		0.56 (0.04)	0.61 (0.15)	0.68 (0.19)	5.33
$\lambda_{01}$	1.51			1.59 (0.10)	1.42 (0.13)	1.40
$\lambda_{02}$	-1.42			-1.48 (0.09)	-1.52 (0.10)	1.15
$\lambda_{03}$	-0.66			-0.64 (0.08)	-0.67 (0.09)	1.13
$\lambda_{04}$	0.50			0.49 (0.08)	0.51 (0.12)	1.42

Model 5: 500 respondents & Simple Q-matrix & Four time points						
	True	Observed	Growth Curve	Mixed Logit	GC-DINA	$\sqrt{\text{R.E. - GC}}$
$\psi_{11}$	0.40	0.38	0.38 (0.05)	0.39 (0.15)	0.28 (0.10)	1.86
$\psi_{22}$	0.02	0.02	0.01 (0.01)	0.02 (0.04)	0.01 (0.01)	1.11
$\psi_{12}$	0.02	0.01	0.03 (0.02)	0.03 (0.06)	0.06 (0.02)	1.16
$\beta$	0.30		0.31 (0.01)	0.31 (0.03)	0.30 (0.04)	2.47
$\sigma^2$	0.60		0.61 (0.03)	0.77 (0.12)	0.90 (0.14)	5.03
$\lambda_{01}$	1.51			1.59 (0.09)	1.53 (0.12)	1.39
$\lambda_{02}$	-1.42			-1.46 (0.08)	-1.39 (0.09)	1.13
$\lambda_{03}$	-0.66			-0.68 (0.08)	-0.64 (0.09)	1.14
$\lambda_{04}$	0.50			0.52 (0.08)	0.61 (0.11)	1.40

with the generating values with 95% CIs for Model 1 to Model 6.

First, Figure 4.1 suggests that the estimated item parameters became worse in Model 2 than Model 1, which corresponds to the effect of the complex Q-matrix. In Model 1, 19 CIs contain the generating values for both the guessing and slipping parameters. In Model 2, however, 14 CIs include the generating values for the guessing parameter. In the estimation process in Mplus, the logit threshold for the slipping parameter of Item 3 approached the extreme value of -15. Except for Item 3, the scatter plots for slipping parameters show increased standard errors for Model 2 than for Model 1. The averaged standard errors of the slipping parameter estimates, excluding Item 3, were 0.013 for Model 1 and 0.018 for Model 2. The averaged standard errors for the guessing parameter estimates were not very different between Model 1 and Model 2: 0.015 for Model 1 and 0.013 for Model 2.

Based on Figure 4.2, when the sample size is 500, we did not observe any severe deterio-

ration due to the complex Q-matrix. Model 4 shows increased length of error bars especially for the slipping parameter estimates—the averaged standard errors of the slipping parameter estimates were 0.018 for Model 3 and 0.023 for Model 4, but the errors in Model 4 may not be trustworthy. When comparing Model 5 and Model 6 for the four time points, there was not significant effects of the complex Q-matrix on the estimated item parameters. The guessing and slipping parameter estimates were not very different between Model 5 and Model 6, and showed overall good recovery.

Comparing Model 1 and Model 3, to examine the effect of sample size, it is noticeable that standard errors increased for Model 3 than for Model 1—the averaged standard errors for the estimated guessing parameters were 0.015 for Model 1 and 0.021 for Model 3 and the averaged errors for the estimated slipping parameters were 0.013 for Model 1 and 0.018 for Model 3. The point estimates, however, did not deteriorate with the smaller sample size.

We then compared Model 3 and Model 5 to investigate changes in the item parameter estimates depending on the number of time points. For both models, the estimated item parameters and the generating values were both close to each other. It appeared the GC-DINA estimates recovered the generating values well enough in both cases with three and four time points.

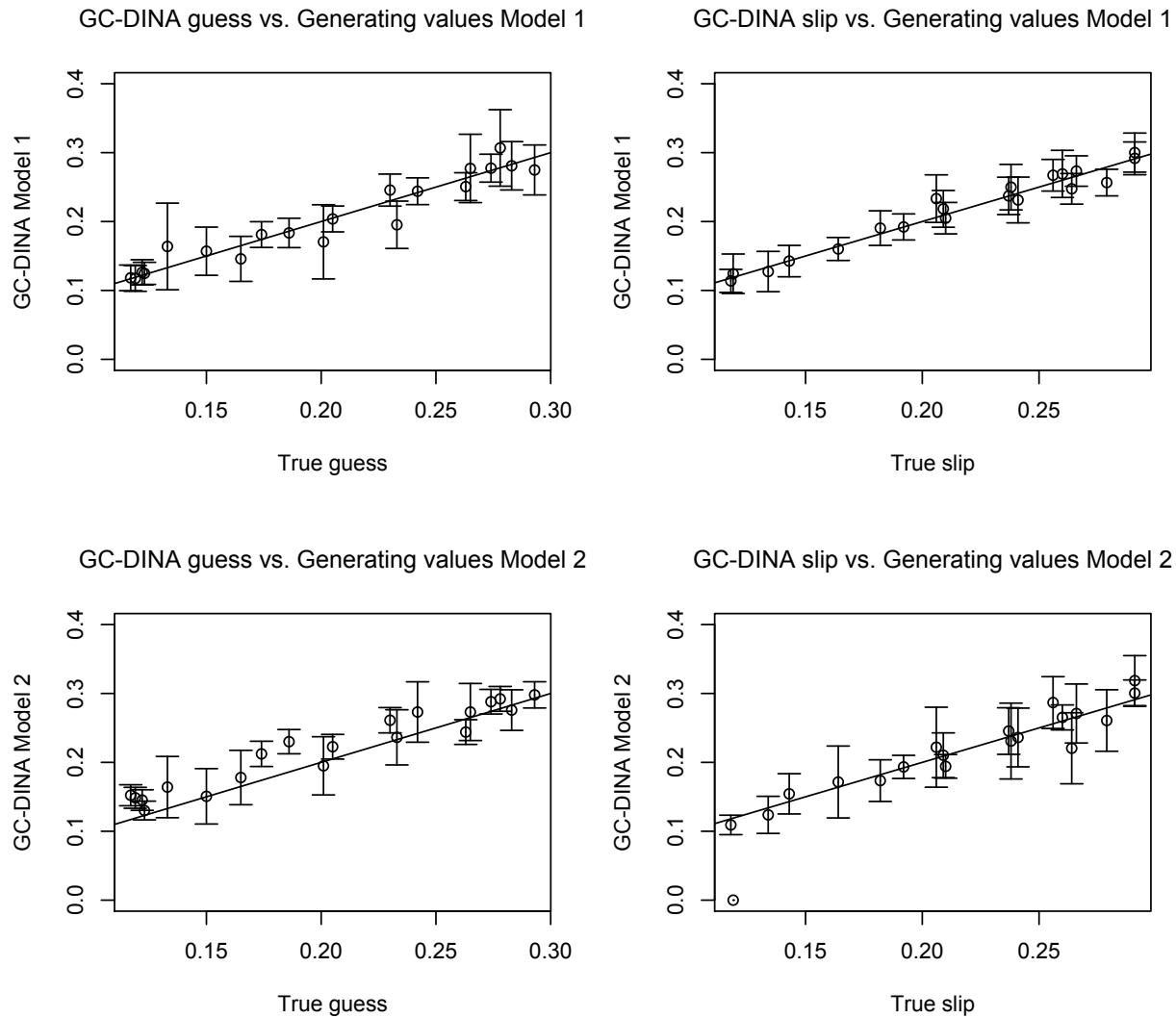


Figure 4.1: Scatter plots of GC-DINA item parameter estimates vs. generating values for Model 1 and Model 2.

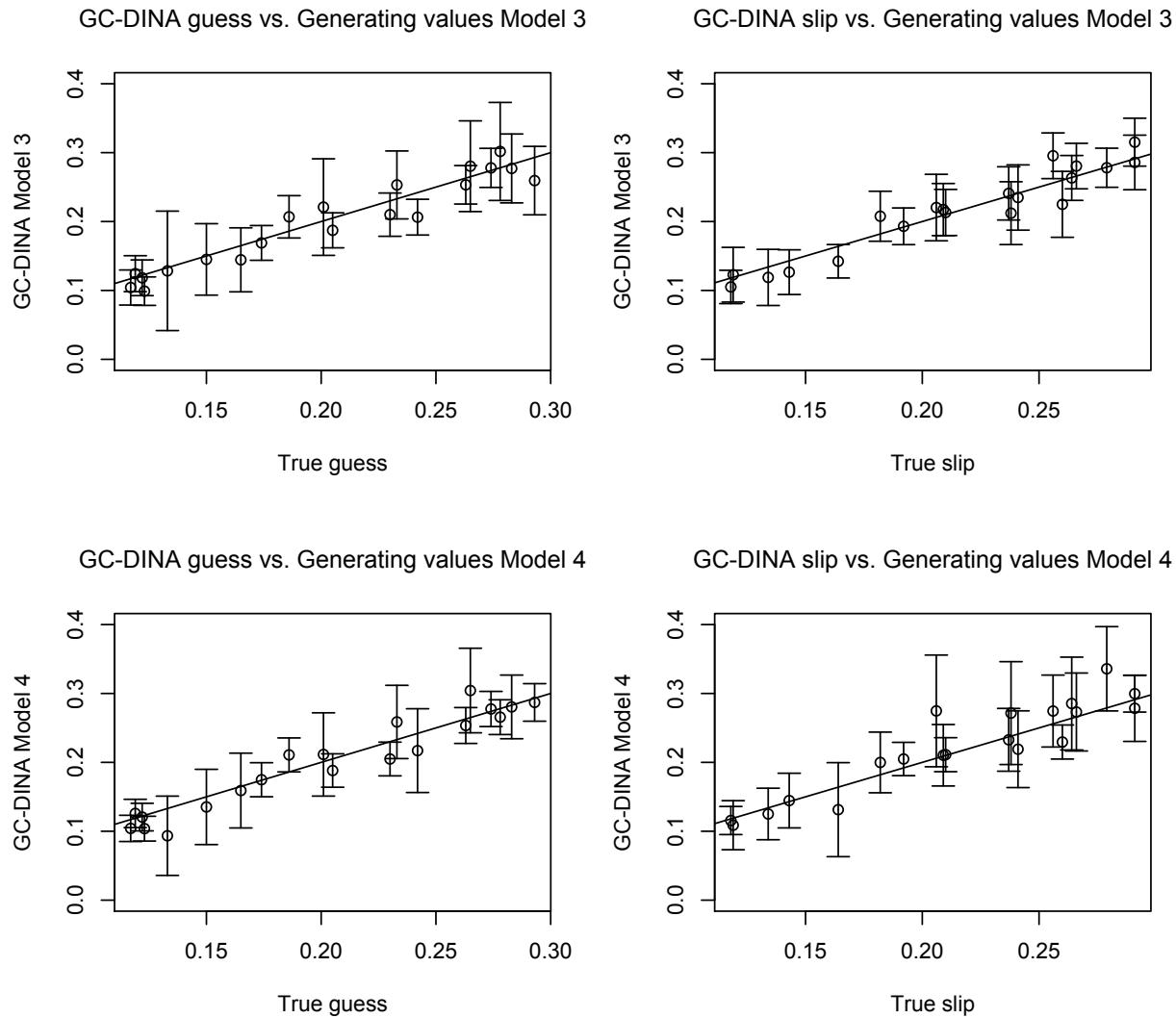


Figure 4.2: Scatter plots of GC-DINA item parameter estimates vs. generating values for Model 3 and Model 4.

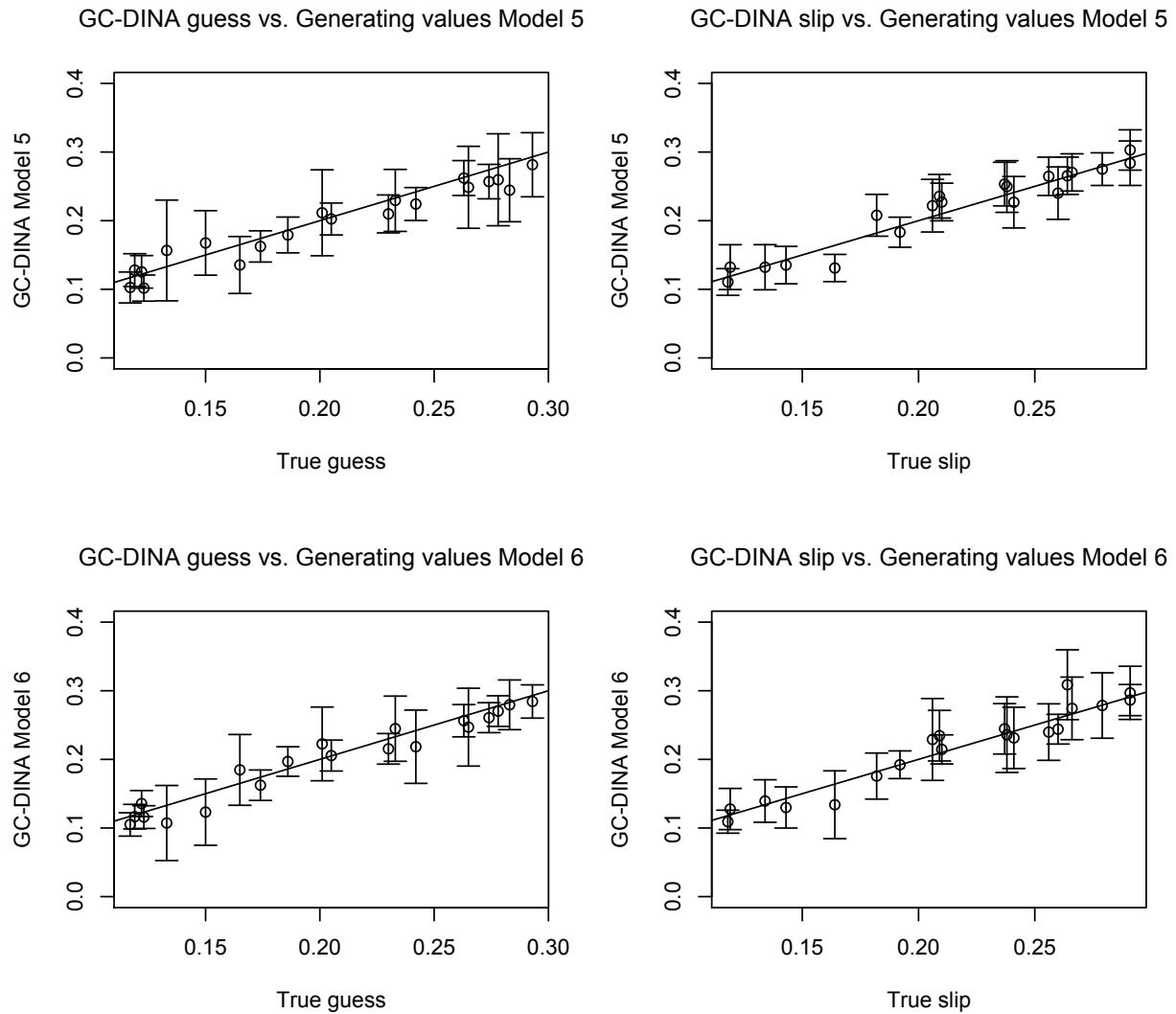


Figure 4.3: Scatter plots of GC-DINA item parameter estimates vs. generating values for Model 5 and Model 6.

## 4.2 Simulation Study 2

### Method

In this section, we consider a pre-post assessment scenario in which the same tests are administered at two time points ( $T=2$ ). Using a simulated dataset, we investigated parameter recovery of the GC-CDM defined in Section 3.3. We particularly used the DINA model for the measurement model. We generated responses of 1,000 respondents to 20 items designed to measure 4 skills. The Q-matrix defined as the simple design in the first simulation study was used, and it is given in Table 4.1. The guessing and slipping parameters were randomly generated from a uniform distribution on 0.1 to 0.3, and were constant over time. The higher-order latent traits for two time points were generated from the following distribution:

$$\begin{pmatrix} \theta_{j1} \\ \theta_{j2} \end{pmatrix} \sim N \left[ \begin{pmatrix} 0 \\ 0.3 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \right].$$

For the higher-order structural parameters, the following values were used:  $(\lambda_{01}, \lambda_{02}, \lambda_{03}, \lambda_{04}) = (1.51, -1.42, -0.66, 0.5)$ . The R code generating the data is available in Appendix.

The model was implemented in Mplus and the code is in Appendix. The parameters were estimated by MLE with 15 integration points. The default convergence criterion of Mplus was used: 0.001 for the absolute change in log-likelihood.

