Statistical Learning for Latent Attribute Models

by

Chenchen Ma

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2022

Doctoral Committee:

Associate Professor Gongjun Xu, Chair
Assistant Professor Kean Ming Tan
Assistant Professor Zhenke Wu
Professor Ji Zhu

Chenchen Ma

chenchma@umich.edu

ORCID iD: 0000-0003-2784-9920

© Chenchen Ma 2022

ACKNOWLEDGEMENTS

First and foremost, I want to express my deepest gratitude to Professor Gongjun Xu, my advisor. This dissertation would not have been possible without his continuous guidance, mentoring, and encouragement during the last five years. Gongjun is a super supportive, inspirational, and patient advisor, and I have learned so much from working with him. Moreover, his passion for research and positive attitude also have profound influences on me, not only in academics but also in my personal life. He has always encouraged me to pursue my dreams, to be brave, and provided a lot of support along the way. Gongjun is the best advisor I could ever ask for.

My gratitude also goes to my amazing collaborators and coauthors: Professor Jimmy de la Torre, Professor Chun Wang, Jing Ouyang, and Chengcheng Li. I have enjoyed it a lot during our discussions and learned so much from our collaborations. I would like to thank Professor Ji Zhu, Professor Kean Ming Tan, and Professor Zhenke Wu for serving on the dissertation committee and providing insightful feedback as well.

I want to thank all the faculty members and staff members who have helped to make the department feel like a huge warm family. Moreover, my amazing cohort, our research group, my greatest friends, my roommates, and our kitties all helped and supported me tremendously. I would not be here without them.

Finally, I would like to thank my parents and grandparents for their unconditional and unwavering love over the years. I feel deeply indebted to them.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Latent variable models are popularly used in unsupervised learning to uncover the latent structures underlying observed data and have seen great successes in representation learning in many applications and scientific disciplines. Latent attribute models, also known as cognitive diagnosis models or diagnostic classification models, are a special family of discrete latent variable models that have been widely applied in modern psychological and biomedical research with diagnostic purposes. Despite the wide usage in various fields, the models' discrete nature and complex restricted structures pose many new challenges for efficient learning and statistical inference. Moreover, with the large-scale item and subject pools emerging in modern educational and psychological measurements, efficient algorithms for uncovering latent structures of both items and subjects are desired. This dissertation studies four important problems that arise in this context.

(I) The first part develops novel methodologies and efficient algorithms to learn the latent and hierarchical structures in latent attribute models. Specifically, researchers in many applications are interested in hierarchical structures among the latent attributes, such as prerequisite relationships among target skills in educational settings. However, in most cognitive diagnosis applications, the number of latent attributes, the attribute-attribute hierarchical structures, the item-attribute dependence structures, as well as the item-level diagnostic models, need to be fully or partially pre-specified, which may be subjective and misspecified as noted by many recent studies. In this part, we consider the problem of jointly learning these latent quantities and hierarchical structures from observed data with minimal model assumptions. A penalized likelihood approach is proposed for joint learning, an Expectation-Maximization (EM) algorithm is developed for efficient computation, and statistical consistency theory is established under mild conditions.

(II) The second part generalizes the methodologies in part I to simultaneously infer the subgroup structures of both subjects and items. We consider the model-based co-clustering algorithms and aim to automatically select numbers of clusters and uncover latent block structures. Specifically, based on latent block models, we propose a penalized co-clustering approach that is capable of learning the numbers of clusters and inner block structures simultaneously. Efficient EM algorithms have been developed and comprehensive simulation studies demonstrate their superiority.

(III) The third part concerns the important yet unaddressed problem of testing the latent hierarchical structures in latent attribute models. Testing the hierarchical structures is shown to be equivalent to testing the sparsity structure of the proportion parameter vector. However, due to the irregularity of the problem, the asymptotic distribution of the popular likelihood ratio test becomes nonstandard and tends to provide unsatisfactory finite sample performance under practical conditions. To tackle these challenges, we discuss the conditions of testability issues, provide statistical understandings of the failures, and propose a practical resampling-based procedure.

(IV) The fourth part introduces a unified estimation framework to bridge the gap between parametric and nonparametric methods in cognitive diagnosis to better understand their relationship. In particular, a number of parametric and nonparametric methods for estimating latent attribute models have been developed and applied in a wide range of contexts. However, in the literature, a wide chasm exists between these two families of methods, and their relationship to each other is not well understood. Driven by this divide, we propose a unified framework and provide both theoretical analysis and practical recommendations under various cognitive diagnosis settings.

CHAPTER I

Introduction

In the past several decades, latent variable models have gained increasing interest in many machine learning problems and found a wide range of applications in many scientific disciplines. By supplementing a set of observed variables with additional latent, or hidden, variables, it allows a way to model unmeasurable structures underlying the observed data, such as latent subgroups in social and behavioral sciences (Templin et al., 2010; de la Torre et al., 2018), and disease etiology in epidemiology studies (O'Brien et al., 2019). Moreover, latent variable models are also used for dimension reduction in machine learning applications (Tipping and Bishop, 1999; Reynolds, 2009), and have gained huge successes in representation learning in this "Big Data" era (Bengio et al., 2013; Van Den Oord et al., 2017; Tschannen et al., 2018).

Latent Attribute Models (LAMs), as known as Cognitive Diagnosis Models (CDMs) in psychometrics, are a special family of latent variable models widely used in educational assessment, psychological measurements, and scientifically-structured clustering based on noisy observations. In particular, the goal of cognitive diagnosis is arriving at a classification-based decision about an individual's latent attribute pattern, based on his or her observed responses to a set of designed diagnostic items. Such diagnostic information plays an important role in constructing efficient and focused remedial strategies for individuals' improvement. For instance, in educational assessment, using LAMs to identify the profiles of mastery/deficiency of the target abilities of students would help better design curriculum and teaching strategies (Templin et al., 2010). In psychiatric evaluation, LAMs can help detect the profiles of the presence/absence of a set of underlying psychological disorders based on the manifested symptoms of the patients (de la Torre et al., 2018). In disease etiology, LAMs can be used to determine the configurations of the existence of a set of viruses based on patients' biological measurements (O'Brien et al., 2019).

The topic of cognitive diagnosis modeling has gained great popularity in recent years due to the models' desirable diagnostic nature of providing informative cognitive profiles for every respondent. Various cognitive diagnosis models have been developed with different cognitive diagnostic restrictions on how the responses depend on the underlying latent attributes. With the rapid developments in information technology during the past several decades, large-scale item pools and response data tend to emerge. In this new "big data" era, efficient algorithms for estimation, inference, and evaluation are desired given the large volume and the complexity of the data. Moreover, despite the wide use in various fields, the models' discrete nature and complex restricted structure make many traditional statistical inference procedures perform poorly or even become invalid. It is essential and challenging to develop methodologies under such irregular situations. Lastly, although many different models and estimation methods have been developed in the past decades, there is a lack of a unified understanding of them. This dissertation is mostly motivated by these challenges.

Learning Latent Hierarchical Structures One important yet challenging problem in latent attribute models is to identify the structure of the latent attributes from the observed data. This is a critical issue in the applications since the latent structures are unobserved and misspecification of the latent structure will result in misleading inference and wrong conclusions. Recently researchers are particularly interested in the hierarchical structures among the latent attributes (Templin and Bradshaw, 2014). For example, in education, curricula are typically structured based on students' hierarchical learning process, where the lower-level skills are prerequisites for the higher-level skills so that the instructors can teach sequentially and progressively. In chapter II, taking identifiability conditions into consideration, we propose a penalized likelihood approach starting from an exploratory latent class model to learn latent structures with minimal model assumptions. This is the first method in the literature to simultaneously estimate multiple quantities, including the number of latent attributes, the item-attribute Q-matrix, the latent hierarchical structure, and item-level diagnosis models. In this chapter, we develop an efficient Expectation-Maximization (EM) algorithm and establish the consistency theory of the proposed model under mild conditions. The good performance of the proposed methodology is illustrated by simulation studies in various settings and applications to two real data sets. This chapter is based on Ma et al. (2022b).

Learning Latent Block Structures With large-scale item pools and response data emerging in modern educational and psychological measurements, it also gains increasing interest in simultaneously inferring the subgroup structures of both subjects and items. This motivates us in chapter III to develop co-clustering algorithms that simultaneously cluster subjects and items into homogeneous blocks, such as clustering both test takers and test questions in educational assessment jointly to form clusters with similar target skills. One essential yet difficult problem in clustering applications is determining the number of clusters. The popularly used information criteria to select the number of clusters, such as AIC and BIC, can be very computationally expensive since it needs to explore many possible values. It is especially true for the co-clustering setting, where it needs to compare all the possible combinations of numbers for both row clusters and column clusters. Moreover, in many applications in cognitive diagnosis, the inner structures of the latent blocks are of great interest. Under the latent attribute model framework, one important and common assumption is that subjects with the same latent attribute pattern have the same response probabilities to the items. With a large pool of items, we can make a further assumption that items targeting on the same latent attributes also share the same response distributions. Therefore, among the latent blocks, there are subsets of blocks sharing the same block distributions, which is also an interesting latent structure of learning. Motivated by the methods in chapter II, we propose a penalized co-clustering method, which is capable of learning the numbers of clusters and the inner block structures simultaneously.

Hypothesis Testing for Latent Hierarchical Structures In previous chapters, we have developed new methodologies and efficient algorithms to learn latent structures from observed data. In many applications, these latent structures are often posited by some domain experts. A natural question is then to test such structure of the latent attributes of interest, which we study in chapter IV. Specifically, we consider the statistical hypothesis testing for latent hierarchical structures in latent attribute models. Such testing can be formulated as testings for nested models, and a popularly used tool in statistics is the fundamental likelihood ratio test (LRT). However, due to the induced non-regularity by the hierarchical structures, the LRT may lose its well-behaved large sample property represented by the famous Wilk's theorem, and the asymptotic behaviors of LRT for testing hierarchies need to be further investigated. Moreover, an even more fundamental issue is to understand when the latent structure is identifiable and testable since if the model is not identifiable, the latent hierarchical structures cannot be learned no matter how large the data size is.

To address this problem, in chapter IV, we specify the conditions needed so that the latent structure is identifiable and testable. In the testable cases, we further study the limiting distribution of the LRT and provide insights into why the conventional theory of the LRT fails in such tests. In addition, we propose a resampling-based method and demonstrate its effectiveness through comprehensive simulation studies. This chapter is based on Ma and Xu (2021).

Bridging Parametric and Nonparametric Methods In latent attribute models, there are two popular families of estimation methods, including parametric and nonparametric ones. For parametric methods, certain distributional functions in a parametric form for the item response probabilities need to be assumed, which may raise validity concerns about the assumed model and the related diagnostic process. As an alternative, researchers have also explored nonparametric methods for assigning subjects to latent groups without relying on parametric model assumptions. Despite the popularity of the parametric and nonparametric methods in cognitive diagnosis, the relationship between these methods has not been studied in the literature. Although seemingly divergent from the surface, these frameworks are in fact closely related. To help better understand their relationship, in chapter V, we propose a unified framework for cognitive diagnosis that subsumes both parametric and nonparametric methods. In particular, we use a general loss function to measure the distance between a subject's responses and the centroid of a latent class. By using different loss functions, the method can assume different parametric and nonparametric forms. We further develop a unified iterative joint estimation algorithm and establish the consistency properties of the corresponding estimators. In addition, we conduct comprehensive simulation studies to compare different methods under a wide variety of settings and provide practical recommendations. The theoretical analysis and numerical studies in this work bridge the gap between the two families of methods and provide a novel point of view to better understand latent attribute models. This chapter is based on Ma et al. (2022a).

CHAPTER II

Learning Latent and Hierarchical Structures

2.1 Introduction

In many applications of latent attribute models, researchers are interested in hierarchical structures among the latent attributes. For example, in a learning context, the possession of lower-level skills is often assumed to be the prerequisite for the possession of higher-level skills in education (Dahlgren et al., 2006; Jimoyiannis and Komis, 2001; Simon and Tzur, 2004; Wang and Gierl, 2011). Learning such latent hierarchical structures among the latent attributes is not only useful for educational research but can also be used to design learning materials and generate recommendations or remedy strategies based on the prerequisite relationships among the latent attributes. Leighton et al. (2004) proposed the Attribute Hierarchy Model, a variant of Tatsuoka's rule-space approach (Tatsuoka, 1983), which explicitly defined the hierarchical attribute structures through an adjacency matrix. Under the cognitive diagnosis modeling framework, Templin and Bradshaw (2014) proposed the Hierarchical Cognitive Diagnosis Models (HCDMs), in which a Directed Acyclic Graph (DAG) was essentially used to impose hard constraints on possible latent attribute profiles under hierarchies. In this chapter, we also term HCDMs as Hierarchical Latent Attribute Models (HLAMs) in the following.

In the cognitive diagnosis modeling framework, the dependence structure between the observed variables and the latent attributes is encoded through a binary design matrix, the so-called Q-matrix (Tatsuoka, 1990). Under different item models, the interactions between the observed variables and the latent attributes are modeled differently. Two basic models are the Deterministic Input Noisy Output "AND" gate (DINA; Haertel, 1989) model and the Deterministic Input Noisy Output "OR" gate (DINO; Templin and Henson, 2006) model, where there are only two levels of item parameters for each item. de la Torre (2011) proposed the Generalized DINA (GDINA) model, where the interactions among all the latent attributes were included. Other popularly used latent attribute models include the General Diagnostic Model (GDM; von Davier, 2019), the reduced Reparameterized Unified Model (reduced-RUM; DiBello et al., 1995), and the Log-linear Cognitive Diagnosis Models (LCDM; Henson et al., 2009).

To fit HLAMs, the Q-matrix, the hierarchical structures among the attributes, the item-level models, and the number of latent attributes all need to be pre-specified by domain experts, which however can be subjective and inaccurate. Moreover, in exploratory data analysis, these prior quantities may be even entirely unknown. An important problem in cognitive diagnosis modeling then becomes how to efficiently and accurately learn such latent and hierarchical structures and model specifications from noisy observations with minimal prior knowledge and assumptions.

In the literature, many methods have been recently developed to learn the Q-matrix, including methods to directly estimate the Q-matrix from the observational data, via either frequentist approaches (Liu et al., 2012; Chen et al., 2015; Xu and Shang, 2018; Li et al., 2022) or Bayesian approaches (Chung and Johnson, 2018; Chen et al., 2018; Culpepper, 2019), and methods to validate the pre-specified Q-matrix (de la Torre, 2008; DeCarlo, 2012; Chiu, 2013; de la Torre and Chiu, 2016; Gu et al., 2018). Many of these Q-matrix learning or validation methods, however, do not consider the hierarchical structures, or they implicitly assume the hierarchical structure is known; moreover, the number of attributes and the item-level diagnostic models are often assumed to be known.

In terms of learning attribute hierarchies from observational data, Wang and Lu (2021) recently studied two exploratory approaches including the latent variable selection (Xu and Shang, 2018) approach and the regularized latent class modeling (regularized LCM, Chen et al., 2017b) approach. However, the latent variable selection approach in Wang and Lu (2021) requires specification of the number of latent attributes and a known identity sub-matrix in the Q-matrix. The regularized LCM approach may not require the number of latent attributes, but the number of latent classes needs to be selected. Based on the simulation in Wang and Lu (2021), the performance of the regularized LCM was less satisfactory – the accuracy of selecting a correct number of latent classes was often below 50% and the accuracy of recovering latent hierarchy was almost 0 in some cases. In Gu and Xu (2019a), the authors proposed a two-step algorithm for structure learning of HLAMs. However, their algorithm also assumed that the number of latent attributes was known and they only considered the DINA and DINO models.

In this chapter, to overcome the limitations of the aforementioned methods, we propose a regularized maximum likelihood approach with minimal model assumptions to achieve the following four goals simultaneously: (1) estimate the number of latent attributes; (2) learn the latent hierarchies among the attributes; (3) learn the Q-matrix; and (4) recover item-level diagnostic models. Specifically, we employ two regularization terms: one penalty on the population proportion parameters to select significant latent classes, and the other on the differences of item parameters for each item to learn the structures of the item parameters. After the significant latent classes and the structure of the item parameters are learned, a latent structure recovery algorithm is used to estimate the number of latent attributes, the latent hierarchies among the attributes, the Q-matrix, and the item-level models. For the estimation, we develop an efficient Penalized EM algorithm using the Difference Convex (DC) programming and the Alternating Direction Method of Multipliers (ADMM) method. Consistent learning theory is established under mild regularity conditions. We also conduct simulation studies to show the good performance of the proposed method. Finally, we demonstrate the application of our method to two real datasets and obtain interpretable results which are consistent with the previous research.

The remaining of this chapter proceeds as follows: the model setup of HLAMs is provided in Section 2.2. The proposed penalized likelihood approach and its theoretical properties are presented in Section 2.3. An efficient algorithm is developed and related computational issues are discussed in Section 2.4. Simulation studies are presented in Section 2.5. In Section 2.6, the model is applied to two real data sets of educational assessment. Finally, Section 2.7 concludes with some discussions. The proof of the main theorem and detailed derivations for the proposed algorithm are presented in Appendix A.

2.2 Model Setup

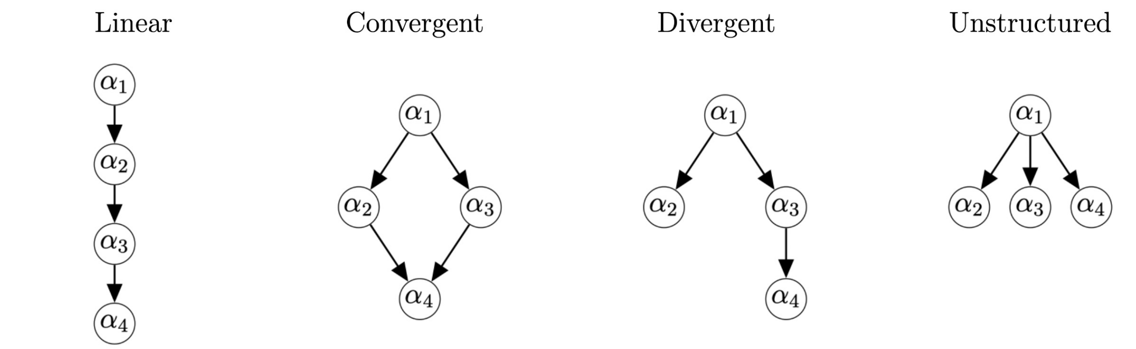
In this section, we introduce the general model setup of HLAMs and illustrate the connections between HLAMs and restricted latent class models, which motivates the proposed approach in Section 2.3. In the following, for an integer K, we use $[K]$ to denote the set $\{1, 2, \dots, K\}$, and we use $|\cdot|$ to denote the cardinality of a set.

2.2.1 Hierarchical Latent Attribute Models

In a latent attribute model with J items which depend on K latent attributes of interest, two types of subject-specific variables are considered, including the responses $\boldsymbol{R} = (R_1, \dots, R_J)$, and the latent binary attribute profile $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$. In this chapter, both the responses $\boldsymbol{R}$ and the latent attribute profile $\boldsymbol{\alpha}$ are assumed to be binary. The J-dimensional vector $\boldsymbol{R} \in \{0, 1\}^J$ denotes the binary responses to a set of J items, and the K-dimensional vector $\boldsymbol{\alpha} \in \{0,1\}^K$ denotes a profile of possession of K latent attributes of interest. Since each latent attribute α_k is binary, the total number of possible latent attribute profiles $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ is 2^K. For each latent attribute profile, we use $\pi_{\boldsymbol{\alpha}}$ to denote its proportion parameter, and the latent attribute profile $\boldsymbol{\alpha}$ is modeled to follow a categorical distribution with the proportion parameter vector $\boldsymbol{\pi} = (\pi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \{0,1\}^K)$. The proportion parameter vector lies in the $(2^K - 1)$-simplex and satisfies $\pi_{\boldsymbol{\alpha}} \in [0,1]$ and $\sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \pi_{\boldsymbol{\alpha}} = 1$.

A key feature of HLAMs is that there exist certain hierarchical structures among the latent attributes. For example, in cognitive diagnosis modeling, the possession of lower-level skills is often regarded as the prerequisite for the possession of higher-level skills (Leighton et al., 2004; Templin and Bradshaw, 2014). With such an attribute hierarchy, any latent attribute profile $\boldsymbol{\alpha}$ that does not respect the hierarchy will not exist and have population proportion $\pi_{\boldsymbol{\alpha}} = 0$. For $1 \leq k \neq l \leq K$, we use $\alpha_k \to \alpha_l$ (or $k \to l$) to denote the hierarchy that attribute α_k is a prerequisite of attribute α_l. We assume such hierarchy $\alpha_k \to \alpha_l$ (or $k \to l$) exists if and only if there are no latent attribute profiles such that $\alpha_l = 1$ but $\alpha_k = 0$, or equivalently, we have $\pi_{\boldsymbol{\alpha}} = 0$ if $\alpha_l = 1$ but $\alpha_k = 0$. We denote an attribute hierarchy by a set of prerequisite relations $\mathcal{E} = \{k \to l : \text{ attribute } k \text{ is a prerequisite for attribute } l,\ 1 \leq k \neq l \leq K\}$, and denote the induced set of existent latent attribute profiles by $\mathcal{A} = \{\boldsymbol{\alpha} \in \{0,1\}^K : \pi_{\boldsymbol{\alpha}} \neq 0 \text{ under } \mathcal{E}\}$. One can see that an attribute hierarchy results in the sparsity of the proportion parameter vector $\boldsymbol{\pi}$, which will significantly reduce the number of model parameters especially when K is large. Example hierarchical structures and the corresponding induced sets of existent attribute profiles are shown in Figure II.1.

In latent attribute models, the structural matrix $\boldsymbol{Q} = (q_{j,k}) \in \{0,1\}^{J \times K}$ is an important component that imposes constraints on items to reflect the dependence between the items and the latent attributes. To be specific, $q_{j,k} = 1$ if item j requires (or depends on) attribute k. Then the jth row vector of $\boldsymbol{Q}$ denoted by $\boldsymbol{q}_j$ describes the



$$\mathcal{A}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \mathcal{A}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \mathcal{A}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \mathcal{A}_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$|\mathcal{A}_1| = 5 \qquad |\mathcal{A}_2| = 6 \qquad |\mathcal{A}_3| = 7 \qquad |\mathcal{A}_4| = 9$$

Figure II.1: Examples of hierarchical structures of latent attributes. For $i = 1, \ldots, 4$, each $\mathcal{A}_i$ represents the induced set of existent attribute profiles under the hierarchical structure above it, where each row in $\mathcal{A}_i$ represents an attribute profile $\boldsymbol{\alpha}$ with $\pi_{\boldsymbol{\alpha}} \neq 0$.

full dependence of item j on K latent attributes. In many applications, the matrix $\boldsymbol{Q}$ is pre-specified by domain experts (George and Robitzsch, 2015; Junker and Sijtsma, 2001; von Davier, 2005) to reflect some scientific assumptions. See Figure II.2 for an illustration of the Q-matrix and the corresponding graphical representation.

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

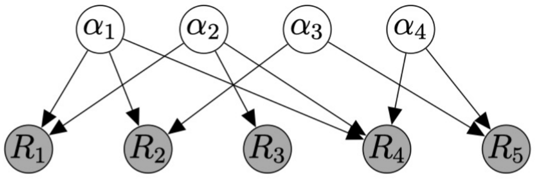


Figure II.2: Illustration of Q-matrix.

As in classical latent class analysis, given a subject's latent attribute profile $\boldsymbol{\alpha}$, the responses to J items are assumed to be independent, which is known as the local independence assumption, and follow Bernoulli distributions with parameters $\theta_{1,\boldsymbol{\alpha}}, \ldots, \theta_{J,\boldsymbol{\alpha}}$, which are called item parameters. Specifically, we have $\theta_{j,\boldsymbol{\alpha}} := \mathbb{P}(R_j = 1 \mid \boldsymbol{\alpha})$. We use $\boldsymbol{\Theta} = (\theta_{j,\boldsymbol{\alpha}})$ to denote the item parameter matrix. Under the local independence assumption, the probability mass function of a subject's response vector $\boldsymbol{R} = (R_1, \ldots, R_J) \in \{0,1\}^J$ can be written as

$$\mathbb{P}(\boldsymbol{R} \mid \boldsymbol{\Theta}, \boldsymbol{\pi}) = \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \pi_{\boldsymbol{\alpha}} \prod_{j=1}^{J} \theta_{j,\boldsymbol{\alpha}}^{R_j} (1 - \theta_{j,\boldsymbol{\alpha}})^{1-R_j}. \tag{2.1}$$

So far we have a latent attribute profile $\boldsymbol{\alpha}$ for a subject to indicate the subject's possession of K attributes, and a structural vector $\boldsymbol{q}$ for an item to reflect the item's dependence on K latent attributes. Moreover, the structural matrix $\boldsymbol{Q}$ puts constraints on item parameters to reflect the diagnostic model assumptions. One important common assumption is that the item parameters $\theta_{j,\boldsymbol{\alpha}}$ only depends on whether the latent attribute profile $\boldsymbol{\alpha}$ contains the required attributes by item j, that is, the attributes in the set $\mathcal{K}_j = \{k \in [K] : q_{j,k} = 1\}$, which is the set of the required attributes of item j. Therefore, for item j, the latent attribute profiles which are only different in the attributes outside of $\mathcal{K}_j$ would have the same item parameters. In this way, the structural matrix $\boldsymbol{Q}$ forces some entries in the item parameter matrix $\boldsymbol{\Theta}$ to be the same. The dependence of item parameters on the required attributes are modeled differently in different LAMs, as shown in Example II.1 and Example II.2.

Example II.1 (DINA and DINO Models)**.** *We first introduce the Deterministic Input Noisy output "And" (DINA, Junker and Sijtsma, 2001) and the Deterministic Input Noisy output "Or" (DINO, Templin and Henson, 2006) models, where there are only two levels of item parameters for each item. Specifically, we use θ_j^+ and θ_j^- to denote the two levels for item j. We introduce a binary indicator matrix $\boldsymbol{\Gamma} = (\Gamma_{j,\boldsymbol{\alpha}} : j \in [J], \boldsymbol{\alpha} \in \{0,1\}^K) \in \{0,1\}^{J \times 2^K}$, which corresponds to the ideal responses under the DINA and DINO models. Under the DINA model, which assumes a conjunctive*

"And" relationship among the binary attributes, the indicator matrix is defined as

$$\Gamma_{j,\boldsymbol{\alpha}}^{DINA} = \prod_{k}^{K} \alpha_k^{q_{j,k}} = \prod_{k \in \mathcal{K}_j} \alpha_k. \tag{2.2}$$

Under the DINO model assuming a conjunctive "Or" relationship among the latent attributes, we have

$$\Gamma_{j,\boldsymbol{\alpha}}^{DINO} = \mathbb{I}\big(\exists\ k \in [K], q_{j,k} = \alpha_k = 1\big). \tag{2.3}$$

The indicator $\Gamma_{j,\alpha}$ in the DINA model indicates whether a subject possesses all the required attributes of item j, while that in the DINO model indicates whether a subject possesses any of the required attributes of item j. In both models, the item parameters only depend on the set of the required attributes of an item $\mathcal{K}_j$, and they are defined as:

$$\theta_j^+ = \mathbb{P}(R_j = 1 \mid \Gamma_{j,\boldsymbol{\alpha}} = 1), \quad \theta_j^- = \mathbb{P}(R_j = 1 \mid \Gamma_{j,\boldsymbol{\alpha}} = 0),$$

where θ_j^- is also called the guessing parameter and $1 - \theta_j^+$ the slipping parameter.