### Results

Table 4.10 shows the generating values and estimated parameter values for the GC-DINA model with standard errors. In the table, overall, the parameters were successfully recovered. The estimates are very close to the generating values; the differences are less than one standard error.

The estimated guessing and slipping parameter values were also compared to the generating values. Figure 4.2 shows two scatter plots comparing the generating values and the estimated values with 95% CIs for the guessing and the slipping parameters, respectively. For the guessing parameters, all CIs contain the generating values; for slipping parameters, 19 out of 20 ICs include the generating values. The item parameters were also sufficiently recovered by the estimated GC-DINA model.

In addition to the estimated model parameters, we examined the predicted skill mastery profiles and the predicted latent traits. For the skill mastery, we calculated the PCS and PCV as the classification accuracy measure. The calculated PCSs were 94% for  $t=1$  and 95% for  $t=2$ . In terms of PCV, at least two skills out of four skills were correctly classified for all respondents at both occasions; the proportion of at least three skills were correctly

classified was 98% for both occasions; and the proportion of all of the four skills were correctly identified was 79% for occasion 1 and 81% for occasion 2.

The predicted higher-order latent traits were compared to the generating values. Figure 4.5 presents scatter plots comparing the predicted values to the generating values for each time point. For both occasions, the correlation between the generating values and the predicted values was 0.67, which is good amount of correspondence.

Table 4.10: Generating values and GC-DINA estimates for simulation study 2.

	$\mu_1$	$\mu_2$	$\sigma_1^2$	$\sigma_2^2$	$\sigma_{12}$	$\lambda_{01}$	$\lambda_{02}$	$\lambda_{03}$	$\lambda_{04}$
True	0	0.30	1.00	1.00	0.8	1.51	-1.42	-0.66	0.50
GC-DINA (SE)	0 (fixed)	0.27 (0.06)	1.06 (0.18)	0.88 (0.16)	0.82 (0.11)	1.55 (0.11)	-1.43 (0.09)	-0.66 (0.08)	0.40 (0.10)

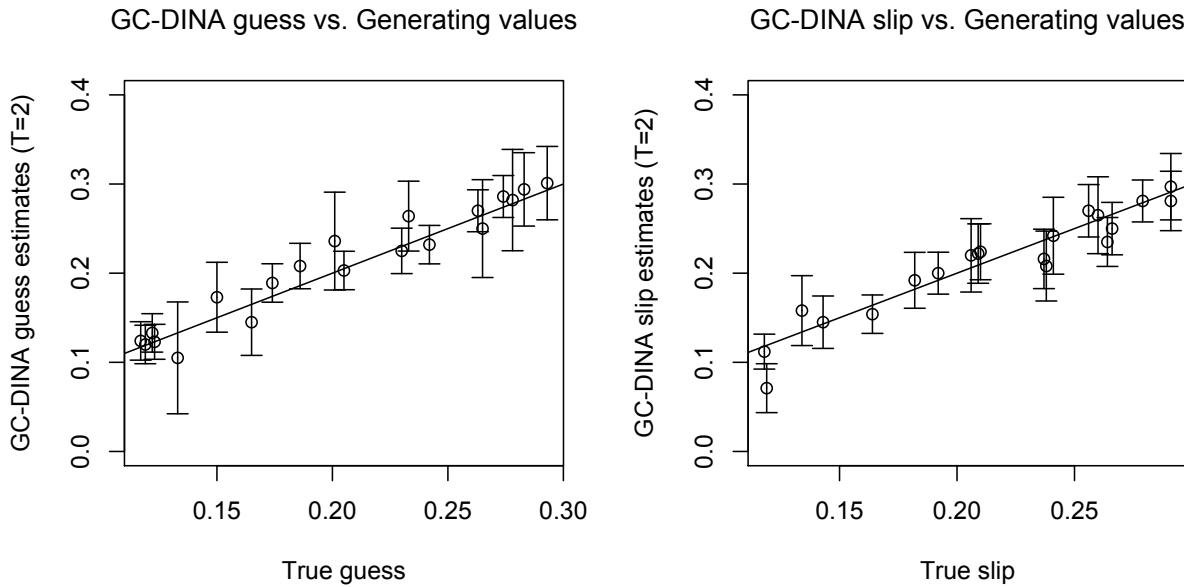


Figure 4.4: Scatter plots of GC-DINA item parameter estimates vs. generating values for simulation study 2.

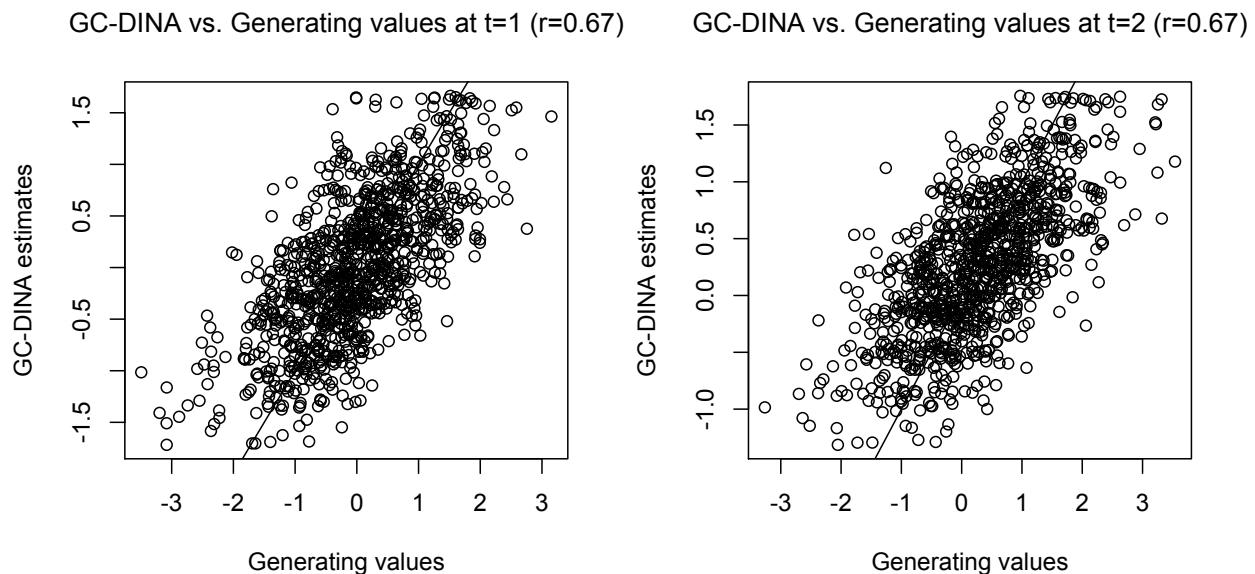


Figure 4.5: Scatter plots of predicted higher-order latent traits from the GC-DINA model vs. generating values for simulation study 2. The estimated correlation,  $r$ , is in parenthesis.

# Chapter 5

## Empirical Study

In this chapter, we use two examples to illustrate application of the GC-DINA model. The data sets are from multi-wave experiments designed to assess the effects of an instructional treatment called Enhanced Anchored Instruction (EAI; Bottge et al., 2003) on mathematics achievement of middle school students with learning disabilities (LD) and without learning disabilities (NLD).

### 5.1 Fraction of the Cost (FOC) test data

#### Data

A study was designed to assess the effects of two different EAI instructions: Kim's Koment (KK) and Fraction of the Cost (FOC) (Bottge et al., 2007). The KK instruction had the primary goal to guide students to develop informal understanding of pre-algebraic concepts, including linear function, line of best fit, variables, rate of change (slope), and reliability and measurement error. KK includes video instruction depicting two girls competing in pentathlon events. Here, with instruction from the video anchor, students learn to identify the fastest cars in the race, based on times and distances and also learn to construct the “line of best fit” to predict the speed of the cars when released from various points on the ramp. The FOC, on the other hand, depicts three middle school students trying to buy materials for a skateboard ramp. The aim is that students learn various concepts and skills and apply them holistically to solve a problem. The skills include (a) calculate the percent of money in a savings account and sales tax on a purchase, (b) read a tape measure, (c) convert feet to inches, (d) decipher building plans, (e) construct a table of materials, (f) compute whole numbers and mixed fractions, (g) estimate and compute combinations, and (h) calculate total cost.

For the study, 109 students were selected from six math classrooms in a school district in a small upper Midwestern town. The sample consisted of 50 males and 59 females in the seventh grade. 9 students were with LD and the remainder of the students were NLD. The KK and FOC instructions were administered during the academic year as follows: between Week 1 and 4, the KK instruction was taught for 13 days; between Week 4 and 19, the regular math curriculum was taught; and between Week 19 and 24, the FOC instruction was administered. In the regular instruction, teachers taught units on concepts related to geometry and proportional reasoning.

The FOC test, which is designed to assess the effects of the FOC instruction, was administered at four time points, at Weeks 1, 4, 19 and 24, before and after each of the three instructional sequences. The FOC test consists of 23 items that measure four skills. The Q-matrix for the FOC test is given in Table 5.1. This Q-matrix, apart from the rows for items 11, 12, and 13, was used as the complex Q-matrix for the simulation study in Section 4.1.

## Results

The GC-DINA model was fitted to the FOC test data using Mplus. The parameters were estimated with 15 integration points, the convergence criterion of  $10^{-7}$  for the absolute change in the log-likelihood and the maximum number of EM iterations was 1,000. Note that the default for the number of EM iterations is 500 in Mplus, but the estimation did not converge within 500 iterations.

The estimated parameters are given in Table 5.2. The variance of the person-specific random intercept was estimated as 7.20 which implies large variation between students in their overall higher-order latent traits. This also means that skill mastery is highly correlated across skills; the estimated intraclass correlation of the latent responses for skill mastery is 0.60. The correlation between the random intercept and the random slope of time is close to -1. This negative relationship corresponds to the idea that the EAI treatments were developed to be effective for students with LD. Thus, the low achieving students might have benefited more from the EAI than high achieving students.

The estimated average growth,  $\hat{\beta} = 1.81$ , suggests that students' overall abilities improved over time on average. The corresponding odds-ratio of 6.1 means that, for a median student (with random slope equal to the median of 0), the odds of mastering each skill increases by a factor of six between testing occasions. Figure 5.1 displays the predicted higher-order latent traits over time, which were calculated by  $\hat{\theta}_{jt} = \hat{\zeta}_{1j} + (\hat{\beta} + \hat{\zeta}_{2j}) \times \text{time}_t + \hat{\epsilon}_{jt}$ . It shows that all students' overall mathematics abilities were predicted to improve over the four time points.

The FOC test data were also analyzed by Li et al. (2016) using the LTA-DINA model. We compared the estimated item parameter values from the GC-DINA model with the LTA-DINA estimates reported in their paper. Figure 5.2 presents scatter plots comparing the

Table 5.1: Q-matrix for Fraction of Cost (FOC) test.

Items	Number & Operation	Measurement	Problem Solving	Presentation
1	1	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	0	0
5	1	1	0	0
6	1	1	0	0
7	1	1	0	0
8	1	1	0	0
9	0	1	0	0
10	0	1	0	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	0	0	1	0
17	1	1	1	1
18	0	1	0	1
19	1	0	0	1
20	1	0	0	0
21	1	0	0	0
22	1	0	0	0
23	1	0	1	0

GC-DINA estimates with the LTA-DINA estimates for guessing and slipping parameters. Most points lie near the 45-degree line, indicating that the item parameter estimates were similar between the GC-DINA model and the LTA-DINA model. Li et al. (2016) examined the diagnostic quality of each item by using the following indicator:

$$\frac{(1 - s_i)/s_i}{g_i/(1 - g_i)},$$

which is the ratio of the odds of responding correctly to Item  $i$  when respondents possess the required skill and the odds of responding correctly to Item  $i$  although they have not mastered the required skill. Items with large values of this odds ratio are considered as diagnostic items that distinguish well between people who have the required skills and those

Table 5.2: Estimated GC-DINA model parameters for the FOC data.

	$\hat{\psi}_{11}$	$\hat{\psi}_{22}$	$\hat{\psi}_{12}$	$\hat{\beta}$	$\hat{\sigma}^2$	$\hat{\lambda}_{01}$	$\hat{\lambda}_{02}$	$\hat{\lambda}_{03}$	$\hat{\lambda}_{04}$
Est	7.20	0.42	-1.74	1.81	1.41	-0.30	-1.61	-6.52	-4.20
SE	2.85	0.25	0.84	0.25	0.91	0.39	0.49	0.98	0.69

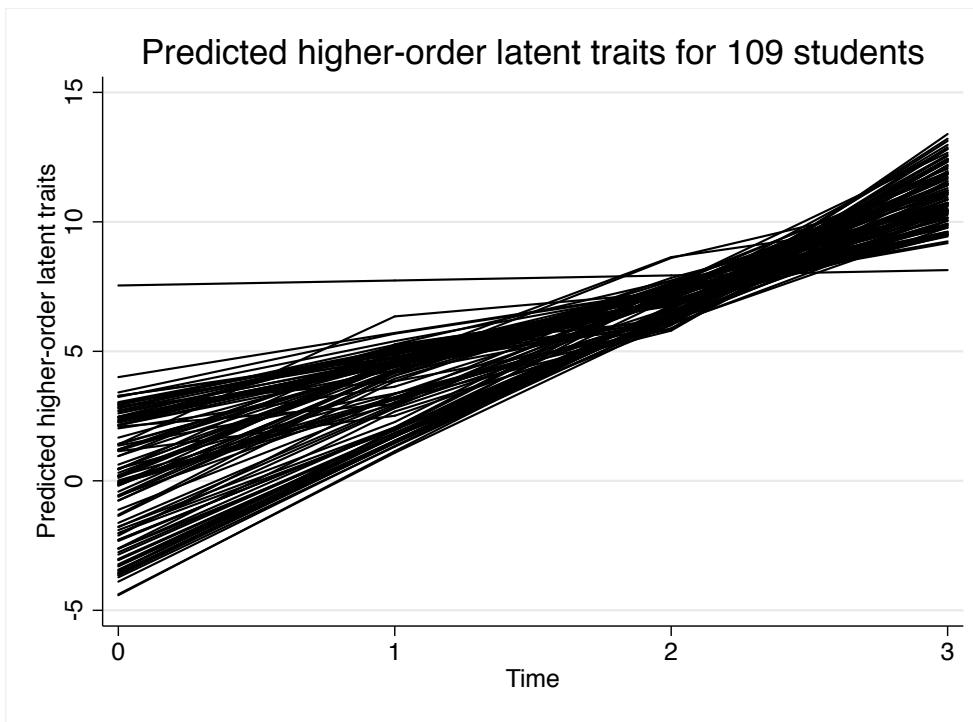


Figure 5.1: Predicted growth trajectories in the higher-order latent trait across four time points for the FOC data.

who do not have the required skills. If the odds ratio is lower than 2, the item is considered **problematic**. Based on the GC-DINA item parameter estimates, there was no item flagged in terms of discrimination ability: the smallest estimated odds ratio was 3.21 for Item 18. Also, for all items, monotonicity,  $1 - s_i > g_i$ , held.

In addition, we estimated the proportions of students that mastered each skill at each time point based on the predicted skill mastery from the GC-DINA model. We also compared those proportions with the skill mastery proportions reported in Li et al. (2016). The computed proportions are given in Table 5.3. Overall, the proportions are similar between the GC-DINA model and the LTA-DINA model. In both models, students' knowledge on all

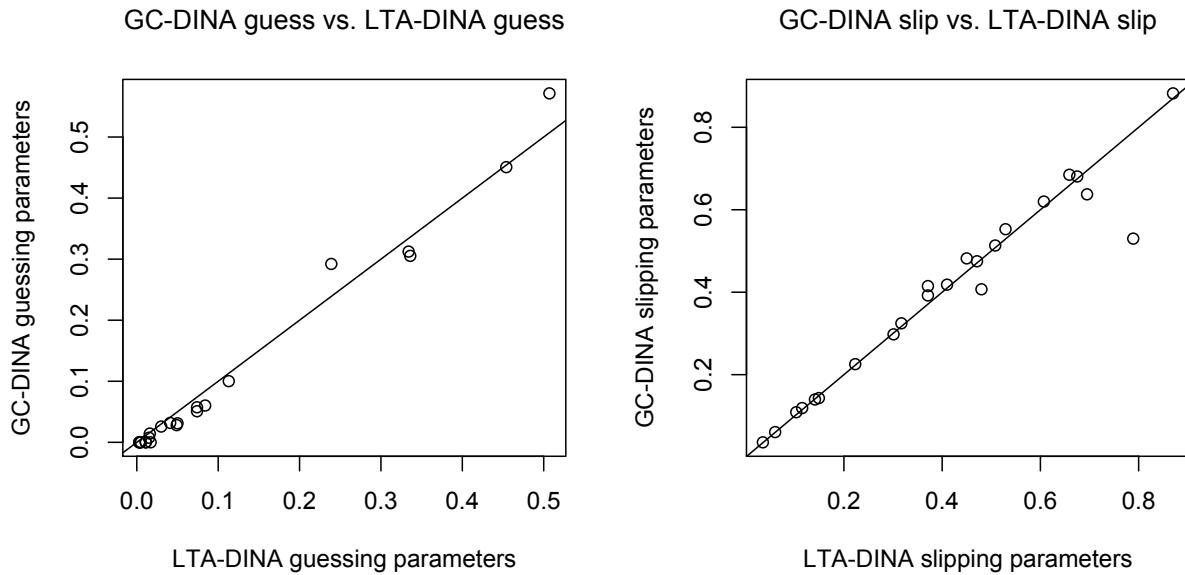


Figure 5.2: Scatter plots of GC-DINA item parameter estimates vs. LTA-DINA item parameter estimates reported in Li et al. (2016) for the FOC data.

of the four skills improved at the last time point compared to the first time point. Among the four skills, Problem Solving appeared to be the most difficult skill to master: only 32% (30% in the LTA-DINA model) mastered Problem Solving at the fourth time point.