Example II.2 (GDINA model)**.** *The Generalized DINA model (GDINA, de la Torre, 2011) is a more general model where all the interactions among the latent attributes are considered. The item parameters for the GDINA model are written as*

$$\begin{aligned}\theta_{j,\boldsymbol{\alpha}}^{GDINA} &= \beta_{j,0} + \sum_{k=1}^{K} \beta_{j,k}\alpha_k q_{j,k} + \sum_{k=1}^{K}\sum_{k'=k+1}^{K} \beta_{j,k,k'}\alpha_k\alpha_{k'}q_{j,k}q_{j,k'} + \cdots + \beta_{j,1,2,\dots,K}\prod_{k=1}^{K}\alpha_k q_{j,k} \\ &= \beta_{j,0} + \sum_{k\in\mathcal{K}_j} \beta_{j,k}\alpha_k + \sum_{k,k'\in\mathcal{K}_j, k\neq k'} \beta_{j,k,k'}\alpha_k\alpha_{k'} + \cdots + \beta_{j,\mathcal{K}_j}\prod_{k\in\mathcal{K}_j}\alpha_k.\end{aligned}$$

The coefficients in the GDINA model can be interpreted as follows: $\beta_{j,0}$ is the probability of a positive response for the most incapable subjects with no required attributes present; $\beta_{j,k}$ is the increase in the probability due to the main effect of latent attribute α_k; $\beta_{j,1,2,\dots,K}$ is the change in the positive probability due to the interaction of all the latent attributes. In the GDINA model, the intercept and main effects are typically *assumed to be nonnegative to satisfy the monotonicity assumption, while the interactions may take negative values. By incorporating all the interactions among the required attributes, the GDINA model is one of the most general cognitive diagnosis models.*

2.2.2 LAMs as Restricted Latent Class Models

Latent attribute models in fact can also be viewed as Restricted Latent Class Models (RLCM, Xu, 2017), a special family of more general Latent Class Models (LCMs). We first briefly describe the general model setup of LCMs (Goodman, 1974). In an LCM, we assume that each subject belongs to one of M latent classes. For each latent class, we use π_m to denote its proportion parameter for $m \in [M]$. The latent classes follow a categorical distribution with the proportion parameter vector $\boldsymbol{\pi} = (\pi_m : m \in [M], \pi_m \geq 0, \sum_{m=1}^{M} \pi_m = 1)$. As in classical finite mixture models, responses to items are assumed to be independent of each other given the latent class membership, and we use $\boldsymbol{\Theta} = (\theta_{jm}) \in [0,1]^{J\times M}$ to denote the item parameter matrix. To be specific, for a subject's response $\boldsymbol{R} = (R_1, R_2, \dots, R_J)$, we have $\theta_{jm} = \mathbb{P}(R_j = 1 \mid m)$. Then the probability mass function of an LCM can be written as

$$\mathbb{P}(\boldsymbol{R} \mid \boldsymbol{\Theta}, \boldsymbol{\pi}) = \sum_{m=1}^{M} \pi_m \prod_{j=1}^{J} \theta_{jm}^{R_j} (1-\theta_{jm})^{1-R_j}. \tag{2.4}$$

This unrestricted LCM is saturated in the sense that no constraints are imposed on the latent classes' response distributions.

LAMs can be viewed as special cases of LCMs with $M = 2^K$ latent classes and additional constraints imposed on the components' distributions. Recall that in LAMs with K latent attributes, each latent attribute profile $\boldsymbol{\alpha}$ is a K-dimensional binary vector and has a proportion parameter $\pi_{\boldsymbol{\alpha}}$. The relationship between the latent attribute profiles in LAMs and the latent classes in LCMs can be seen by some one to one correspondence from $\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \{0,1\}^K\}$ to $\{m : m = 1, \dots, 2^K\}$, such as

$m = \sum_{k=1}^{K} \alpha_k \cdot 2^{k-1} + 1$. Therefore in a LAM with K latent attributes and no hierarchical structure, we have $M = 2^K$ latent classes. In HLAMs, the number of allowed latent attribute profiles is smaller than 2^K and we have $M = |\mathcal{A}|$, as discussed in Section 2.2.1. Moreover, in LAMs, there are additional restrictions on the item parameter matrix $\mathbf{\Theta}$ through the Q-matrix. Under these restrictions, for each item, certain subsets of item-level response probabilities will be constrained to be the same. Thus, a latent attribute model with or without any hierarchical structure can be viewed as a sub-model of a saturated LCM.

2.3 Regularized Estimation Method

2.3.1 Motivation and Proposed Method

To fit HLAMs, the Q-matrix, the hierarchical structures among the attributes, the item-level models, and the number of latent attributes are often needed to be pre-specified by domain experts, which however can be subjective and inaccurate. An important problem in cognitive diagnosis modeling then becomes how to efficiently and accurately learn these quantities from noisy observations.

In this section, we propose a unified modeling and inference approach to learning the latent structures, including the number of latent attributes K, the attribute-attribute hierarchical structure, the item-attribute Q-matrix, and the item-level diagnostic models. In particular, based on the observation in Section 2.2.2, we propose to learn an HLAM with minimal model assumptions starting with an unrestricted LCM. We use the following discussion and examples to further illustrate the key idea.

- As discussed in Section 2.2.1, when there exist hierarchical structures among the latent attributes, the number of truly existing latent attribute profiles is smaller than 2^K. For example, when $K = 4$, the total number of possible attribute profiles without any hierarchical structure is $2^K = 16$. Under different hierarchical structures as shown in Figure II.1, the numbers of existing attribute profiles are all smaller than 16. Therefore, to learn a hierarchical cognitive diagnosis model, we need first select significant latent attribute profiles that truly exist in the population.

- Furthermore, to reconstruct the Q-matrix and item models in hierarchical latent attribute models, it is also essential to examine the inner structure of the item parameter matrix. One key challenge here is that under certain model assumptions, there may exist some equivalent Q-matrices. Here we say two Q-matrices are equivalent under certain hierarchical structure $\mathcal{E}$, denoted by $\boldsymbol{Q}_1 \overset{\mathcal{E}}{\sim} \boldsymbol{Q}_2$, if they give the same item parameter matrices, that is, $\boldsymbol{\Theta}(\boldsymbol{Q}_1, \mathcal{A}_{\mathcal{E}}) = \boldsymbol{\Theta}(\boldsymbol{Q}_2, \mathcal{A}_{\mathcal{E}})$, where $\mathcal{A}_{\mathcal{E}}$ is the induced latent attribute profile set under hierarchy $\mathcal{E}$.

  As we introduced in Example II.1, for the DINA model, the item parameters only depend on the highest interactions among the required latent attributes. For such models, we have equivalent Q-matrices under hierarchical structures. For example, consider three latent attributes with a linear hierarchy, that is, $\mathcal{E} = \big\{1 \to 2 \to 3\big\}$. We have

$$\boldsymbol{Q}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \overset{\mathcal{E}}{\sim} \boldsymbol{Q}^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \overset{\mathcal{E}}{\sim} \boldsymbol{Q}^{(*)} = \begin{pmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{pmatrix}, \tag{2.5}$$

  where "$*$" can be either 0 or 1. However, when the underlying model is the GDINA model, since all the interactions among the latent attributes are considered, there would not exist such equivalent Q-matrices. For example, consider the second item of the Q-matrices in (2.5), for $\boldsymbol{Q}^{(1)}$, $\boldsymbol{q}_2^{(1)} = (0, 1, 0)$ and the corresponding item parameter vector under the GDINA model is $\boldsymbol{\theta}_2^{(1)} = (\beta_0,\ \beta_0,\ \beta_0 + \beta_2,\ \beta_0 + \beta_2)$. For $\boldsymbol{Q}^{(2)}$, $\boldsymbol{q}_2^{(2)} = (1, 1, 0)$ and the corresponding item parameter vector under the GDINA model is $\boldsymbol{\theta}_2^{(2)} = (\ \beta_0,\ \beta_0+\beta_1,\ \beta_0+\beta_1+\beta_2+\beta_{1,2},\ \beta_0+\beta_1+\beta_2+\beta_{1,2})$, which is different from that of $\boldsymbol{Q}^{(1)}$, and thus, the equivalence no longer holds under the GDINA model. Therefore, to learn the Q-matrix and infer the item models in HLAMs, it is also necessary to learn the item parameter matrix and investigate its inner constraint structure of it. Moreover, after learning the item parameter matrix, we can get the partial orders among the selected latent classes, which would in turn enable us to recover the latent hierarchies, the Q-matrix, and item models. We leave the details of the reconstruction of these quantities in Section 2.4.2.

Motivated by the above discussions and the fact that LAMs are a special family of LCMs with additional restrictions, we propose to start with an unrestricted latent class model and then put additional regularization terms, to select significant latent classes and learn the item parameter matrix simultaneously. Specifically, we start with a latent class model with M latent classes, where M is a large number, serving as an upper bound for the true number of latent classes. If the true number of latent classes is smaller than M, some of the proportion parameters will be zeros. Let $\boldsymbol{\pi} = (\pi_1, \dots, \pi_M)$ be the proportion parameter vector, and $\boldsymbol{\Theta} = (\theta_{jk}) \in (0,1)^{J\times M}$ be the item parameter matrix of the LCM, with $\boldsymbol{\theta}_j = (\theta_{jk}, k = 1, \dots, M)$ being the jth item's parameter vector. Then for a response data matrix $\mathcal{R} = (R_{ij} : i \in [N], j \in [J]) \in \{0,1\}^{N\times J}$, where N is the sample size, the likelihood can be written as

$$L_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R}) = \prod_{i=1}^{N} \Big(\sum_{k=1}^{M} \big(\pi_k \prod_{j=1}^{J} \theta_{jk}^{R_{ij}} (1-\theta_{jk})^{1-R_{ij}} \big) \Big). \tag{2.6}$$

And the log-likelihood is

$$l_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R}) = \sum_{i=1}^{N} \log \Big(\sum_{k=1}^{M} \big(\pi_k \prod_{j=1}^{J} \theta_{jk}^{R_{ij}} (1-\theta_{jk})^{1-R_{ij}} \big) \Big) \tag{2.7}$$

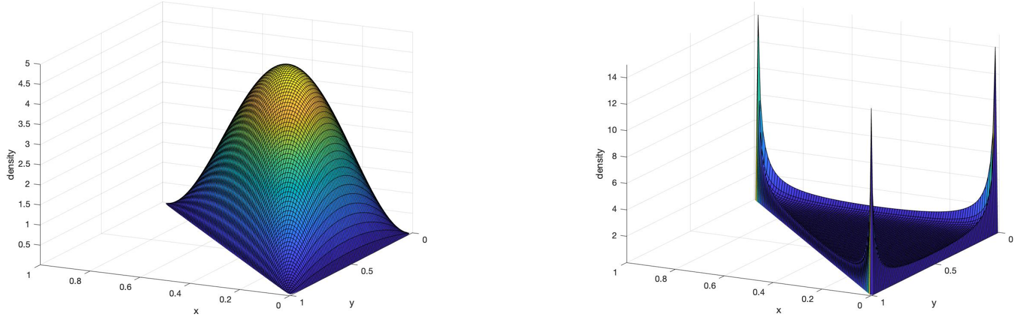
We consider the following objective function with two additional penalty terms:

$$l_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R}) - \lambda_1 \sum_{k=1}^{M} \log_{[\rho_N]} \pi_k - \lambda_2 \sum_{j=1}^{J} \mathcal{J}(\boldsymbol{\theta}_j), \tag{2.8}$$

where λ_1 and λ_2 are two nonnegative tuning parameters. The terms $\log_{[\rho_N]} \pi_k$ and $\mathcal{J}(\boldsymbol{\theta}_j)$ are two penalties and we discuss them one by one as follows.

The term $\log_{[\rho_N]} \pi_k = \log \pi_k \cdot \mathbb{I}\big(\pi_k > \rho_N\big) + \log \rho_N \cdot \mathbb{I}\big(\pi_k \leq \rho_N\big)$, is a log-type penalty (Gu and Xu, 2019b) on the proportion parameters, where ρ_N is a small threshold to circumvent the singularity of the log function at zero. Following Gu and Xu (2019b), we can take ρ_N to be a small value, such as N^{-d} for some $d \geq 1$. The log penalty is imposed on the proportion parameters, which forces small values in the proportion parameters to be zero. This log-type penalty also makes computation efficient in the E-step, as shown in our EM algorithm in Section 2.4.1. We can also interpret this log penalty from a Bayesian perspective, where we use a Dirichlet prior with parameter $1 - \lambda_1$ for the proportions. When $1 - \lambda_1 < 0$, it's not a proper Dirichlet distribution. But allowing $1 - \lambda_1 < 0$ would help us select significant proportion parameters more efficiently compared to the traditional proper Dirichlet priors. As shown in Figure II.3, when $\lambda_1 < 1$, the density concentrates more in the interior of the parameter space, while with $\lambda_1 > 1$ the density concentrates more on the boundary, encouraging sparsity of the proportion parameter vector. Therefore, it is essential to allow λ_1 to be nonnegative and even larger than 1 for the purpose of selecting the significant latent classes.

The penalty $\mathcal{J}(\boldsymbol{\theta}_j)$ is enforced on the item parameters for different latent classes item-wisely, which aims to learn the inner structure of the item parameter matrix. In particular, as discussed in Section 2.2.1, due to the restrictions of the Q-matrix and item model assumptions, for each item, some subset of latent attribute profiles have the same item parameters; and such constraint structure of the item parameter



(a)

(b)

Figure II.3: Illustration of Dirichlet Prior. (a): probability density function of 3-dimensional Dirichlet distribution with parameters all equal to $1 - \lambda_1 = 1$; (b): (improper) probability density function of 3-dimensional Dirichlet distribution with parameters all equal to $1 - \lambda_1 = -0.9$.

matrix can be used to further estimate the hierarchical structure and the Q-matrix. Therefore, to learn the set of latent classes that share the same item parameters, we put the penalty function $\mathcal{J}(\cdot)$ on the differences among the item parameters for each item. A popular choice for shrinkage estimation is the Lasso penalty, which however is known to produce biased estimation results. To overcome this issue, we propose to use the grouped truncated Lasso penalty (Shen et al., 2012),

$$\mathcal{J}_\tau(\boldsymbol{\theta}_j) = \sum_{1 \le k < l \le M} \mathrm{TLP}(|\theta_{jk} - \theta_{jl}|; \tau),$$

where $\mathrm{TLP}(x;\tau) = \min(|x|, \tau)$, and τ here is a positive tuning parameter. Figure II.4 (a) provides an example for the TLP with $\tau = 1$. Moreover, since we only focus on the item parameters for significant latent classes, we use

$$\mathcal{J}_{\tau,\rho_N}(\boldsymbol{\theta}_j) = \sum_{\substack{1 \le k < l \le M, \\ \pi_l > \rho_N, \pi_k > \rho_N}} \mathrm{TLP}(|\theta_{jk} - \theta_{jl}|; \tau).$$

A key feature of the truncated Lasso penalty is that it can be regarded as a L_1 penalty for a small $|x| \le \tau$, while it does not put further penalization for a large $|x| > \tau$.

In this way, it corrects the Lasso bias through adaptive shrinkage combined with thresholding. It discriminates small from large differences through thresholding and consequently is capable of handling low-resolution differences through tuning τ.

In Chen et al. (2017b), the authors used the Smoothly Clipped Absolute Deviation (SCAD) penalty (Fan and Li, 2001), which can also be used to merge similar item probabilities. The SCAD penalty is similar to the TLP, while there is a quadratic spline function between the L_1 penalty for small values and the constant penalty for large values. Specifically, the SCAD penalty is expressed as below

$$p_{\lambda,a}^{\text{SCAD}}(x) = \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda, \\ -\left(\frac{|x|^2-2a\lambda|x|+\lambda^2}{2(a-1)}\right) & \text{if } \lambda < x < a\lambda, \\ \frac{(a+1)\lambda^2}{2} & \text{if } |x| \geq a\lambda. \end{cases}$$

Figure II.4 (c) plots the SCAD penalty with $\lambda = 0.5$ and $a = 2$. Here, we want to mention several additional advantages of using the truncated Lasso penalty. First, it performs the model selection task of the L_0 function by providing a computationally efficient surrogate. When τ is sufficiently small, the truncated Lasso penalty has a good approximation to the L_0 penalty. Moreover, although it is not a convex function, it is piecewise linear and can be decomposed into a difference of two convex functions as illustrated in Figure II.4 (a) and Figure II.4 (b), which allows us to use Difference Convex (DC) programming (Tuy, 1995), gaining computational advantages. TLP also has nice likelihood oracle properties studied in previous literature (Shen et al., 2012).

Remark II.1. *Our approach shares some similarities with the regularized LCM approach in Chen et al. (2017b) that both use exploratory LCMs to estimate the latent structures. However, in Chen et al. (2017b), the number of latent classes is pre-specified or selected in a way that all the possible values should be considered. This*

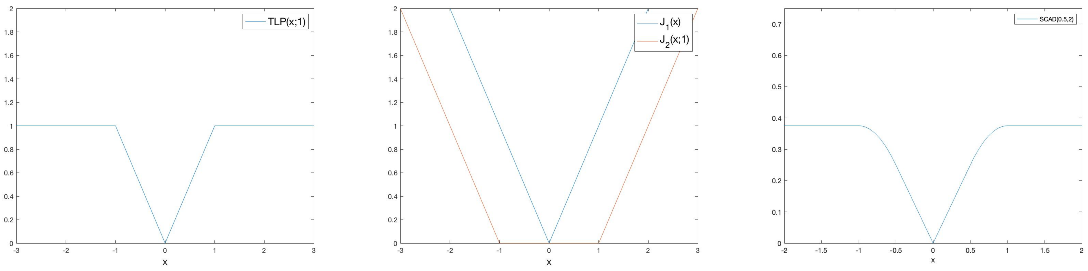


Figure II.4: Illustration of TLP and SCAD. (a): truncated Lasso function TLP$(x; \tau)$ with $\tau = 1$; (b): the DC decomposition into a difference of two convex functions $J_1(x)$ and $J_2(x; 1)$; (c): SCAD penalty with $\lambda = 0.5$ and $a = 2$.

could require significantly more computational efforts when the number of latent attributes K is large since there will be 2^K possible latent classes. For instance $K = 10$ would lead to $2^K = 1024$ possible candidate M values. On the contrary, in our method, we only need an upper bound for the number of latent classes, and our model would perform the selection of significant latent classes more efficiently through the added log penalty. Moreover, in Section 2.4 we also develop a novel estimation algorithm to recover the number of latent attributes, the hierarchical structures among the attributes, and the Q-matrix, based on the proposed regularization estimation results.

Remark II.2. *In Wang and Lu (2021), the authors also studied the latent variable selection approach, which, however, required a pre-specified number of latent attributes and a known identity sub-matrix in the Q-matrix. Moreover, a hard cutoff for proportion parameters was used to select significant latent classes. For example, they chose 0.05 as the cutoff when $K = 3$ and 0.025 when $K = 4$. This hard cutoff in fact played a decisive role in determining the significant latent classes. However, there is neither a systematical way nor theoretical guarantee to select this cutoff, making it less practical in real applications.*

2.3.2 Theoretical Proporties

In this section, we present the statistical properties of the regularized estimator obtained from (2.8). We first present some identifiability results of hierarchical latent attribute models from Gu and Xu (2019b). Then we will show that, under suitable conditions, the regularized estimator is consistent for model selection. In the following, for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of dimension n, we say $\boldsymbol{a} \preceq \boldsymbol{b}$ if $a_i \leq b_i$ for $i = 1, \ldots, n$, and $\boldsymbol{a} \succeq \boldsymbol{b}$ if $a_i \geq b_i$ for $i = 1, \ldots, n$.

The identifiability of the model parameters depends on the restrictions of the item parameter matrix. To characterize the identifiability conditions, we introduce an indicator matrix of most capable classes as $\boldsymbol{\Gamma} := \big(\mathbb{I}\{\theta_{j,m} = \max_{m' \in [M]} \theta_{j,m'}\}, j \in [J], m \in [M]\big) \in \{0,1\}^{J \times M}$, indicating whether the latent classes possess the highest level of each item's positive response probability. Let $\boldsymbol{\Gamma}_{\cdot,m}$ denote the mth column vector of the $\boldsymbol{\Gamma}$ matrix. Based on the indicator matrix, we can define a partial order among latent classes. For $1 \leq m_1 \neq m_2 \leq M$, we say latent class m_1 is of a larger order than latent class m_2 under $\boldsymbol{\Gamma}$ if $\boldsymbol{\Gamma}_{\cdot,m_1} \succeq \boldsymbol{\Gamma}_{\cdot,m_2}$. See Figure II.5 for an illustrative example, where we use a Directed Acyclic Graph (DAG) to represent partial orders, and $\boldsymbol{\Gamma}_{\cdot,m_1}$ points to $\boldsymbol{\Gamma}_{\cdot,m_2}$ if $\boldsymbol{\Gamma}_{\cdot,m_1} \preceq \boldsymbol{\Gamma}_{\cdot,m_2}$.

$$\boldsymbol{\Theta} = \begin{pmatrix} 0.2 & 0.8 & 0.8 \\ 0.2 & 0.2 & 0.8 \\ 0.2 & 0.2 & 0.8 \end{pmatrix}, \qquad \boldsymbol{\Gamma} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

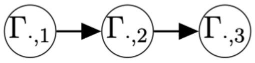


Figure II.5: Indicator matrix and partial orders.

As in Gu and Xu (2019b), for LAMs, we define the indicator matrix for a set of latent attribute profiles $\mathcal{A}$ as $\boldsymbol{\Gamma}^{\mathcal{A}} := \big(\mathbb{I}\{\theta_{j,\boldsymbol{\alpha}} = \max_{\boldsymbol{\alpha}' \in \mathcal{A}} \theta_{j,\boldsymbol{\alpha}'}\} : \ j \in [J], \ \boldsymbol{\alpha} \in \mathcal{A}\big) \in \{0,1\}^{J \times |\mathcal{A}|}$. Note that if we take the set of latent classes as the set of attribute profiles, the indicator matrix of an LCM is equivalent to that of a LAM. Similarly, we define the proportion parameter vector and item parameter matrix for a set of latent attribute profiles $\mathcal{A}$ as $\boldsymbol{\pi}^{\mathcal{A}} = \big(\pi_{\boldsymbol{\alpha}} : \ \boldsymbol{\alpha} \in \mathcal{A}\big)$ and $\boldsymbol{\Theta}^{\mathcal{A}} = \big(\theta_{j,\boldsymbol{\alpha}} : \ j \in [J], \ \boldsymbol{\alpha} \in \mathcal{A}\big)$. Following Gu and Xu (2019b), for any subset of items $S \subset [J]$, we define a partial order among the latent attribute profiles. For $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}$, we say $\boldsymbol{\alpha} \succeq_S \boldsymbol{\alpha}'$ under $\mathbf{\Gamma}^{\mathcal{A}}$ if $\Gamma^{\mathcal{A}}_{j,\boldsymbol{\alpha}} \geq \Gamma^{\mathcal{A}}_{j,\boldsymbol{\alpha}'}$ for $j \in S$. And for two item sets S_1 and S_2, we say "$\succeq_{S_1} = \succeq_{S_2}$" if for any $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}$, we have $\boldsymbol{\alpha} \succeq_{S_1} \boldsymbol{\alpha}'$ if and only if $\boldsymbol{\alpha} \succeq_{S_2} \boldsymbol{\alpha}'$. Note that if we take the item set to be the set of all items, the definitions of indicator matrix and partial orders are the same as those in Section 2.3.1. For a subset of items $S \subset [J]$ and a set of attribute profiles $\mathcal{A}$, we define the corresponding indicator matrix $\mathbf{\Gamma}^{(S,\ \mathcal{A})} = \big(\Gamma_{j,\boldsymbol{\alpha}} :\ j \in S,\ \boldsymbol{\alpha} \in \mathcal{A}\big)$.

We first state the definition of strict identifiability for latent hierarchy and model parameters.

Definition 2.3.1 (strict identifiability, Gu and Xu (2019b))**.** Consider an LAM with a hierarchy $\mathcal{E}_0$ and the induced latent attribute profile set $\mathcal{A}_0$. $\mathcal{A}_0$ is said to be (strictly) identifiable if for any indicator matrix $\mathbf{\Gamma}^{\mathcal{A}}$ of size $J \times |\mathcal{A}|$ with $|\mathcal{A}| \leq |\mathcal{A}_0|$, any proportion parameter vector $\boldsymbol{\pi}^{\mathcal{A}}$ and any valid item parameter matrix $\mathbf{\Theta}^{\mathcal{A}}$ respecting constraints given by $\mathbf{\Gamma}^{\mathcal{A}}$, the following equality

$$\mathbb{P}(\boldsymbol{R} \mid \boldsymbol{\pi}^{\mathcal{A}}, \mathbf{\Theta}^{\mathcal{A}}) = \mathbb{P}(\boldsymbol{R} \mid \boldsymbol{\pi}^{\mathcal{A}_0}, \mathbf{\Theta}^{\mathcal{A}_0}) \tag{2.9}$$

implies $\mathcal{A} = \mathcal{A}_0$. Moreover, if (2.9) implies $\big(\boldsymbol{\pi}^{\mathcal{A}}, \mathbf{\Theta}^{\mathcal{A}}\big) = \big(\boldsymbol{\pi}^{\mathcal{A}_0}, \mathbf{\Theta}^{\mathcal{A}_0}\big)$, then we say the model parameters $\big(\boldsymbol{\pi}^{\mathcal{A}_0}, \mathbf{\Theta}^{\mathcal{A}_0}\big)$ are (strictly) identifiable.

The following theorem provides sufficient conditions for strict identifiability of latent hierarchies and model parameters.

Theorem 2.3.2 (strict identifiability, Gu and Xu (2019b))**.** *Consider a LAM with a hierarchy $\mathcal{E}_0$. The hierarchy is identifiable if the following conditions of the indicator matrix $\mathbf{\Gamma}^{\mathcal{A}_0}$ corresponding to the induced latent attribute profile set $\mathcal{A}_0$ are satisfied:*

(1) There exist two disjoint item sets S_1 and S_2, such that $\mathbf{\Gamma}^{(S_i,\ \mathcal{A}_0)}$ has distinct column vectors for $i = 1, 2$ and "$\succeq_{S_1}$"="$\succeq_{S_2}$" under $\mathbf{\Gamma}^{\mathcal{A}_0}$.

(2) For any $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}_0$ where $\boldsymbol{\alpha}' \succeq_{S_i} \boldsymbol{\alpha}$ under $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ for $i = 1$ or 2, there exists some $j \in \left(S_1 \cup S_2\right)^c$ such that $\Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}} \neq \Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}'}$.

(3) Any column vector of $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ is different from any column vector of $\boldsymbol{\Gamma}^{\mathcal{A}_0^c}$, where $\mathcal{A}_0^c = \{0,1\}^K \setminus \mathcal{A}_0$.

Moreover, under Conditions (1) - (3), the model parameters $\left(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}\right)$ associated with $\mathcal{A}_0$ are also identifiable.

Theorem 2.3.2 provides conditions for strict identifiability of hierarchical structures and model parameters. The strict identifiability can be relaxed to generic identifiability, where the hierarchy and model parameters can be identified except for a zero-measure set. The definition of generic identifiability is defined below.

Definition 2.3.3 (generic identifiability, Gu and Xu (2019b))**.** Consider an LAM with a hierarchy $\mathcal{E}_0$ and the induced latent attribute profile set $\mathcal{A}_0$. Denote the parameter space of $(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0})$ constrained by $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ by Ω. We say $\mathcal{A}_0$ is generically identifiable, if there exists a subset $\mathcal{V} \subset \Omega$ that has a Lebesgue measure zero, such that for any $(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}) \in \Omega \setminus \mathcal{V}$, Equation (2.9) implies $\mathcal{A} = \mathcal{A}_0$. Moreover, for any $(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}) \in \Omega \setminus \mathcal{V}$, if (2.9) implies $\left(\boldsymbol{\pi}^{\mathcal{A}}, \boldsymbol{\Theta}^{\mathcal{A}}\right) = \left(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}\right)$, then we say $\left(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}\right)$ are generically identifiable.

Theorem 2.3.4 (generic identifiability, Gu and Xu (2019b))**.** *Consider an LAM with a hierarchy $\mathcal{E}_0$. The hierarchy is generically identifiable, if the following conditions of the indicator matrix $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ corresponding to the induced latent attribute profile set $\mathcal{A}_0$ are satisfied:*

1. *There exist two disjoint item sets S_1 and S_2, such that altering some entries from 0 to 1 in $\boldsymbol{\Gamma}^{(S_1 \cup S_2,\ \mathcal{A}_0)}$ yields a $\tilde{\boldsymbol{\Gamma}}^{(S_1 \cup S_2,\ \mathcal{A}_0)}$ satisfying that $\tilde{\boldsymbol{\Gamma}}^{(S_i,\ \mathcal{A}_0)}$ has distinct column vectors for $i = 1, 2$ and "$\succeq_{S_1}$"="$\succeq_{S_2}$" under $\tilde{\boldsymbol{\Gamma}}^{\mathcal{A}_0}$.*

2. *For any $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}_0$ where $\boldsymbol{\alpha}' \succeq_{S_i} \boldsymbol{\alpha}$ under $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ for $i = 1$ or 2, there exists some $j \in \left(S_1 \cup S_2\right)^c$ such that $\tilde{\Gamma}^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}} \neq \tilde{\Gamma}^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}'}$.*

3. *Any column vector of $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ is different from any column vector of $\boldsymbol{\Gamma}^{\mathcal{A}_0^c}$, where $\mathcal{A}_0^c = \{0,1\}^K \setminus \mathcal{A}_0$.*

Moreover, under conditions (A) - (C), the model parameters $\left(\boldsymbol{\pi}^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}\right)$ associated with $\mathcal{A}_0$ are also generically identifiable.

Theorem 2.3.4 establishes generic identifiability conditions where the hierarchical structure and model parameters can be identified except for a zero-measure set of parameters. To establish consistency results, we need to make the following assumption.

Assumption 2.3.5. $\left[l_N(\hat{\boldsymbol{\pi}}^*, \hat{\boldsymbol{\Theta}}^*) - l_N(\hat{\boldsymbol{\pi}}_0, \hat{\boldsymbol{\Theta}}_0)\right]/N = O_p(N^{-\delta})$, *for some* $1/2 < \delta \leq 1$, *where* $(\hat{\boldsymbol{\pi}}^*, \hat{\boldsymbol{\Theta}}^*)$ *is the maximum likelihood estimator (MLE) directly obtained from* (2.7), *and* $(\hat{\boldsymbol{\pi}}_0, \hat{\boldsymbol{\Theta}}_0)$ *is the Oracle MLE obtained under the condition that the number of latent attributes, the hierarchical structure, the Q-matrix and item-level diagnostic models are known.*

When $\delta = 1$, Assumption 1 corresponds to the usual root-N convergence rate of the estimators, while $1/2 < \delta \leq 1$ corresponds to a slower convergence rate. Here we make a general assumption to cover different situations. In Gu and Xu (2019b), the authors made a similar assumption about the convergence rate of the likelihood.

We use $(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$ to denote the true model parameter and $M_0 := |\mathcal{A}_0|$ to denote the true number of latent classes, where $\mathcal{A}_0$ is the reduced latent attribute profile set under the true hierarchical structure $\mathcal{E}_0$. For $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$ obtained from optimizing (2.8), we define the selected latent classes as $\{m : \hat{\pi}_m > \rho_N,\ m \in [M]\}$, and the number of selected latent classes $\hat{M} := \left|\{m : \hat{\pi}_m > \rho_N,\ m \in [M]\}\right|$. For the true item parameter matrix $\boldsymbol{\Theta}^0$, we defined the set $S^0 = \big\{(j, k_1, k_2) : \theta^0_{j,k_1} = \theta^0_{j,k_2}, 1 \leq$

$k_1 < k_2 \leq M_0, 1 \leq j \leq J\big\}$ to indicate the constraint structure of the item parameter matrix. Similarly, for $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$, we define $\hat{S} = \big\{(j, k_1, k_2) : \hat{\theta}_{j,k_1} = \hat{\theta}_{j,k_2},\ 1 \leq k_1 < k_2 \leq M,\ \hat{\pi}_{k_1} > \rho_N,\ \hat{\pi}_{k_2} > \rho_N\big\}$. We say $\hat{S} \sim S^0$ if there exists a column permutation σ of $\hat{\boldsymbol{\Theta}}$ such that $\hat{S}_\sigma = S^0$. Given the above assumptions, we have the following consistency results.

Theorem 2.3.6 (consistency)**.** *Suppose the identifiability conditions in Theorem 2.3.2 are satisfied and Assumption 2.3.5 holds. For λ_1, λ_2, τ and ρ_N satisfying $N^{1-\delta}|\log \rho_N|^{-1} = o(\lambda_1)$, $\lambda_1 = o(N|\log \rho_N|^{-1})$ and $\lambda_2\tau = o(\lambda_1|\log \rho_N|)$, we can select the true number of latent classes consistently, that is,*

$$\mathbb{P}\big(\hat{M} \neq M_0\big) \to 0, \ \textit{as } N \to \infty. \tag{2.10}$$

Moreover, the estimated parameter $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$ is also consistent of $(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$. If we further assume $\lambda_1 = o(N^{1/2})$, $\lambda_2\tau = o(N^{1/2})$, $\lambda_2 N^{-1/2} \to \infty$ and $\tau N^{1/2} \to \infty$, up to a column permutation, the identical item parameter pair set S^0 is also consistently estimated,

$$\mathbb{P}\big(\hat{S} \nsim S^0\big) \to 0, \ \textit{as } N \to \infty. \tag{2.11}$$

Theorem 2.3.6 implies that with suitable choices of hyperparameters, we can correctly select the number of latent classes and learn the inner structure of the item parameter matrix consistently as sample size N goes to infinity. For example, we can take $\rho_N \sim N^{-d}$ for some $d \geq 1$, $\lambda_1 \sim N^{\frac{1}{2}-\epsilon_1}$, $\lambda_2 \sim N^{\frac{1}{2}+\epsilon_2}$ and $\tau \sim N^{-\epsilon_3}$ for some small positive constants $\epsilon_1, \epsilon_2, \epsilon_3$ satisfying that $0 < \epsilon_1 < \delta - 1/2,\ 0 < \epsilon_2 < \epsilon_3 < 1/2$ and $\epsilon_3 - \epsilon_2 > \epsilon_1$. Moreover, if the conditions in Theorem 2.3.4 are satisfied, we can consistently estimate the true number of latent classes and inner structure of the item parameter matrix except for a zero-measure set of model parameters. In practice, we can use information criteria, such as the Bayesian Information Criterion (BIC, Schwarz et al., 1978), to help select the tuning parameters, which will be further discussed in Section 2.4.1. The proof of the theorem is presented in Appendix A.

Based on the learned latent classes and estimated item parameter matrix, we develop a latent structure recovery algorithm outlined in Algorithm II.2 in Section 2.4. Specifically, we recover the number of latent attributes, latent hierarchies, and the Q-matrix based on the partial orders among the selected latent classes. Under the identifiability conditions, due to the consistency of the item parameter estimator $\hat{\boldsymbol{\Theta}}$ and the inner structure $\hat{S}$ established, we can also consistently recover the partial orders among the latent classes, which ultimately leads to the consistency of the estimated number of latent attributes, the hierarchical structures and the Q-matrix. For algorithm details, please see Section 2.4.

2.4 Learning Algorithms

2.4.1 Penalized EM Algorithm

In this section, we develop an efficient EM algorithm for the proposed model. For an LCM, the complete data log-likelihood function can be written as

$$l_C(\mathcal{R}, \boldsymbol{z}; \boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{i=1}^{N} \sum_{k=1}^{M} z_{ik} \log \big(\pi_k \varphi(\boldsymbol{R}_i; \boldsymbol{\theta}_k) \big), \tag{2.12}$$

where $\varphi(\boldsymbol{R}_i; \boldsymbol{\theta}_k) = \prod_{j=1}^{J} \theta_{jk}^{R_{ij}} (1-\theta_{jk})^{1-R_{ij}}$ and $\boldsymbol{z} \in \{0,1\}^{N \times M}$ is the latent variable in which z_{ik} indicates whether the ith subject belongs to the kth latent class. Then in an EM algorithm without additional penalty, we maximize the following objective function at the $(c+1)$th iteration:

$$\max_{\boldsymbol{\pi}, \boldsymbol{\Theta}} Q(\boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)}) = \sum_{i=1}^{N} \sum_{k=1}^{M} s_{ik}^{(c)} \big(\log \pi_k + \log \varphi_k(\boldsymbol{R}_i; \boldsymbol{\theta}_k) \big), \tag{2.13}$$

where

$$s_{ik}^{(c)} = \mathbb{E}_{\boldsymbol{\pi}^{(c)},\boldsymbol{\Theta}^{(c)}}[z_{ik} = 1 \mid \boldsymbol{R}_i] = \frac{\pi_k^{(c)}\varphi_k(\boldsymbol{R}_i;\boldsymbol{\theta}_k^{(c)})}{\sum_{k'}\pi_{k'}^{(c)}\varphi_{k'}^{(c)}(\boldsymbol{R}_i;\boldsymbol{\theta}_{k'}^{(c)})}.$$

With the additional penalty terms in (2.8), the new objective function denoted as $G(\boldsymbol{\pi},\boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)},\boldsymbol{\Theta}^{(c)})$ becomes:

$$\begin{aligned}\min_{\boldsymbol{\pi},\boldsymbol{\Theta}} G(\boldsymbol{\pi},\boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)},\boldsymbol{\Theta}^{(c)}) = &-\frac{1}{N}Q(\boldsymbol{\pi},\boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)},\boldsymbol{\Theta}^{(c)}) \\ &+ \tilde{\lambda}_1\sum_{k=1}^{M}\log_{[\rho_N]}\pi_k + \tilde{\lambda}_2\sum_{j=1}^{J}\mathcal{J}_{\tau,\rho_N}(\boldsymbol{\theta}_j),\end{aligned}$$

where $\tilde{\lambda}_1 = \lambda_1/N$ and $\tilde{\lambda}_2 = \lambda_2/N$.

As we mentioned in Section 2.3.1, the truncated Lasso penalty can be decomposed into a difference between two convex functions. Therefore we can utilize DC programming (Tuy, 1995) to optimize G. Moreover, we also exploit the Alternating Direction Method of Multipliers (ADMM, Boyd et al., 2011) method to facilitate solving the problem. There are several advantages of using ADMM to perform optimization here. Updating the parameters in an alternating or sequential fashion takes advantage of the decomposability of dual ascent while using the method of multipliers enables superior convergence properties (Boyd et al., 2011). In practice, we also observe that the ADMM algorithm converges within a few tens of iterations in our simulation and real data studies. The algorithm is summarized in Algorithm II.1 and the derivations of the algorithm are presented in Appendix A.

We want to note that our algorithm can naturally handle missing values. If $\mathcal{R} = (\mathcal{R}_{\text{obs}},\mathcal{R}_{\text{miss}})$ is the decomposition of the full data matrix into the observed part $\mathcal{R}_{\text{obs}}$ and the missing part $\mathcal{R}_{\text{miss}}$, then after marginalization over the missing values, the initial likelihood $l(\mathcal{R};\boldsymbol{\pi},\boldsymbol{\Theta})$ simplifies to $l(\mathcal{R}_{\text{obs}};\boldsymbol{\pi},\boldsymbol{\Theta})$. Then a natural implementation could be based on indexing the inference procedure so that the posterior conditionals only involve sums over the observed values. The detailed Penalized EM

Algorithm II.1: PEM: Penalized EM with log-penalty and TLP

```
Data: Binary response matrix $\mathcal{R} = (R_{i,j})_{N\times J}$.
Set hyperparameters $\tilde{\lambda}_1,\ \tilde{\lambda}_2,\ \tau,\ \gamma$ and $\rho$.
Set an upper bound for the number of latent classes $M$.
Initialize parameters $\boldsymbol{\pi}$, $\boldsymbol{\Theta}$, and the conditional expectations $\boldsymbol{s}$.
while not converged do
    In the $(c+1)th$ iteration,
    for $(i,k) \in [N]\times[M]$ do
        $s_{ik}^{(c+1)} = \frac{\pi_k^{(c)}\varphi_k(\boldsymbol{R}_i;\boldsymbol{\theta}_k^{(c)})}{\sum \pi_{k'}^{(c)}\varphi_{k'}(\boldsymbol{R}_i;\boldsymbol{\theta}_{k'}^{(c)})}$, where $\varphi(\boldsymbol{R}_i;\boldsymbol{\theta}_k) = \prod_{j=1}^{J}\theta_{jk}^{R_{ij}}(1-\theta_{jk})^{1-R_{ij}}$.
    end
    for $k \in [M]$ and $\pi_k^{(c)} > \rho$ do
        $\pi_k^{(c+1)} = \frac{\sum_{i=1}^{N} s_{ik}^{(c+1)}/N - \tilde{\lambda}_1}{1 - M\tilde{\lambda}_1}$.
    end
    for $(j,k) \in [J]\times[M]$ and $\pi_k^{(c+1)} > \rho$ do
        $\theta_{jk}^{(c+1)} = \underset{\theta_{jk}}{\operatorname{argmin}}\Big\{ -\frac{\sum_{i=1}^{N} s_{ik}^{(c)} R_{ij}}{N}\log(\theta_{jk})$
            $- \frac{\sum_{i=1}^{N} s_{ik}^{(c)}(1-R_{ij})}{N}\log(1-\theta_{jk})$
            $+ \frac{\gamma}{2}\sum_{l>k}\big(\hat{d}_{jkl}^{(c)} - (\theta_{jk} - \hat{\theta}_{jl}^{(c)}) + \hat{\mu}_{jkl}^{(c)}\big)^2$
            $+ \frac{\gamma}{2}\sum_{l<k}\big(\hat{d}_{jlk}^{(c)} - (\hat{\theta}_{jl}^{(c+1)} - \theta_{jk}) + \hat{\mu}_{jlk}^{(c)}\big)^2\Big\}$
    end
    for $j \in [J]$, $1 \le k < l \le M$ and $\pi_k^{(c+1)} > \rho$, $\pi_l^{(c+1)} > \rho$ do
        $\hat{d}_{jkl}^{(c+1)} = \begin{cases} \big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}\big)\cdot\mathbb{I}\big(|\hat{d}_{jkl}^{(c)}| \ge \tau\big) \\ \mathrm{ST}\big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)};\tilde{\lambda}_2/\gamma\big)\cdot\mathbb{I}\big(|\hat{d}_{jkl}^{(c)}| < \tau\big), \end{cases}$
            where $\mathrm{ST}(x;\gamma) = (|x|-\gamma)_+ x/|x|$.
        $\hat{\mu}_{jkl}^{(c+1)} = \hat{\mu}_{jkl}^{(c)} + \hat{d}_{jkl}^{(c+1)} - \big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)}\big)$
    end
end
Output: $\{\hat{\boldsymbol{\pi}},\ \hat{\boldsymbol{\Theta}},\ \hat{\boldsymbol{s}}\}$
```

algorithm with missing values is also summarized in Appendix A.

To address the computational bottleneck when faced with large-scale datasets, we can also use a stochastic version of the aforementioned EM algorithm. In each iteration, we randomly subsample a subset S_r of rows (subjects), and a subset S_c of columns (items). Then update the conditional expectation $s_i^{(c)}$ for $i \in S_r$ with items in S_c. Updates in M-step remain the same, which will give us an intermediate model parameter $(\hat{\boldsymbol{\pi}}^{(c+1/2)}, \hat{\mathbf{\Theta}}^{(c+1/2)})$. Then we use a weighted average of $(\hat{\boldsymbol{\pi}}^{(c)}, \hat{\mathbf{\Theta}}^{(c)})$ and $(\hat{\boldsymbol{\pi}}^{(c+1/2)}, \hat{\mathbf{\Theta}}^{(c+1/2)})$ to update the model parameters. Appropriate weights will provably lead to convergence to a local optimum (Delyon et al., 1999).

In terms of hyperparameter tuning, we use BIC defined as below:

$$\mathrm{BIC}(\boldsymbol{\pi}, \mathbf{\Theta}) = -2l_N(\boldsymbol{\pi}, \mathbf{\Theta}) + \log N\big(M_{\rho_N} - 1 + \sum_{j=1}^{J} \dim(\boldsymbol{\theta}_j)\big) \tag{2.14}$$

where l_N is the log-likelihood, $M_{\rho_N} := \big|\{m : \pi_m > \rho_N, m \in [M]\}\big|$ is the selected number of latent classes, and $\dim(\boldsymbol{\theta}_j)$ is the number of distinct values in the set $\{\theta_{j,m} : \pi_m > \rho_N,\ m \in [M]\}$, that is, the number of distinct item parameters for item j corresponding to the selected latent classes. Our simulation results in Section 2.5 show that BIC performed well. We can also use other selection criteria such as EBIC (Chen and Chen, 2008) when the number of latent attributes K is large. From the matrix completion perspective, we may also perform cross-validation to choose tuning parameters.

2.4.2 Recover Latent Hierarchies and Q-matrix

Once we fit the model and get the estimates of the model parameters including the number of significant latent classes $\hat{M}$, proportion parameters $\hat{\boldsymbol{\pi}}$ and item parameter matrix $\hat{\mathbf{\Theta}}$, our next goal is to recover the number of latent attributes, the latent hierarchical structure, the Q-matrix, and item models.

To this end, we develop an algorithm based on the indicator matrix

$$\mathbf{\Gamma} := \left(\mathbb{I}\{\hat{\theta}_{j,m} = \max_{l \in [\hat{M}]} \hat{\theta}_{j,l}\} : \ j \in [J], \ m \in [\hat{M}] \right) \in \{0,1\}^{J \times \hat{M}},$$

indicating whether a latent class possesses the highest level of an item's parameters. One common assumption in LAMs is that more capable subjects have higher item parameters and thus larger indicator vectors, that is, $\mathbf{\Gamma}_{\cdot,\boldsymbol{\alpha}} \succeq \mathbf{\Gamma}_{\cdot,\boldsymbol{\alpha}^*}$, if $\boldsymbol{\alpha} \succeq \boldsymbol{\alpha}^*$. Based on this assumption, we can get partial orders among the latent classes. Then we can find the smallest integer K such that some binary representations with K digits satisfy these partial orders, and the binary representations can be treated as the learned latent attribute profiles. With these reconstructed latent attribute profiles, we can subsequently recover the hierarchical structures among the latent attributes and the Q-matrix.

Specifically, based on the indicator matrix, we get the partial orders among the latent classes. We use a matrix $\boldsymbol{P} \in \{0,1\}^{\hat{M} \times \hat{M}}$ to represent the partial orders, where $P_{m_1,m_2} = 1$ indicates that $\mathbf{\Gamma}_{\cdot,m_1} \preceq \mathbf{\Gamma}_{\cdot,m_2}$. Since we only want to include direct partial orders, for any (m_1, m_2) such that $\left(\boldsymbol{P}^2\right)_{m_1,m_2} > 0$, we set $P_{m_1,m_2} = 0$. For example, if $\mathbf{\Gamma}_{\cdot,m_1} \preceq \mathbf{\Gamma}_{\cdot,m_2}$, $\mathbf{\Gamma}_{\cdot,m_1} \preceq \mathbf{\Gamma}_{\cdot,m_3}$ and $\mathbf{\Gamma}_{\cdot,m_2} \preceq \mathbf{\Gamma}_{\cdot,m_3}$, since m_2 here is an intermediate latent class between m_1 and m_3, we will not include the partial order $\mathbf{\Gamma}_{\cdot,m_1} \preceq \mathbf{\Gamma}_{\cdot,m_3}$ in $\boldsymbol{P}$. From $\boldsymbol{P}$, we can get a partial order set $\{m_1 \to m_2 : P_{m_1,m_2} = 1\}$, based on which a DAG can be plotted, where $\mathbf{\Gamma}_{\cdot,m_1}$ points to $\mathbf{\Gamma}_{\cdot,m_2}$ if $\mathbf{\Gamma}_{\cdot,m_1} \preceq \mathbf{\Gamma}_{\cdot,m_2}$. One can see the partial order matrix $\boldsymbol{P}$ in fact is the adjacency matrix of the DAG. An example of the indicator matrix, the partial order matrix, and the corresponding DAG is shown in Figure II.6. In a DAG, we call a node at the start of an arrow as a parent node and a node at the end of an arrow as a child node. Note that since we always include the most basic attribute profile with all attributes being 0 and the most capable attribute profile with all attributes being 1, and any other latent attribute profile will lie between them, there is always a path passing each latent attribute profile from the most basic one to the most capable one. After we plot the DAG, we then recover the

$$\mathbf{\Gamma} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

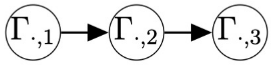

$\mathbf{\Gamma}_{\cdot,1}$
$\mathbf{\Gamma}_{\cdot,2}$
$\mathbf{\Gamma}_{\cdot,3}$

Figure II.6: Indicator matrix, partial order matrix, and corresponding DAG.

binary representations of the latent classes. We start from the most basic one and move forward layer by layer. Specifically, when we construct binary representations, we can find its parent nodes, and follow two rules below:

- If the node has only one parent node, then we need to add a dimension in the binary representations.
- If the node has several parent nodes, then we set the binary representation of the node to be the union of all of its parent nodes.

We use examples in Figure II.7 and Figure II.8 to illustrate the procedures of recovering binary representations. In the upper plot of Figure II.7, $\mathbf{\Gamma}_{\cdot,2}$ only has one parent node $\mathbf{\Gamma}_{\cdot,1}$, then we need to add a dimension in the binary representations. In the middle plot of Figure II.7, $\mathbf{\Gamma}_{\cdot,3}$ has two parent nodes $\mathbf{\Gamma}_{\cdot,1}$ and $\mathbf{\Gamma}_{\cdot,2}$. Since there is no partial order between $\mathbf{\Gamma}_{\cdot,1}$ and $\mathbf{\Gamma}_{\cdot,2}$, then there are at least two dimensions in which $\mathbf{\Gamma}_{\cdot,1}$ and $\mathbf{\Gamma}_{\cdot,2}$ have different values. Then we should set $\mathbf{\Gamma}_{\cdot,3}$ to be the union of $\mathbf{\Gamma}_{\cdot,1}$ and $\mathbf{\Gamma}_{\cdot,2}$, which will be larger than $\mathbf{\Gamma}_{\cdot,1}$ and $\mathbf{\Gamma}_{\cdot,2}$. A more general case is shown in the lower plot of Figure II.7. In Figure II.8, we provide a more complicated example. Since $\mathbf{\Gamma}_{\cdot,2}$ only has one parent node $\mathbf{\Gamma}_{\cdot,1}$, we need one binary digit for $\mathbf{\Gamma}_{\cdot,2}$ and set $\mathbf{\Gamma}_{\cdot,1} = (0)$ and $\mathbf{\Gamma}_{\cdot,2} = (1)$. Since $\mathbf{\Gamma}_{\cdot,3}$ and $\mathbf{\Gamma}_{\cdot,4}$ also have only one parent node, we need two additional dimensions, and set $\mathbf{\Gamma}_{\cdot,3} = (1,1,0)$ and $\mathbf{\Gamma}_{\cdot,4} = (1,0,1)$. Next because $\mathbf{\Gamma}_{\cdot,5}$ has two parent nodes $\mathbf{\Gamma}_{\cdot,3}$ and $\mathbf{\Gamma}_{\cdot,4}$, we set $\mathbf{\Gamma}_{\cdot,5} = (1,1,1)$. Lastly since $\mathbf{\Gamma}_{\cdot,6}$ has one parent node $\mathbf{\Gamma}_{\cdot,5}$, we need one more dimension and set $\mathbf{\Gamma}_{\cdot,6} = (1,1,1,1)$. Therefore, in total we have four latent attributes and the reconstruction process is highlighted in blue in Figure II.8. We want to point out that when we recover the latent structures using Algorithm II.2, we choose the smallest K such that the corresponding binary representations of the latent classes satisfy the partial orders. A larger value of K is possible and may not be unique, but here we use the smallest one to make the latent structure concise. Moreover, researchers can also use their domain knowledge to help specify these binary representations.

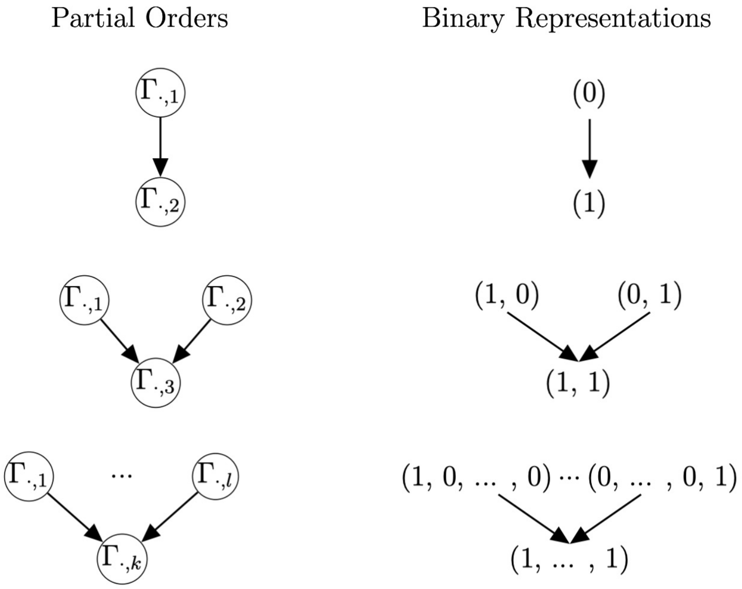


Figure II.7: Examples of binary representations from partial orders.

After we reconstruct binary representations of the latent classes, we can infer the attribute hierarchy accordingly. Specifically, we can get partial orders among latent attributes. For the example in Figure II.8, our reconstructed latent attribute profile matrix $\mathcal{A}$ is shown in Figure II.9, where rows of $\mathcal{A}$ are the binary representations of the latent classes. We can see that $\mathcal{A}_{\cdot,1} \succeq \mathcal{A}_{\cdot,k}$ for all $k \in [K]$, which indicates that the first latent attribute is the most basic one and the prerequisite for all the other latent attributes. Moreover, the fourth attribute is 1 only if all the other attributes are 1, indicating that the fourth attribute is the

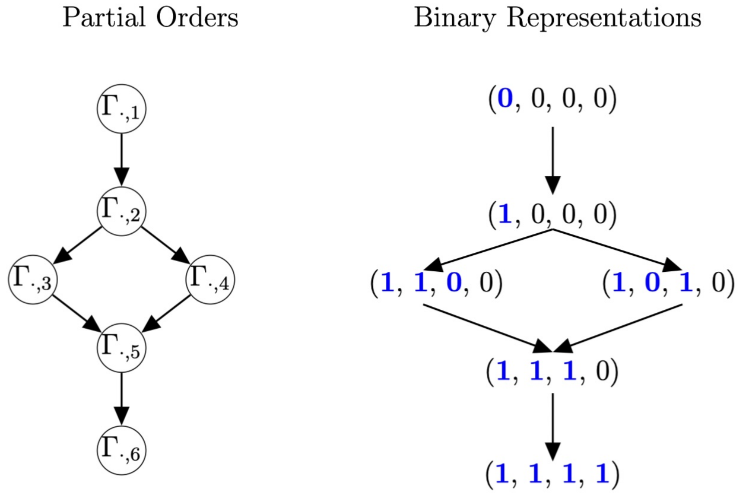


Figure II.8: A more complicated example of binary representations from partial orders.

highest and requires all the other attributes as prerequisites. Formally, we can use $\mathcal{E} = \{k \to l : \text{attribute } k \text{ is a prerequisite for attribute } l\}$ introduced in Section 2.2.1 to denote the prerequisite relationship set, where $k \to l$ if $\mathcal{A}_{\cdot,k} \succeq \mathcal{A}_{\cdot,l}$. For latent attribute profile matrix $\mathcal{A}$ in Figure II.9, we have $\mathcal{E} = \{1 \to 2,\ 1 \to 3,\ 2 \to 4,\ 3 \to 4\}$. We can also plot a DAG according to the prerequisite relationship set $\mathcal{E}$ as shown in the right plot of Figure II.9.

Latent Attribute Profile Matrix

$$\mathcal{A} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Attribute Hierarchy

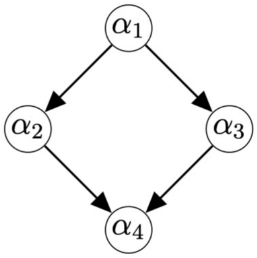


Figure II.9: Latent attribute profile matrix and attribute hierarchy; rows of $\mathcal{A}$ are the binary representations of the selected latent classes in Figure II.8.