Table 5.3: Proportion of students predicted to have mastered each skill at each occasion for the FOC data (LTA results from Li et al. (2016)).

	t=1		t=2		t=3		t=4	
	GC	LTA	GC	LTA	GC	LTA	GC	LTA
Number & Operation	0.49	0.47	0.72	0.72	0.90	0.78	0.99	1
Measurement	0.32	0.51	0.56	0.71	0.77	0.68	0.97	0.95
Problem Solving	0.01	0.02	0.03	0.02	0.07	0.03	0.32	0.30
Presentation	0.08	0.28	0.18	0.17	0.35	0.25	0.78	0.95

Although the proportions were, in general, similar in both models, there were cases where the probability increased between adjacent occasions for the GC-DINA model and decreased for the LTA-DINA model. This may be because the LTA-DINA model allows different transition probabilities for each skill and for each pair of occasions, whereas the

GC-DINA model only allows linear growth in the higher-order latent trait and the changes in the skill mastery are driven by the changes in the latent trait. Thus, the LTA-DINA model has a larger number of parameters than the GC-DINA model:  $6 \times 4$  transition probabilities, 4 prior probabilities and 46 item parameters, compared with 9 structural model parameters and 46 item parameters for GC-DINA. We calculated Akaike information criterion (AIC) to compare the GC-DINA model and the LTA-DINA model. The AIC is defined as  $-2 \times \log\text{-likelihood} + 2 \times \text{number of parameters}$ . Using the reported deviance of 6,590 for the LTA-DINA model in Li et al. (2016), the computed AICs were 6,738 for the LTA-DINA model and 7,467.59 for the GC-DINA model. The AIC values indicate that the models differ appreciably, suggesting a preference for the LTA-DINA model.

### Simulation Study Using Parameters Recovered from the FOC data

To confirm that the estimates can be trusted, we conducted a simulation study using the estimated parameter values from the FOC data application. We simulated data with the same characteristics as the FOC data. Item responses of 109 students to 23 items measuring 4 skills were generated. The same Q-matrix in Table 5.1 was used. For the model parameters, we used the point estimates obtained from the FOC data.

Table 5.4 presents the estimated parameter values from the simulated data. The data has a small number of students ( $J=109$ ) and the Q-matrix has the complex structure; thus, it was a relatively tough case for estimation. The standard errors for the estimated parameters were a bit large, but the point estimates were not very far from the generating values.

Figure 5.3 provides scatter plots comparing the GC-DINA item parameter estimates to the generating values with 95% CIs. The logit thresholds were estimated for the guessing and slipping parameters with standard errors on the logit scale. We first calculated 95% CIs on the logit scale and then transformed the confidence limits to probabilities. Although large standard errors were observed when the generating values are large, the parameters were sufficiently recovered by the GC-DINA estimation.

Table 5.4: GC-DINA parameter estimates and standard errors for the simulated data using parameters recovered from the FOC data.

	$\psi_{11}$	$\psi_{22}$	$\psi_{12}$	$\beta$	$\sigma^2$	$\lambda_{01}$	$\lambda_{02}$	$\lambda_{03}$	$\lambda_{04}$
True	7.20	0.42	-1.74	1.81	1.41	-0.30	-1.61	-6.52	-4.20
Est	7.70	0.60	-2.16	2.29	2.57	-1.08	-2.76	-8.55	-5.47
SE	3.47	0.36	1.10	0.40	1.63	0.47	0.70	1.55	1.09

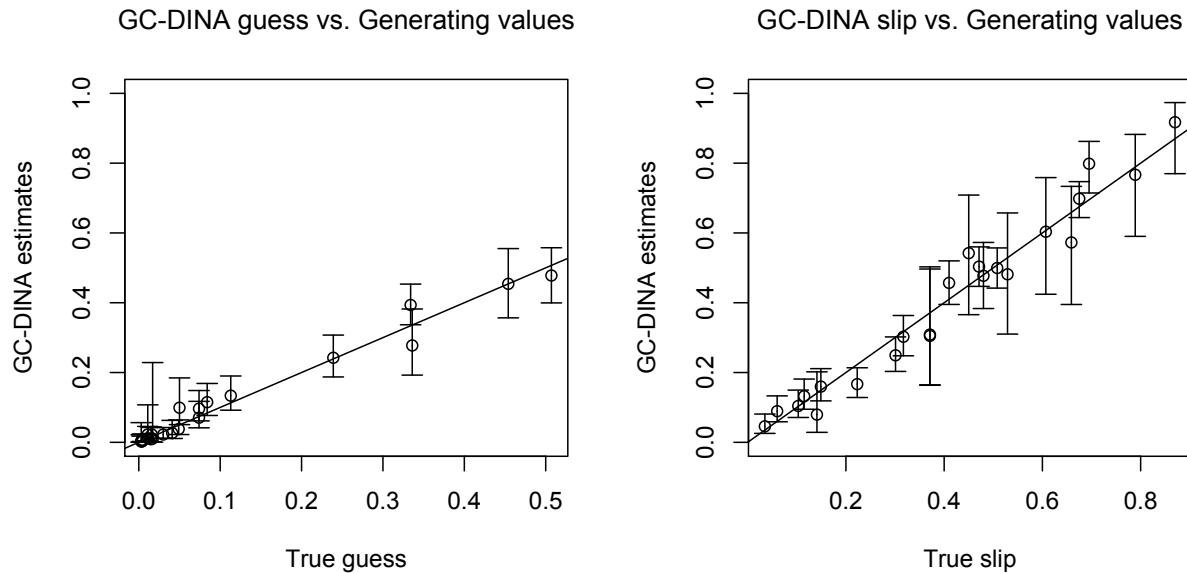


Figure 5.3: Scatter plots of GC-DINA item parameter estimates vs. generating values for the simulated data using parameters recovered from the FOC data.

## 5.2 Problem Solving Test (PST) data

### Data

The second dataset was obtained from a pre-post test cluster-randomized trial designed to estimate the effects of combining explicit and anchored instruction on fraction computation and problem solving. The experiment considered two instructional conditions: EAI and business as usual (BAU), and random assignment was conducted at the school level. EAI instruction includes five EAI units: KK, FOC, Fractions at Work (FAW), Hovercraft Project (HOV), and Grand Pentathlon (GP). Teachers assigned to the BAU condition followed their regular school math curriculum. To examine the effects of the EAI on students computation and problem solving skills, two researcher-developed tests (Fractions Computation Test, Problem-Solving Test) and two standardized achievement tests (two math sub-tests of the ITBS) were administered prior to and following the instructional period.

We analyzed students' responses of the Problem-Solving Test (PST). The test consists of 21 items measuring "Ratios and Proportional Relationships", "Measurement and Data", "Number and Operations-Fractions", "Geometry" and "Statistics and Probability". The Q-matrix was developed with the simple structure where all items measure a single skill.

Among 21 items, only one item measured “Statistics and Probability”, thus we excluded that pair of skill and item. The final Q-matrix with 20 items and 4 skills given in Table 5.5 was used for our analysis.

The data consist of two samples. The first sample was obtained from special education resource rooms in 31 middle schools and included 335 students with disabilities. The other sample was obtained from regular education math classrooms in 24 middle schools and included 471 students of whom 134 had disabilities in mathematics.

Table 5.5: Q-matrix for Problem Solving Test (PST).

Items	Ratios and Proportional Relationships	Measurement and Data	Number and Operations-Fractions	Geometry
1	1	0	0	0
2	0	1	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	1	0
6	0	0	1	0
7	0	0	1	0
8	0	0	1	0
9	0	1	0	0
10	0	1	0	0
11	0	1	0	0
12	0	1	0	0
13	1	0	0	0
14	1	0	0	0
15	0	0	0	1
16	0	0	0	1
17	1	0	0	0
18	0	0	0	1
19	0	0	0	1
20	0	0	0	1

## Results

We fitted the GC-DINA model for two time points, which was described in Section 3.3. The parameters were estimated with 15 integration points, the convergence criterion of 0.001 for

the absolute change in the log-likelihood and the maximum number of EM iterations set to 500.

The estimated parameters for the structural part are given in Table 5.6, and the estimated item parameters are given in Table 5.7. The estimated variances were very large at 11.07 and 14.50 for  $\theta_{j1}$  and  $\theta_{j2}$ , respectively, and the estimated covariance between  $\theta_{j1}$  and  $\theta_{j2}$  was 11.53, corresponding to a correlation of 0.91. The estimated mean of  $\theta_{j2}$ ,  $\hat{\mu}_2 = 2.05$ , indicates that the higher-order latent traits increased on average from pre to post. In this data, the higher-order latent trait can be considered as the overall mathematics knowledge related to the four skills: Ratios and Proportional Relationships, Measurement and Data, Number and Operations-Fractions and Geometry. The discrimination index of the items was also investigated. As can be seen in the last column of Table 5.7, all odds ratios are greater than 2 so no item was flagged. Also, monotonicity held for all items.

We estimated the proportions of students who mastered each skill at each time point based on the predicted skill mastery from the GC-DINA model. These proportions are given in Table 5.8. The result shows that the proportions of students who mastered each skill increased from pre to post. For Ratios and Proportional Relationships, the proportion increased to 0.59 from 0.39; for Measurement and Data, the proportion increased to 0.73 from 0.58; for Number and Operations-Fractions, the proportion increased to 0.52 from 0.31; for Geometry, the proportion increased to 0.69 from 0.51.

Table 5.6: GC-DINA parameter estimates and standard errors for the PST data.

	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\sigma}_{12}$	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\lambda}_{01}$	$\hat{\lambda}_{02}$	$\hat{\lambda}_{03}$	$\hat{\lambda}_{04}$
Est	11.07	14.50	11.53	0	2.05	-1.07	0.75	-1.91	0.16
SE	1.65	2.38	1.53	fixed	0.17	0.24	0.21	0.24	0.28

### Simulation Study Using Parameters Recovered from the PST data

We conducted a simulation study to check whether our estimates appear to be trustworthy. Item responses for 800 respondents to 20 items measuring four skills were generated. The same Q-matrix as used for the PST data analysis was used. The higher-order latent traits were generated from the following bivariate normal distribution:

$$\begin{pmatrix} \theta_{1j} \\ \theta_{2j} \end{pmatrix} \sim N \left[ \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 11 & 11 \\ 11 & 14 \end{pmatrix} \right].$$

For other parameters, the estimated GC-DINA model parameters from the PST data were used as generating values.

Table 5.7: GC-DINA item parameter estimates and standard errors for the PST data.

Item	$\hat{g}$	SE ( $\hat{g}$ )	$\hat{s}$	SE ( $\hat{s}$ )	$\frac{(1-\hat{s}_i)/\hat{s}_i}{\hat{g}_i/(1-\hat{g}_i)}$
1	0.10	0.01	0.42	0.03	12.52
2	0.49	0.03	0.13	0.01	7.19
3	0.06	0.01	0.65	0.02	9.17
4	0.17	0.02	0.39	0.03	7.71
5	0.29	0.03	0.05	0.01	44.17
6	0.35	0.02	0.07	0.02	26.51
7	0.03	0.01	0.37	0.04	66.69
8	0.30	0.02	0.35	0.03	4.35
9	0.32	0.02	0.17	0.02	10.67
10	0.18	0.03	0.07	0.01	66.43
11	0.29	0.04	0.02	0.01	107.27
12	0.16	0.02	0.13	0.02	37.47
13	0.39	0.03	0.09	0.01	14.95
14	0.04	0.01	0.26	0.03	63.02
15	0.25	0.03	0.21	0.02	11.31
16	0.35	0.03	0.19	0.02	8.36
17	0.02	0.01	0.76	0.02	13.64
18	0.44	0.03	0.05	0.01	26.68
19	0.11	0.02	0.38	0.02	13.12
20	0.10	0.02	0.51	0.02	8.48

Table 5.8: Proportion of students predicted to have mastered each skill at each occasion for the PST data.

Time	Ratios and Proportional Relationships		Measurement and Data	Number and Operations-Fractions	Geometry
	Time 1	Time 2			
Time 1	0.39		0.58	0.31	0.51
Time 2		0.59	0.73	0.52	0.69

Table 5.9 presents the generating values and the estimated parameter values from the GC-DINA model. The estimated parameter values were close enough to the generating values for all parameters. Based on the scatter plots comparing the estimated item parameters with the generating values in Figure 5.4, we can see that the item parameters were also sufficiently recovered. This simulated data is a similar case to the case considered in Section 4.2. The data had a sufficient number of respondents ( $J=800$ ) and the Q matrix was the

simple Q-matrix used in the simulations, so this case is considered a relatively easy case for estimation. Overall, the GC-DINA estimates show good parameter recovery even for the large covariance matrix.

Table 5.9: GC-DINA parameter estimates and standard errors for the simulated data using parameters recovered from the PST data.

	$\sigma_1^2$	$\sigma_2^2$	$\sigma_{12}$	$\mu_1$	$\mu_2$	$\lambda_{01}$	$\lambda_{02}$	$\lambda_{03}$	$\lambda_{04}$
True	11	14	11	0	2	-1.07	0.75	-1.91	0.16
Est	10.98	12.84	10.5	0	2.11	-1.08	0.62	-2.25	0.11
SE	1.44	1.74	1.16	fixed	0.15	0.20	0.16	0.20	0.18

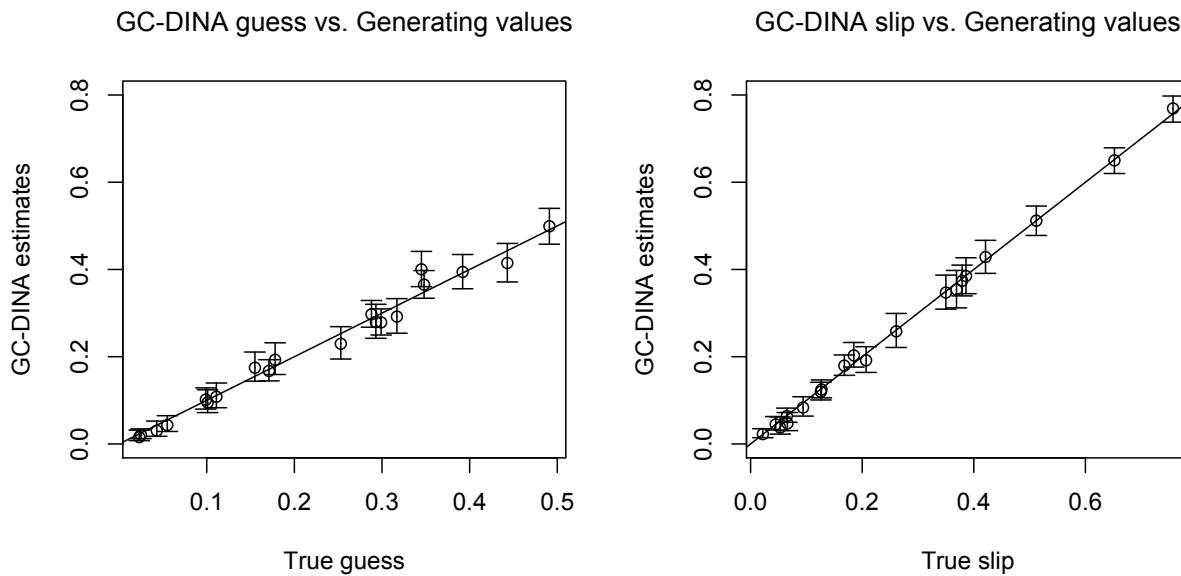


Figure 5.4: Scatter plots of GC-DINA item parameter estimates vs. generating values for the simulated data using parameters recovered from the PST data.