Finally, we need to reconstruct the Q-matrix, which can be done by comparing the indicator matrix $\mathbf{\Gamma}$ and the reconstructed latent attribute profile $\mathcal{A}$. Specifically, since capable subjects have the same highest item parameters, for each item, the row in the Q-matrix will equal the smallest latent attribute profile such that the corresponding indicator is 1. To be more formal, let $\boldsymbol{q}_j$ be the jth row of the Q-matrix, we have

$$\boldsymbol{q}_j = \mathcal{A}_{m,\cdot} \text{ such that } \Gamma_{j,m} = 1 \text{ and for any } m' \text{ with } \Gamma_{j,m'} = 1, \mathcal{A}_{m,\cdot} \preceq \mathcal{A}_{m',\cdot},$$

where $\mathcal{A}_{m,\cdot}$ denotes the mth row vector of the latent attribute profile matrix $\mathcal{A}$, i.e., the binary representation of the mth latent class in $\boldsymbol{\Gamma}$. The procedures are summarized in Algorithm II.2.

```
Algorithm II.2: Recover Latent Attribute Profiles, Hierarchical Structure
and Q-matrix
  Input  : Item parameter Matrix $\boldsymbol{\Theta}$
  Step 1 : Construct the indicator matrix $\boldsymbol{\Gamma} = \big(\mathbb{I}\{\theta_{j,m} = \max_{l\in[M]} \theta_{j,l}\}\big)$.
  Step 2 : Construct $\boldsymbol{P}$ based on the partial orders among the columns of $\boldsymbol{\Gamma}$;
           plot a DAG based on $\boldsymbol{P}$.
  Step 3 : Reconstruct binary representations and get latent attribute profile
   set $\mathcal{A}$:
    for node from top to bottom do
        if the node has only one parent node then
            add a dimension in the binary representations
        end
        if the node has more than one parent node then
            set the binary representation to be the union of all of its parent nodes
        end
  end
  Step 4 : Construct prerequisite relationship set $\mathcal{E}$ and thus recover latent
   hierarchy.
  Step 5 : Reconstruct the Q-matrix $\boldsymbol{Q} = \big(\boldsymbol{q}_j\big)_{j=1}^{J}$:

    $\boldsymbol{q}_j = \mathcal{A}_{m,\cdot}$ such that $\Gamma_{j,m} = 1$ and for any $m'$ with $\Gamma_{j,m'} = 1, \mathcal{A}_{m,\cdot} \preceq \mathcal{A}_{m',\cdot}$.

  Output: Latent attribute profile set $\mathcal{A}$, prerequisite relationship set $\mathcal{E}$ and
          the Q-matrix $\boldsymbol{Q}$.
```

2.5 Simulation Studies

In this section, we conducted comprehensive simulation studies under various settings to evaluate the performance of the proposed method.

For the underlying models, we considered three settings. In the first setting, all the items conformed to the DINA model. In the second setting, half of the items were from the DINA model and the others followed the DINO model. In the third setting, we considered the GDINA model as the underlying data-generating model. To satisfy identifiability conditions (Xu and Zhang, 2016; Gu and Xu, 2019b,c), in the DINA setting, the Q-matrix contained two identity sub-matrices and the remaining items were randomly generated. In the DINA + DINO setting, the Q-matrix contained an identity sub-matrix for each type of the model and the remains were randomly generated. For the GDINA setting, we had two identity sub-matrices, and the remaining items were randomly generated and required at most 3 latent attributes.

We considered four hierarchical structures shown in Figure II.1 with $K = 4$. The test length was set to 30 ($J = 30$). For the DINA and DINA + DINO settings, we considered two sample sizes with $N = 500$ or 1000. Two different signal strengths for true item parameters were included: $\{\theta_j^+ = 0.9,\ \theta_j^- = 0.1;\ j \in [J]\}$ and $\{\theta_j^+ = 0.8,\ \theta_j^- = 0.2;\ j \in [J]\}$. For the GDINA setting, we considered two different sample sizes, $N = 1000$ or 2000. The sample sizes considered in the GDINA settings are relatively larger than those for the DINA and DINA + DINO settings since in the GDINA model there are more item parameters to be estimated. As before, we set two different signal strengths, where the highest item parameters were 0.9 or 0.8, and the lowest item parameter was 0.1 or 0.2. The other item parameters in between were equally spaced. For each scenario, we performed 50 independent repetitions. All model parameters were randomly initialized and the implementations were done in Matlab.

To tune hyper-parameters for the proposed method, we used a two-stage training strategy. In the first training stage, we primarily tuned $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ to select significant latent classes, and used a fixed relatively large τ. In the second stage, we did not put penalty on the proportion parameters (i.e. $\tilde{\lambda}_1$ was set to 0), and fine-tuned $\tilde{\lambda}_2$ and τ for the TLP term to further merge identical item parameters. Specifically, in the first stage, the candidates for $\tilde{\lambda}_2$ were set to relatively small and the value of the threshold τ was set to relatively large. In this work, we chose $\tau = 0.3$ and selected $\tilde{\lambda}_1 \in \{0.01, 0.015, \ldots, 0.05\}$ and $\tilde{\lambda}_2 \in \{0.001, 0.005, 0.01, 0.015\}$. The reason to use a small penalty coefficient and a relatively large threshold for the TLP during the first training stage is that we mainly aim to select the correct number of latent classes instead of learning identical item parameters. A small TLP would facilitate the shrinkage of the proportion parameters, while a large TLP would merge the latent classes too fast. After we selected the significant latent classes from the first stage, we next moved to the second stage where we used a larger $\tilde{\lambda}_2$ and a smaller threshold τ for the TLP to further merge identical item parameters. Specifically, the penalty for the proportion parameters $\tilde{\lambda}_1$ was set to 0, and we selected $\log(\tilde{\lambda}_2) \in \{-1, 0, 1, 2, 3\}$ and $\tau \in \{0.03, 0.05, 0.1\}$. For the γ parameter, similarly to Wu et al. (2016), we used a fixed γ for simplicity with $\gamma = 0.02$. If computation allows, we could also tune for γ or adaptively select it in each iteration (Wang and Liao, 2001). The candidate sets for all tuning parameters were the same across the simulation settings. In total there were 480 possible combinations of tuning parameters, while using the two-stage training procedure, the number of combinations was reduced to around 50. On average, the computation time in our simulation study was less than 2.0 seconds per repetition per set of hyper-parameters. We can also try larger candidate sets for these hyper-parameters, but our simulation results below showed that the aforementioned candidate sets were enough to provide good results.

Following Chen et al. (2017b) and Wang and Lu (2021), we also fitted the regularized LCMs under the same settings for comparison. For the regularized LCM method, the number of latent classes and the coefficient for the penalty term need to be selected according to some information criteria. As suggested in Chen et al. (2017b), we used GIC_2 to select these tuning parameters in regularized LCMs. In our simulation, for the number of latent classes, we chose $M \in \{M_0-2, M_0-1, M_0, M_0+1, M_0+2\}$, where M_0 is the true number of latent classes. We also conducted a sensitivity analysis to investigate the impacts of different specifications of the upper bound M on our algorithm. The results show that our method is robust to the choice of different M. The detailed results of the sensitivity analysis are included in Appendix A. For the penalty term, we selected $\lambda \in \{0.01, 0.02, \ldots, 0.1\}$ as in Wang and Lu (2021).

We inspect the results from different aspects. Firstly we examine the accuracy of selecting the number of latent classes $\hat{M}$, which is denoted as $\mathrm{Acc}(\hat{M})$. Based on the learned item parameters, we reconstruct the indicator matrix $\hat{\mathbf{\Gamma}} = \big(\mathbb{I}\{\theta_{j,m} = \max_{l\in[\hat{M}]}\theta_{j,l}\}\big) \in \{0,1\}^{J\times\hat{M}}$ and the corresponding partial order matrix $\hat{\boldsymbol{P}}$. It's worth noting that when we extract the partial orders among the latent classes, single misspecification of the elements in the indicator matrix may lead to different ordering results, making the method of directly estimating the partial orders not robust. Based on this observation, we shall allow for certain tolerance on the estimation errors of the indicator matrix when reconstructing the partial orders. In particular, we relax the construction condition of the partial order such that we regard $\mathbf{\Gamma}_{\cdot,k} \succeq \mathbf{\Gamma}_{\cdot,j}$, if $\Gamma_{j,k} \geq \Gamma_{j,l}$ except for a small proportion t of $j \in [J]$. In our simulation, we used $t = 5\%$ when the noise was small, and $t = 10\%$ when the noise was large. Another issue to note here is that directly comparing two indicator matrices is not straightforward due to the label switching issue. To address this issue, we apply the Hungarian algorithm (Kuhn, 1955) to find the best match of the columns of the estimated indicator matrix and the true indicator matrix, based on which the following comparisons can be made accordingly. We use $\mathrm{Acc}(\hat{\boldsymbol{P}})$ to denote the accuracy of reconstruction of the partial orders. If all the partial orders among the columns of the indicator matrix are correctly recovered, then we will successfully reconstruct the binary latent pattern representations and accordingly the hierarchical structures among the latent attributes. We use Acc($\hat{\mathcal{E}}$) to denote the recovery rate of the hierarchical structure. Here we count it a success only if the entire hierarchical structure is recovered. If the number of latent classes is successfully selected, we also compute the mean squared error of the item parameters MSE($\hat{\mathbf{\Theta}}$). Finally, if the hierarchical structure is correctly recovered, we compute the accuracy of the estimated Q-matrix, denoted by Acc($\hat{\boldsymbol{Q}}$). In summary, we have five evaluation metrics : Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$), Acc($\hat{\mathcal{E}}$), MSE($\hat{\mathbf{\Theta}}$) and Acc($\hat{\boldsymbol{Q}}$).

The simulation results of the DINA, and DINA + DINO settings are presented in Table 2.1 and Table 2.2. The results of the GDINA model are shown in Table 2.3. The simulations show that compared with the regularized LCM approach, our method provided much better results in almost all the settings and from all the evaluation aspects. In many settings, the proposed method could achieve a nearly perfect selection of the number of latent classes, reconstruction of the partial orders and hierarchies, and the estimation of the Q-matrix, especially when the noise was small or there was a sufficiently large data size. Among the four hierarchical structures, the unstructured hierarchy was the most difficult one, especially when the noise was large but the sample size was relatively small. This is expected since under the unstructured hierarchy, there are 9 latent classes, and the hierarchical structure is more complicated compared with the others. However, with increasing sample sizes, the proposed method also provided satisfactory results, while the regularized LCM approach did not. In terms of the underlying data-generating model, the DINA setting was the most difficult one to learn. This is because the DINA models the conjunctive "AND" relationship among the latent attributes, which makes it hard to distinguish the latent classes under hierarchical structures. For example, consider latent classes $\boldsymbol{\alpha} = (1, 0, 0, 0)$ and $\boldsymbol{\alpha}' = (1, 1, 0, 0)$. Under the DINA model, only the items with

Hierarchy	N	r	Method	Acc($\hat{M}$)	Acc($\hat{\boldsymbol{P}}$)	Acc($\hat{\mathcal{E}}$)	MSE($\hat{\boldsymbol{\Theta}}$)	Acc($\hat{\boldsymbol{Q}}$)
Linear	500	0.1	Proposed	1	1	1	0.0004	0.99
			RLCM	0.72	0.71	0.60	0.0006	0.96
		0.2	Proposed	0.68	0.68	0.66	0.0012	0.95
			RLCM	0.42	0.42	0.42	0.0012	0.99
	1000	0.1	Proposed	1	1	1	0.0002	1
			RLCM	0.80	0.80	0.80	0.0013	0.99
		0.2	Proposed	0.96	0.96	0.96	0.0004	0.99
			RLCM	0.54	0.53	0.52	0.0025	0.99
Convergent	500	0.1	Proposed	1	1	0.98	0.0005	0.99
			RLCM	0.62	0.61	0.48	0.0013	0.96
		0.2	Proposed	0.56	0.56	0.50	0.0014	0.93
			RLCM	0.20	0.18	0.08	0.0245	0.93
	1000	0.1	Proposed	1	1	1	0.0002	1
			RLCM	0.48	0.48	0.40	0.0003	0.98
		0.2	Proposed	0.84	0.84	0.84	0.0005	0.98
			RLCM	0.38	0.37	0.34	0.0062	0.98
Divergent	500	0.1	Proposed	1	1	0.97	0.0005	0.97
			RLCM	0.44	0.43	0.28	0.0047	0.93
		0.2	Proposed	0.48	0.47	0.34	0.0016	0.88
			RLCM	0.22	0.20	0.08	0.0194	0.95
	1000	0.1	Proposed	0.98	0.98	0.98	0.0002	1
			RLCM	0.48	0.48	0.44	0.0003	0.97
		0.2	Proposed	0.86	0.86	0.80	0.0006	0.96
			RLCM	0.26	0.25	0.20	0.0108	0.97
Unstructured	500	0.1	Proposed	0.82	0.82	0.66	0.0006	0.93
			RLCM	0.22	0.21	0.10	0.0103	0.90
		0.2	Proposed	0.06	0.06	0.02	0.0031	0.90
			RLCM	0.14	0.13	0.04	0.0126	0.87
	1000	0.1	Proposed	0.92	0.92	0.92	0.0002	0.99
			RLCM	0.36	0.35	0.18	0.0074	0.98
		0.2	Proposed	0.48	0.48	0.48	0.0006	0.94
			RLCM	0.28	0.26	0.14	0.0124	0.93

Table 2.1: DINA Results; Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$) and Acc($\hat{\mathcal{E}}$) are calculated for all the cases; MSE($\hat{\boldsymbol{\Theta}}$) is calculated for the cases when the number of latent classes are correctly selected; Acc($\hat{\boldsymbol{Q}}$) is calculated for the cases when the hierarchical structure is successfully recovered.

the q-vector $\boldsymbol{q}_j = (0, 1, 0, 0)$ or $(1, 1, 0, 0)$ can distinguish these two latent classes. By contrast, under the DINO model, the items with $\boldsymbol{q}_j = (0, 1, *, *)$ where $*$ can be either 0 or 1, will distinguish them. And under the GDINA model, the two latent classes can be differentiated by the items with $\boldsymbol{q}_j = (*, 1, *, *)$. Therefore, if the underlying data-generating model is the DINA model, it requires a larger sample size to achieve good performance. It is also noted that for the Q-matrix estimation, Acc($\hat{\boldsymbol{Q}}$) for both

Hierarchy	N	r	Method	Acc($\hat{M}$)	Acc($\hat{\boldsymbol{P}}$)	Acc($\hat{\mathcal{E}}$)	MSE($\hat{\boldsymbol{\Theta}}$)	Acc($\hat{\boldsymbol{Q}}$)
Linear	500	0.1	Proposed	1	1	1	0.0004	0.99
			RLCM	0.96	0.93	0.68	0.0006	0.96
		0.2	Proposed	0.98	0.98	0.96	0.0010	0.94
			RLCM	0.72	0.72	0.70	0.0013	0.97
	1000	0.1	Proposed	1	1	1	0.0002	1
			RLCM	0.96	0.96	0.94	0.0002	0.98
		0.2	Proposed	1	1	1	0.0004	0.99
			RLCM	0.78	0.78	0.78	0.0004	0.99
Convergent	500	0.1	Proposed	1	1	0.86	0.0004	0.98
			RLCM	0.88	0.84	0.52	0.0012	0.93
		0.2	Proposed	0.96	0.94	0.76	0.0013	0.89
			RLCM	0.60	0.60	0.54	0.0017	0.92
	1000	0.1	Proposed	1	1	1	0.0002	1
			RLCM	0.88	0.87	0.76	0.0003	0.97
		0.2	Proposed	1	0.99	0.82	0.0004	0.99
			RLCM	0.64	0.64	0.58	0.0006	0.98
Divergent	500	0.1	Proposed	0.98	0.98	0.96	0.0005	0.97
			RLCM	0.80	0.77	0.40	0.0009	0.92
		0.2	Proposed	0.86	0.84	0.46	0.0016	0.86
			RLCM	0.40	0.39	0.26	0.0023	0.88
	1000	0.1	Proposed	1	1	1	0.0002	1
			RLCM	0.82	0.81	0.56	0.0003	0.96
		0.2	Proposed	1	0.99	0.78	0.0005	0.97
			RLCM	0.48	0.48	0.40	0.0009	0.95
Unstructured	500	0.1	Proposed	0.92	0.91	0.70	0.0006	0.94
			RLCM	0.54	0.51	0.14	0.0039	0.88
		0.2	Proposed	0.28	0.27	0	0.0010	0.75
			RLCM	0.28	0.27	0.08	0.0112	0.88
	1000	0.1	Proposed	0.98	0.98	0.96	0.0002	1
			RLCM	0.58	0.57	0.34	0.0005	0.94
		0.2	Proposed	0.82	0.81	0.48	0.0007	0.92
			RLCM	0.18	0.17	0.06	0.0066	0.88

Table 2.2: DINA+DINO Results; Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$) and Acc($\hat{\mathcal{E}}$) are calculated for all the cases; MSE($\hat{\boldsymbol{\Theta}}$) is calculated for the cases when the number of latent classes are correctly selected; Acc($\hat{\boldsymbol{Q}}$) is calculated for the cases when the hierarchical structure is successfully recovered.

methods are similar from the tables. However, since we calculate the accuracy of the Q-matrix only if the hierarchical structure is correctly recovered, given the worse performance on hierarchical structure recovery of the regularized LCM method, the proposed method in fact provided a much better overall Q-matrix estimation.

2.6 Real Data Analysis

Hierarchy	N	r	Method	Acc($\hat{M}$)	Acc($\hat{\boldsymbol{P}}$)	Acc($\hat{\mathcal{E}}$)	MSE($\hat{\boldsymbol{\Theta}}$)	Acc($\hat{\boldsymbol{Q}}$)
Linear	1000	0.1	Proposed	0.98	0.98	0.98	0.0005	1
			RLCM	0.76	0.76	0.76	0.0005	0.99
		0.2	Proposed	0.96	0.96	0.96	0.0010	0.99
			RLCM	0.52	0.51	0.48	0.0036	0.97
	2000	0.1	Proposed	0.94	0.94	0.94	0.0003	1
			RLCM	0.92	0.92	0.92	0.0002	1
		0.2	Proposed	0.96	0.96	0.96	0.0005	1
			RLCM	0.62	0.62	0.62	0.0009	1
Convergent	1000	0.1	Proposed	0.98	0.98	0.98	0.0006	1
			RLCM	0.68	0.68	0.66	0.0008	0.97
		0.2	Proposed	0.90	0.90	0.86	0.0013	0.98
			RLCM	0.36	0.35	0.30	0.0161	0.96
	2000	0.1	Proposed	0.98	0.98	0.98	0.0003	1
			RLCM	0.82	0.82	0.80	0.0003	0.99
		0.2	Proposed	1	1	1	0.0005	1
			RLCM	0.38	0.38	0.36	0.0029	0.99
Divergent	1000	0.1	Proposed	0.98	0.98	0.98	0.0006	1
			RLCM	0.84	0.83	0.66	0.0061	0.94
		0.2	Proposed	0.86	0.86	0.82	0.0014	0.96
			RLCM	0.38	0.36	0.26	0.0148	0.89
	2000	0.1	Proposed	1	1	1	0.0003	1
			RLCM	0.76	0.76	0.74	0.0003	0.99
		0.2	Proposed	0.98	0.98	0.92	0.0006	0.99
			RLCM	0.52	0.51	0.48	0.0018	0.97
Unstructured	1000	0.1	Proposed	1	1	0.98	0.0007	0.99
			RLCM	0.62	0.61	0.44	0.0013	0.92
		0.2	Proposed	0.48	0.47	0.36	0.0021	0.89
			RLCM	0.36	0.34	0.20	0.0220	0.88
	2000	0.1	Proposed	1	1	1	0.0003	1
			RLCM	0.66	0.66	0.60	0.0005	0.96
		0.2	Proposed	0.86	0.85	0.78	0.0008	0.99
			RLCM	0.38	0.37	0.24	0.0056	0.94

Table 2.3: GDINA Results; Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$) and Acc($\hat{\mathcal{E}}$) are calculated for all the cases; MSE($\hat{\boldsymbol{\Theta}}$) is calculated for the cases when the number of latent classes are correctly selected; Acc($\hat{\boldsymbol{Q}}$) is calculated for the cases when the hierarchical structure is successfully recovered.

2.6.1 Analysis of ECPE Data

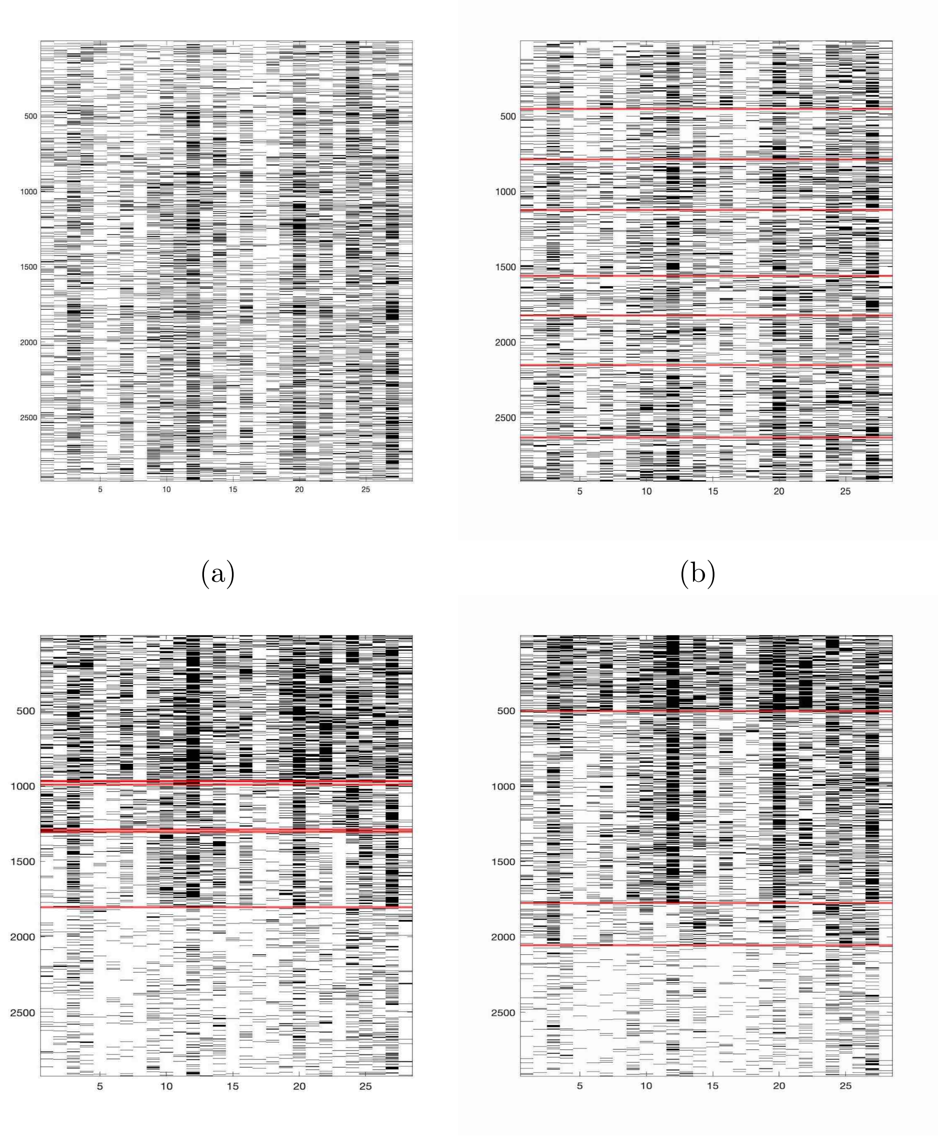
In this section, we apply the proposed approach to the Examination for the Certificate of Proficiency in English (ECPE) data to learn the latent hierarchical structure. The ECPE data was collected by the English Language Institute of the University of Michigan, and there were 2,922 examinees and 28 ECPE items. There were three target attributes including lexical rules, cohesive rules, and morphosyntactic rules.

In the literature on the analysis of the ECPE data, Templin and Bradshaw (2014) fitted an HLAM with the Q-matrix pre-specified by exam designers and tested the presence of the linear hierarchy through bootstrap, which supports the linear hierarchy among the three attributes under the cognitive diagnosis modeling framework. In Wang and Lu (2021), the authors also studied this ECPE data using the latent variable selection approach and regularized LCM approach respectively. In the latent variable selection approach, they used three "anchor" items which formed a known identity sub-matrix in the Q-matrix. The latent variable selection approach selected 5 significant latent classes, and the learned model implied a convergent structure, that is, two latent attributes were prerequisites for the third one. Though estimations of the ECPE data have been widely studied under the cognitive diagnosis setting, von Davier and Haberman (2014) pointed out that ECPE data appeared to have mainly a unidimensional structure, which may not be suitable for cognitive diagnosis modeling.

Our proposed method uses a penalized exploratory latent class analysis approach, which does not depend on the cognitive diagnosis models' settings such as the Q-matrix structure and multi-dimensionality of the attributes. The proposed method does not require any prior information except for an upper bound of the number of latent classes M. Here we took $M = 8$ and used spectral clustering to initialize model parameters. Specifically, given the data matrix $\mathcal{R} \in \{0,1\}^{N\times J}$, we calculated the symmetric normalized Laplacian matrix $L^{\text{norm}} := I - D^{-1/2}\mathcal{R}D^{-1/2}$, where $D = \text{diag}\{\sum_j R_{1j}, \sum_j R_{2j}, \ldots, \sum_j R_{Nj}\}$. Then we took the first M eigenvectors of L and performed k-means clustering on the eigenvectors. Based on the clustering results, we had an initialization of the partition of the subjects to M classes and then used class proportions and mean responses to the items as the model initializations. The clustered data from spectral initialization is shown in Figure II.10b and the final estimation results with spectral initialization are in Figure II.10d.

For a comparison purpose, we also used the pre-specified Q-matrix to fit a GDINA model with 2^K latent classes, and then used the learned GDINA estimation results as initialization, which is shown in Figure II.10c. We found that the GDINA model initialization using the pre-specified Q-matrix resulted in the same learned models as the spectral initialization, which does not require the pre-specified Q-matrix information. It is also noted that by directly fitting a GDINA model with the pre-designed Q-matrix and 2^K latent classes, it learned four latent groups with large proportions and all the other proportion parameters were very small, but not exactly zeros. Comparing Figure II.10c and Figure II.10d, we can also see that the clustered data based on our method shows a much clearer ordered structure among the latent classes. Specifically, using the proposed method, we obtained four significant latent classes, as shown in Figure II.10d. In the plot, each row represents the response vector from a subject and each column represents an item, with dark cells standing for “1”’s and white cells standing for “0”’s. The resulting clusters are separated by red lines. For ease of visualization, we have rearranged the rows of data to form clusters. From the clustered results, there seems to be an ordered structure: the subjects in the first cluster are more likely to give positive responses than those in the second cluster, the second cluster tends to have more positive responses than the third cluster, and the same for the results in the third and the fourth clusters. To better identify the hierarchical structure, we further calculated the indicator matrix. The estimated item parameter matrix $\hat{\mathbf{\Theta}}$ and the reconstructed indicator matrix $\hat{\mathbf{\Gamma}}$ are shown in Figure II.11. It is easy to see $\hat{\mathbf{\Gamma}}_{\cdot,1} \prec \hat{\mathbf{\Gamma}}_{\cdot,2} \prec \hat{\mathbf{\Gamma}}_{\cdot,3} \prec \hat{\mathbf{\Gamma}}_{\cdot,4}$, which indicates a unidimensional located latent class model structure, or in other words, a model structure with strictly ordered latent classes (von Davier and Haberman, 2014). This finding is consistent with the observation in von Davier and Haberman (2014).

To present the latent class structure under the HLAM framework, we can apply the proposed Algorithm II.2. Since there are four latent classes and $\hat{\mathbf{\Gamma}}_{\cdot,1} \prec \hat{\mathbf{\Gamma}}_{\cdot,2} \prec \hat{\mathbf{\Gamma}}_{\cdot,3} \prec \hat{\mathbf{\Gamma}}_{\cdot,4}$, the smallest K will be 3 and the corresponding binary representations of



(a)

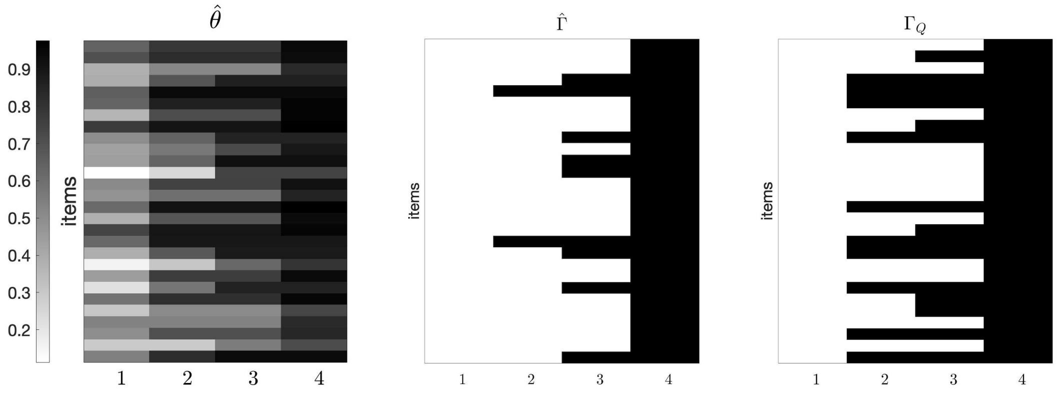
(b)

(c)

(d)

Figure II.10: Clustering Results for ECPE. (a): the original data; (b): clustered data from spectral initialization; (c): clustered data from GDINA initialization with known Q-matrix; (d): clustered data from the proposed method. Note that the rows of the data matrices in (b), (c), and (d) are permuted differently to better show the clustering structures. The black points stand for response value 1, and the white ones stand for response value 0.

the latent classes will be $(0,0,0),(1,0,0),(1,1,0),(1,1,1)$, which is consistent with the analysis in Templin and Bradshaw (2014) under the cognitive diagnosis modeling framework. Moreover, we also fitted GDINA models with three latent attributes and linear hierarchy based on the inferred Q-matrix from our model and the original designed Q-matrix, respectively. The corresponding indicator matrices are shown in Figure II.11b and II.11c. From the fitted GDINA models, we found that BIC for the original designed Q-matrix was 86,117, while BIC for our learned Q-matrix was 86,000, indicating that our learned $\boldsymbol{Q}$ fits the data better in terms of BIC.



(a) (b) (c)

Figure II.11: Recovered Structures of ECPE. (a): estimated $\hat{\boldsymbol{\Theta}}$ matrix; (b): reconstructed indicator matrix $\hat{\boldsymbol{\Gamma}}$; (c): the indicator matrix based on the pre-specified Q-matrix. Black blocks indicate value 1, and white blocks indicate value 0.

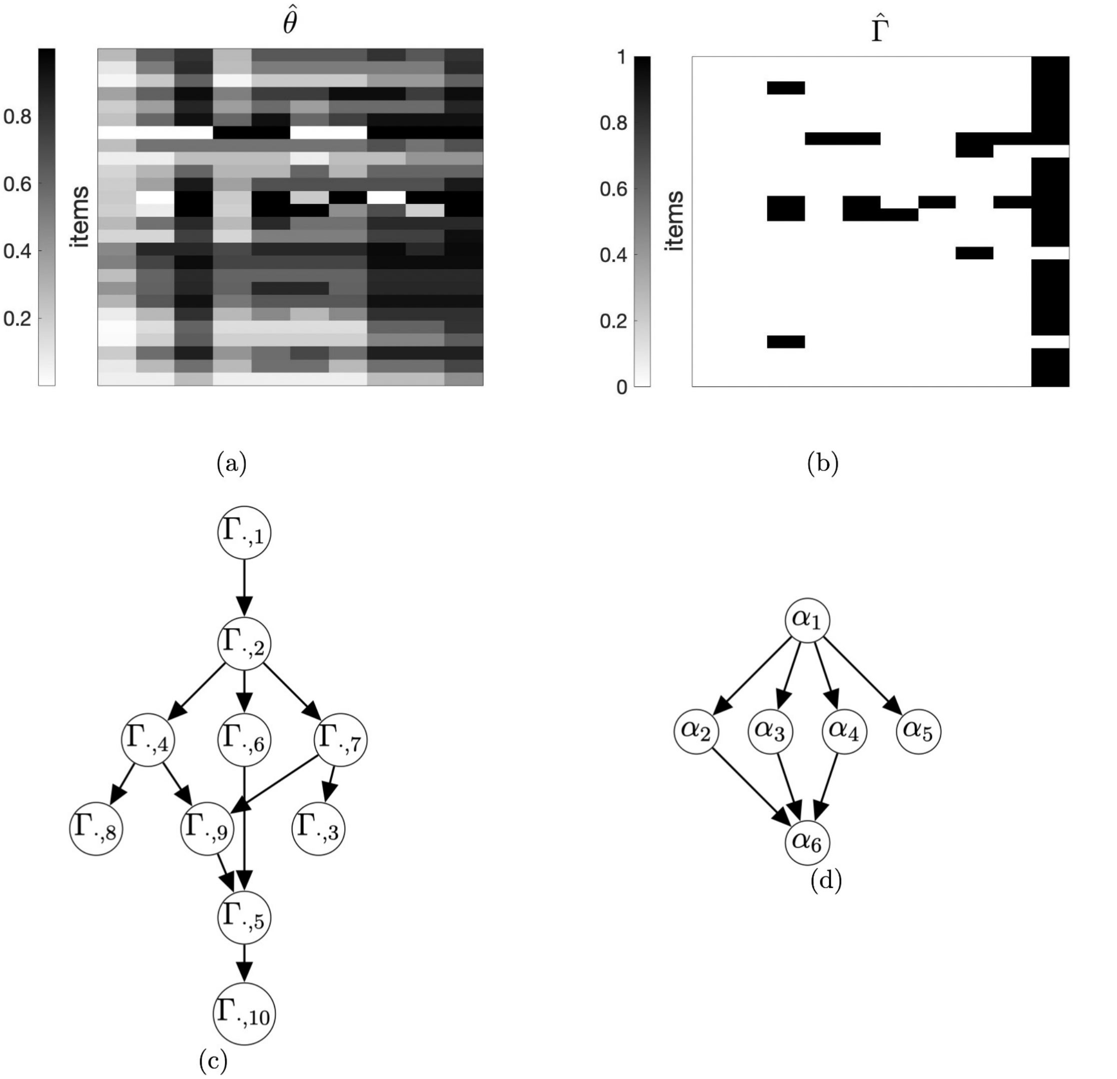
2.6.2 Analysis of PISA Data

To test in a more complex and realistic setting, we also applied the proposed approach to a dataset from Programme for International Student Assessment (PISA), an international reading assessment for 15-year-old students. In particular, we used a PISA 2000 dataset from R package `CDM`, which was previously studied in Chen and de la Torre (2014). This dataset contains $J = 26$ items from six independent articles assessing 1096 examinees' reading abilities. Most of the 26 items are dichotomous items except for some trichotomous items. We converted the trichotomous items to dichotomous by combining all non-zero response values as one category, where we regarded any partial or full credit case as a success and no credit as a failure. In Chen and de la Torre (2014), the authors specified six latent attributes for the PISA 2000 data: (1) locating information; (2) forming a broad general understanding; (3) developing a logical interpretation; (4) evaluating a number-rich text with number sense; (5) evaluating the quality or appropriateness of a text; (6) test speededness.

To apply the proposed method, we set the initial number of latent classes $M = 2^6 = 64$ and initialized the model parameters using the pre-specified Q-matrix in Chen and de la Torre (2014). Specifically, we first fitted a GDINA model using the pre-specified Q-matrix and then used the estimated item and mixture proportion parameters as the initial values for the item parameter matrix $\boldsymbol{\Theta}$ and the proportion parameter vector $\boldsymbol{\pi}$.

After applying the proposed method to the PISA 2000 data, we learned 10 significant latent classes. On average, for each set of hyperparameters, the computation time was 14.87 seconds. The estimated item parameter matrix $\hat{\boldsymbol{\Theta}}$ and the reconstructed indicator matrix $\hat{\boldsymbol{\Gamma}}$ are shown in Figure II.12(a) and Figure II.12(b), respectively. Based on the indicator matrix, we recovered the partial orders of these 10 latent classes in Figure II.12(c), which suggests a multi-dimensional latent structure. With partial orders recovered, we applied Algorithm II.2 and recovered six latent attributes with a hierarchical structure as shown in Figure II.12(d).

The six latent attributes and their hierarchical structures learned from the data may match the prior study in Chen and de la Torre (2014) as follows. The recovered attributes α_1 to α_6 may correspond to "locating information", "forming a broad general understanding", "evaluating a number-rich text with number sense", "evaluating the quality or appropriateness of a text", "Test speededness", and "Developing a logical interpretation", respectively. From the hierarchical structure in Figure II.12(d), α_1 can be viewed as a basic prerequisite for other attributes, which makes sense



(a)

(b)

(c)

(d)

Figure II.12: Recovered Structures of PISA. (a): Estimated $\hat{\mathbf{\Theta}}$ matrix; (b): Reconstructed indicator matrix $\hat{\mathbf{\Gamma}}$. (c): Partial orders; (d): Constructed hierarchical structure of latent attributes.

because examinees need to first understand the item and correctly identify the key information in the article before forming an understanding or evaluating the text, while developing a logical interpretation (α_6) can be interpreted as a more advanced skill.

Besides interpreting the hierarchical structure, we also assessed the model performance using the BIC. Specifically, the BIC of our estimation is 7045, while the BIC of the full GDINA model with the pre-specified Q-matrix is 31495, indicating that the proposed method improves the model fitting in terms of BIC.

2.7 Discussion

In this chapter, we propose a penalized likelihood approach to simultaneously learn the number of latent attributes, the hierarchical structure, the item-attribute Q-matrix, and item-level diagnostic models in HLAMs. We achieve these goals by imposing two regularization terms on an exploratory latent class model: one is a log-type penalty on proportion parameters and the other is a truncated Lasso penalty on the differences among item parameters. The nice form of the penalty terms facilitates the computation and an efficient EM-type algorithm is developed. A latent structure recovery algorithm is also provided based on the learned model parameters. The simulation study and real data analysis demonstrate the good performance of the proposed method.

In most existing works of learning LAMs, the hierarchical structures of latent attributes are either not considered or pre-specified by domain experts. Moreover, related works using exploratory approaches to learn latent hierarchies also require additional pre-specifications such as the number of latent attributes. By contrast, in this work, we develop an exploratory regularized likelihood approach with minimal model specifications. In particular, we estimate the number of latent attributes and recover the hierarchical structure, the Q-matrix, and the item-level diagnostic models simultaneously. The price we have to pay for the minimal model assumptions is that a set of hyperparameters need to be tuned, while our simulation results show that we can achieve it computationally efficiently.

In addition to computational efficiency, our proposed method also has theoretical guarantees. Specifically, we show that the number of latent classes, the model parameters, and the constraint structure of the item parameter matrix can be consistently estimated. Moreover, based on the assumption that more capable subjects have higher item parameters, we develop procedures to recover the number of latent attributes, hierarchical structures, and Q-matrix from the introduced indicator matrix. Due to the consistency of the item parameter matrix and its constraint structure, under the identifiability conditions, the indicator matrix is also consistently estimated, which leads to the consistency of these latent specifications as well. Although the method is purely data-driven, our analysis also provides sound theoretical support. With the theoretical foundation established, our method is consistent and robust in learning the hierarchical structure and other cognitive diagnosis modeling characteristics.

A natural follow-up question would be how we conduct hypothesis testing for the learned hierarchies. Since the existence of hierarchical structures would result in the sparsity structure of the proportion parameter vector, it is equivalent to testing the zero elements in the proportion vector. However, due to the irregularity of the problem since the true parameter now is lying on the boundary of the parameter space, the limiting distribution of the likelihood ratio statistic would be complicated. As noted in the literature (Ma and Xu, 2021), such nonstandard tests need to be further investigated theoretically. We tackle this problem in Chapter IV.

Currently, the proposed model is applied to a static setting where we only have a data set for a fixed time point. It would be also interesting to extend it to the dynamic setting, where multiple measurement data sets for a sequence of time points are available. We can also learn such hierarchical structures by inferring the learning trajectories of the subjects. Moreover, considering the hierarchical structures, we can generate recommendations for learning materials or test items by formulating a sequential decision problem. We leave these interesting directions for future work.

CHAPTER III

Learning Latent Block Structures

3.1 Introduction

With large-scale item pools emerging in modern educational and psychological measurements, it's gaining increasing interest in simultaneously inferring the subgroup structures of both subjects and items, which motivates us to consider co-clustering algorithms. Co-clustering is a data mining technique that allows simultaneous clustering of the rows and columns of a data matrix. We encounter such co-clustering tasks in a wide range of applications. For example, in gene expression studies, the rows of a data matrix represent genes and columns correspond to various environmental conditions or samples such as tissues. Then the data matrix contains gene expression values for the genes under different environmental conditions or of different samples. Simultaneously clustering rows and columns allows us to discover both gene groups and condition similarities (Cheng and Church, 2000; Cho et al., 2004). In collaborative filterings, such as movie rating data, the rows are viewers, columns are movies, and entries in the data are the corresponding ratings for the movies from these viewers. Co-clustering then simultaneously clusters movies into subgroups with similar attractiveness levels and viewers into subgroups of similar viewing patterns (George and Merugu, 2005; Khoshneshin and Street, 2010). Moreover, such data matrices are also studied in the context of graphs, where they are viewed as adjacency matrices of bipartite graphs. In bipartite network modeling, there are two types of nodes, and only nodes of different types can be connected. Co-clustering then can be used to detect communities in these two sets of nodes at the same time (Barber, 2007; Wyse et al., 2017). Despite the wide usage of co-clustering algorithms in many applications, this chapter is mainly motivated by applications in cognitive diagnosis and education assessments that we have introduced previously in Chapter II, while the amounts of both subjects and items can be very large. Specifically, in such contexts, the rows represent students who take the test, the columns represent the test questions, and the matrix contains indicators of whether the students answer the questions correctly. The students are expected to form groups with similar skills and the questions are expected to form groups testing similar aspects (Chen and Li, 2019).

One of the first co-clustering approaches was proposed by Hartigan (1972) and since then, many have been developed. Generally speaking, approaches to co-clustering fall into two classes. In the first class, an objective function measuring discrepancy from an ideal block structure is minimized to produce the clusters of the data matrix, which is often called deterministic. Example works include Doreian et al. (2004), Brusco and Steinley (2006), Brusco and Steinley (2011) and Doreian et al. (2013). The other type is stochastic, which is also referred to as model-based, where the blocks are modeled by a parameterized distribution. Among them, one attractive probabilistic model is the Latent Block Model (LBM, Govaert and Nadif, 2010). In LBMs, each row belongs to a row cluster and each column belongs to a column cluster. Given the row cluster membership and column cluster membership, the entries in the corresponding block are conditionally independent and follow the same distribution. Treating the cluster memberships as latent variables, LBMs are also formulated as mixture models. Many related works based on LBMs have been developed (Govaert and Nadif, 2003, 2005, 2008; Keribin et al., 2012, 2015; Rohe et al., 2012; Wyse and

Friel, 2012). A popular probabilistic model in network analysis called the Stochastic Block Model (SBM, Nowicki and Snijders, 2001) is very similar to LBMs. However, as pointed out in Wyse et al. (2017), the LBMs aim to discover relationships between rows and columns (representing two different sets of objects) while the SBMs focus on modeling interactions within a set of graph nodes.

In this chapter, we mainly focus on the model-based co-clustering methods, specifically the latent block models. The motivations come from two aspects. Firstly, we are interested in learning block inner structures, especially subsets of blocks sharing the same block parameters. This is mostly motivated by cognitive diagnosis applications, where it is often assumed that the item parameters only depend on whether the subject possesses the required attributes by the item. In other words, the subjects with the same attributes have the same item response probabilities. Therefore, there are subsets of blocks sharing the same block parameters, which is however not assumed in classical LBMs. Secondly, when fitting an LBM, one needs to specify both the number of row clusters and the number of column clusters. For clustering, choosing the appropriate number of clusters is of essential importance. Some information criteria such as the Bayesian Information Criterion (BIC, Schwarz et al., 1978) may be used for model selection, and specifically for LBMs, Keribin et al. (2012) has proposed to use the Integrated Completed Likelihood (ICL) criterion to perform the model selection. However in the co-clustering setting, since we need to select both the number of row clusters and the number of column clusters, it is computationally expensive to try all the possible combinations. Therefore, it is desired to develop more efficient algorithms of model selection for LBMs.

In this chapter, motivated by the aforementioned challenges, we propose a two-stage method to achieve the goals of selecting numbers of clusters and learning the inner structure of the blocks. Specifically, similar to the developed method in Chapter II, we employ two regularization terms: log-type penalty on the population row and column proportion parameters to select significant clusters and truncated Lasso penalty on the block parameter pair differences to learn the inner structure. Efficient EM-type algorithms are developed based on the Difference Convex (DC) programming and the Alternating Direction Method of Multipliers (ADMM) method. Simulation studies and a real data application have been conducted to demonstrate the effectiveness of the method.

This chapter proceeds as follows: the model setup of LBMs and the motivations of the proposed method are provided in Section 3.2. The proposed approach and computational algorithms are developed in Section 3.3. Simulation studies are presented in Section 3.4 and an application to a real data set is demonstrated in Section 3.5. Finally, Section 3.6 concludes with some discussions.

3.2 Model Setup and Motivations

In this section, we first give a brief review of latent block models and then provide motivations for our method developed in Section 3.3.

3.2.1 Latent Block Models

In LBMs, the distribution of a data matrix $\boldsymbol{R} = (r_{ij})$ is specified by a latent structure on its rows and columns. Specifically, we assume that each row belongs to a row cluster, and each column belongs to a column cluster. The underlying row clusters and column clusters form the latent block structure of the observed data matrix. Let $\mathcal{I}$ be a set of N rows and $\mathcal{J}$ be a set of M columns. Assume that there are K row clusters and L column clusters, and let $\mathcal{U}$ denote the set of all possible assignments of $\mathcal{I} \times \mathcal{J}$. We use Ξ to denote the parameter of the model. As in classical mixture models, given the block memberships, the entries in $\boldsymbol{R}$ are assumed to be conditionally independent. Then the marginal probability mass function for $\boldsymbol{R}$ is specified as follows:

$$\mathbb{P}(\boldsymbol{R}; \Xi) = \sum_{\boldsymbol{U} \in \mathcal{U}} \mathbb{P}(\boldsymbol{U}; \Xi) \times \mathbb{P}(\boldsymbol{R} \mid \boldsymbol{U}; \Xi). \tag{3.1}$$

Note that the total number of possible assignments of $\mathcal{I} \times \mathcal{J}$ is $K^N \times L^M$, making Eq. (3.1) intractable. Even though this combinatorial challenge is shared by many models with latent structures, the sum here would require a huge computation cost even for a small data set. For example, considering a data matrix with size $N = M = 10$ and $K = L = 2$, there will be 2^{20} unique configurations, which is not computationally feasible. To resolve this issue, we can first restrict the assignments of $\mathcal{I} \times \mathcal{J}$ to be the independent product of assignments of $\mathcal{I}$ and $\mathcal{J}$, and assume local independence as in classical mixture models, which gives the following probability mass function:

$$\mathbb{P}(\boldsymbol{R}; \Xi) = \sum_{(\boldsymbol{Z}, \boldsymbol{W}) \in \mathcal{Z} \times \mathcal{W}} \mathbb{P}(\boldsymbol{Z}; \Xi) \times \mathbb{P}(\boldsymbol{W}; \Xi) \times \mathbb{P}(\boldsymbol{R} \mid \boldsymbol{Z}, \boldsymbol{W}; \Xi), \tag{3.2}$$

where $\boldsymbol{Z} \in \{0,1\}^{N \times K}$ denotes the row cluster assignments of $\mathcal{I}$, $\boldsymbol{W} \in \{0,1\}^{M \times L}$ denotes the column cluster assignments of $\mathcal{J}$, $\mathcal{Z}$ is the set of all possible assignments of rows and $\mathcal{W}$ is the set of all possible assignments of columns. Further, as in mixture models, we assume the row cluster assignment and column cluster assignment follow categorical distributions with proportion parameters $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$ and $\boldsymbol{\rho} = (\rho_1, ..., \rho_L)$, respectively. Moreover, given the memberships, the entries in $\boldsymbol{R}$ are conditionally independent and follow distribution $\psi(\cdot)$ with parameter $\boldsymbol{\Theta} = (\theta_{k,l};\ k = 1, \ldots, K,\ l = 1, \ldots, L)$. Specifically, we have the following probability mass function:

$$\mathbb{P}(\boldsymbol{R}; \Xi) = \sum_{(\boldsymbol{Z}, \boldsymbol{W}) \in \mathcal{Z} \times \mathcal{W}} \Big[\prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} \psi(r_{ij}; \theta_{kl})^{z_{ik} w_{jl}} \Big], \tag{3.3}$$

where $\Xi = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Theta})$ is the model parameter, in which $\boldsymbol{\pi}$ is the row cluster proportion vector, $\boldsymbol{\rho}$ is the column cluster proportion vector and $\boldsymbol{\Theta}$ is the block parameter matrix with $\theta_{k,l}$ being the parameter for the (k, l)th block. The corresponding log-likelihood function is

$$\mathcal{L}(\Xi; \boldsymbol{R}) = \log\Big(\sum_{(\boldsymbol{Z},\boldsymbol{W})\in\mathcal{Z}\times\mathcal{W}} \Big[\prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} \psi(r_{ij}; \theta_{kl})^{z_{ik}w_{jl}} \Big] \Big), \tag{3.4}$$

which is generally intractable due to the complex dependence structure among the rows and columns as we need to sum up all the probability masses over a large set $\mathcal{Z} \times \mathcal{W}$. In the following of this work, we mainly consider a binary data matrix $\boldsymbol{R} \in \{0, 1\}^{N\times M}$, and assume the block distributions are Bernoulli distributions, that is, $\psi(r_{ij}; \theta_{kl}) = \theta_{kl}^{r_{ij}} \cdot (1 - \theta_{kl})^{1-r_{ij}}$.

3.2.2 Motivations

The motivations behind this work come from two aspects: model selection and block inner structure learning.

On the one hand, for clustering applications, it is of essential importance to choose a proper number of clusters, as different choices may lead to different clustering results. In some deterministic methods such as spectral graph clustering (Chen and Li, 2019), the so-called eigengap heuristic (Azran and Ghahramani, 2006; Von Luxburg, 2007) can be used where a number is identified corresponding to the "gap" of the eigenvalues of the graph Laplacian matrix. For probabilistic models such as LBMs, some information criteria can be used. Since the marginal likelihood for an LBM is intractable and defining statistical units in LBMs can be tricky due to the asymptotic of rows and columns, Keribin et al. (2012) proposed to use the Integrated Computed Likelihood criterion (ICL) as the selection criterion. However, selecting the numbers of clusters in a co-clustering setting could be very computationally expensive, since we need to consider both row clusters and column clusters simultaneously. For example, if we consider 10 possible values for the number of row clusters and the number of column clusters respectively, then the total number of possible combinations is 10×10. Moreover, to select the best values of the numbers of clusters, it is essential to compare as many combinations as possible, which is very computationally challenging in practice. Therefore, a more efficient way for cluster selection in co-clustering settings is needed.

On the other hand, in classical LBMs, there are no restrictions on the block parameters. However, the relationships among the blocks are of interest in many applications, especially in the cognitive diagnosis contexts. As we have mentioned in Section 2.2, one important and common assumption in cognitive diagnosis models is that the item parameters only depend on whether a subject possesses the required attributes by the item. In other words, the subjects with the same attributes have the same item response probabilities. For example, in the DINA model, for each item, subjects with all the required attributes have the same item parameter θ_j^+, while subjects missing one of the required attributes share the same item parameter θ_j^-. Recently with large-scale item pools emerging in modern educational and psychological measurements, it gains increasing interest in simultaneously inferring the subgroup structures of both subjects and items (Chen et al., 2017a). In such a setting, we can further assume that items depending on the same set of latent attributes also share the same item parameters, which suggests a latent block structure underlying the response data matrix. More importantly, because of the shared item parameter phenomenon in cognitive diagnosis, we are interested in learning such inner structures of the latent block parameters as well. Below we show an example from the DINA model.

Example III.1 (block structure under the DINA model)**.** *Assume there are three binary latent attributes of interest and a set of J items are designed to make inference on the latent attribute profiles of N subjects. For the subjects, there are 8 possible latent attribute profiles that are in the set $\{0, 1\}^3$, while for the items, there are 7* *possible targeting attribute patterns in* $\{0,1\}^3 \setminus \{(0,0,0)\}$*, since an item should target at least one attribute. Therefore, we have* 7×8 *blocks in total with 7 item clusters and 8 subject clusters. Moreover, assuming that for the items targeting the same subset of latent attributes, the subjects with the same latent attribute profiles share the same item parameters, we have equal block structures as shown in Figure III.1.*

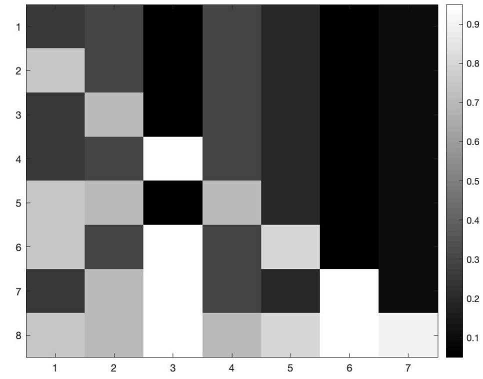


Figure III.1: Block Structure under the DINA Model: rows are for subjects and columns are for items.

In summary, based on the aforementioned motivations, we aim to achieve two goals in this chapter: selecting the numbers of clusters efficiently and learning the inner block structures in co-clustering. Motivated by these two goals, we propose a new method that will be introduced in Section 3.3.

3.3 Proposed Method and Learning Algorithms

In this section, we introduce the proposed method and develop an EM-type learning algorithm.

3.3.1 Proposed Method

Based on the motivations in Section 3.2.2, we propose a two-step procedure to select the numbers of clusters first and learn the block inner structures subsequently.

In specific, similarly to the regularized likelihood approach proposed in chapter II, starting with relatively large numbers of clusters, we first apply log-type penalties (Gu and Xu, 2019b) on both row proportion parameters and column proportion parameters, to select the numbers of clusters for rows and columns. Then truncated Lasso penalties (TLP, Shen et al., 2012) are imposed on the differences among the block parameters to learn the shared parameter structure of latent blocks. Specifically, the objective function for the first step is specified as below:

Step 1.

$$(\hat{\boldsymbol{\pi}}_1, \hat{\boldsymbol{\rho}}_1, \hat{\boldsymbol{\Theta}}_1) = \arg\max_{\boldsymbol{\pi},\boldsymbol{\rho},\boldsymbol{\Theta}} \Big\{\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Theta} \mid \boldsymbol{R}) - \lambda_1 \sum_{k=1}^{K} \log_{[\epsilon_N]} \pi_k - \lambda_2 \sum_{l=1}^{M} \log_{[\delta_M]} \rho_l \Big\}, \quad (3.5)$$

where $\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Theta} \mid \boldsymbol{R})$ is the log-likelihood function and $\log_{[\cdot]}(\cdot)$ is a log-type penalty. Specifically, $\log_{[\epsilon_N]} \pi_k = \log \pi_k \cdot \mathbb{I}\big(\pi_k > \epsilon_N\big) + \log \epsilon_N \cdot \mathbb{I}\big(\pi_k \leq \epsilon_N\big)$ and $\log_{[\delta_M]} \rho_l = \log \rho_l \cdot \mathbb{I}\big(\rho_l > \delta_M\big) + \log \delta_M \cdot \mathbb{I}\big(\rho_l \leq \delta_M\big)$, are log-type penalties on the proportion parameters, where ϵ_N and δ_M are small thresholds to circumvent the singularity of the log function at zero. Similarly to Gu and Xu (2019b), we take ϵ_N and δ_M to be small values, such as N^{-d} and M^{-d} for some $d \geq 1$. The purpose of log-type penalties on the proportion parameters is to push small values to be zeros, and thus achieve the goal of selecting the numbers of row and column clusters. As in Chapter II, we will later show that incorporating log-type penalties in the EM-type algorithm only requires minor changes in the estimates of proportions.

After selecting the numbers of clusters for rows and columns, we then move forward to learn the inner structures of the latent blocks. Specifically, we would like to merge the blocks with equal underlying block parameters. To achieve this goal, we apply the truncated Lasso penalty to the differences for each pair of block parameters. Therefore, we perform the following optimization in Step 2:

Step 2.

$$(\hat{\boldsymbol{\pi}}_2, \hat{\boldsymbol{\rho}}_2, \hat{\boldsymbol{\Theta}}_2) = \arg\max_{\boldsymbol{\pi},\boldsymbol{\rho},\boldsymbol{\Theta}} \Big\{ \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Theta} \mid \boldsymbol{R}) - \lambda_{\mathrm{TLP}} \sum_{(k,l)\neq(k',l'),} \mathrm{TLP}\big(|\theta_{kl} - \theta_{k'l'}|; \tau\big) \Big\}, \quad (3.6)$$

where $\mathrm{TLP}(x;\tau) = \min(|x|, \tau)$ and λ_{TLP} is a nonnegative coefficient for the TLP. It is noted that the TLP only penalizes small values which are smaller than the threshold τ and does not put further penalties on larger differences, which corrects the bias of Lasso estimates. Other penalties such as the SCAD (Fan and Li, 2001) is also applicable here. As pointed out in Chapter II, there are additional advantages of using TLP, including sound theoretical guarantees (Shen et al., 2012) and computational efficiency. Similar to Section 2.4, we also use Difference Convex (DC) programming (Tuy, 1995) to perform the optimization due to the fact that the TLP can be decomposed into a difference of two convex functions.