# Chapter 6

## Comparison of Longitudinal Psychometric Models

### 6.1 Introduction

This chapter considers different longitudinal psychometric models described in Chapter 2 to examine their similarity when fitted to the same data. For simplicity and clarity, we focus on the situation where there are two occasions,  $T = 2$ . We compare our GC-CDM to the LTA-CDM (Li et al., 2016) and Andersen's longitudinal IRT model (Andersen, 1985), henceforth referred to as the longitudinal Rasch model. The LTA-CDM and the longitudinal IRT model have similarities with the GC-CDM in that the LTA-CDM traces skill mastery over time and the longitudinal Rasch model measures growth in the latent trait.

Using a data set generated from the GC-DINA model, our study focuses on the following research questions:

1. What are the parameter estimates of the GC-DINA, LTA-DINA and longitudinal IRT models?
2. How well do the models fit the data?
3. What are the predicted skill mastery profiles from the GC-DINA and the LTA-DINA and how do they differ?
4. What are the predicted latent traits from the GC-DINA and the longitudinal IRT model and how do they differ?

## 6.2 Method

For the simulation study, the same dataset generated for simulation study 2 in Chapter 4 was used. We give a brief summary of data generation here. Item responses of 1,000 persons to 20 items at 2 occasions were generated from the GC-DINA model. The assessment measures four skills with the simple Q-matrix in Table 4.1. The latent traits for the two occasions were generated from the following bivariate normal distribution:

$$\begin{pmatrix} \theta_{1j} \\ \theta_{2j} \end{pmatrix} \sim N \left[ \begin{pmatrix} 0 \\ 0.3 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \right].$$

For the higher-order structural model, the Rasch model was used with the intercepts,  $\lambda_0 = (1.51, -1.42, -0.66, 0.50)'$ . The guessing and slipping parameters were randomly generated from an uniform distribution from 0.1 to 0.3, and were constant over time.

For the comparison, we considered the following models: the GC-DINA model described in Section 3.3 with time-constant guessing and slipping parameters, the LTA-DINA model with different transition probabilities for each skill and time-constant guessing and slipping parameters, and the longitudinal Rasch model with a bivariate normal distribution of ability and time-constant item difficulty parameters. The longitudinal IRT and GC-DINA model therefore both assumed a bivariate normal distribution for the latent traits because a full growth-curve model is not identified with two occasions. All models were fitted in Mplus, and the codes for the LTA-DINA model and the longitudinal Rasch model are presented in Appendix.

## 6.3 Results

### Parameter Estimates and Model Fit

The estimated model parameters from the GC-DINA, LTA-DINA and longitudinal Rasch model are given in Tables 6.1 to 6.4. Table 6.1 shows the estimated mean and covariance matrix of  $\boldsymbol{\theta}$  for the GC-DINA and longitudinal Rasch models. The GC-DINA model estimates are close to the generating values whereas the estimates from the longitudinal Rasch model differ a little. In particular, the longitudinal Rasch model gives a smaller variance of 0.52 for  $\theta_1$  and 0.49 of  $\theta_2$  and a smaller mean of  $\theta_2$  of 0.09. In Table 6.2, the GC-DINA model shows good parameter recovery for the higher-order structural parameters  $\lambda_0$ .

Table 6.3 presents the estimated transition probabilities for the LTA-DINA model.  $p_{m1}$  indicates the probability of skill mastery at the first occasion;  $p_{m|n}$  is the transition probability from non-mastery to mastery;  $p_{m|m}$  is the transition probability from mastery to mastery;  $p_{n|m}$  is the transition probability from mastery to non-mastery;  $p_{n|n}$  is the transition probability from non-mastery to non-mastery. The model implies that, for the first time point,

76% of respondents have mastered skill 1, 23% of respondents have mastered skill 2, 38% of respondents have mastered skill 3, and 59% of respondents have mastered skill 4.  $p_{m|n}$  can be interpreted as the learning rate, which can be used as an index when evaluating a certain educational intervention performed between two time points—skill 1 has the highest learn rate of 0.77 and skill 2 has the lowest rate of 0.23.

Table 6.4 shows the generating values and estimates of item parameters. The estimated guessing and slipping parameters for the GC-DINA and LTA-DINA are very similar to each other (the difference is less than 0.01), and they sufficiently recover the generating values, taking into account the estimated standard errors except for the slipping parameter of item 3—in the GC-DINA, the point estimate is 0.07 and differs from the true value of 0.12 by more than three standard errors. The estimated item difficulty parameters from the longitudinal Rasch model are also reported, but they cannot be directly compared with the guessing and slipping parameters.

We examined the model fit of each model. Table 6.5 presents the following model fit information: log-likelihood (=the maximum log-likelihood for the model), number of parameters, Akaike information criterion (AIC) and Bayesian information criterion (BIC). The AIC is defined as  $-2 \times \log\text{-likelihood} + 2 \times \text{number of parameters}$ ; and the BIC is defined as  $-2 \times \log\text{-likelihood} + \ln(N) \times \text{number of parameters}$  where  $N$  is the number of observations. For the AIC and BIC, the smallest value suggests the preferred model so the GC-DINA is the best-fitting model among the three models—the AIC is 47,100.47 for the GC-DINA, 47,272.96 for the LTA-DINA and 50,765.54 for the longitudinal Rasch model; and the BIC is 47,336.04 for the GC-DINA, 47,528.16 for the LTA-DINA and 50,883.33 for the longitudinal Rasch model. The AIC and BIC suggest that the GC-DINA is more similar to the LTA-DINA and is more different from the longitudinal Rasch model in terms of the model fit.

Table 6.1: Parameter estimates for the GC-DINA model and the longitudinal Rasch model.

	$\widehat{E}(\theta_1)$	$\widehat{E}(\theta_2)$	$\widehat{\text{var}}(\theta_1)$	$\widehat{\text{var}}(\theta_2)$	$\widehat{\text{cov}}(\theta_1, \theta_2)$
Generating values	0	0.3	1	1	0.8
GC-DINA	0 (fixed)	0.27	1.06	0.88	0.82
Longitudinal Rasch	0 (fixed)	0.09	0.52	0.49	0.17

**Table 6.2:** Higher-order structural parameter estimates for the GC-DINA model.

	$\hat{\lambda}_{01}$ (SE)	$\hat{\lambda}_{02}$ (SE)	$\hat{\lambda}_{03}$ (SE)	$\hat{\lambda}_{04}$ (SE)
Generating values	1.51	-1.42	-0.66	0.50
GC-DINA	1.55 (0.11)	-1.43 (0.09)	-0.66 (0.08)	0.40 (0.10)

**Table 6.3:** Parameter estimates for the LTA-DINA model.

	$\hat{p}_{m1}$	$\hat{p}_{m n}$	$\hat{p}_{m m}$	$\hat{p}_{n m}$	$\hat{p}_{n n}$
Skill 1	0.76 (0.02)	0.77 (0.04)	0.86 (0.02)	0.14 (0.02)	0.23 (0.04)
Skill 2	0.23 (0.02)	0.23 (0.02)	0.41 (0.04)	0.59 (0.04)	0.77 (0.02)
Skill 3	0.38 (0.02)	0.39 (0.02)	0.44 (0.03)	0.56 (0.03)	0.61 (0.02)
Skill 4	0.59 (0.02)	0.51 (0.04)	0.72 (0.03)	0.28 (0.03)	0.49 (0.04)

Table 6.4: Item parameter estimates for the GC-DINA model, the LTA-DINA model, and the longitudinal Rasch model.

	True		GC-DINA		LTA-DINA		L-Rasch <sup>a</sup>
	Guess	Slip	Guess	Slip	Guess	Slip	Difficulty
Item1	0.20	0.19	0.24 (0.03)	0.20 (0.01)	0.23 (0.03)	0.20 (0.01)	0.85 (0.06)
Item2	0.24	0.26	0.23 (0.01)	0.27 (0.02)	0.23 (0.01)	0.26 (0.02)	-0.68 (0.05)
Item3	0.26	0.12	0.27 (0.01)	0.07 (0.01)	0.27 (0.01)	0.07 (0.02)	-0.32 (0.05)
Item4	0.12	0.29	0.13 (0.01)	0.30 (0.02)	0.13 (0.01)	0.30 (0.02)	-0.70 (0.06)
Item5	0.23	0.14	0.23 (0.01)	0.15 (0.02)	0.22 (0.01)	0.15 (0.02)	-0.16 (0.05)
Item6	0.19	0.18	0.21 (0.01)	0.19 (0.02)	0.21 (0.01)	0.20 (0.02)	-0.29 (0.05)
Item7	0.12	0.24	0.12 (0.01)	0.22 (0.02)	0.12 (0.01)	0.22 (0.02)	-0.58 (0.05)
Item8	0.12	0.21	0.12 (0.01)	0.22 (0.02)	0.12 (0.01)	0.23 (0.02)	-0.58 (0.05)
Item9	0.17	0.13	0.19 (0.01)	0.16 (0.02)	0.19 (0.01)	0.16 (0.02)	-0.71 (0.06)
Item10	0.20	0.24	0.20 (0.01)	0.24 (0.02)	0.20 (0.01)	0.24 (0.02)	-0.76 (0.06)
Item11	0.27	0.24	0.29 (0.01)	0.21 (0.02)	0.29 (0.01)	0.21 (0.02)	-0.43 (0.05)
Item12	0.12	0.21	0.12 (0.01)	0.22 (0.02)	0.12 (0.01)	0.22 (0.02)	-1.04 (0.06)
Item13	0.26	0.28	0.25 (0.03)	0.28 (0.01)	0.25 (0.03)	0.28 (0.01)	0.54 (0.05)
Item14	0.28	0.16	0.28 (0.03)	0.15 (0.01)	0.28 (0.03)	0.15 (0.01)	1.10 (0.06)
Item15	0.29	0.27	0.30 (0.02)	0.25 (0.02)	0.30 (0.02)	0.25 (0.02)	0.29 (0.06)
Item16	0.23	0.26	0.26 (0.02)	0.27 (0.02)	0.26 (0.02)	0.27 (0.02)	0.17 (0.05)
Item17	0.13	0.12	0.11 (0.03)	0.11 (0.01)	0.11 (0.03)	0.11 (0.01)	1.09 (0.06)
Item18	0.16	0.29	0.15 (0.02)	0.28 (0.02)	0.14 (0.02)	0.28 (0.02)	-0.07 (0.06)
Item19	0.15	0.21	0.17 (0.02)	0.22 (0.02)	0.17 (0.02)	0.23 (0.02)	0.14 (0.06)
Item20	0.28	0.26	0.29 (0.02)	0.24 (0.01)	0.29 (0.02)	0.24 (0.01)	0.32 (0.05)

Note: Standard errors in parentheses

<sup>a</sup> L-Rasch = Longitudinal Rasch

Table 6.5: Model fit information of the GC-DINA model, the LTA-DINA model, and the longitudinal Rasch model.

Model	Log-likelihood	# of parameters	AIC	BIC
GC-DINA	-23,502.23	48	47,100.47	47,336.04
LTA-DINA	-23,584.48	52	47,272.96	47,528.16
Longitudinal Rasch	-25,358.77	24	50,765.54	50,883.33

## Predicted Skill Mastery from the GC-DINA and LTA-DINA

As a comparison between the GC-DINA and LTA-DINA, we evaluated the ability of the models to predict respondents' skill mastery. The predicted skill mastery vector for respondent  $j$  can be represented as  $\hat{\alpha}_j = (\hat{\alpha}_{j11}, \hat{\alpha}_{j21}, \hat{\alpha}_{j31}, \hat{\alpha}_{j41}, \hat{\alpha}_{j12}, \hat{\alpha}_{j22}, \hat{\alpha}_{j32}, \hat{\alpha}_{j42})'$ , where  $\alpha_{jkt}$  indicates respondent  $j$ 's mastery state of skill  $k$  at time  $t$  (1: mastery, 0: non-mastery),  $k = 1, \dots, 4$ ,  $t = 1, 2$ .

We first considered the proportion of correctly classified skills (i.e., skill level classification rate; referred to as *PCS*) and the proportion of correctly classified skill profile vectors (i.e., vector level classification rate; referred to as *PCV*). The PCS and PCV were calculated (1) for both two time points ( $t=1$  &  $t=2$ ) based on  $(\hat{\alpha}_{j11}, \hat{\alpha}_{j21}, \hat{\alpha}_{j31}, \hat{\alpha}_{j41}, \hat{\alpha}_{j12}, \hat{\alpha}_{j22}, \hat{\alpha}_{j32}, \hat{\alpha}_{j42})'$ , (2) for the first occasion ( $t=1$ ) based on  $(\hat{\alpha}_{j11}, \hat{\alpha}_{j21}, \hat{\alpha}_{j31}, \hat{\alpha}_{j41})'$ , and (3) for the second occasion ( $t=2$ ) based on  $(\hat{\alpha}_{j12}, \hat{\alpha}_{j22}, \hat{\alpha}_{j32}, \hat{\alpha}_{j42})'$ . The PCS and PCV were calculated using the `ClassRate()` function in the GDINA R package.

Table 6.6: Proportion of correctly classified skills (PCS) and last element of the proportion of correctly classified skill profile vectors (PCV) for the GC-DINA model and the LTA-DINA model.

Model	t=1 & t=2		t=1		t=2	
	PCS	PCV	PCS	PCV	PCS	PCV
GC-DINA	0.95	0.64	0.94	0.79	0.95	0.81
LTA-DINA	0.94	0.63	0.94	0.79	0.94	0.79

Table 6.6 shows that the GC-DINA and the LTA-DINA have very similar accuracy rates in terms of skill mastery prediction. At the skill level, based on the PCS, 94 to 95% of skill mastery indicators were correctly classified in both models at each occasion and across occasions. At the vector level, based on the last element of PCV which indicates the proportion of all elements in the vector being correctly classified, 63 to 64% of the skill mastery profiles were correctly classified across the time point; 79% were correct for the first occasion; and

79 to 81% were correct for the second occasion. Overall, the GC-DINA shows slightly better accuracy rates than the LTA-DINA.

In addition, we also compared the model-implied marginal probability of having each skill at each time point for the GC-DINA and LTA-DINA models. The model-implied marginal probabilities of skill mastery for the GC-DINA can be obtained using numerical integration over  $\theta$ . This calculation is available using the `xtrhoi` function in Stata. For the LTA-DINA model, these probabilities are obtained by multiplying prior and transition probabilities for each skill. The calculated probabilities are given in Table 6.7. The model-implied probabilities of having each skill at each time point were nearly the same for the GD-DINA and LTA-DINA. This confirms that the two models provide similar information in terms of prediction of skill mastery although they measure learning over time in different ways.

Table 6.7: Model-implied probability of having each skill at each time point for the GC-DINA model and the LTA-DINA model.

	t=1		t=2	
	GC-DINA	LTA-DINA	GC-DINA	LTA-DINA
Skill 1	0.78	0.77	0.83	0.84
Skill 2	0.23	0.23	0.27	0.28
Skill 3	0.37	0.38	0.42	0.41
Skill 4	0.58	0.59	0.64	0.64

## Predicted Latent Traits from the GC-DINA and Longitudinal Rasch

de la Torre & Douglas (2004) discussed that the higher-order latent trait can be viewed as the analog of the latent trait in traditional IRT models. They demonstrated a high correlation between the predicted higher-order latent traits and the abilities from the 2PL IRT model—the correlation of 0.96 was reported in the application using the fraction subtraction test data (Tatsuoka, 1990). This is one of the advantages of the HO-DINA model as it estimates both skill mastery and latent trait in the same analysis.

To examine if this high correlation holds for the longitudinal models, we also compared the predicted latent traits from the GC-DINA and the longitudinal Rasch models at each time point. One difference from the correlation study of de la Torre & Douglas (2004) is that we used the Rasch model instead of the 2PL model.

Figure 6.1 shows scatter plots of the latent trait predictions versus the generating values at each occasion. In the first row, the GC-DINA estimates were compared to the generating

values; the correlation is 0.67 at both time points. The second row shows the comparison of the longitudinal Rasch estimates to the generating values; the correlations are 0.59 at time 1 and 0.57 at time 2. These results suggest that our GC-DINA estimates have higher correlation with the generating values than the longitudinal Rasch estimates, which was expected as the data were generated from the GC-DINA model. In Figure 6.2, the GC-DINA estimates and the longitudinal Rasch estimates were compared each other. The correlations are 0.87 at time 1 and 0.85 at time 2, which indicates high correspondence between the two estimates.