3.3.2 Learning Algorithms

3.3.2.1 EM Algorithm for Step 1

For latent variable model estimation, we usually rely on Expectation-Maximization (EM) type algorithms. Specifically, in the EM algorithm, we consider the following complete data log-likelihood:

$$\begin{aligned} \mathcal{L}_C(\Xi \mid \boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{W}) = &\sum_{i,k} z_{ik} \log \pi_k + \sum_{j,l} w_{jl} \log \rho_l \\ &+ \sum_{i,j,k,l} z_{ik} w_{jl} \big[r_{ij} \log \theta_{kl} + (1 - r_{ij}) \log(1 - \theta_{kl}) \big]. \end{aligned}$$

The EM algorithm iteratively applies two steps: E-step and M-step.

E-step: in the $(c+1)$th iteration, from the current estimate $\Xi^{(c)}$, we calculate the expected complete log-likelihood with respect to the conditional distribution of

$(\boldsymbol{Z}, \boldsymbol{W})$ given observation $\boldsymbol{R}$:

$$\begin{aligned}\mathcal{Q}(\Xi \mid \Xi^{(c)}) &:= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{W}\mid\boldsymbol{R};\Xi^{(c)}}\big[\mathcal{L}_C(\Xi \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W})\big] \\ &= \sum_{i,k} s_{ik}^{(c)} \log \pi_k + \sum_{j,l} t_{j,l}^{(c)} \log \rho_l \\ &\quad + \sum_{i,j,k,l} e_{ijkl}^{(c)} \big[r_{ij} \log \theta_{kl} + (1 - r_{ij}) \log(1 - \theta_{kl})\big], \end{aligned} \tag{3.7}$$

where $s_{ik}^{(c)} = \mathbb{E}_{\boldsymbol{Z},\boldsymbol{W}\mid\boldsymbol{R};\Xi^{(c)}}[z_{i,k}]$, $t_{ik}^{(c)} = \mathbb{E}_{\boldsymbol{Z},\boldsymbol{W}\mid\boldsymbol{R};\Xi^{(c)}}[w_{j,k}]$ and $e_{ijkl}^{(c)} = \mathbb{E}_{\boldsymbol{Z},\boldsymbol{W}\mid\boldsymbol{R};\Xi^{(c)}}[z_{i,k} w_{j,l}]$.

M-step: find the parameters that maximize $\mathcal{Q}(\Xi \mid \Xi^{(c)})$:

$$\Xi^{(c+1)} = \arg\max_{\Xi}\ \mathcal{Q}(\Xi \mid \Xi^{(c)}). \tag{3.8}$$

In LBMs, the computation of the expectation in the E-step requires the posterior distribution of the latent variables $\boldsymbol{Z}$ and $\boldsymbol{W}$ given the observed data $\boldsymbol{R}$. However, this conditional distribution is hard to compute as the marginal probability mass function in Eq (3.3) for an LBM involves a summation over a large set $\mathcal{Z} \times \mathcal{W}$, which makes it intractable. Following Govaert and Nadif (2008), we consider the mean field approximation where $\boldsymbol{z}_i$ and $\boldsymbol{w}_j$ given $\boldsymbol{R}$ are assumed to be independent. Moreover, using Neal and Hinton's fuzzy criterion (Neal and Hinton, 1998), an alternative view of the EM algorithm, we consider the new objective function denoted as $\mathcal{G}$:

$$\mathcal{G}(\boldsymbol{s}, \boldsymbol{t}, \Xi) = \mathcal{L}_C(\boldsymbol{s}, \boldsymbol{t}, \Xi) + \mathcal{H}(\boldsymbol{s}) + \mathcal{H}(\boldsymbol{t}), \tag{3.9}$$

where $\boldsymbol{s} := (s_{ik}; i = 1, \ldots, N, k = 1, \ldots, K)$ with $s_{ik} = \mathbb{P}(z_{ik} = 1)$, $\boldsymbol{t} := (t_{jl}; j = 1, \ldots, M, l = 1, \ldots, L)$ with $t_{jl} = \mathbb{P}(w_{jl} = 1)$, and $\mathcal{H}(\boldsymbol{s}) = -\sum_{ik} s_{ik} \log s_{ik}$, the entropy function. To optimize the new objective function (3.9), we use the Block EM algorithm developed in Govaert and Nadif (2008) where the rows' and columns' parameters are updated alternately. Specifically, in the $(c+1)$th iteration, we perform the following alternate updates:

1. update row parameters $\boldsymbol{s}^{(c+1)}$, $\boldsymbol{\pi}^{(c+1)}$ and block parameter $\boldsymbol{\Theta}^{(c+1/2)}$ using the EM algorithm, given that column parameters $\boldsymbol{t}^{(c)}$ and $\boldsymbol{\rho}^{(c)}$ are fixed;

2. update column parameters $\boldsymbol{t}^{(c+1)}$, $\boldsymbol{\rho}^{(c+1)}$ and block parameter $\boldsymbol{\Theta}^{(c+1)}$ using the EM algorithm, given that row parameters $\boldsymbol{s}^{(c+1)}$ and $\boldsymbol{\pi}^{(c+1)}$ are fixed.

With column parameters $\boldsymbol{t}^{(c)}$ and $\boldsymbol{\rho}^{(c)}$ fixed, optimizing function (3.9) results in the following updates:

$$\begin{aligned}
s_{ik}^{(c+1)} &= \frac{\pi_k^{(c)} \prod_l (\theta_{kl}^{(c)})^{u_{il}} (1-\theta_{kl}^{(c)})^{m_l - u_{il}}}{\sum_{k'} \pi_{k'}^{(c)} \prod_l (\theta_{k'l}^{(c)})^{u_{il}} (1-\theta_{k'l}^{(c)})^{m_l - u_{il}}}, \\
\rho_l^{(c+1)} &= \frac{\sum_j t_{jl}^{(c)}}{M}, \\
\theta_{kl}^{(c+1)} &= \frac{\sum_{i,j} s_{ik}^{(c+1)} t_{jl}^{(c)} r_{ij}}{\sum_{i,j} s_{ik}^{(c+1)} t_{jl}^{(c)}},
\end{aligned}$$

where $u_{il} = \sum_j t_{jl}^{(c)} r_{ij}$, $m_l = \sum_j t_{jl}^{(c)}$. Similarly, when the row parameters $\boldsymbol{s}^{(c+1)}$ and $\boldsymbol{\pi}^{(c+1)}$ are fixed, we obtain updates for the column parameters and the block parameters as follows:

$$\begin{aligned}
t_{jl}^{(c+1)} &= \frac{\rho_l^{(c+1)} \prod_k (\theta_{kl}^{(c+1/2)})^{v_{kj}} (1-\theta_{kl}^{(c+1/2)})^{n_k - v_{kj}}}{\sum_{l'} \rho_{l'}^{(c+1)} \prod_k (\theta_{kl'}^{(c+1/2)})^{v_{kj}} (1-\theta_{kl'}^{(c+1/2)})^{n_k - v_{kj}}}, \\
\pi_k^{(c+1)} &= \frac{\sum_i s_{ik}^{(c+1)}}{N}, \\
\theta_{kl}^{(c+1)} &= \frac{\sum_{i,j} s_{ik}^{(c+1)} t_{jl}^{(c+1)} r_{ij}}{\sum_{i,j} s_{ik}^{(c+1)} t_{jl}^{(c+1)}},
\end{aligned}$$

where $v_{kj} = \sum_i s_{ik}^{(c+1)} r_{ij}$, and $n_k = \sum_i s_{ik}^{(c+1)}$. In the following, $\boldsymbol{u} := (u_{il})$ where $u_{il} = \sum_i t_{jl} r_{ij}$, $\boldsymbol{m} := (m_l)$ where $m_l = \sum_j t_{jl}$, $\boldsymbol{v} := (v_{kj})$ where $v_{kj} = \sum_i s_{ik} r_{ij}$ and $\boldsymbol{n} := (n_k)$ where $n_k = \sum_i s_{ik}$. Note that $\boldsymbol{u} = \boldsymbol{R} \cdot \boldsymbol{t}^\top$, $\boldsymbol{m} = \boldsymbol{t}^\top \cdot \mathbf{1}_M$, where $\mathbf{1}_M$ is a column vector of length M with all elements being 1; $\boldsymbol{v} = \boldsymbol{s}^\top \cdot \boldsymbol{R}$, and $\boldsymbol{n} = \boldsymbol{s} \cdot \mathbf{1}_N$, where $\mathbf{1}_N$ is a column vector of length N with all elements being 1.

In Step 1 of our method, we apply the log-type penalties on the proportion parameters, and thus consider the following objective function:

$$\min -\mathcal{G}(\boldsymbol{s}, \boldsymbol{t}, \Xi)/NM + \tilde{\lambda}_1/M \sum_{k=1}^{K} \log_{[\epsilon_N]} \pi_k + \tilde{\lambda}_2/N \sum_{l=1}^{L} \log_{[\delta_M]} \rho_l, \tag{3.10}$$

where $\tilde{\lambda}_1 = \lambda_1/N$ and $\tilde{\lambda}_2 = \lambda_2/M$. We follow the same estimation procedures as in the block EM algorithm for LBMs. The log-type penalty here in fact provides us with computational convenience, in that only minor modifications for estimating $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ in the block EM algorithm are needed:

$$\pi_k = \frac{\sum_{i=1}^{N} s_{ik}/N - \tilde{\lambda}_1}{1 - K\tilde{\lambda}_1}, \quad \rho_l = \frac{\sum_{j=1}^{M} t_{jl}/M - \tilde{\lambda}_2}{1 - L\tilde{\lambda}_2}. \tag{3.11}$$

The corresponding algorithm is summarized in Algorithm III.1.

3.3.2.2 EM Algorithm for Step 2

After we perform the estimation for Step 1, we have selected row and column clusters. In Step 2, we apply the truncated Lasso penalties on the block parameters to learn the inner structure of them. Specifically, we consider the criterion (3.9) plus the truncated Lasso penalties as the new objective function:

$$\min \ -\mathcal{G}(\boldsymbol{s}, \boldsymbol{t}, \Xi) + \tilde{\lambda}_{\text{TLP}} \sum_{(k,l)\neq(k',l')} \text{TLP}(|\theta_{kl} - \theta_{k'l'}|; \tau). \tag{3.12}$$

Similar to the block EM algorithm for LBMs, we also develop an alternated optimization algorithm that updates the row and column parameters alternately. As we mentioned in Section 3.3.1, the truncated Lasso penalty can be decomposed into a difference of two convex functions, which allows us to utilize the DC programming (Tuy, 1995) to perform the optimization. Moreover, we also exploit the Alternating Direction Method of Multipliers (ADMM, Boyd et al., 2011) method to facilitate

Algorithm III.1: PBEM-1: Penalized Block EM for Step 1

```
Data: Binary data matrix $\boldsymbol{R} = (r_{i,j})_{N\times M}$.
Set hyperparameters $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$.
Set upper bounds for the numbers of row clusters $K$ and column clusters $L$.
Initialize parameters $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\rho}^{(0)}$, $\boldsymbol{\Theta}^{(0)}$ and posterior expectations $\boldsymbol{s}^{(0)}$ and $\boldsymbol{t}^{(0)}$.
while not converged do
    In the $(c+1)th$ iteration,
    1. update row parameters:
     $\boldsymbol{u} = \boldsymbol{R}\cdot(\boldsymbol{t}^{(c)})^\top,\ \ \boldsymbol{m} = (\boldsymbol{t}^{(c)})^\top\cdot\mathbf{1}_M$.
     for $(i,k)\in[N]\times[K]$ do
         $s_{ik}^{(c+1)} = \frac{\pi_k^{(c)}\prod_l(\theta_{kl}^{(c)})^{u_{il}}(1-\theta_{kl}^{(c)})^{m_l-u_{il}}}{\sum_{k'}\pi_{k'}^{(c)}\prod_l(\theta_{k'l}^{(c)})^{u_{il}}(1-\theta_{k'l}^{(c)})^{m_l-u_{il}}}$.
     end
     $\boldsymbol{\pi}^{(c+1)} = \big[(\boldsymbol{s}^{(c+1)})^\top\cdot\mathbf{1}_N/N - \tilde{\lambda}_1\mathbf{1}_N\big]/(1-K\tilde{\lambda}_1)$.
    2. update block parameters:
     for $(k,l)\in[K]\times[L]$ do
         $\theta_{kl}^{(c+1/2)} = \sum_{i,j}s_{ik}^{(c+1)}t_{jl}^{(c)}r_{ij}\big/\sum_{i,j}s_{ik}^{(c+1)}t_{jl}^{(c)}$.
     end
    3. update column parameters:
     $\boldsymbol{v} = (\boldsymbol{s}^{(c+1)})^\top\cdot\boldsymbol{R},\ \ \boldsymbol{n} = \boldsymbol{s}^{(c+1)}\cdot\mathbf{1}_N$.
     for $(j,l)\in[M]\times[L]$ do
         $t_{jl}^{(c+1)} = \frac{\rho_l^{(c)}\prod_k(\theta_{kl}^{(c+1/2)})^{v_{kj}}(1-\theta_{kl}^{(c+1/2)})^{n_k-v_{kj}}}{\sum_{l'}\rho_{l'}^{(c)}\prod_k(\theta_{kl'}^{(c+1/2)})^{v_{kj}}(1-\theta_{kl'}^{(c+1/2)})^{n_k-v_{kj}}}$.
     end
     $\boldsymbol{\rho}^{(c+1)} = \big[(\boldsymbol{t}^{(c+1)})^\top\cdot\mathbf{1}_M/M - \tilde{\lambda}_2\mathbf{1}_M\big]/(1-K\tilde{\lambda}_2)$.
    4. update block parameters:
     for $(k,l)\in[K]\times[L]$ do
         $\theta_{kl}^{(c+1)} = \sum_{i,j}s_{ik}^{(c+1)}t_{jl}^{(c+1)}r_{ij}\big/\sum_{i,j}s_{ik}^{(c+1)}t_{jl}^{(c+1)}$.
     end
end
Output: $\{\hat{\boldsymbol{\pi}}_1,\ \hat{\boldsymbol{\rho}}_1,\ \hat{\boldsymbol{\Theta}}_1,\ \hat{\boldsymbol{s}}_1,\ \hat{\boldsymbol{t}}_1\}$
```

solving the problem.

In the following we derive the updates for the row parameters $\boldsymbol{s}$, $\boldsymbol{\pi}$ and block parameter $\boldsymbol{\Theta}$ with fixed column parameters $\boldsymbol{t}$ and $\boldsymbol{\rho}$. The updates for the column parameters can be obtained similarly. Specifically, we consider the following objective function:

$$\begin{aligned}
&\min -\mathcal{G}(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta} \mid \boldsymbol{t},\boldsymbol{\rho}) + \lambda_{\mathrm{TLP}} \sum_{(k,l)\neq(k',l')} \mathrm{TLP}(|\theta_{kl}-\theta_{k'l'}|;\tau) \\
&= -\mathcal{L}_C(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta} \mid \boldsymbol{t},\boldsymbol{\rho}) - \mathcal{H}(\boldsymbol{s}) + \lambda_{\mathrm{TLP}} \sum_{(k,l)\neq(k',l')} \mathrm{TLP}(|\theta_{kl}-\theta_{k'l'}|;\tau) \\
&= -\sum_{i,k} s_{ik}\log\pi_k - \sum_{i,j,k,l} s_{ik}t_{jl}\big[r_{ij}\log\theta_{kl} + (1-r_{ij})\log(1-\theta_{kl})\big] - \sum_{ik} s_{ik}\log s_{ik} \\
&\quad + \lambda_{\mathrm{TLP}} \sum_{(k,l)\neq(k',l')} \mathrm{TLP}(|d_{kk'll'}|;\tau),
\end{aligned}$$

such that $d_{kk'll'} = \theta_{kl} - \theta_{k'l'}$, for $1 \le k,k' \le K$ and $1 \le l,l' \le L$. We use $\boldsymbol{d} := (d_{kk'll'}; 1 \le k,k' \le K, 1 \le l,l' \le L)$ to denote the block parameter pair difference tensor. In the following, we denote the new objective function in Step 2 as $\mathcal{G}_{\mathrm{TLP}}(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d} \mid \boldsymbol{t},\boldsymbol{\rho})$ and we decompose it into a difference of two convex functions:

$$\mathcal{G}_{\mathrm{TLP}}(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d} \mid \boldsymbol{t},\boldsymbol{\rho}) = \mathcal{G}_1(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d} \mid \boldsymbol{t},\boldsymbol{\rho}) - \mathcal{G}_2(\boldsymbol{d}),$$

where

$$\begin{aligned}
\mathcal{G}_1(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d} \mid \boldsymbol{t},\boldsymbol{\rho}) &= -\mathcal{L}_C\big(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta} \mid \boldsymbol{t},\boldsymbol{\rho}\big) + \lambda_{\mathrm{TLP}} \sum_{(k,l)\neq(k',l')} |d_{kk'll'}|, \\
\mathcal{G}_2(\boldsymbol{d}) &= \lambda_{\mathrm{TLP}} \sum_{(k,l)\neq(k',l')} (|d_{kk'll'}| - \tau)_+.
\end{aligned}$$

Then we construct a sequence of upper approximation of $\mathcal{G}_{\mathrm{TLP}}(\boldsymbol{s},\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d} \mid \boldsymbol{t},\boldsymbol{\rho})$

iteratively by replacing $\mathcal{G}_2(\boldsymbol{d})$ at iteration $c+1$ with its piecewise affine minorization:

$$\mathcal{G}_2^{(c)}(\boldsymbol{d}) = \mathcal{G}_2(\hat{\boldsymbol{d}}^{(c)}) + \lambda_{\text{TLP}} \sum_{(k,l)\neq(k',l')} \big(|d_{kk'll'}| - |\hat{d}_{kk'll'}^{(c)}|\big) \cdot \mathbb{I}\big(|\hat{d}_{kk'll'}^{(c)}| \geq \tau\big),$$

at the current estimate $\hat{\boldsymbol{d}}^{(c)}$, which leads to an upper convex approximation:

$$\begin{aligned}&\mathcal{G}_{\text{TLP}}^{(c+1)}(\boldsymbol{s}, \boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d} \mid \boldsymbol{t}, \boldsymbol{\rho})\\ =&-\mathcal{L}_C\big(\boldsymbol{s}, \boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{t}, \boldsymbol{\rho}\big) + \lambda_{\text{TLP}} \sum_{(k,l)\neq(k',l')} \big(|d_{kk'll'}|\big) \cdot \mathbb{I}\big(|\hat{d}_{kk'll'}^{(c)}| < \tau\big)\\ &+ \lambda_{\text{TLP}} \sum_{(k,l)\neq(k',l')} \tau \cdot \mathbb{I}\big(|\hat{d}_{kk'll'}^{(c)}| \geq \tau\big).\end{aligned}$$

Now we apply the ADMM to the above objective. Specifically, at iteration $c+1$, the augmented Lagrangian is

$$\begin{aligned}&\mathcal{L}_p(\boldsymbol{s}, \boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d} \mid \boldsymbol{t}, \boldsymbol{\rho})\\ =&-\mathcal{L}_C\big(\boldsymbol{s}, \boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{t}, \boldsymbol{\rho}\big) + \lambda_{\text{TLP}} \sum_{(k,l)\neq(k',l')} \big(|d_{kk'll'}|\big) \cdot \mathbb{I}\big(|\hat{d}_{kk'll'}^{(c)}| < \tau\big)\\ &+ \lambda_{\text{TLP}} \sum_{(k,l)\neq(k',l')} \tau \cdot \mathbb{I}\big(|\hat{d}_{kk'll'}^{(c)}| \geq \tau\big) + y' \sum_{(k,l)\neq(k',l')} \big(d_{kk'll'} - (\theta_{kl} - \theta_{k'l'})\big)\\ &+ \frac{\gamma}{2} \sum_{(k,l)\neq(k',l')} \big(d_{kk'll'} - (\theta_{kl} - \theta_{k'l'})\big)^2,\end{aligned}$$

where y is the dual variable and γ is a nonnegative penalty parameter. Using the scaled Lagrangian multiplier $\mu = y/\gamma$, we update the parameters as follows:

$$
\begin{aligned}
\hat{\pi}_k^{(c+1)} &= \sum_{i=1}^{N} s_{ik}^{(c+1)}/N; \\
\hat{\theta}_{kl}^{(c+1)} &= \underset{a}{\operatorname{argmin}} \Big\{ - \big(\sum_{i=1}^{N} \sum_{j=1}^{M} s_{ik}^{(c+1)} t_{jl}^{(c+1)} r_{ij} \big) \log a \\
&\qquad - \big(\sum_{i=1}^{N} \sum_{j=1}^{M} s_{ik}^{(c+1)} t_{jl}^{(c+1)} (1 - r_{ij}) \big) \log(1-a) \\
&\qquad + \frac{\gamma}{2} \sum_{(k,l) \neq (k',l')} \big(\hat{d}_{kk'll'}^{(c)} - (a - \hat{\theta}_{k'l'}^{(c)}) + \hat{\mu}_{kk'll'}^{(c)} \big)^2 \\
&\qquad + \frac{\gamma}{2} \sum_{(k,l) \neq (k',l')} \big(\hat{d}_{kk'll'}^{(c)} - (a - \hat{\theta}_{k'l'}^{(c+1)}) + \hat{\mu}_{kk'll'}^{(c)} \big) \Big\}; \qquad (3.13) \\
\hat{d}_{kk'll'}^{(c+1)} &= \begin{cases} \hat{\theta}_{kl}^{(c+1)} - \hat{\theta}_{k'l'}^{(c+1)} - \hat{\mu}_{kk'll'}^{(c)}, & \text{if } |\hat{d}_{kk'll'}^{(c)}| \geq \tau \\ \mathrm{ST}\big(\hat{\theta}_{kl}^{(c+1)} - \hat{\theta}_{k'l'}^{(c+1)} - \hat{\mu}_{kk'll'}^{(c)}; \lambda_{\mathrm{TLP}}/\gamma\big), & \text{if } |\hat{d}_{kk'll'}^{(c)}| < \tau, \end{cases}; \\
&\qquad \text{where } \mathrm{ST}(x;\gamma) = (||x||_2 - \gamma)_+/||x||_2. \\
\hat{\mu}_{kk'll'}^{(c+1)} &= \hat{\mu}_{kk'll'}^{(c)} + \hat{d}_{kk'll'}^{(c+1)} - \big(\hat{\theta}_{kl}^{(c)} - \hat{\theta}_{k'l'}^{(c)}\big).
\end{aligned}
$$

We summarize the above updates in Algorithm III.2 and denote the function on the right hand side of Eq. (3.13) as $\mathcal{A}_{kl}(a \mid \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{\Theta}, \boldsymbol{d}, \boldsymbol{\mu})$.

Remark III.1. *The algorithms we have developed in this section can also be considered as variational EM algorithms. In variational inference, we optimize the evidence lower bound ELBO(q)* $:= \mathbb{E}_q\big[\log p(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{R}; \Xi)\big] - \mathbb{E}_q\big[\log q(\boldsymbol{Z}, \boldsymbol{W})\big]$*, where* q *is the variational distribution over the latent variables* $(\boldsymbol{Z}, \boldsymbol{W})$*. Here we consider the mean field approximation where* $\boldsymbol{z}_i$ *and* $\boldsymbol{w}_j$ *given* $\boldsymbol{R}$ *are independent, that is,* $q(\boldsymbol{Z}, \boldsymbol{W}) = \prod_i q_i(\boldsymbol{z}_i) \times \prod_j q_j(\boldsymbol{w}_j)$*. Then* $q_i(\boldsymbol{z}_i) = \prod_{k=1}^{K} s_{ik}^{z_{ik}} \cdot (1 - s_{ik})^{1 - z_{ik}}$*,* $q_j(\boldsymbol{w}_j) = \prod_{l=1}^{L} t_{jl}^{w_{jl}} \cdot (1 - t_{jl})^{1 - w_{jl}}$*, where* $s_{ik} := \mathbb{P}(z_{ik} = 1)$ *and* $t_{jl} = \mathbb{P}(w_{jl} = 1)$*, and ELBO(q)* $= \mathcal{L}_C(\boldsymbol{\Xi} \mid \boldsymbol{R}, \boldsymbol{S}, \boldsymbol{T}) - \mathbb{E}_q\big[\log q(\boldsymbol{Z}, \boldsymbol{W})\big]$*. In the variational EM algorithm, it is shown that the variational distributions* $q_i(\boldsymbol{z_i})$ *and* $q_j(\boldsymbol{w_j})$ *take the following forms (Blei et al., 2017):* $q_i(\boldsymbol{z_i}) \propto \exp\big\{\mathbb{E}_{-\boldsymbol{z}_i}\big[\log p(\boldsymbol{z}_i \mid \boldsymbol{Z}_{-\boldsymbol{z}_i}, \boldsymbol{W}, \boldsymbol{R})\big]\big\}$ *and*

Algorithm III.2: PBEM-2: Penalized Block EM with TLP for Step 2

```
Data: Binary data matrix $\boldsymbol{R} = (r_{i,j})_{N\times M}$.
Set hyperparameters $\lambda_{\text{TLP}}$, $\tau$ and $\gamma$.
Initialize parameters using the estimates from Step 1 $\big(\hat{\boldsymbol{\pi}}_1,\ \hat{\boldsymbol{\rho}}_1,\ \hat{\boldsymbol{\Theta}}_1,\ \hat{\boldsymbol{s}}_1,\ \hat{\boldsymbol{t}}_1\big)$.
Let $\hat{K} := |\{\hat{\pi}_k \in \hat{\boldsymbol{\pi}}_1 :\ \hat{\pi}_k > \epsilon_N\}|$ and $\hat{L} := |\{\hat{\rho}_l \in \hat{\boldsymbol{\rho}}_1 :\ \hat{\rho}_l > \delta_M\}|$.
while not converged do
    In the $(c+1)th$ iteration,
    1. update row parameters:
        $$\boldsymbol{u} = \boldsymbol{R}\cdot(\boldsymbol{t}^{(c)})^\top, \boldsymbol{m} = (\boldsymbol{t}^{(c)})^\top \cdot \mathbf{1}_M$$
        $$s_{ik}^{(c+1)} = \frac{\pi_k^{(c)}\prod_l(\theta_{kl}^{(c)})^{u_{il}}(1-\theta_{kl}^{(c)})^{m_l-u_{il}}}{\sum_{k'}\pi_{k'}^{(c)}\prod_l(\theta_{k'l}^{(c)})^{u_{il}}(1-\theta_{k'l}^{(c)})^{m_l-u_{il}}}$$
        $$\boldsymbol{\pi}^{(c+1)} = (\boldsymbol{s}^{(c+1)})^\top \cdot \mathbf{1}_N/N$$
    2. update block parameters:
        $$\theta_{kl}^{(t+1/2)} = \underset{a}{\text{argmin}}\ \mathcal{A}_{kl}(a \mid \boldsymbol{R}, \boldsymbol{s^{(c+1)}}, \boldsymbol{t^{(c)}}, \boldsymbol{\Theta^{(c)}}, \boldsymbol{d^{(c)}}, \boldsymbol{\mu^{(c)}})$$
        $$\hat{d}_{kk'll'}^{(c+1/2)} = \begin{cases}\hat{\theta}_{kl}^{(c+1/2)} - \hat{\theta}_{k'l'}^{(c+1/2)} - \hat{\mu}_{kk'll'}^{(c)}, & \text{if } |\hat{d}_{kk'll'}^{(c)}| \geq \tau;\\ \text{ST}\big(\hat{\theta}_{kl}^{(c+1/2)} - \hat{\theta}_{k'l'}^{(c+1/2)} - \hat{\mu}_{kk'll'}^{(c)}; \lambda_{\text{TLP}}/\gamma\big), & \text{if } |\hat{d}_{kk'll'}^{(c)}| < \tau,\end{cases}$$
            where $\text{ST}(x;\gamma) = (||x||_2 - \gamma)_+/||x||_2$.
        $$\hat{\mu}_{kk'll'}^{(c+1/2)} = \hat{\mu}_{kk'll'}^{(c)} + \hat{d}_{kk'll'}^{(c+1/2)} - \big(\hat{\theta}_{kl}^{(c+1/2)} - \hat{\theta}_{k'l'}^{(c+1/2)}\big).$$
    3. update column parameters:
        $$\boldsymbol{v} = (\boldsymbol{s}^{(c+1)})^\top \cdot \boldsymbol{R},\ \boldsymbol{n} = \boldsymbol{s}^{(c+1)} \cdot \mathbf{1}_N$$
        $$t_{jl}^{(c+1)} = \frac{\rho_l^{(c)}\prod_k(\theta_{kl}^{(c+1/2)})^{v_{kj}}(1-\theta_{kl}^{(c+1/2)})^{n_k-v_{kj}}}{\sum_{l'}\rho_{l'}^{(c)}\prod_k(\theta_{kl'}^{(c+1/2)})^{v_{kj}}(1-\theta_{kl'}^{(c+1/2)})^{n_k-v_{kj}}}$$
        $$\boldsymbol{\rho}^{(c+1)} = (\boldsymbol{t}^{(c+1)})^\top \cdot \mathbf{1}_M/M$$
    4. update block parameters:
        $$\theta_{kl}^{(t+1)} = \underset{a}{\text{argmin}}\ \mathcal{A}_{kl}(a \mid \boldsymbol{R}, \boldsymbol{s^{(c+1)}}, \boldsymbol{t^{(c+1)}}, \boldsymbol{\Theta^{(c+1/2)}}, \boldsymbol{d^{(c+1/2)}}, \boldsymbol{\mu^{(c+1/2)}})$$
        $$\hat{d}_{kk'll'}^{(c+1)} = \begin{cases}\hat{\theta}_{kl}^{(c+1)} - \hat{\theta}_{k'l'}^{(c+1)} - \hat{\mu}_{kk'll'}^{(c+1/2)}, & \text{if } |\hat{d}_{kk'll'}^{(c+1/2)}| \geq \tau;\\ \text{ST}\big(\hat{\theta}_{kl}^{(c+1)} - \hat{\theta}_{k'l'}^{(c+1)} - \hat{\mu}_{kk'll'}^{(c+1/2)}; \lambda_{\text{TLP}}/\gamma\big), & \text{if } |\hat{d}_{kk'll'}^{(c+1/2)}| < \tau,\end{cases}$$
            where $\text{ST}(x;\gamma) = (||x||_2 - \gamma)_+/||x||_2$.
        $$\hat{\mu}_{kk'll'}^{(c+1)} = \hat{\mu}_{kk'll'}^{(c)} + \hat{d}_{kk'll'}^{(c+1)} - \big(\hat{\theta}_{kl}^{(c+1)} - \hat{\theta}_{k'l'}^{(c+1)}\big).$$
end
Output: $\big\{\hat{\boldsymbol{\pi}}_2,\ \hat{\boldsymbol{\rho}}_2,\ \hat{\boldsymbol{\Theta}}_2,\ \hat{\boldsymbol{s}}_2,\ \hat{\boldsymbol{t}}_2\big\}$
```

$q_j(\boldsymbol{w_j}) \propto \exp\left\{\mathbb{E}_{-\boldsymbol{w}_j}\left[\log p(\boldsymbol{w}_j \mid \boldsymbol{Z}, \boldsymbol{W}_{-\boldsymbol{w}_j}, \boldsymbol{R})\right]\right\}$, *which leads to the same updates for* $\boldsymbol{s}$ *and* $\boldsymbol{t}$ *as in the Block EM algorithms.*

3.3.2.3 Initialization

Initialization plays an important role in EM-type algorithms. To get a good initialization for our algorithms, we use the spectral co-clustering algorithm (Chen and Li, 2019) for Step 1. Then we use the estimates from Step 1 as the initializations for Step 2.

3.3.2.4 Model Selection

As we discussed previously, due to the complex dependence structure among the rows and columns, the likelihood of an LBM is not numerically tractable. Therefore the traditional information criteria relying on the marginal log-likelihood are not applicable here. Instead, following Keribin et al. (2012), we use the Integrated Completed Likelihood (ICL) criterion to perform model selection for LBMs.

Assuming a factorized prior for the model parameters, $p(\Xi) = p(\boldsymbol{\pi})p(\boldsymbol{\rho})p(\boldsymbol{\theta})$, with a non-informative Dirichlet distribution $\mathcal{D}(a, \ldots, a)$ for $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$, and a Beta distribution $\mathcal{B}(b, \ldots, b)$ for $\boldsymbol{\Theta}$, the ICL for an LBM has the following close form:

$$\begin{aligned}\log p(\boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{W}) &= \log\Gamma(Ka) + \log\Gamma(La) - (K+L)\log\Gamma(a) \\ &+ KL(\log\Gamma(2b) - 2\log\Gamma(b)) - \log\Gamma(N+Ka) - \log\Gamma(M+La) \\ &+ \sum_k \log\Gamma(N_k + a) + \sum_l \log\Gamma(M_l + a) \\ &+ \sum_{k,l}[\log\Gamma(N_{kl}+b) + \log\Gamma(N_kM_l - N_{kl} + b) - \log\Gamma(N_kM_l + 2b)],\end{aligned} \tag{3.14}$$

where $N_k = \sum_i z_{ik}$ is the number of rows in kth row cluster, $M_l = \sum_j w_{jl}$ is the number of columns in lth column cluster and $N_{kl} = \sum_i r_{ij}z_{ik}w_{jl}$ is the number of black cells in the klth block. ICL can be easily computed with $(\hat{\boldsymbol{Z}}, \hat{\boldsymbol{W}})$ being the estimated cluster membership.

In Step 1, we use the ICL to tune the coefficients for the log-type penalties to select the numbers of clusters. In Step 2, since we aim to learn the block inner structure where subsets of blocks share the same block parameters, to incorporate the information of distinct block parameters, we propose to modify the ICL by adding a penalty term:

$$ICL_{\text{modified}} = ICL - \lambda_{ICL}|\boldsymbol{\Theta}|, \tag{3.15}$$

where $|\boldsymbol{\Theta}|$ is the number of unique values in $\boldsymbol{\Theta}$. In this work, we choose λ_{ICL} to be $\log(NM)/2$, which is similar to BIC (Schwarz et al., 1978).

3.3.2.5 Missing Values

The EM algorithms that we have developed can handle missing values naturally by marginalizing the likelihood over the missing observations. More precisely, if $\boldsymbol{R} = (\boldsymbol{R}_{\text{obs}}, \boldsymbol{R}_{\text{miss}})$ is the decomposition of the full matrix into the observed part $\boldsymbol{R}_{\text{obs}}$ and the missing part $\boldsymbol{R}_{\text{miss}}$, then after marginalization, the initial likelihood $\mathcal{L}(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{Z}, \Xi)$ simplifies to $\mathcal{L}(\boldsymbol{R}_{\text{obs}} \mid \boldsymbol{W}, \boldsymbol{Z}, \Xi)$. Then a naive implementation could be based on indexing the inference procedure so that the posterior conditionals of $\boldsymbol{Z}$ and $\boldsymbol{W}$ involve only sums over observed elements. To be specific, let $\boldsymbol{M}$ be the mask matrix which indicates missing data such that $M_{ij} = 0$ if r_{ij} is missing, and $M_{ij} = 1$ if r_{ij} is observed. Then we only need to make minor modifications in our algorithm:

$$u_{il} = \sum_j M_{ij} t_{jl} r_{ij}, \quad m_l = \sum_j M_{ij} t_{jl},$$
$$v_{kj} = \sum_i M_{ij} s_{ik} r_{ij}, \quad n_k = \sum_i M_{ij} s_{ik}.$$

3.4 Simulation Studies

In this section, we conducted comprehensive simulation studies under various settings to evaluate the performance of the proposed method. For the data generation process, we considered two settings. In the first setting, the block parameters were randomly sampled from a set of values. We refer to this setting as "random blocks". In the second setting, the block structure followed the DINA model as we introduced in Example III.1. We refer to the second setting as "DINA blocks". Different numbers of row clusters and column clusters and different sample sizes were considered in our simulation studies. To choose tuning parameters, we used the ICL introduced in Section 3.3.2.4. Specifically, in Step 1, the ICL was used to choose $\lambda_1 = \lambda_2 \in \{0.001, 0.003, 0.005, 0.01, 0.015, 0.02, 0.025, 0.03\}$. In Step 2, we applied the modified ICL in Eq (3.15) to choose $\lambda \in \{1, 10, 100\}$, $\tau \in \{0.01, 0.02, 0.03, 0.05\}$, $\gamma \in \{1, 10, 100\}$. For each setting, we repeated 100 times.

Three related methods were compared with the proposed one:

1. the original LBM with true numbers of clusters (R package `blockcluster`);
2. the original LBM initialized at the true values;
3. LBM with truncated Lasso penalty only.

The EM-type algorithms are known to be highly dependent on a good initialization and the block-EM algorithm for LBMs has a marked tendency to produce empty clusters using the maximum a posterior (MAP) classification rule (Brault and Mariadassou, 2015). Therefore in our simulation study, we also reported the results using the true values as initializations to evaluate the model performance with the optimal initials. Moreover, essentially only imposing truncated Lasso penalties on the block parameters could also achieve the goal of selecting numbers of clusters and learning inner block structures, since such penalties would merge similar blocks. Therefore, we also present the results using TLP only to demonstrate the necessity of the log-type penalties on the proportions.

To evaluate the results, we considered the following evaluation metrics:

1. $\boldsymbol{N}_{\text{blocks}}$: the accuracy of the selected number of blocks. We counted it a success only if both the number of row clusters and the number of column clusters were correctly selected. For original LBMs using R package `blockcluster`, we counted it a success if after using the MAP rule, there were correct numbers of row clusters and column clusters.

2. MAE($\hat{\boldsymbol{\Theta}}$): Mean Absolute Error of the estimated block parameters $\hat{\boldsymbol{\Theta}}$.

3. TNR($\hat{\boldsymbol{\Theta}}$): specificity/true negative rate, TNR = TN/TN+FP = TN/Actual N, that is, the proportion of estimated equal block parameter pairs out of true equal block pairs in $\boldsymbol{\Theta}$.

4. FNR($\hat{\boldsymbol{\Theta}}$): miss rate/false negative rate, FNR = FN/FN+TP = FN/Actual P, that is, the proportion of wrongly estimated equal block parameter pairs out of true unequal block pairs in $\boldsymbol{\Theta}$.

5. MER: we determined the cluster memberships by MAP, then calculated the Misclassification Error Rate (MER) of the cluster memberships.

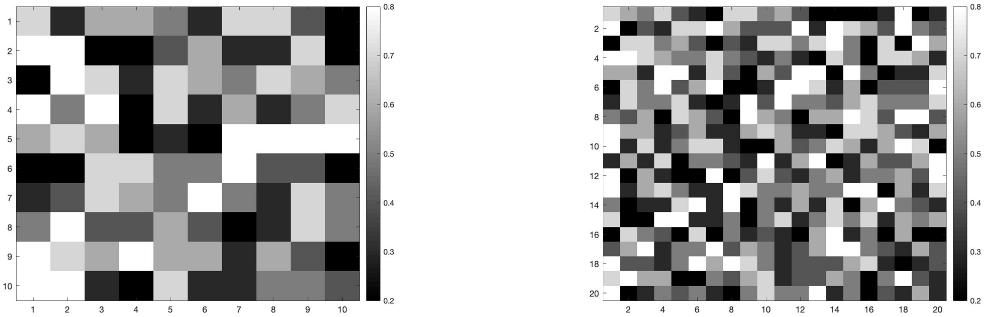
6. MAE($\hat{\boldsymbol{p}}$): Mean Absolute Error of of the estimated positive probabilities $\hat{\boldsymbol{p}} = (\hat{p}_{ij})_{N\times M}$, where $p_{ij} = \mathbb{P}(r_{ij} = 1)$, $i = 1, \ldots, N$, $j = 1, \ldots, M$.

Among the above evaluation metrics, we use $\boldsymbol{N}_{\text{blocks}}$ to measure the accuracy of the selected number of clusters and MAE($\hat{\boldsymbol{\Theta}}$) to measure the accuracy of the estimated block parameters. To evaluate the learned block inner structure, that is, the blocks that share the same block parameters, we use TNR($\hat{\boldsymbol{\Theta}}$) and FNR($\hat{\boldsymbol{\Theta}}$). Moreover, to characterize the estimate for a single entry in the data matrix, we use MAE($\hat{\boldsymbol{p}}$) to measure the accuracy of the estimated probability of a single data point. Lastly, we also calculate the Misclassification Error Rate (MER) of the cluster membership using MAP to evaluate the clustering results. Note that we only calculated MAE($\hat{\mathbf{\Theta}}$), TNR($\hat{\mathbf{\Theta}}$), FNR($\hat{\mathbf{\Theta}}$) and MER for the cases where the numbers of row clusters and column clusters were estimated correctly.

Remark III.2. *In clustering applications, there is always the label switching issue. Therefore to evaluate the estimated block parameter* $\mathbf{\Theta}$ *and the clustering membership accuracy, we need to match the learned clusters with the original clusters. One way to do such matching is to minimize the mean squared error between the learned block parameters and the original block parameters, among all permutations of rows and columns. However, the number of possible combinations of such permutations is huge, especially in the co-clustering setting. Therefore, in our simulation, we matched the clusters using the learned cluster memberships. Specifically, we compared the learned cluster memberships and the original memberships and determined the matching using the majority of original clusters in each learned cluster.*

3.4.1 Random Blocks

As we introduced in Section 3.4, in the first setting, the block parameters were randomly generated. More specifically, we randomly chose $\theta_{k,l} \sim U\{0.2, 0.3, \ldots, 0.8\}$. Since the block parameters were from a finite set, subsets of them shared the same values as shown in Figure III.2. For the random block setting, we considered two different numbers of clusters: the true number of clusters was 10×10 and we started with 20×20 clusters in Step 1; the true number of clusters was 20×20 and we started with 40×40 clusters. When the numbers of clusters were relatively small ($K = L = 10$), we experimented with relatively small sample sizes (200×200 and 500×500). When the numbers of clusters were relatively large ($K = L = 20$), we considered larger sample sizes (500×500 and 1000×1000). The simulation results are presented in Table 3.2. In our empirical experiments, when the sample size and the number of clusters were large ($N = J = 1000$, $K = L = 20$), the R package for LBM sometimes failed (6 failures out of 100 repetitions).



(a)

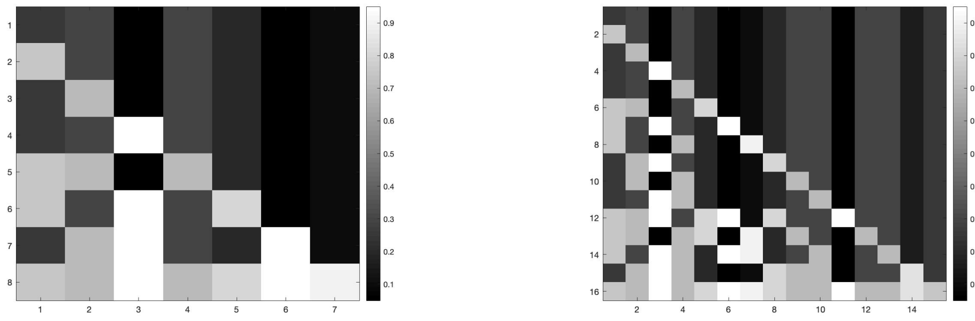
(b)

Figure III.2: Random Block Structures. (a) 10 row clusters × 10 column clusters; (b) 20 row clusters × 20 column clusters.

In terms of selecting numbers of clusters, from the results in Table 3.2, one can see that the LBMs using block EM algorithms produced empty clusters in most cases as we expected. In fact, when we looked into the first two cases with relatively small numbers of clusters ($K = L = 10$), even though it may not produce empty clusters using MAP, the biases of the estimated proportions were large. Specifically, when the sample size was 200×200, there were 83% of cases where at least one estimated proportion had an absolute bias larger than the true proportion (i.e. $|\hat{\pi} - \pi_0| > \pi_0$); while when the sample size was 500×500, all of the cases had such large biases. Therefore, even though using the correctly-specified numbers of clusters, the original LBMs performed poorly in learning true clusters. By contrast, our method achieved high accuracy of selecting the correct numbers of clusters across different settings. Moreover, only using TLP did not perform as well as our method either, demonstrating that the log-type penalties on the proportions were of essential importance in Step 1. For block inner structure learning, the proposed method performed well in terms of both TNR and FNR, while using TLP only produced larger FNRs, meaning that TLP itself would over-merge blocks. Moreover, the LBMs even starting with true initializations had very small TNRs, which is also expected since we did not put any restrictions on the block parameters in LBMs. The results here have shown that our method is capable of selecting the correct numbers of clusters and learning block inner structures effectively.

3.4.2 DINA Blocks

In this section, we considered the block structures under the DINA model. Specifically, the columns can be seen as the questions targeting certain sets of skills and the rows are the subjects answering these questions. In the DINA setting, the subjects with all the required attributes of the question will have the same high probability of getting a correct response, while the subjects missing some required attributes will have the same low probability of getting a correct response. Therefore, in the block structure considered under the DINA setting, for each column cluster, there will be two levels of positive response probability, as shown in Figure III.3. We considered two different numbers of latent attributes in the DINA model: $K = 3$ or 4. When $K = 3$, we experimented with different sample sizes $N = J = 100$, 300, or 500. When we had relatively more attributes ($K = 4$), we considered larger sample sizes $N = J = 300$, 500, or 800. The simulation results are presented in Table 3.3.



(a)

(b)

Figure III.3: Block Structures under DINA. (a) $K = 3$, 7 row clusters × 8 column clusters; (b) $K = 4$, 15 row clusters × 16 column clusters.

Similarly to the results in Section 3.4.1, the LBMs using R package `blockcluster` produced empty clusters under all considered settings, although starting with correctly specified numbers of clusters. The LBMs with true initializations failed to learn the block inner structures and only using TLP would have high FNRs. On the contrary, the proposed method performed consistently well under most of the considered cases and achieved good results in terms of all the evaluation metrics.

3.5 Real Data Analysis

In this section, we applied our method to the Cattell's 16 Personality Factors Test (Cattell and Mead, 2008) data set[1]. There were 49159 subjects and 163 items in total. The 163 items were designed to detect 16 personalities, as shown in Table 3.1. Therefore, this design can serve as a baseline for the clustering of questions. In this test, all the questions have 5 scales: 1 for strongly disagree, 2 for slightly disagree, 3 for neither agree nor disagree, 4 for slightly agree and 5 for strongly agree. In the designed questions, there are some reverse questions. For example, "I take an interest in other people's lives" and "I am not really interested in others". Therefore, for the positive questions, we binarized the responses to be 1 if the responses were bigger or equal to 3, while for the negative questions, we binarized them to be 1 if the responses were smaller than 3.

Warmth	Intellect	Emotional Stability	Assertiveness
Q1-10	Q11-23	Q24-33	Q34-43
Gregariousness	Dutifulness	Friendliness	Sensitivity
Q44-53	Q54-63	Q64-73	Q74-83
Distrust	Imagination	Reserve	Anxiety
Q84-93	Q94-103	Q104-113	Q114-123
Complexity	Introversion	Orderliness	Emotionality
Q124-133	Q134-143	Q144-153	Q154-163

Table 3.1: Clusters of items according to the target personalities.

[1] https://openpsychometrics.org

We applied our two-stage method to this personality data and used the same candidate sets of tuning parameters as in the simulation study. We started with 40 row clusters and 40 column clusters respectively. The only difference is that since the number of subjects was very large, we used the fixed effects for the row cluster memberships instead of calculating the posteriors. After fitting the model, we obtained 30 row clusters and 16 column clusters. The resulting number of column clusters coincided with the original design but the resulting clusters were not exactly the same. See Figure III.4 for the original data matrix and the rearranged data according to the clustering results.

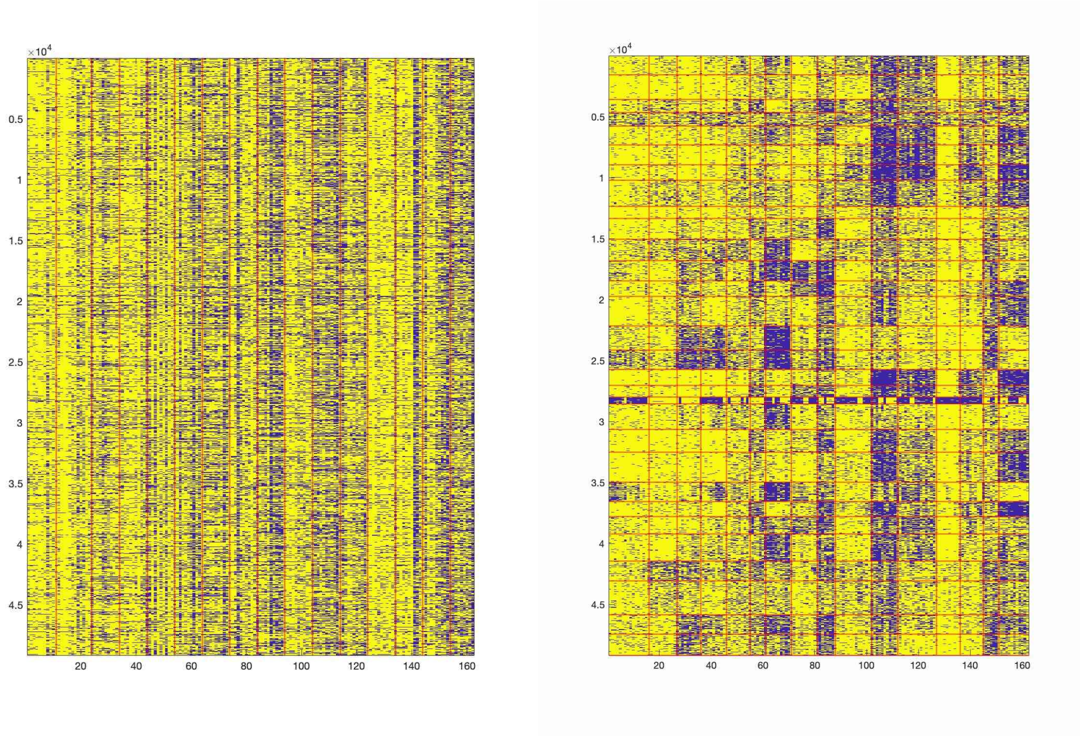


Figure III.4: Original and clustered Personality data.

To evaluate clustering performance quantitatively, we consider some internal measures since the ground truth is unknown. One such measure is called Dunn index (Dunn, 1974). Specifically, let $C = \{C_1, \ldots, C_K\}$ be K clusters of vectors. The

Dunn index is defined as

$$DI = \frac{\min_{1 \le k < k' \le K} \delta(C_k, C_{k'})}{\max_{1 \le k \le K} \Delta(C_k)},$$

where $\delta(C_k, C_{k'})$ is the intercluster distance metric and $\Delta(C_k)$ is intracluster distance metric. The larger the Dunn index, the better the clustering is. Here we define these distances as

$$\delta(C_k, C_{k'}) = \min_{i \in C_k, j \in C_{k'}} d(x_i, x_j),$$

$$\Delta(C_k) = \max_{i,j \in C_k} d(x_i, x_j),$$

where $d(\cdot, \cdot)$ is a distance measure between two vectors and here we use the Jaccard similarity. For the originally designed clusters, the Dunn index is 0.061, while for the clusters obtained by the proposed method, the Dunn index is 0.1281, which indicates that our result is better than the original design from a quantitative perspective.

3.6 Discussion

In this chapter, we propose a two-step procedure based on the latent block models to select the numbers of clusters and learn block inner structures in the co-clustering setting, where we infer the subgroups of items and subjects at the same time. Specifically, log-type penalties are put on the proportion parameters to select the significant row and column clusters, and truncated Lasso penalties are imposed on the block parameter pair differences to learn equal block parameter structures. We develop EM-type algorithms for these two steps and show its good performance through comprehensive simulation studies and a real data analysis.

In this work, we are mainly concerned with learning equal block parameter pairs as the inner block structure. More complicated block structures such as hierarchical structures may be considered for future work. For large-scale problems, the developed

EM algorithms can be easily adapted to stochastic versions where in each iteration, we randomly choose a subset of rows and a subset of columns to update the model parameters. Moreover, due to the complex dependence structures in LBMs, theoretical results are difficult to derive without making further independence assumptions. More theoretical understandings are desired in this double asymptotic regime where both the number of rows and the number of columns go to infinity, which we leave as a future direction.

Setting	Method	N_{blocks}	MAE($\hat{\boldsymbol{\Theta}}$)	TNR($\hat{\boldsymbol{\Theta}}$)	FNR($\hat{\boldsymbol{\Theta}}$)	MER	MAE($\hat{\boldsymbol{p}}$)
$N = J = 200$ $K = L = 10$	LBM (R package)	0.93	0.0597	0.0228	0.0037	0.1527	0.0620
	LBM (true initial)	1.00	0.0019	0.0265	0.0001	0.0093	0.0198
	TLP	0.17	0.0353	0.6055	0.0697	0.0322	0.0480
	Log + TLP	0.97	0.0122	0.9325	0.0228	0.0072	0.0132
$N = J = 500$ $K = L = 10$	LBM (R package)	0.43	0.0797	0.0412	0.0054	0.1831	0.0930
	LBM (true initial)	1.00	0.0007	0.0638	0.0000	0.0000	0.0072
	TLP	0.66	0.0113	0.9208	0.0279	0.0002	0.0148
	Log + TLP	1.00	0.0020	1.0000	0.0000	0.00002	0.0020
$N = J = 500$ $K = L = 20$	LBM (R package)	0.01	0.0296	0.0191	0.0011	0.1370	0.0745
	LBM (true initial)	1.00	0.0007	0.0247	0.0000	0.00005	0.0146
	TLP	0.11	0.0687	0.8082	0.2484	0.0040	0.0551
	Log + TLP	0.80	0.0028	0.9841	0.0026	0.0000	0.0035
$N = J = 1000$ $K = L = 20$	LBM (R package)	0.00	–	–	–	–	0.1492
	LBM (true initial)	1.00	0.0004	0.0612	0.0000	0.0000	0.0074
	TLP	0.96	0.0124	0.9989	0.0477	0.0000	0.0153
	Log + TLP	0.98	0.0010	1.0000	0.0000	0.0000	0.0011

Table 3.2: Performance of different methods under the random block structures.

Setting	Method	N_{blocks}	MAE($\hat{\boldsymbol{\Theta}}$)	TNR($\hat{\boldsymbol{\Theta}}$)	FNR($\hat{\boldsymbol{\Theta}}$)	MER	MAE($\hat{\boldsymbol{p}}$)
$K = 3$ $N = J = 100$	LBM (R package)	0.00	–	–	–	–	0.3461
	LBM (true initial)	1.00	0.0250	0.0234	0.0012	0.0627	0.0266
	TLP	0.30	0.0883	0.8790	0.4012	0.0953	0.0880
	Log + TLP	0.37	0.0279	0.6068	0.0595	0.0719	0.0311
$K = 3$ $N = J = 300$	LBM (R package)	0.00	–	–	–	–	0.3172
	LBM (true initial)	1.00	0.0061	0.0732	0.0000	0.0000	0.0059
	TLP	1.00	0.0661	0.9935	0.3546	0.0023	0.0655
	Log + TLP	1.00	0.0044	0.9665	0.0061	0.0014	0.0045
$K = 3$ $N = J = 500$	LBM (R package)	0.00	–	–	–	–	0.3912
	LBM (true initial)	1.00	0.0045	0.0984	0.0000	0.0001	0.0045
	TLP	1.00	0.0779	0.9955	0.4187	0.0001	0.0777
	Log + TLP	1.00	0.0020	0.9897	0.0000	0.0000	0.0021
$K = 4$ $N = J = 300$	LBM (R package)	0.00	–	–	–	–	0.3227
	LBM (true initial)	1.00	0.0154	0.0299	0.0011	0.0086	0.0149
	TLP	0.00	–	–	–	–	– 0.0461
	Log + TLP	0.10	0.0229	0.9278	0.1389	0.0397	0.0237
$K = 4$ $N = J = 500$	LBM (R package)	0.00	–	–	–	–	0.3023
	LBM (true initial)	1.00	0.0093	0.0513	0.0001	0.0002	0.0090
	TLP	0.73	0.0530	0.9678	0.3526	0.0146	0.0511
	Log + TLP	0.77	0.0111	0.9726	0.0659	0.0039	0.0114
$K = 4$ $N = J = 800$	LBM (R package)	0.00	–	–	–	–	0.3216
	LBM (true initial)	1.00	0.0061	0.0698	0.0000	0.0003	0.0060
	TLP	1.00	0.0803	0.9990	0.5133	0.0009	0.0799
	Log + TLP	0.97	0.0014	0.9958	0.0015	0.0005	0.0017

Table 3.3: Performance of different methods under the random block structures.