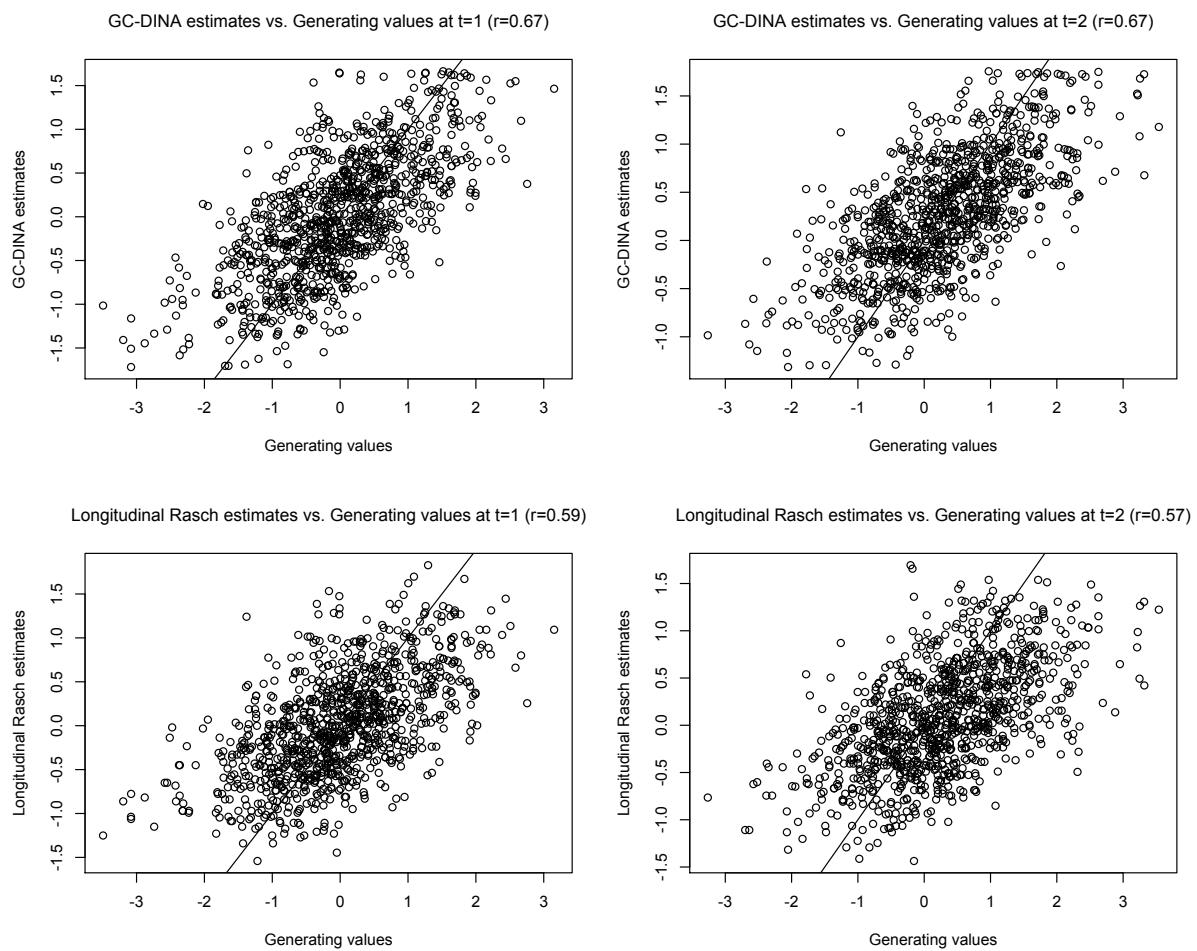


Figure 6.1: Scatter plots of predicted higher-order latent traits vs. generating values: (1) GC-DINA estimates vs. generating values and (2) longitudinal Rasch estimates vs. generating values. The estimated correlation,  $r$ , is in parenthesis.

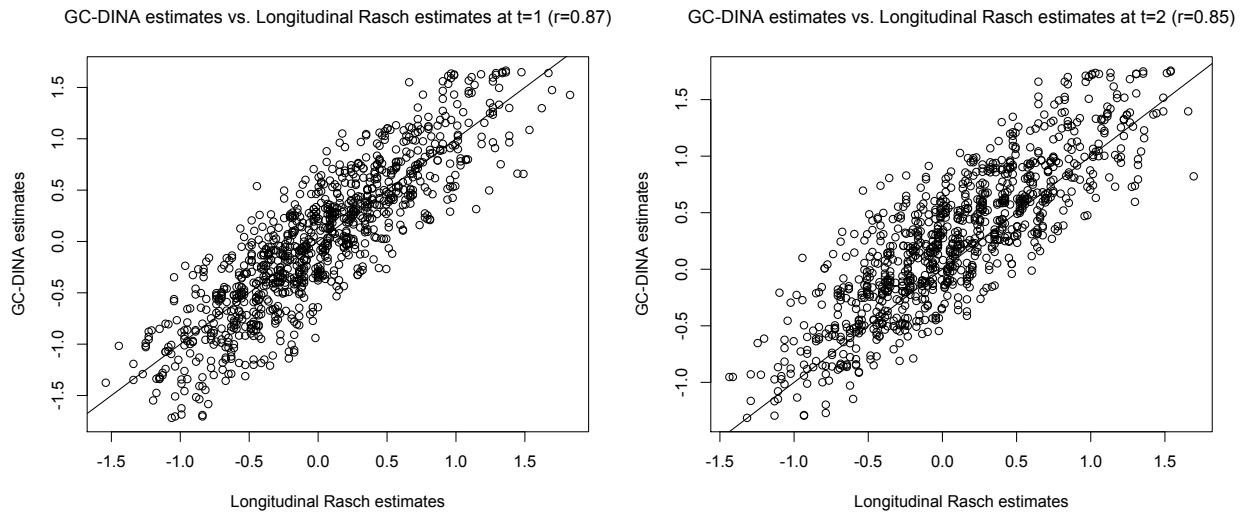


Figure 6.2: Scatter plots of predicted higher-order latent traits: GC-DINA estimates vs. longitudinal Rasch estimates. The estimated correlation,  $r$ , is in parenthesis.

# Chapter 7

## Conclusion

In this dissertation, GC-CDMs were proposed to incorporate learning over time into the cognitive assessment framework. GC-CDMs include a higher-order latent trait interpretable as a more broadly defined general ability associated with skills, and the relationship between the higher-order latent trait and skills can be specified by the Rasch model, i.e., skills play the role of items and skill masteries for different skills are conditionally independent given the higher-order latent trait. GC-CDMs trace changes in the higher-order latent trait over time by latent growth modeling, and simultaneously trace students' skill mastery through the CDM measurement model.

For estimation of GC-CDMs, we used marginal likelihood estimation with the nested integration to lessen the high-dimensional computation problem. By using the nested integration, the likelihood has the same structure as a multilevel model in which occasions are nested within persons, and the MLE requires only 3-dimensional integration regardless of the number of time points.

Relevant issues for estimating GC-CDMs were also discussed. In terms of modeling the relationship between the higher-order latent trait and the multiple skills, we showed that, for the case of a single time point, parameters of the HO-DINA model were estimated more precisely when using the Rasch model than when using the 2PL model. In addition, we addressed the model identification issue when only two time points are available. In such cases, we suggested an alternative approach of using a bivariate normal distribution to model the correlation between the higher-order latent traits at the two occasions.

Through simulation studies, parameter recovery of the GC-DINA model was examined. In the first simulation study, we focused on parameter recovery under differing conditions to investigate the effects of the following factors on model estimates: the design of the Q-matrix, the number of respondents and the number of time points. We found that the point estimates deteriorated when switching from the simple to the complex Q-matrix; and the standard errors of the estimates increased for the smaller sample size. We did not observe a

significant effect of the number of time points based on the comparison between three time points and four time points. Overall, across all conditions, the GC-DINA model estimates showed good recovery, especially the average growth (which is the parameter of interest). The results imply that the simpler design of the Q-matrix and larger sample size would be preferable for better estimates of GC-CDMs. However, this conclusion is tentative because these results were based on only one replication per model. In the second simulation study, we focused on the analysis of pre-post assessment data when only two time points are available. The GC-DINA defined for two time points was applied, and it had good parameter recovery.

The application of the GC-DINA model to real data was illustrated by using two datasets from multi-wave experiments designed to assess the effects of EAI treatments on mathematics achievement. Both the regular GC-DINA model and the GC-DINA model for two time points were applied and provided useful interpretations in terms of both growth in the higher-order latent trait and changes in skill mastery status over time.

Another contribution of this dissertation is a review of the literature on other longitudinal psychometric models—longitudinal IRT models, latent transition analysis with CDMs and dynamic Bayesian Networks. Then we examined similarities among the GC-CDM, the LTA-CDM and the longitudinal IRT model in a simulation study. Using a simulated data for two occasions from the GC-DINA model, we compared the GC-DINA model, the LTA-DINA model and the longitudinal IRT model. We found similarities between the GC-DINA model and the LTA-DINA model in terms of the predicted skill mastery; and between the GC-DINA model and the longitudinal IRT model in terms of the predicted higher-order latent trait.

There are several potential avenues for future work. In terms of model performance, future research could investigate how to improve the efficiency of estimation and develop model diagnostics for GC-CDMs. In addition, the possible extensions discussed in Section 3 could be further investigated. Specifically, it would be interesting to (1) allow skill and time specific learning and (2) incorporate respondents' covariates into the model.

# Appendix A

## R and Mplus Codes for Chapter 3

### A.1 R code generating data from the HO-DINA with the Rasch model

```

# number of respondents
J <- 1000
# number of items
I <- 20
# number of skills
K <- 4

# Q-matrix
Q <- t(matrix(c(1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,
0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,
0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1),
K,I))
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

# skill profile patterns
alpha_patt <- as.matrix(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)))
colnames(alpha_patt) <- paste0("A", 1:4)

# slip and guess
set.seed(1234)
slip <- runif(I,0.1,0.3)
guess <- runif(I,0.1,0.3)

```

```

# higher-order structural parameters
lambda0 <- c(1.51, -1.42, -0.66, 0.5)

# higher-order latent trait
theta <- rnorm(J,0,1)

# find the prob of respondent j having skill k
eta.jk <- matrix(0,J,K)
for (j in 1:J){
  for (k in 1:K){
    eta.jk[j,k] <- exp(theta[j]+lambda0[k])
    /(1+exp(theta[j]+lambda0[k]))
  }
}

# generate a respondent's true latent class (skill profile)
A <- matrix(0,J,K)
for (j in 1:J){
  for (k in 1:K){
    A[j,k] <- rbinom(1,1,eta.jk[j,k])
  }
}

# calculate if respondents have all skills needed for each item
xi_ind <- matrix(0,J,I)
for (j in 1:J){
  for (i in 1:I){
    xi_ind[j,i] <- prod(A[j,]^Q[i,])
  }
}

# generate prob of correct response and sample responses
prob.correct <- matrix(0,J,I)
y <- matrix(0,J,I)
for (j in 1:J){
  for (i in 1:I){
    prob.correct[j,i] <- ((1-slip[i])^xi_ind[j,i])
    * (guess[i]^(1-xi_ind[j,i]))
    y[j,i] <- rbinom(1,1,prob.correct[j,i])
  }
}

```

## A.2 R code generating data from the HO-DINA with the 2PL model

```

# number of respondents
J <- 1000
# number of items
I <- 20
# number of skills
K <- 4

# Q-matrix
Q <- t(matrix(c(1,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,
0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,
0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,1),
K,I))
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

# skill profile patterns
alpha_patt <- as.matrix(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)))
colnames(alpha_patt) <- paste0("A", 1:4)

# slip and guess
set.seed(1234)
slip <- runif(I,0.1,0.3)
guess <- runif(I,0.1,0.3)

# higher-order structural parameters
lambda0 <- c(1.51, -1.42, -0.66, 0.5)
lambda1 <- c(1.34, 0.65, 1.11, 0.97)

# higher-order latent trait
theta <- rnorm(J,0,1)

# find the prob of respondent j having skill k
eta.jk <- matrix(0,J,K)
for (j in 1:J){
  for (k in 1:K){
    eta.jk[j,k] <- exp(lambda1[k]*theta[j]+lambda0[k])
    /(1+exp(lambda1[k]*theta[j]+lambda0[k]))
  }
}

```

```

}

# generate a respondent's true latent class (skill profile)
A <- matrix(0,J,K)
for (j in 1:J){
  for (k in 1:K){
    A[j,k] <- rbinom(1,1,eta.jk[j,k])
  }
}

# calculate if respondents have all skills needed for each item
xi_ind <- matrix(0,J,I)
for (j in 1:J){
  for (i in 1:I){
    xi_ind[j,i] <- prod(A[j,]^Q[i,])
  }
}

# generate prob of correct response and sample responses
prob.correct <- matrix(0,J,I)
y <- matrix(0,J,I)

for (j in 1:J){
  for (i in 1:I){
    prob.correct[j,i] <- ((1-slip[i])^xi_ind[j,i])
    * (guess[i]^(1-xi_ind[j,i]))
    y[j,i] <- rbinom(1,1,prob.correct[j,i])
  }
}
}

```

### A.3 Mplus code of the HO-DINA with the Rasch model

```

TITLE: !HO-DINA example: 20 item, 4 attribute data set.
DATA:
  FILE IS data_rasch.txt;
VARIABLE:
  NAMES = x1-x20;
  USEVARIABLE = x1-x20;
  CATEGORICAL = x1-x20;

```

```

CLASSES = c1(2) c2(2) c3(2) c4(2);
ANALYSIS:
  TYPE=MIXTURE;
  STARTS=0;
  ALGORITHM=INTEGRATION;
  INTEGRATION=20;
  PROC=4;
  CONVERGENCE = 0.000001;
  H1CONVERGENCE = 0.000001;
  LOGCRITERION = 0.000001;
  RLOGCRITERION = 0.000001;
  MCONVERGENCE = 0.000001;
  MCCONVERGENCE = 0.000001;
  MUCONVERGENCE = 0.000001;

MODEL:
%OVERALL%
f BY;
c1 c2 c3 c4 ON f@1;
[f@0];
f*1;

%c1#1.c2#1.c3#1.c4#1% !for attribute pattern 1 [0,0,0,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);

```

```
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#1% !for attribute pattern 2 [1,0,0,0];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#1% !for attribute pattern 3 [0,1,0,0];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
```

```
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#2.c4#1% !for attribute pattern 4 [0,0,1,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#1.c4#2% !for attribute pattern 5 [0,0,0,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
```

```
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#1.c4#1% !for attribute pattern 6 [1,1,0,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#2.c4#1% !for attribute pattern 7 [1,0,1,0];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
```

```
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#2% !for attribute pattern 8 [1,0,0,1];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#2.c3#2.c4#1% !for attribute pattern 9 [0,1,1,0];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
```

```
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#2% !for attribute pattern 10 [0,1,0,1];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#1.c3#2.c4#2% !for attribute pattern 11 [0,0,1,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
```

```
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#2.c4#1% !for attribute pattern 12 [1,1,1,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#2.c3#1.c4#2% !for attribute pattern 13 [1,1,0,1];
```

```
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#1.c3#2.c4#2% !for attribute pattern 14 [1,0,1,1];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
```

```
[x20$1] (t20_2);

%c1#1.c2#2.c3#2.c4#2% !for attribute pattern 15 [0,1,1,1];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#2.c4#2% !for attribute pattern 16 [1,1,1,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
```

```
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

MODEL CONSTRAINT:
! ITEM 1:
NEW(l1_0 l1_e);
t1_1=-(l1_0);
t1_2=-(l1_0+l1_e);
l1_e>0;

! ITEM 2:
NEW(l2_0 l2_e);
t2_1=-(l2_0);
t2_2=-(l2_0+l2_e);
l2_e>0;

! ITEM 3:
NEW(l3_0 l3_e);
t3_1=-(l3_0);
t3_2=-(l3_0+l3_e);
l3_e>0;

! ITEM 4:
NEW(l4_0 l4_e);
t4_1=-(l4_0);
t4_2=-(l4_0+l4_e);
l4_e>0;

! ITEM 5:
NEW(l5_0 l5_e);
t5_1=-(l5_0);
t5_2=-(l5_0+l5_e);
l5_e>0;

! ITEM 6:
NEW(l6_0 l6_e);
t6_1=-(l6_0);
t6_2=-(l6_0+l6_e);
```

```
16_e >0;

! ITEM 7:
NEW(17_0 17_e);
t7_1=-(17_0);
t7_2=-(17_0+17_e);
17_e >0;

! ITEM 8:
NEW(18_0 18_e);
t8_1=-(18_0);
t8_2=-(18_0+18_e);
18_e >0;

! ITEM 9:
NEW(19_0 19_e);
t9_1=-(19_0);
t9_2=-(19_0+19_e);
19_e >0;

! ITEM 10:
NEW(110_0 110_e);
t10_1=-(110_0);
t10_2=-(110_0+110_e);
110_e >0;

! ITEM 11:
NEW(111_0 111_e);
t11_1=-(111_0);
t11_2=-(111_0+111_e);
111_e >0;

! ITEM 12:
NEW(112_0 112_e);
t12_1=-(112_0);
t12_2=-(112_0+112_e);
112_e >0;

! ITEM 13:
NEW(113_0 113_e);
t13_1=-(113_0);
```

```
t13_2=-(l13_0+l13_e);
l13_e>0;

! ITEM 14:
NEW(l14_0 l14_e);
t14_1=-(l14_0);
t14_2=-(l14_0+l14_e);
l14_e>0;

! ITEM 15:
NEW(l15_0 l15_e);
t15_1=-(l15_0);
t15_2=-(l15_0+l15_e);
l15_e>0;

! ITEM 16:
NEW(l16_0 l16_e);
t16_1=-(l16_0);
t16_2=-(l16_0+l16_e);
l16_e>0;

! ITEM 17:
NEW(l17_0 l17_e);
t17_1=-(l17_0);
t17_2=-(l17_0+l17_e);
l17_e>0;

! ITEM 18:
NEW(l18_0 l18_e);
t18_1=-(l18_0);
t18_2=-(l18_0+l18_e);
l18_e>0;

! ITEM 19:
NEW(l19_0 l19_e);
t19_1=-(l19_0);
t19_2=-(l19_0+l19_e);
l19_e>0;

! ITEM 20:
NEW(l20_0 l20_e);
```

```
t20_1=-(120_0);
t20_2=-(120_0+120_e);
120_e>0;
```

#### A.4 MODEL command syntax of the Mplus code of the HO-DINA with the 2PL model

The same code as A.3 except for the following command:

```
MODEL:
%OVERALL%
f BY;
c1 c2 c3 c4 ON f*1;
[f*0];
f@1;
```

## Appendix B

### R and Mplus Codes for Chapter 4

#### B.1 R code for the simulation study for comparison between the simple and complex Q-matrix for the simple DINA model

```
# load the GDINA package
library(GDINA)
# load the mclust package
library(mclust)

# number of respondents
J <- 1000
# number of items
I <- 20
# number of skills
K <- 4
# number of skill mastery profiles
C <- 2^K

# skill profile patterns
alpha_patt <- as.matrix(expand.grid(c(0,1), c(0,1), c(0,1), c(0,1)))
colnames(alpha_patt) <- paste0("A", 1:4)

# probabilities that respondents master each skill
eta <- c()
eta[1] <- 0.3
```

```

eta[2] <- 0.6
eta[3] <- 0.8
eta[4] <- 0.2
eta[5] <- 0.7

# probabilities for the 16 skill profiles
alpha_prob <- rep(1, nrow(alpha_patt))
for (i in 1:nrow(alpha_patt)) {
  for (j in 1:ncol(alpha_patt)) {
    alpha_prob[i] <- alpha_prob[i] * eta[j]^alpha_patt[i, j]
    * (1 - eta[j])^(1 - alpha_patt[i, j])
  }
}

# slip and guess
# following values were generated from runif(I,0.1,0.3)
slip <- c(0.192,0.260,0.119,0.291,0.143,0.182,0.237,0.209,0.134,0.241,
0.238,0.206,0.279,0.164, 0.266,0.256,0.118,0.291,0.210,0.264)
guess <- c(0.201,0.242,0.263,0.122,0.230,0.186,0.119,0.117,0.174,0.205,
0.274,0.123,0.265,0.278, 0.293,0.233,0.133,0.165,0.150,0.283)

# Generate a respondent's true latent skill profile
ind <- sample(x = 1:C, size = J, replace = TRUE, prob = alpha_prob)
A <- alpha_patt[ind, ] # true skill profiles

# 1. DINA with the simple Q-matrix
# the simple Q-matrix
Q_simple <- t(matrix(c(1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,
0,1,0,0,0,1,0,0,0,1,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1),K,I))
rownames(Q_simple) <- paste0("Item", 1:I)
colnames(Q_simple) <- paste0("A", 1:K)

# calculate an indicator whether respondents have all skills
# needed for each item
xi_ind_simple <- matrix(0, J, I)
for (j in 1:J) {
  for (i in 1:I) {
    xi_ind_simple[j, i] <- prod(A[j, ]^Q_simple[i, ])
  }
}

```

```

}

# generate probability correct and sample responses
prob_correct_simple <- matrix(0, J, I)
y_simple <- matrix(0, J, I)
for (j in 1:J) {
  for (i in 1:I) {
    prob_correct_simple[j, i] <- ((1 - slip[i])^xi_ind_simple[j, i])
    * (guess[i]^(1 - xi_ind_simple[j, i]))
    y_simple[j, i] <- rbinom(1, 1, prob_correct_simple[j, i])
  }
}

# fit the DINA model
mod_simpleQ <- GDINA(dat = y_simple, Q = Q_simple, model = "DINA")

# calculate PCS and PCV
CR_simple <- ClassRate(A, personparm(mod_simpleQ))
CR_simple

# 2. DINA with the complex Q-matrix
# the complex Q-matrix
Q_complex<-matrix(0,I,K)
Q_complex[c(1:8,11,12,14,16:20),1]<-1
Q_complex[c(3:12,14,15),2]<-1
Q_complex[c(11:14,20),3]<-1
Q_complex[c(10:12,14:16),4]<-1
rownames(Q_complex) <- paste0("Item", 1:I)
colnames(Q_complex) <- paste0("A", 1:K)

# calculate an indicator whether respondents have all skills
# needed for each item
xi_ind_complex <- matrix(0, J, I)
for (j in 1:J) {
  for (i in 1:I) {
    xi_ind_complex[j, i] <- prod(A[j, ]^Q_complex[i, ])
  }
}

# generate probability correct and sample responses
prob_correct_complex <- matrix(0, J, I)

```

```

y_complex <- matrix(0, J, I)
for (j in 1:J) {
  for (i in 1:I) {
    prob_correct_complex[j, i] <- ((1 - slip[i])^xi_ind_complex[j, i])
    * (guess[i]^(1 - xi_ind_complex[j, i]))
    y_complex[j, i] <- rbinom(1, 1, prob_correct_complex[j, i])
  }
}

# fit the DINA model
mod_complexQ <- GDINA(dat = y_complex, Q = Q_complex, model = "DINA")

# calculate PCS and PCV
CR_complex <- ClassRate(A, personparm(mod_complexQ))
CR_complex

# create skill profile labels for the true profile
A_label<-c()
for (i in 1:J){
  A_label[i] <- as.numeric(paste(A[i,], collapse = ""))
}

# create labels for the predicted skill profiles with the simple Q
model_simple_label<-c()
for (i in 1:J){
  model_simple_label[i]
    <- as.numeric(paste(personparm(mod_simpleQ)[i,], collapse = ""))
}

# create labels for the predicted skill profiles with the complex Q
model_complex_label<-c()
for (i in 1:J){
  model_complex_label[i]
    <- as.numeric(paste(personparm(mod_complexQ)[i,], collapse = ""))
}

# calculate ARI
adjustedRandIndex(A_label, model_simple_label)
adjustedRandIndex(A_label, model_complex_label)

```

## B.2 R code generating six data sets for Model 1 to Model 6

### Model 1

```

# load required packages
library(MASS)
library(boot)

# number of respondents
J<-1000
# number of items
I<-20
# number of skills
K<-4
# number of time points
t<-3

# id & time
id<-rep(1:J,each=t)
time<-rep(c(0:(t-1)),J)

# covariance matrix for random effects
psi_11<-0.4
psi_22<-0.02
cov_12<-0.02
var_re<-matrix(c(psi_11,cov_12,cov_12,psi_22),2,2)

# generate random effects
set.seed(1)
re<-mvrnorm(n=J,mu=c(0,0.3),Sigma=var_re)

fb <- rep(re[,1],each=t) # random intercept
rs <- rep(re[,2],each=t) # beta+random slope
fw <- rnorm(J*t,0,sqrt(0.6)) #level-1 errors

# intercepts for the higher-order Rascho model
lambda0 <- c(1.51, -1.42, -0.66, 0.5)
lambda1 <- 1

```

```

# Q-matrix
Q <- t(matrix(c(1,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,
0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,
0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1),
K,I))
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

# skill profile patterns
alpha_patt <- as.matrix(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)))
colnames(alpha_patt) <- paste0("A", 1:4)
alpha_patt

# slip and guess
slip <- c(0.192,0.260,0.119,0.291,0.143,0.182,0.237,0.209,0.134,
0.241,0.238,0.206,0.279,0.164, 0.266,0.256,0.118,0.291,0.210,0.264)
guess <- c(0.201,0.242,0.263,0.122,0.230,0.186,0.119,0.117,0.174,
0.205,0.274,0.123,0.265,0.278,0.293,0.233,0.133,0.165,0.150,0.283)

# true higher-order latent trait
ability <- fb + fw + rs*time

# prob of respondent j having skill k
eta_jk <- matrix(NA, J*t, K)
for (j in 1:(J*t)){
    eta_jk[j,] <- inv.logit(ability[j]+lambda0)
}

# generate a respondent's true skill profile
A <- matrix(NA, J*t, K)
for (j in 1:(J*t)){
    for (k in 1:K) {
        A[j, k] <- rbinom(1, 1, eta_jk[j, k])
    }
}

# calculate if respondents have all skills needed for each item
xi_ind <- matrix(NA, J*t, I)
for (j in 1:(J*t)) {
    for (i in 1:I) {
        xi_ind[j, i] <- prod(A[j, ]^Q[i, ])
    }
}

```

```

}

# generate prob correct and sample responses
prob.correct <- matrix(NA, J*t, I)
y <- matrix(NA, J*t, I)
for (j in 1:(J*t)){
  for (i in 1:I) {
    prob.correct[j, i] <- ((1 - slip[i])^xi_ind[j, i])
    * (guess[i]^(1 - xi_ind[j,i]))
    y[j,i] <- rbinom(1,1,prob.correct[j,i])
  }
}

```

## Model 2 to Model 6

- For Model 2, the same R code as Model 1 was used except for the Q-matrix specification.

```

# Q-matrix
Q<-matrix(0,I,K)
Q[c(1:8,11,12,14,16:20),1]<-1
Q[c(3:12,14,15),2]<-1
Q[c(11:14,20),3]<-1
Q[c(10:12,14:16),4]<-1
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

```

- For Model 3, the same R code as Model 1 was used except for the number of respondents.

```

# number of respondents
J<-500

```

- For Model 4, the same R code as Model 1 was used except for the Q-matrix and the number of respondents.

```

# number of respondents
J<-500

# Q-matrix
Q<-matrix(0,I,K)
Q[c(1:8,11,12,14,16:20),1]<-1
Q[c(3:12,14,15),2]<-1

```

```

Q[c(11:14,20),3]<-1
Q[c(10:12,14:16),4]<-1
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

```

- For Model 5, the same R code as Model 1 was used except for the number of time points and the number of respondents.

```

# number of respondents
J<-500

```

```

# number of time points
t<-4

```

- For Model 6, the same R code as Model 1 was used except for the Q-matrix, the number of time points and the number of respondents.

```

# number of respondents
J<-500

```

```

# number of time points
t<-4

```

```

# Q-matrix
Q<-matrix(0,I,K)
Q[c(1:8,11,12,14,16:20),1]<-1
Q[c(3:12,14,15),2]<-1
Q[c(11:14,20),3]<-1
Q[c(10:12,14:16),4]<-1
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

```

### B.3 Mplus code for fitting the GC-DINA model for Model 1 to Model 6

Mplus code used for Model 1, Model 3, and Model 5 (with the simple Q-matrix)

```
TITLE: !GC-DINA: 20 item, 4 attribute data set.
```

```

DATA:
  FILE IS responses.txt;
VARIABLE:
  NAMES = id time x1-x20;
  USEVARIABLE = id time x1-x20;
  CLUSTER = id;
  CATEGORICAL = x1-x20;
  CLASSES = c1(2) c2(2) c3(2) c4(2);
  WITHIN = time x1-x20;
ANALYSIS:
  TYPE=MIXTURE TWOLEVEL RANDOM;
  ALGORITHM=INTEGRATION;
  ESTIMATOR IS ML;
  LINK IS LOGIT;
  PROC=4; STARTS=0;
  CONVERGENCE = 0.000001;
  H1CONVERGENCE = 0.000001;
  LOGCRITERION = 0.000001;
  RLOGCRITERION = 0.000001;
  MCONVERGENCE = 0.000001;
  MCCONVERGENCE = 0.000001;
  MUCONVERGENCE = 0.000001;

MODEL:
%WITHIN%
%OVERALL%
fw BY;
c1 c2 c3 c4 on fw@1;
s | fw ON time;

%BETWEEN%
%OVERALL%
fb BY;
c1 c2 c3 c4 on fb@1;
[s] (1);
fb WITH s;

%WITHIN%
%c1#1.c2#1.c3#1.c4#1% !for attribute pattern 1 [0,0,0,0];
[x1$1] (t1_1);
[x2$1] (t2_1);

```

```
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#1% !for attribute pattern 2 [1,0,0,0];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);
```

```
%c1#1.c2#2.c3#1.c4#1% !for attribute pattern 3 [0,1,0,0];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#2.c4#1% !for attribute pattern 4 [0,0,1,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
```

```
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#1.c4#2% !for attribute pattern 5 [0,0,0,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#1.c4#1% !for attribute pattern 6 [1,1,0,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
```

```
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#2.c4#1% !for attribute pattern 7 [1,0,1,0];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#2% !for attribute pattern 8 [1,0,0,1];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
```

```
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#2.c3#2.c4#1% !for attribute pattern 9 [0,1,1,0];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#2% !for attribute pattern 10 [0,1,0,1];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
```

```
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#1.c3#2.c4#2% !for attribute pattern 11 [0,0,1,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#2.c4#1% !for attribute pattern 12 [1,1,1,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
```

```
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#2.c3#1.c4#2% !for attribute pattern 13 [1,1,0,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#1.c3#2.c4#2% !for attribute pattern 14 [1,0,1,1];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
```

```
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#2.c3#2.c4#2% !for attribute pattern 15 [0,1,1,1];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#2.c4#2% !for attribute pattern 16 [1,1,1,1];
```

```
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

MODEL CONSTRAINT:
! ITEM 1:
NEW(l1_0 l1_e);
t1_1=-(l1_0);
t1_2=-(l1_0+l1_e);
l1_e>0;

! ITEM 2:
NEW(l2_0 l2_e);
t2_1=-(l2_0);
t2_2=-(l2_0+l2_e);
l2_e>0;

! ITEM 3:
NEW(l3_0 l3_e);
t3_1=-(l3_0);
t3_2=-(l3_0+l3_e);
l3_e>0;

! ITEM 4:
```

```
NEW(14_0 14_e);
t4_1=-(14_0);
t4_2=-(14_0+14_e);
14_e>0;

! ITEM 5:
NEW(15_0 15_e);
t5_1=-(15_0);
t5_2=-(15_0+15_e);
15_e>0;

! ITEM 6:
NEW(16_0 16_e);
t6_1=-(16_0);
t6_2=-(16_0+16_e);
16_e>0;

! ITEM 7:
NEW(17_0 17_e);
t7_1=-(17_0);
t7_2=-(17_0+17_e);
17_e>0;

! ITEM 8:
NEW(18_0 18_e);
t8_1=-(18_0);
t8_2=-(18_0+18_e);
18_e>0;

! ITEM 9:
NEW(19_0 19_e);
t9_1=-(19_0);
t9_2=-(19_0+19_e);
19_e>0;

! ITEM 10:
NEW(110_0 110_e);
t10_1=-(110_0);
t10_2=-(110_0+110_e);
110_e>0;
```

```
! ITEM 11:  
NEW(l11_0 l11_e);  
t11_1=-(l11_0);  
t11_2=-(l11_0+l11_e);  
l11_e>0;  
  
! ITEM 12:  
NEW(l12_0 l12_e);  
t12_1=-(l12_0);  
t12_2=-(l12_0+l12_e);  
l12_e>0;  
  
! ITEM 13:  
NEW(l13_0 l13_e);  
t13_1=-(l13_0);  
t13_2=-(l13_0+l13_e);  
l13_e>0;  
  
! ITEM 14:  
NEW(l14_0 l14_e);  
t14_1=-(l14_0);  
t14_2=-(l14_0+l14_e);  
l14_e>0;  
  
! ITEM 15:  
NEW(l15_0 l15_e);  
t15_1=-(l15_0);  
t15_2=-(l15_0+l15_e);  
l15_e>0;  
  
! ITEM 16:  
NEW(l16_0 l16_e);  
t16_1=-(l16_0);  
t16_2=-(l16_0+l16_e);  
l16_e>0;  
  
! ITEM 17:  
NEW(l17_0 l17_e);  
t17_1=-(l17_0);  
t17_2=-(l17_0+l17_e);
```

```

l17_e>0;

! ITEM 18:
NEW(l18_0 l18_e);
t18_1=-(l18_0);
t18_2=-(l18_0+l18_e);
l18_e>0;

! ITEM 19:
NEW(l19_0 l19_e);
t19_1=-(l19_0);
t19_2=-(l19_0+l19_e);
l19_e>0;

! ITEM 20:
NEW(l20_0 l20_e);
t20_1=-(l20_0);
t20_2=-(l20_0+l20_e);
l20_e>0;

OUTPUT:
  TECH1 TECH3 TECH4 TECH5 TECH8;
SAVEDATA:
  FORMAT IS f10.5;
  FILE IS predicted.dat;
  SAVE = CPROBABILITIES FScores;;

```

### Mplus code used for Model 2, Model 4, and Model 6 (with the complex Q-matrix)

```

TITLE: !GC-DINA: 20 item, 4 attribute data set.
DATA:
  FILE IS responses.txt;
VARIABLE:
  NAMES = id time x1-x20;
  USEVARIABLE = id time x1-x20;
  CLUSTER = id;
  CATEGORICAL = x1-x20;
  CLASSES = c1(2) c2(2) c3(2) c4(2);
  WITHIN = time x1-x20;
ANALYSIS:

```

```

TYPE=MIXTURE TWOLEVEL RANDOM;
ALGORITHM=INTEGRATION;
ESTIMATOR IS ML;
LINK IS LOGIT;
PROC=4; STARTS=0;
CONVERGENCE = 0.000001;
H1CONVERGENCE = 0.000001;
LOGCRITERION = 0.000001;
RLOGCRITERION = 0.000001;
MCONVERGENCE = 0.000001;
MCCONVERGENCE = 0.000001;
MUCONVERGENCE = 0.000001;

MODEL:
%WITHIN%
%OVERALL%
fw BY;
c1 c2 c3 c4 on fw@1;
s | fw ON time;

%BETWEEN%
%OVERALL%
fb BY;
c1 c2 c3 c4 on fb@1;
[s] (1);
fb WITH s;

%WITHIN%
%c1#1.c2#1.c3#1.c4#1% !for attribute pattern 1 [0,0,0,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);

```

```
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#1% !for attribute pattern 2 [1,0,0,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#1% !for attribute pattern 3 [0,1,0,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
```

```
[x9$1] (t9_2);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

% c1#1.c2#1.c3#2.c4#1% !for attribute pattern 4 [0,0,1,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

% c1#1.c2#1.c3#1.c4#2% !for attribute pattern 5 [0,0,0,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
```

```
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

% c1#2.c2#2.c3#1.c4#1% !for attribute pattern 6 [1,1,0,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_1);
```

```
%c1#2.c2#1.c3#2.c4#1% !for attribute pattern 7 [1,0,1,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#1.c3#1.c4#2% !for attribute pattern 8 [1,0,0,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
```

```
[x19$1] (t19_2);
[x20$1] (t20_1);

%c1#1.c2#2.c3#2.c4#1% !for attribute pattern 9 [0,1,1,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#2% !for attribute pattern 10 [0,1,0,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
```

```
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#2.c4#2% !for attribute pattern 11 [0,0,1,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#2.c3#2.c4#1% !for attribute pattern 12 [1,1,1,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
```

```
[x9$1] (t9_2);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#1.c4#2% !for attribute pattern 13 [1,1,0,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_1);

%c1#2.c2#1.c3#2.c4#2% !for attribute pattern 14 [1,0,1,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_1);
```

```
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

% c1#1.c2#2.c3#2.c4#2% !for attribute pattern 15 [0,1,1,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);
```

```
%c1#2.c2#2.c3#2.c4#2% !for attribute pattern 16 [1,1,1,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

MODEL CONSTRAINT:
! ITEM 1:
NEW(l1_0 l1_e);
t1_1=-(l1_0);
t1_2=-(l1_0+l1_e);
l1_e>0;

! ITEM 2:
NEW(l2_0 l2_e);
t2_1=-(l2_0);
t2_2=-(l2_0+l2_e);
l2_e>0;

! ITEM 3:
NEW(l3_0 l3_e);
t3_1=-(l3_0);
t3_2=-(l3_0+l3_e);
l3_e>0;
```

```
! ITEM 4:  
NEW(14_0 14_e);  
t4_1=-(14_0);  
t4_2=-(14_0+14_e);  
14_e>0;  
  
! ITEM 5:  
NEW(15_0 15_e);  
t5_1=-(15_0);  
t5_2=-(15_0+15_e);  
15_e>0;  
  
! ITEM 6:  
NEW(16_0 16_e);  
t6_1=-(16_0);  
t6_2=-(16_0+16_e);  
16_e>0;  
  
! ITEM 7:  
NEW(17_0 17_e);  
t7_1=-(17_0);  
t7_2=-(17_0+17_e);  
17_e>0;  
  
! ITEM 8:  
NEW(18_0 18_e);  
t8_1=-(18_0);  
t8_2=-(18_0+18_e);  
18_e>0;  
  
! ITEM 9:  
NEW(19_0 19_e);  
t9_1=-(19_0);  
t9_2=-(19_0+19_e);  
19_e>0;  
  
! ITEM 10:  
NEW(110_0 110_e);  
t10_1=-(110_0);
```

```
t10_2=-(l10_0+l10_e);
l10_e>0;

! ITEM 11:
NEW(l11_0 l11_e);
t11_1=-(l11_0);
t11_2=-(l11_0+l11_e);
l11_e>0;

! ITEM 12:
NEW(l12_0 l12_e);
t12_1=-(l12_0);
t12_2=-(l12_0+l12_e);
l12_e>0;

! ITEM 13:
NEW(l13_0 l13_e);
t13_1=-(l13_0);
t13_2=-(l13_0+l13_e);
l13_e>0;

! ITEM 14:
NEW(l14_0 l14_e);
t14_1=-(l14_0);
t14_2=-(l14_0+l14_e);
l14_e>0;

! ITEM 15:
NEW(l15_0 l15_e);
t15_1=-(l15_0);
t15_2=-(l15_0+l15_e);
l15_e>0;

! ITEM 16:
NEW(l16_0 l16_e);
t16_1=-(l16_0);
t16_2=-(l16_0+l16_e);
l16_e>0;

! ITEM 17:
NEW(l17_0 l17_e);
```

```

t17_1=-(l17_0);
t17_2=-(l17_0+l17_e);
l17_e>0;

! ITEM 18:
NEW(l18_0 l18_e);
t18_1=-(l18_0);
t18_2=-(l18_0+l18_e);
l18_e>0;

! ITEM 19:
NEW(l19_0 l19_e);
t19_1=-(l19_0);
t19_2=-(l19_0+l19_e);
l19_e>0;

! ITEM 20:
NEW(l20_0 l20_e);
t20_1=-(l20_0);
t20_2=-(l20_0+l20_e);
l20_e>0;

OUTPUT:
TECH1 TECH3 TECH4 TECH5 TECH8;
SAVEDATA:
  FORMAT IS f10.5;
  FILE IS predicted.dat;
  SAVE = CPROBABILITIES FSCORES;;

```

## B.4 R code generating data for simulation study 2

```

# load required packages
library(MASS)
library(boot)

# number of respondents
J<-1000
# number of items
I<-20
# number of skills
K<-4

```

```

# Q-matrix
Q <- t(matrix(c(1,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,
0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,
0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1),K,I))
rownames(Q) <- paste0("Item", 1:I)
colnames(Q) <- paste0("A", 1:K)

# skill profile patterns
alpha_patt <- as.matrix(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)))
colnames(alpha_patt) <- paste0("A", 1:4)
alpha_patt

# slip and guess
slip <- c(0.192,0.260,0.119,0.291,0.143,0.182,0.237,0.209,0.134,
0.241,0.238,0.206,0.279,0.164, 0.266,0.256,0.118,0.291,0.210,0.264)
guess <- c(0.201,0.242,0.263,0.122,0.230,0.186,0.119,0.117,0.174,
0.205,0.274,0.123,0.265,0.278,0.293,0.233,0.133,0.165,0.150,0.283)

# generate higher-order latent traits at two time points
set.seed(1234)
theta <- mvrnorm(n=J,mu=c(0,0.3),Sigma=matrix(c(1,.8,.8,1),2,2))

# structural model parameters
lambda0 <- c(1.51, -1.42, -0.66, 0.5)

# generate true skill mastery profiles and sample responses
resp<-array(NA, dim=c(J,I,2))
A_all<-array(NA, dim=c(J,K,2))

for (t in 1:2){
# find the prob of respondent j having skill k
eta.jk <- matrix(0, J, K)
for (j in 1:J) {
  for (k in 1:K) {
    eta.jk[j, k]
    <- exp(theta[j,t] + lambda0[k])/(1 + exp(theta[j,t] + lambda0[k]))
  }
}
# generate a respondent's true skill profile

```

```

A <- matrix(0, J, K)
for (j in 1:J) {
  for (k in 1:K) {
    A[j, k] <- rbinom(1, 1, eta.jk[j, k])
  }
}

# calculate if respondents have all skills needed for each item
xi_ind <- matrix(0, J, I)
for (j in 1:J) {
  for (i in 1:I) {
    xi_ind[j, i] <- prod(A[j, ]^Q[i, ])
  }
}

# generate prob correct and sample responses
prob.correct <- matrix(0, J, I)
y <- matrix(0, J, I)

for (j in 1:J) {
  for (i in 1:I) {
    prob.correct[j, i] <- ((1 - slip[i])^xi_ind[j, i])
    * (guess[i]^(1 - xi_ind[j, i]))
    y[j, i] <- rbinom(1, 1, prob.correct[j, i])
  }
}

A_all[,,t]<-A
resp[,,t]<-y
}

skill_data<-cbind(A_all[,1],A_all[,2])
resp_data<-cbind(resp[,1],resp[,2])

```

## B.5 Mplus code for fitting the GC-DINA in simulation study 2

```

TITLE: !GC-DINA model for T=2
DATA:
FILE IS responses_t2.txt;

```

## VARIABLE:

```

NAMES = x1-x20 y1-y20; !list of variables in input file
USEVARIABLE = x1-x20 y1-y20;
CATEGORICAL = x1-x20 y1-y20;
CLASSES = c1(2) c2(2) c3(2) c4(2) c5(2) c6(2) c7(2) c8(2);

```

## ANALYSIS:

```

TYPE=MIXTURE;
STARTS=0;
ALGORITHM=INTEGRATION;

```

## MODEL:

## %OVERALL%

```

f1 BY;
f2 BY;
c1 c2 c3 c4 ON f1@1;
c5 c6 c7 c8 ON f2@1;
[f1@0];
[f2*0.5];
f1*1;
f2*1;
f1-f2 WITH f1-f2;

```

```

[c1#1] (inter1);
[c2#1] (inter2);
[c3#1] (inter3);
[c4#1] (inter4);
[c5#1] (inter1);
[c6#1] (inter2);
[c7#1] (inter3);
[c8#1] (inter4);

```

## MODEL c1.c2.c3.c4:

```

%c1#1.c2#1.c3#1.c4#1% !for attribute pattern 1 [0,0,0,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);

```

```
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#1% !for attribute pattern 2 [1,0,0,0];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#1% !for attribute pattern 3 [0,1,0,0];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
```

```
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#2.c4#1% !for attribute pattern 4 [0,0,1,0];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#1.c3#1.c4#2% !for attribute pattern 5 [0,0,0,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
```

```
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#1.c4#1% !for attribute pattern 6 [1,1,0,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);
```

```
%c1#2.c2#1.c3#2.c4#1% !for attribute pattern 7 [1,0,1,0];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#1.c3#1.c4#2% !for attribute pattern 8 [1,0,0,1];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
```

```
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#2.c3#2.c4#1% !for attribute pattern 9 [0,1,1,0];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_1);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#1.c2#2.c3#1.c4#2% !for attribute pattern 10 [0,1,0,1];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
```

```
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#1.c3#2.c4#2% !for attribute pattern 11 [0,0,1,1];
[x1$1] (t1_1);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#2.c4#1% !for attribute pattern 12 [1,1,1,0];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
```

```
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_1);
[x16$1] (t16_1);
[x17$1] (t17_2);
[x18$1] (t18_1);
[x19$1] (t19_1);
[x20$1] (t20_1);

%c1#2.c2#2.c3#1.c4#2% !for attribute pattern 13 [1,1,0,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_1);
[x5$1] (t5_1);
[x6$1] (t6_1);
[x7$1] (t7_1);
[x8$1] (t8_1);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#1.c3#2.c4#2% !for attribute pattern 14 [1,0,1,1];
[x1$1] (t1_2);
[x2$1] (t2_1);
[x3$1] (t3_1);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_1);
```

```
[x10$1] (t10_1);
[x11$1] (t11_1);
[x12$1] (t12_1);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#1.c2#2.c3#2.c4#2% !for attribute pattern 15 [0,1,1,1];
[x1$1] (t1_1);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_1);
[x14$1] (t14_1);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_1);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

%c1#2.c2#2.c3#2.c4#2% !for attribute pattern 16 [1,1,1,1];
[x1$1] (t1_2);
[x2$1] (t2_2);
[x3$1] (t3_2);
[x4$1] (t4_2);
[x5$1] (t5_2);
[x6$1] (t6_2);
```

```
[x7$1] (t7_2);
[x8$1] (t8_2);
[x9$1] (t9_2);
[x10$1] (t10_2);
[x11$1] (t11_2);
[x12$1] (t12_2);
[x13$1] (t13_2);
[x14$1] (t14_2);
[x15$1] (t15_2);
[x16$1] (t16_2);
[x17$1] (t17_2);
[x18$1] (t18_2);
[x19$1] (t19_2);
[x20$1] (t20_2);

MODEL c5.c6.c7.c8:
%c5#1.c6#1.c7#1.c8#1% !for attribute pattern 1 [0,0,0,0];
[y1$1] (t1_1);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_1);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#2.c6#1.c7#1.c8#1% !for attribute pattern 2 [1,0,0,0];
[y1$1] (t1_2);
[y2$1] (t2_1);
```

```
[y3$1] (t3_1);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_2);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#1.c6#2.c7#1.c8#1% !for attribute pattern 3 [0,1,0,0];
[y1$1] (t1_1);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_1);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);
```

```
%c5#1.c6#1.c7#2.c8#1% !for attribute pattern 4 [0,0,1,0];
[y1$1] (t1_1);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_1);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#1.c6#1.c7#1.c8#2% !for attribute pattern 5 [0,0,0,1];
[y1$1] (t1_1);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_1);
[y18$1] (t18_2);
```

```
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#2.c6#2.c7#1.c8#1% !for attribute pattern 6 [1,1,0,0];
[y1$1] (t1_2);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_2);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#2.c6#1.c7#2.c8#1% !for attribute pattern 7 [1,0,1,0];
[y1$1] (t1_2);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_1);
```

```
[y16$1] (t16_1);
[y17$1] (t17_2);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#2.c6#1.c7#1.c8#2% !for attribute pattern 8 [1,0,0,1];
[y1$1] (t1_2);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_2);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#1.c6#2.c7#2.c8#1% !for attribute pattern 9 [0,1,1,0];
[y1$1] (t1_1);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
```

```
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_1);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#1.c6#2.c7#1.c8#2% !for attribute pattern 10 [0,1,0,1];
[y1$1] (t1_1);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_1);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#1.c6#1.c7#2.c8#2% !for attribute pattern 11 [0,0,1,1];
[y1$1] (t1_1);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_1);
```

```
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_1);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#2.c6#2.c7#2.c8#1% !for attribute pattern 12 [1,1,1,0];
[y1$1] (t1_2);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_1);
[y16$1] (t16_1);
[y17$1] (t17_2);
[y18$1] (t18_1);
[y19$1] (t19_1);
[y20$1] (t20_1);

%c5#2.c6#2.c7#1.c8#2% !for attribute pattern 13 [1,1,0,1];
[y1$1] (t1_2);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_1);
[y5$1] (t5_1);
[y6$1] (t6_1);
```

```

[y7$1] (t7_1);
[y8$1] (t8_1);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_2);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#2.c6#1.c7#2.c8#2% !for attribute pattern 14 [1,0,1,1];
[y1$1] (t1_2);
[y2$1] (t2_1);
[y3$1] (t3_1);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_1);
[y10$1] (t10_1);
[y11$1] (t11_1);
[y12$1] (t12_1);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_2);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#1.c6#2.c7#2.c8#2% !for attribute pattern 15 [0,1,1,1];
[y1$1] (t1_1);
[y2$1] (t2_2);
[y3$1] (t3_2);

```

```
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_1);
[y14$1] (t14_1);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_1);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);

%c5#2.c6#2.c7#2.c8#2% !for attribute pattern 16 [1,1,1,1];
[y1$1] (t1_2);
[y2$1] (t2_2);
[y3$1] (t3_2);
[y4$1] (t4_2);
[y5$1] (t5_2);
[y6$1] (t6_2);
[y7$1] (t7_2);
[y8$1] (t8_2);
[y9$1] (t9_2);
[y10$1] (t10_2);
[y11$1] (t11_2);
[y12$1] (t12_2);
[y13$1] (t13_2);
[y14$1] (t14_2);
[y15$1] (t15_2);
[y16$1] (t16_2);
[y17$1] (t17_2);
[y18$1] (t18_2);
[y19$1] (t19_2);
[y20$1] (t20_2);
```

MODEL CONSTRAINT:

```
! ITEM 1:  
NEW(11_0 11_e);  
t1_1=-(11_0);  
t1_2=-(11_0+11_e);  
11_e>0;  
  
! ITEM 2:  
NEW(12_0 12_e);  
t2_1=-(12_0);  
t2_2=-(12_0+12_e);  
12_e>0;  
  
! ITEM 3:  
NEW(13_0 13_e);  
t3_1=-(13_0);  
t3_2=-(13_0+13_e);  
13_e>0;  
  
! ITEM 4:  
NEW(14_0 14_e);  
t4_1=-(14_0);  
t4_2=-(14_0+14_e);  
14_e>0;  
  
! ITEM 5:  
NEW(15_0 15_e);  
t5_1=-(15_0);  
t5_2=-(15_0+15_e);  
15_e>0;  
  
! ITEM 6:  
NEW(16_0 16_e);  
t6_1=-(16_0);  
t6_2=-(16_0+16_e);  
16_e>0;  
  
! ITEM 7:  
NEW(17_0 17_e);  
t7_1=-(17_0);  
t7_2=-(17_0+17_e);  
17_e>0;
```

```
! ITEM 8:  
NEW(18_0 18_e);  
t8_1=-(18_0);  
t8_2=-(18_0+18_e);  
18_e>0;  
  
! ITEM 9:  
NEW(19_0 19_e);  
t9_1=-(19_0);  
t9_2=-(19_0+19_e);  
19_e>0;  
  
! ITEM 10:  
NEW(110_0 110_e);  
t10_1=-(110_0);  
t10_2=-(110_0+110_e);  
110_e>0;  
  
! ITEM 11:  
NEW(111_0 111_e);  
t11_1=-(111_0);  
t11_2=-(111_0+111_e);  
111_e>0;  
  
! ITEM 12:  
NEW(112_0 112_e);  
t12_1=-(112_0);  
t12_2=-(112_0+112_e);  
112_e>0;  
  
! ITEM 13:  
NEW(113_0 113_e);  
t13_1=-(113_0);  
t13_2=-(113_0+113_e);  
113_e>0;  
  
! ITEM 14:  
NEW(114_0 114_e);  
t14_1=-(114_0);  
t14_2=-(114_0+114_e);
```

```
l14_e>0;

! ITEM 15:
NEW(l15_0 l15_e);
t15_1=-(l15_0);
t15_2=-(l15_0+l15_e);
l15_e>0;

! ITEM 16:
NEW(l16_0 l16_e);
t16_1=-(l16_0);
t16_2=-(l16_0+l16_e);
l16_e>0;

! ITEM 17:
NEW(l17_0 l17_e);
t17_1=-(l17_0);
t17_2=-(l17_0+l17_e);
l17_e>0;

! ITEM 18:
NEW(l18_0 l18_e);
t18_1=-(l18_0);
t18_2=-(l18_0+l18_e);
l18_e>0;

! ITEM 19:
NEW(l19_0 l19_e);
t19_1=-(l19_0);
t19_2=-(l19_0+l19_e);
l19_e>0;

! ITEM 20:
NEW(l20_0 l20_e);
t20_1=-(l20_0);
t20_2=-(l20_0+l20_e);
l20_e>0;

OUTPUT:
TECH10;
```

SAVEDATA:

```
FORMAT IS f10.5;
FILE IS GCDINA_t2.dat;
SAVE = CPROBABILITIES FScores;
```

## Appendix C

### R and Mplus Codes for Chapter 6

#### C.1 Mplus code for the LTA-DINA model

The Mplus code for the LTA-DINA model is the same as the code for the GC-DINA model in Appendix B.5 except for ANALYSIS and the MODEL syntax for the overall part:

```
ANALYSIS :
  TYPE=MIXTURE;
  PARAMETERIZATION = PROBABILITY;
  STARTS=0;
  ALGORITHM=INTEGRATION;
  PROCESSORS = 4;
```

```
MODEL :
%OVERALL%
c5 ON c1;
c6 ON c2;
c7 ON c3;
c8 ON c4;
```

#### C.2 Mplus code for the longitudinal Rasch model

```
TITLE: !Longitudinal Rasch model (Andersen's model)
DATA:
  FILE IS responses_t2.txt;
VARIABLE:
  NAMES = x1-x20 y1-y20;
```

```

USEVARIABLE = x1-x20 y1-y20;
CATEGORICAL = x1-x20 y1-y20;
ANALYSIS:
  STARTS=0;
  ALGORITHM=INTEGRATION;

MODEL:
f1 BY x1@1 x2@ x3@1 x4@1 x5@1 x6@1
x7@1 x8@1 x9@1 x10@1 x11@1 x12@1
x13@1 x14@1 x15@1 x16@1 x17@1 x18@1
x19@1 x20@1;

f2 BY y1@1 y2@ y3@1 y4@1 y5@1 y6@1
y7@1 y8@1 y9@1 y10@1 y11@1 y12@1
y13@1 y14@1 y15@1 y16@1 y17@1 y18@1
y19@1 y20@1;

[f1@0];
[f2*0.5];
f1*1;
f2*1;
f1-f2 WITH f1-f2;

[x1$1 y1$1] (21);
[x2$1 y2$1] (22);
[x3$1 y3$1] (23);
[x4$1 y4$1] (24);
[x5$1 y5$1] (25);
[x6$1 y6$1] (26);
[x7$1 y7$1] (27);
[x8$1 y8$1] (28);
[x9$1 y9$1] (29);
[x10$1 y10$1] (30);
[x11$1 y11$1] (31);
[x12$1 y12$1] (32);
[x13$1 y13$1] (33);
[x14$1 y14$1] (34);
[x15$1 y15$1] (35);
[x16$1 y16$1] (36);
[x17$1 y17$1] (37);
[x18$1 y18$1] (38);

```

```
[x19$1 y19$1] (39);  
[x20$1 y20$1] (40);  
  
OUTPUT:  
  TECH10;  
SAVEDATA:  
  FORMAT IS f10.5;  
  FILE IS longitudinal_rasch.dat;  
  SAVE = FScores;
```

# References

- Almond, R. G. (2007a). Cognitive modeling to represent growth (learning) using markov decision processes. *Technology, Instruction, Cognition & Learning*, 5(4).
- Almond, R. G. (2007b). An illustration of the use of markov decision processes to represent student growth (learning). *ETS Research Report Series*, 2007(2).
- Almond, R. G., Mislevy, R. J., Steinberg, L., Yan, D., & Williamson, D. (2015). *Bayesian networks in educational assessment*. Springer.
- Andersen, E. B. (1970). Asymptotic properties of conditional maximum-likelihood estimators. *Journal of the Royal Statistical Society. Series B (Methodological)*, 283–301.
- Andersen, E. B. (1985). Estimating latent correlations between repeated testings. *Psychometrika*, 50(1), 3–16.
- Andrade, D. F., & Tavares, H. R. (2005). Item response theory for longitudinal data: population parameter estimation. *Journal of Multivariate Analysis*, 95(1), 1–22.
- Ayers, E., & Rabe-Hesketh, S. (2011). *Cognitive diagnosis models with longitudinal growth curves for skill knowledge (invited talk)*. (Joint Statistical Meetings. Miami Beach, FL, USA)
- Ayers, E., Rabe-Hesketh, S., & Nugent, R. (2013). Incorporating student covariates in cognitive diagnosis models. *Journal of Classification*, 30(2), 195–224.
- Baker, F. B., & Kim, S.-H. (2004). *Item response theory: Parameter estimation techniques*. CRC Press.
- Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*.
- Bottge, B. A., Heinrichs, M., Chan, S.-Y., Mehta, Z. D., & Watson, E. (2003). Effects of video-based and applied problems on the procedural math skills of average-and low-achieving adolescents. *Journal of Special Education Technology*, 18(2), 5–22.

- Bottge, B. A., Ma, X., Gassaway, L., Toland, M. D., Butler, M., & Cho, S.-J. (2014). Effects of blended instructional models on math performance. *Exceptional children, 80*(4), 423–437.
- Bottge, B. A., Rueda, E., Serlin, R. C., Hung, Y.-H., & Kwon, J. M. (2007). Shrinking achievement differences with anchored math problems: Challenges and possibilities. *The Journal of Special Education, 41*(1), 31–49.
- Bottge, B. A., Toland, M. D., Gassaway, L., Butler, M., Choo, S., Griffen, A. K., & Ma, X. (2015). Impact of enhanced anchored instruction in inclusive math classrooms. *Exceptional Children, 81*(2), 158–175.
- Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research, 11*(1), 94.
- Bunt, A., & Conati, C. (2002). Assessing effective exploration in open learning environments using bayesian networks. In S. A. Cerri, G. Gouardères, & F. Paraguaçu (Eds.), *Intelligent tutoring systems: 6th international conference, its 2002 biarritz, france and san sebastian, spain, june 2–7, 2002 proceedings* (pp. 698–707). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Cai, L. (2010). A two-tier full-information item factor analysis model with applications. *Psychometrika, 75*(4), 581–612.
- Carlin, B. P., & Louis, T. A. (2000). *Bayes and empirical bayes methods for data analysis* (Vol. 17). Chapman & Hall/CRC Boca Raton, FL.
- Cho, S.-J., Cohen, A. S., Kim, S.-H., & Bottge, B. (2010). Latent transition analysis with a mixture item response theory measurement model. *Applied Psychological Measurement, 34*(7), 483–504.
- Collins, L. M., & Wugalter, S. E. (1992). Latent class models for stage-sequential dynamic latent variables. *Multivariate Behavioral Research, 27*(1), 131–157.
- Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction, 4*(4), 253–278.
- Culbertson, M. J. (2016). Bayesian networks in educational assessment: The state of the field. *Applied Psychological Measurement, 40*(1), 3–21.
- Davier, M. (2005). A general diagnostic model applied to language testing data. *ETS Research Report Series, 2005*(2).

- de la Torre, J. (2009). Dina model and parameter estimation: A didactic. *Journal of educational and behavioral statistics, 34*(1), 115–130.
- de la Torre, J. (2011). The generalized dina model framework. *Psychometrika, 76*(2), 179–199.
- de la Torre, J., & Douglas, J. A. (2004). Higher-order latent trait models for cognitive diagnosis. *Psychometrika, 69*(3), 333–353.
- Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational intelligence, 5*(2), 142–150.
- De Boeck, P., & Wilson, M. (2004). *Explanatory item response models: A generalized linear and nonlinear approach*. New York: Springer-Verlag.
- 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. Series B (methodological), 1*–38.
- Embretson, S. E. (1991). A multidimensional latent trait model for measuring learning and change. *Psychometrika, 56*(3), 495–515.
- Gibbons, R. D., & Hedeker, D. R. (1992). Full-information item bi-factor analysis. *Psychometrika, 57*(3), 423–436.
- Graham, J. W., Collins, L. M., Wugalter, S. E., Chung, N., & Hansen, W. B. (1991). Modeling transitions in latent stage-sequential processes: a substance use prevention example. *Journal of consulting and clinical psychology, 59*(1), 48.
- Haertel, E. H. (1990). Continuous and discrete latent structure models for item response data. *Psychometrika, 55*(3), 477–494.
- Harris, D. (1989). Comparison of 1-, 2-, and 3-parameter irt models. *Educational Measurement: Issues and Practice, 8*(1), 35–41.
- Hartz, S. (2002). *A bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality*. (Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL.)
- Heckerman, D. (1998). A tutorial on learning with bayesian networks. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 301–354). Dordrecht: Springer Netherlands.
- Henson, R. A., Templin, J. L., & Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika, 74*(2), 191–210.

- Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1), 193–218.
- Jeon, M., & Rabe-Hesketh, S. (2016). An autoregressive growth model for longitudinal item analysis. *Psychometrika*, 81(3), 830–850.
- Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3), 258–272.
- Kaya, Y., & Leite, W. L. (2016). Assessing change in latent skills across time with longitudinal cognitive diagnosis modeling an evaluation of model performance. *Educational and Psychological Measurement*, 0013164416659314.
- Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3), 239–264.
- LaMar, M. M. (2017). Markov decision process measurement model. *Psychometrika*, 1–22.
- Levy, R., & Mislevy, R. (2004). Specifying and refining a measurement model for a simulation-based assessment. *International Journal of Testing*, 4, 333–369.
- Li, F., Cohen, A., Bottge, B., & Templin, J. (2016). A latent transition analysis model for assessing change in cognitive skills. *Educational and Psychological Measurement*, 76(2), 181–204.
- Ma, W., & de la Torre, J. (2017). GDINA: The generalized DINA model framework [Computer software manual]. Retrieved from <https://CRAN.R-project.org/package=GDINA> (R package version 1.4.2)
- Madison, M. J., & Bradshaw, L. P. (2015). The effects of q-matrix design on classification accuracy in the log-linear cognitive diagnosis model. *Educational and Psychological Measurement*, 75(3), 491–511.
- Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, 64(2), 187–212.
- Matsuda, N., & VanLehn, K. (2000). Decision theoretic instructional planner for intelligent tutoring systems. In *Proceedings workshop on modeling human teaching tactics and strategies (its2000, pp. 72-83)*.

- Mislevy, R., Almond, R., Yan, D., & Steinberg, L. (1999). Bayes nets in educational assessment: Where do the numbers come from? In K. Laskey & H. Prade (Eds.), *Proceedings of the fifteenth conference on uncertainty in artificial intelligence* (p. 437-446). San Francisco: Morgan Kaufmann.
- Mislevy, R. J. (1995). Probability-based inference in cognitive diagnosis. In P. Nichols, S. Chipman, & S. Brennan (Eds.), *Cognitively diagnostic assessment* (p. 43-71). Mahwah, NJ: Erlbaum.
- Muthén, L. K., & Muthén, B. O. (2017). Mplus (version 8). Los Angeles, CA: Muthén & Muthén.
- Pastor, D. A., & Beretvas, S. N. (2006). Longitudinal rasch modeling in the context of psychotherapy outcomes assessment. *Applied Psychological Measurement*, 30(2), 100–120.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Rasch, G. (1961). On general laws and the meaning of measurement in psychology. In *Proceedings of the fourth berkeley symposium on mathematical statistics and probability* (Vol. 4, pp. 321–333).
- Rijmen, F. (2009). Efficient full information maximum likelihood estimation for multidimensional irt models. *ETS Research Report Series*, 2009(1).
- Rijmen, F., Tuerlinckx, F., De Boeck, P., & Kuppens, P. (2003). A nonlinear mixed model framework for item response theory. *Psychological methods*, 8(2), 185.
- Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement*, 14(3), 271–282.
- 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 & M. J. Gierl (Eds.), *Cognitive diagnostic assessment for education: Theory and applications* (p. 275-318). Cambridge University Press.
- Rupp, A. A., Templin, J. L., & Henson, R. A. (2010). *Diagnostic assessment: Theory, methods, and applications*. New York: Guilford.
- Schäfer, R., & Weyrath, T. (1997). Assessing temporally variable user properties with dynamic bayesian networks. In *User modeling* (pp. 377–388).

- Sinharay, S., & Almond, R. G. (2007). Assessing fit of cognitively diagnostic models: A case study. *Educational and Psychological Measurement, 67*(2), 239-257.
- Tatsuoka, K. K. (1985). A probabilistic model for diagnosing misconceptions by the pattern classification approach. *Journal of Educational and Behavioral Statistics, 10*, 55-73.
- Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnosis. *Diagnostic monitoring of skill and knowledge acquisition*, 453-488.
- Templin, J. L., & Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological methods, 11*(3), 287.
- Van Der Linden, W. J., & Hambleton, R. K. (1997). Item response theory: Brief history, common models, and extensions. In *Handbook of modern item response theory* (pp. 1-28). Springer.
- Xu, X., & Davier, M. (2008). Fitting the structured general diagnostic model to naep data. *ETS Research Report Series, 2008*(1).
- Zapata-Rivera, J.-D., & Greer, J. (2004). Inspectable bayesian student modelling servers in multi-agent tutoring systems. *International Journal of Human-Computer Studies, 61*(4), 535 - 563.