CHAPTER IV

Hypothesis Testing for Latent Hierarchical Structures

4.1 Introduction

In chapter II, we have developed an efficient algorithm to learn latent hierarchical structures from observed data. In many applications, such hierarchical structures are often posited by domain experts. Hypothesis testing plays an important role in validating the presence of the suspected attribute hierarchies, which can provide guidance to practitioners for experiment design or data modeling (Templin and Bradshaw, 2014). As we introduced in chapter II, if some hierarchical structure exists, any latent profile $\boldsymbol{\alpha}$ that does not respect the hierarchy is deemed not to exist with the corresponding population proportion $\pi_{\boldsymbol{\alpha}} = 0$. For $1 \leq k \neq l \leq K$, we use $\alpha_k \to \alpha_l$ (or $k \to l$) to denote the hierarchy that attribute α_k is a prerequisite for attribute α_l. Under the hierarchy $\alpha_k \to \alpha_l$, the latent profiles with $\alpha_l = 1$ but $\alpha_k = 0$ will not exist in the population and therefore we have $\pi_{\boldsymbol{\alpha}} = 0$ if $\alpha_l = 1$ but $\alpha_k = 0$. The set of prerequisite relationships is denoted by $\mathcal{E} = \{k \to l : \text{ attribute } k \text{ is a prerequisite for } l,\ 1 \leq k \neq l \leq K\}$, and the induced set of existing latent attribute profiles is denoted by $\mathcal{A} = \{\boldsymbol{\alpha} \in \{0,1\}^K : \pi_{\boldsymbol{\alpha}} \neq 0 \text{ under } \mathcal{E}\}$. It is noted that an attribute hierarchy results in the sparsity of the proportion parameter vector, which will reduce the number of model parameters especially when K is large. Example hierarchical structures and the corresponding induced latent profile sets are shown in Figure II.1.

In this chapter, we consider the problem of hypothesis testing for the existence of such pre-specified latent hierarchical structures. As we illustrated above, the hierarchical structure of the latent attributes results in the sparsity structure of the proportion parameter vector for the latent attribute profiles, since the latent profiles that do not follow the hierarchical structure will not exist in the population. Therefore the problem of testing latent hierarchy is equivalent to testing the sparsity structure of the proportion parameter vector. More formally, we aim to test the following hypothesis:

$$\mathrm{H}_0 : \pi_{\boldsymbol{\alpha}} = 0,\ \forall \boldsymbol{\alpha} \notin \mathcal{A}_0 \text{ under hierarchy } \mathcal{E}_0,$$

where $\mathcal{E}_0$ is the hierarchical structure under the null hypothesis and $\mathcal{A}_0$ is the induced latent attribute profile set under $\mathcal{E}_0$.

Even though LAMs can be viewed as a special family of finite mixture models, there is a key difference between testing hierarchical structures in LAMs and testing the number of components in finite mixture models. When testing the number of components in finite mixture models, there are no restrictions on the components' distributions. See Chen (2017) for a review of testing the number of components in finite mixture models. However, when testing latent hierarchical structures, we are in fact testing whether the proportion parameters of the nonexistent latent attribute profiles corresponding to the hierarchy are zeros. Moreover, the constraints imposed by the structural Q-matrix make it more restrictive and complicated.

To our best knowledge, there are no systematical testing procedures or statistical theories on hypothesis testing for latent hierarchical structures. Two natural questions about such testings are (1) when the hierarchical structures are testable and

(2) how to conduct the hypothesis testing. On the one hand, under the framework of LAMs, if the hierarchical structure under the null hypothesis cannot be distinguished from those under the alternative, we cannot test such a hierarchical structure and therefore it is untestable. In fact, the testability of hierarchical structures is closely related to the identifiability of the models. On the other hand, under the hierarchical constraints, the problem of testing latent hierarchical structures is equivalent to testing the sparsity structure of the set of latent attribute profiles in the population, that is, the sparsity structure of the population proportion parameter vector. However, due to the identifiability and the irregularity issue that the true proportion parameters are on the boundary of the parameter space under hierarchical structures, the conventional asymptotic Chi-squared distribution may not hold for the likelihood ratio test.

Non-regularity issues of the likelihood ratio test are known to exist in many latent variable models such as finite mixture models, factor analysis, structural equation models, and random effects models (Chen, 2017; Chen et al., 2020). In particular, testing the sparsity structure of the proportion parameter vector in LAMs is closely related to the problem of testing the number of components in finite mixture models and latent class models (Nylund et al., 2007; Chen, 2017). However, testing the hierarchical structures in LAMs is even more challenging, since it tests whether a specific set of the proportion parameters specified by the hierarchical structure under the null hypothesis is zero; and such a problem is further complicated due to the restrictions imposed by the structural Q-matrix and the discrete nature of the latent variables in LAMs.

In this chapter, we focus on the problem of hypothesis testing for latent hierarchical structures. We first discuss the testability of latent hierarchical structures and present sufficient conditions under which hierarchical structures are testable in LAMs. Then under such conditions, we examine the asymptotic behaviors of the popularly used likelihood ratio test. Since the true proportion parameter is on the boundary of the parameter space, the asymptotic distribution of LRT becomes nonstandard due to the lack of regularity (Self and Liang, 1987). Moreover, the nonstandard limiting distribution of LRT is observed to not provide satisfactory finite-sample results under practical settings, and we provide statistical insights on such failures. Specifically, we find that when the number of items is large or the item parameters are close to the boundary, the convergence of the nonstandard limiting distribution can be very slow and the test tends to fail. Therefore we do not recommend using the nonstandard limiting distribution to conduct the hypothesis testing in practice. Instead, based on these findings, we propose to use resampling-based methods to test hierarchical structures. We conduct comprehensive simulations and comparisons between parametric bootstrap and nonparametric bootstrap and recommend using parametric bootstrap for testing latent hierarchies in LAMs.

The rest of the chapter is organized as follows: we first discuss some sufficient conditions for the testability of hierarchical structures and provide several illustrative examples in Section 4.2. Studies on the likelihood ratio test and numerical results are presented in Section 4.3. Specifically, section 4.3.1 studies the asymptotic behaviors of LRT and provides insights into its failures in some situations. Section 4.3.2 presents simulation studies that compare parametric bootstrap and nonparametric bootstrap for testing hierarchical structures. In Section 4.4, we perform hypothesis testing for a linear attribute hierarchy in an educational assessment dataset and compare different testing procedures. Finally Section 4.5 concludes with some discussions.

4.2 Testability Requirements and Conditions

Before we introduce concrete testing procedures, we first need to understand when the hierarchical structures are testable. For instance, consider the case when the item parameters are the same for two latent attribute profiles and we want to test the nonexistence of one of them. In this situation, we cannot distinguish these two profiles and thus cannot identify their proportion parameters, not mention testing whether the corresponding proportion is zero. Example IV.1 provides an illustrative example. Therefore, the testability issue is of fundamental importance before performing concrete testing procedures. Moreover, the testability conditions would also provide guidance for practitioners and scientific researchers to design experiments.

Example IV.1. *Assume that there are two latent attributes of interest and there is a linear attribute hierarchy $\mathcal{E}_0 = \{1 \to 2\}$, which results in the induced attribute profile set $\mathcal{A}_0 = \{(0,0),(1,0),(1,1)\}$. If the Q-matrix is specified as,*

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

then under the DINA model assumption, the item parameter vector for the latent profile $(0,0)$, $\boldsymbol{\theta}_{(0,0)}$, would be the same as $\boldsymbol{\theta}_{(0,1)}$. In this case, we cannot distinguish the profiles $(0,0)$ and $(0,1)$, and therefore the proportion parameters $\pi_{(0,0)}$ and $\pi_{(0,1)}$ cannot be identified. Furthermore, the induced profile set $\mathcal{A}_0$ is not identifiable, which makes the latent hierarchical structure untestable.

To ensure the testability of hierarchical structures, some conditions need to be met. Before we dive into these conditions, let's first define the concept of the testability of latent hierarchies.

Definition 4.2.1 (strict testability of $\mathcal{E}_0$)**.** Given the Q-matrix and certain cognitive diagnosis model assumptions, consider the following hypothesis testing:

$$\mathrm{H}_0\text{: the latent attributes respect the hierarchy } \mathcal{E}_0,$$

$$\text{H}_1\text{: the latent attributes do not respect the hierarchy } \mathcal{E}_0.$$

Then the latent hierarchy $\mathcal{E}_0$ is said to be testable if there is no parameter under the alternative hypothesis that gives the same distribution as the parameters under the null hypothesis.

In fact, the testability is closely related to the identifiability of LAMs (e.g., Xu and Zhang, 2016; Xu, 2017; Xu and Shang, 2018). The identifiability refers to that if two parameters give the same distribution, then the two parameters must be the same. Nevertheless, the testability of hierarchical structure is actually less restrictive compared with the identifiability. In testing latent hierarchies, we only need to distinguish the latent attribute profiles under the null hierarchical structure from the others under the alternative, while in terms of the identifiability, we need to identify all the model parameters and all the latent attribute profiles. Therefore the concept of testability is weaker than the definitions of identifiability. In particular, identifiability is a sufficient but not necessary condition for testability.

We first consider the DINA model. For the DINA model, since the item parameters only depend on the highest interactions among the required latent attributes, we have equivalent Q-matrices under hierarchical structures. As introduced in Section 2.3.1, we say two Q-matrices are equivalent under hierarchical structure $\mathcal{E}$, denoted by $\mathbf{Q}_1 \overset{\mathcal{E}}{\sim} \mathbf{Q}_2$, if they give the same item parameter matrices, that is, $\boldsymbol{\Theta}(\mathbf{Q}_1, \mathcal{A}_{\mathcal{E}}) = \boldsymbol{\Theta}(\mathbf{Q}_2, \mathcal{A}_{\mathcal{E}})$, where $\mathcal{A}_{\mathcal{E}}$ is the induced latent attribute profile set under hierarchy $\mathcal{E}$. For example, consider three latent attributes with a linear hierarchy, that is, $\mathcal{E} = \big\{1 \to 2 \to 3\big\}$. We have

$$\mathbf{Q}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \overset{\mathcal{E}}{\sim} \mathbf{Q}^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \overset{\mathcal{E}}{\sim} \mathbf{Q}^{(*)} = \begin{pmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{pmatrix}, \tag{4.1}$$

where "$*$" can be either 0 or 1. Based on this observation, following Gu and Xu

(2022), we introduce two useful operations on the Q-matrix.

Definition 4.2.2. Given an attribute hierarchy $\mathcal{E}$ and a Q-matrix $\boldsymbol{Q}$. For any $q_{j,l} = 1$ and $k \to l$, set $q_{j,k}$ to 0 and obtain a modified matrix $\mathcal{S}^{\mathcal{E}}(\boldsymbol{Q})$, which is called the "sparsified" version of $\boldsymbol{Q}$.

Definition 4.2.3. Given an attribute hierarchy $\mathcal{E}$ and a Q-matrix $\boldsymbol{Q}$. For any $q_{j,l} = 1$ and $k \to l$, set $q_{j,k}$ to 1 and obtain a modified matrix $\mathcal{D}^{\mathcal{E}}(\boldsymbol{Q})$, which is called the "densified" version of $\boldsymbol{Q}$.

As we discussed previously, the identifiability conditions are sufficient conditions for testability. We present some identifiability results in Gu and Xu (2022) for the DINA model.

Proposition 4.2.1 (strict testability for the DINA model)**.** *Consider a DINA model with a given $\boldsymbol{Q}$. A hierarchy $\mathcal{E}_0$ is testable if $\boldsymbol{Q}$ satisfies the following conditions:*

(1) $\boldsymbol{Q}$ contains a $K \times K$ identity submatrix I_K. (Without loss of generality, assume the first K rows of $\boldsymbol{Q}$ form I_K and denote the remaining submatrix of $\boldsymbol{Q}$ by $\boldsymbol{Q}^*$.)

(2) $\mathcal{S}^{\mathcal{E}_0}(\boldsymbol{Q})$, the sparsified version of $\boldsymbol{Q}$, has at least three entries of "1" in each column.

(3) $\mathcal{D}^{\mathcal{E}_0}(\boldsymbol{Q}^*)$, the densified version of $\boldsymbol{Q}^*$, contains K distinct column vectors.

Among the above conditions, condition (1) is in fact necessary to ensure the testability of any hierarchy $\mathcal{E}$, which is however not satisfied in Example IV.1. Since the Q-matrix in Example IV.1 does not contain $(0, 1)$, we can not distinguish the latent attribute profiles $(0, 1)$ and $(0, 0)$, making the hierarchy not testable. Conditions (2) and (3), on the other hand, may be further weakened. For example, Gu and Xu (2022) provided necessary conditions for the identifiability of DINA-based hierarchical cognitive diagnosis models and discussed different necessary conditions for different hierarchical structures. As testability is a less restrictive concept than identifiability, we would expect the identifiability conditions (2) and (3) can be further weakened for testability. We leave such an interesting yet challenging question for future study. We next revisit Example IV.1 with a different Q-matrix and demonstrate its testability.

Example IV.2 (Example IV.1 revisited)**.** *Consider the same setting as in Example IV.1, but with a different Q-matrix specified as below:*

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{I}_2 \\ \boldsymbol{Q}^* \end{pmatrix}, \ \textit{where } \boldsymbol{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \textit{ and } \boldsymbol{Q}^* = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Then the modified Q matrices are:

$$\mathcal{S}^{\mathcal{E}_0}(\boldsymbol{Q}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \mathbf{0} & 1 \end{pmatrix}, \quad \mathcal{D}^{\mathcal{E}_0}(\boldsymbol{Q}^*) = \begin{pmatrix} 1 & 0 \\ \mathbf{1} & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Since the Q-matrix contains an identity matrix $\boldsymbol{I}_2$, the sparsified version $\mathcal{S}^{\mathcal{E}_0}(\boldsymbol{Q})$ has three "1" entries in each column, and the densified version $\mathcal{D}^{\mathcal{E}_0}(\boldsymbol{Q}^*)$ contains two distinct columns, all three conditions in Theorem 1 are satisfied. Therefore, for a DINA model with this Q-matrix, the linear hierarchy $\mathcal{E}_0$ is strictly testable.

Given an attribute hierarchy $\mathcal{E}_0$, if we are interested in testing a subset of prerequisite relations conditioned on that the other prerequisite relations are assumed, we can further relax Condition (1) in Proposition 4.2.1. For example, assume that there are three latent attributes and the full hierarchical structure is $\mathcal{E}_0 = \{1 \to 2 \to \ 3\}$. If we are interested in testing $\mathcal{E} = \{1 \to 2\}$ given $\mathcal{E}_0 \setminus \mathcal{E} = \{2 \to 3\}$, that is $\mathrm{H}_0 : \ \mathcal{E}_0$ vs. $\mathrm{H}_1 : \ \mathcal{E}_0 \setminus \mathcal{E}$, we can relax Condition (1) in Proposition 4.2.1 to Condition (1*) in Corollary 4.2.1.

Corollary 4.2.1. *Consider a DINA model with a given $\boldsymbol{Q}$ and a given attribute hierarchy $\mathcal{E}_0$. Suppose we are interested in testing a subset $\mathcal{E} \subset \mathcal{E}_0$ given that $\mathcal{E}_0 \setminus \mathcal{E}$ has already been assumed. It is testable if Condition (2) and (3) in Proposition 4.2.1 and the following condition are satisfied:*

(1^*) $\mathcal{S}^{\mathcal{E}_0}(\boldsymbol{Q})$, the sparsified version of $\boldsymbol{Q}$, contains an identity submatrix I_K and for any attribute α_k involved in $\mathcal{E}$, there is an item which is only targeted on this attribute.

The proof of Corollary 4.2.1 directly follows Theorem 1 in Gu and Xu (2022). As we mentioned previously, the testability is a weaker requirement than identifiability, in that we only need to differentiate the latent attribute profiles between the null and alternative hypothesis for testability. We next provide examples in which the hierarchical structures are testable but the models are not identifiable.

Example IV.3 (Testability vs. Identifiability)**.** *Consider a DINA model with three latent attributes. Further, assume the slipping and guessing parameters are known. We want to test the linear hierarchy which is specified as $\mathcal{E}_0 = \{1 \to 2 \to 3\}$. Then the induced latent attribute profile set is $\mathcal{A}_0 = \{(0,0,0),(1,0,0),(1,1,0),(1,1,1)\}$. We denote the set of latent attribute profiles that do not exist under the hierarchy as $\mathcal{A}_0^c = \{(0,1,0),(0,0,1),(1,0,1),(0,1,1)\}$. Consider the following Q-matrix and get* *the corresponding ideal response matrices for $\mathcal{A}_0$ and $\mathcal{A}_0^c$:*

$$\boldsymbol{Q} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}; \quad \boldsymbol{\Gamma}^{\mathcal{A}_0} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\Gamma}^{\mathcal{A}_0^c} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then the ideal response vectors for classes in $\mathcal{A}_0$ and $\mathcal{A}_0^c$ are different, making the hierarchical structure testable. However, the ideal response vectors for $(0, 0, 0)$ and $(1, 0, 0)$ are the same, and those for $(0, 0, 1)$ and $(1, 0, 1)$ are also the same, making the model not identifiable.

Example IV.4 (Condition on Alternative)**.** *Consider the same model and hierarchical structure in Example IV.3 but with unknown slipping and guessing parameters. Here we are interested in testing $\{1 \to 2\}$ given $\{2 \to 3\}$. Consider the Q-matrix:*

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

Since $\boldsymbol{\theta}_{(0,0,0)} = \boldsymbol{\theta}_{(0,0,1)}$, we cannot distinguish the latent profiles $(0, 0, 0)$ and $(0, 0, 1)$, and thus not all the latent profiles are identifiable. However, the conditions in Proposition 4.2.1 are satisfied, so the hierarchical structure $\{1 \to 2\}$ given $\{2 \to 3\}$ is *testable.*

For more general LAMs, we adapt the identifiability results from Gu and Xu (2019b) to establish sufficient conditions for the testability of latent attribute hierarchies. Following the same notations in Gu and Xu (2019b) and Section 2.3.2, we use the so-called constraint matrix $\mathbf{\Gamma}$. Recall that the constraint matrix for a set of latent attribute profiles $\mathcal{A}$ is defined as $\mathbf{\Gamma}^{\mathcal{A}} = (\mathbb{I}(\boldsymbol{\alpha} \succeq \boldsymbol{q}_j) : \boldsymbol{\alpha} \in \mathcal{A}, j \in [J]) \in \{0,1\}^{J\times|\mathcal{A}|}$, which is a binary matrix indicating whether an attribute profile $\boldsymbol{\alpha} \in \mathcal{A}$ possesses all the required attributes of item j. Note that for the DINA model, the constraint matrix is also its ideal response matrix, while they are not the same for the DINO model. The defined constraint matrix is used as a tool to study the testability conditions for general LAMs. Based on the constraint matrix, we define a partial order among the latent attribute profiles "$\succeq_S$" for any subset of items $S \subset [J]$. For $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}$, we say $\boldsymbol{\alpha} \succeq_S \boldsymbol{\alpha}'$ under $\mathbf{\Gamma}^{\mathcal{A}}$ if $\Gamma^{\mathcal{A}}_{j,\boldsymbol{\alpha}} \geq \Gamma^{\mathcal{A}}_{j,\boldsymbol{\alpha}'}$ for $j \in S$. And for two item sets S_1 and S_2, we say "$\succeq_{S_1}=\succeq_{S_2}$" if for any $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}$, we have $\boldsymbol{\alpha} \succeq_{S_1} \boldsymbol{\alpha}'$ if and only if $\boldsymbol{\alpha} \succeq_{S_2} \boldsymbol{\alpha}'$. Example IV.5 provides illustrations for the partial orders.

Example IV.5. *Consider two latent attributes with a linear hierarchical structure. The Q-matrix considered and the corresponding constraint matrix for the latent attribute profile set* $\mathcal{A} = \{(0,0),(1,0),(1,1)\}$ *are specified as :*

$$\boldsymbol{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{\Gamma}^{\mathcal{A}} = \begin{array}{c} \begin{matrix} (0,0) & (1,0) & (1,1) \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}.$$

For the item set $S = \{1,2\}$, *we can see that* $\Gamma_{j,(1,0)} \geq \Gamma_{j,(0,0)}$ *and* $\Gamma_{j,(1,1)} \geq \Gamma_{j,(1,0)}$ *for* $j \in S$. *Therefore* $(1,0) \succeq_S (0,0)$ *and* $(1,1) \succeq_S (1,0)$. *Moreover, if we take* $S_1 = \{1,2\}$

and $S_2 = \{3, 4\}$, then for any $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}$, we have $\boldsymbol{\alpha} \succeq_{S_1} \boldsymbol{\alpha}'$ if and only if $\boldsymbol{\alpha} \succeq_{S_2} \boldsymbol{\alpha}'$. Therefore, $\succeq_{S_1} = \succeq_{S_2}$.

In the following testability results for general LAMs, we focus on equal size cases or under-fitted cases when $|\mathcal{A}| \leq |\mathcal{A}_0|$, where $\mathcal{A}_0$ is the set of the latent attribute profiles under the null hypothesis and $\mathcal{A}$ is the set of the latent attribute profiles under the alternative hypothesis. Note that for overfitted cases with $|\mathcal{A}| > |\mathcal{A}_0|$, if $\mathcal{A}$ and $\mathcal{A}_0$ lead to the same distribution, the model complexity of $\mathcal{A}$ is larger than that of $\mathcal{A}_0$, and therefore practically we can still distinguish them using information-based criteria or penalized likelihood methods.

Proposition 4.2.2 (Strict testability for general LAMs)**.** *Consider a general LAM with a given $\boldsymbol{Q}$ and an arbitrary hierarchy $\mathcal{E}_0$. The hierarchy is testable when the alternative is restricted to the latent profile sets of the same or smaller size than that under the null hypothesis, if the following conditions of the constraint matrix $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ corresponding to the induced latent profile set $\mathcal{A}_0$ under the hierarchy $\mathcal{E}_0$ are satisfied:*

(1) There exist two disjoint item sets S_1 and S_2, such that $\boldsymbol{\Gamma}^{(S_i, \mathcal{A}_0)}$ has distinct column vectors for $i = 1, 2$ and "$\succeq_{S_1} = \succeq_{S_2}$" under $\boldsymbol{\Gamma}^{\mathcal{A}_0}$.

(2) For any $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}' \in \mathcal{A}_0$ where $\boldsymbol{\alpha}' \succeq_{S_i} \boldsymbol{\alpha}$ under $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ for $i = 1$ or 2, there exists some $j \in \big(S_1 \cup S_2\big)^c$ such that $\Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}} \neq \Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}'}$.

(3) Any column vector of $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ is different from any column vector of $\boldsymbol{\Gamma}^{\mathcal{A}_0^c}$, where $\mathcal{A}_0^c = \{0, 1\}^K \setminus \mathcal{A}_0$

Based on the conditions in Proposition 4.2.2, one can see that having three identity submatrices in the Q-matrix is sufficient for testability. However, having several identity submatrices is in fact a strong requirement in practice. Under a general LAM, these conditions can be further relaxed if we consider $\mathcal{E}_0$ to be testable with the true model parameter ranging almost everywhere in the restricted parameter space except a set of Lebesgue measure zero. Specifically, we have the following definition of generic testability.

Definition 4.2.4 (Generic testability of $\mathcal{E}_0$)**.** Denote the parameter space under $\mathcal{E}_0$ by $\boldsymbol{\Omega}_0$. The latent hierarchy $\mathcal{E}_0$ is said to be generically testable, if there exists a subset $\boldsymbol{\mathcal{V}}$ of $\boldsymbol{\Omega}_0$ that has Lebesgue measure zero, such that there is no parameter under the alternative hypothesis gives the same distribution as the parameters in $\boldsymbol{\Omega}_0 \setminus \boldsymbol{\mathcal{V}}$.

For generic testability, following the generic identifiability results in Gu and Xu (2019b) and Gu and Xu (2020), a nice corollary can be derived where the requirements are directly characterized by the structure of the Q-matrix.

Corollary 4.2.2. *If the Q-matrix satisfies the following conditions, then for any hierarchy $\mathcal{E}_0$ such that the induced latent attribute profile set $\mathcal{A}_0$ satisfies Condition (3) in Proposition 4.2.2, the hierarchy $\mathcal{E}_0$ is generically testable:*

(1) The Q-matrix contains two $K \times K$ sub-matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, such that for $i = 1, 2$,

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_1 \\ \boldsymbol{Q}_2 \\ \boldsymbol{Q}' \end{pmatrix}_{J\times K}; \quad \boldsymbol{Q}_i = \begin{pmatrix} 1 & * & \cdots & * \\ * & 1 & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & 1 \end{pmatrix}_{K\times K}, i = 1, 2,$$

where each "$*$" can be either zero or one.

(2) With $\boldsymbol{Q}$ in the form as above, $\sum_{j=2K+1}^{J} q_{j,k} \geq 1$ for each $k \in [K]$.

By relaxing strict testability to generic testability, less stringent conditions in Corollary 4.2.2 have been established. Moreover, the requirements in Corollary 4.2.2

can be checked directly from the Q-matrix, making it easier to use in practice. Next, we present an illustrative example of strict testability and generic testability of general LAMs.

Example IV.6. *Consider a general latent attribute model setting with two latent attributes and a linear attribute hierarchy* $\mathcal{E}_0 = \{1 \to 2\}$*. Consider the Q-matrix:*

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{I}_2 \\ \boldsymbol{I}_2 \\ \boldsymbol{Q}' \end{pmatrix}; \quad \boldsymbol{Q}' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

By directly looking at the Q-matrix, we know the conditions in Corollary 4.2.2 are satisfied and therefore the hierarchical structure is generically testable. Moreover, the constraint matrix under attribute hierarchy $\mathcal{E}_0 = \{1 \to 2\}$ *is*

$$\boldsymbol{\Gamma}^{\mathcal{A}_0} = \begin{array}{c} \begin{array}{ccc} (0,0) & (1,0) & (1,1) \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}; \quad \boldsymbol{\Gamma}^{\mathcal{A}_0^c} = \begin{array}{c} (0,1) \\ \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{array}.$$

If we set $S_1 = \{1,2\},\ S_2 = \{3,4\}$*, then* $\boldsymbol{\Gamma}^{(S_i,\mathcal{A}_0)}$ *has distinct columns for* $i = 1, 2$*. Moreover, "*$\succeq_{S_1} = \succeq_{S_2}$*" under* $\boldsymbol{\Gamma}^{\mathcal{A}_0}$*. For* $(1,0) \succeq_{S_i} (0,0)$ *for* $i = 1$ *or 2, we have* $\Gamma^{\mathcal{A}_0}_{5,(1,0)} \neq \Gamma^{\mathcal{A}_0}_{5,(0,0)}$*. For* $(1,1) \succeq_{S_i} (1,0)$ *for* $i = 1$ *or 2, we have* $\Gamma^{\mathcal{A}_0}_{6,(1,1)} \neq \Gamma^{\mathcal{A}_0}_{6,(1,0)}$*. For* $(1,1) \succeq_{S_i} (0,0)$ *for* $i = 1$ *or 2, we have* $\Gamma^{\mathcal{A}_0}_{6,(1,1)} \neq \Gamma^{\mathcal{A}_0}_{6,(0,0)}$*. Finally, the columns of* $\boldsymbol{\Gamma}^{\mathcal{A}_0}$ *are different from that of* $\boldsymbol{\Gamma}^{\mathcal{A}_0^c}$*. Therefore, based on the constraint matrix, we can see that the conditions in Proposition 4.2.2 are met, and thus the linear attribute* *hierarchy is also strictly testable.*

Example IV.7. *Consider a general latent attribute model setting with three latent attributes and a linear hierarchical structure $\mathcal{E}_0 = \{1 \to 2 \to 3\}$. The induced latent attribute profile set under $\mathcal{E}_0$ is $\mathcal{A}_0 = \{(0,0,0),(1,0,0),(1,1,0),(1,1,1)\}$. We denote the complement set of latent attribute profiles as $\mathcal{A}_0^c = \{(0,1,0),(0,0,1),(1,0,1),(0,1,1)\}$. Consider the Q-matrix and the corresponding constraint matrices for $\mathcal{A}_0$ and $\mathcal{A}_0^c$:*

$$\boldsymbol{Q} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \boldsymbol{\Gamma}^{\mathcal{A}_0} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\Gamma}^{\mathcal{A}_0^c} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Based on the specified Q-matrix and the corresponding constraint matrices, one can easily see that the conditions in Corollary 4.2.2 are satisfied and therefore the hierarchical structure is generically testable. However, Condition (1) in Proposition 4.2.2 is not satisfied and the model is not strictly identifiable since $\boldsymbol{\Gamma}^{(0,0,0)}$ and $\boldsymbol{\Gamma}^{(1,0,0)}$ are the same.

4.3 Likelihood Ratio Test

With the sufficient conditions for the testability of the hierarchical structures specified in Section 4.2, the next question becomes how to conduct the hypothesis testing. As we illustrated in Section 2.1 when some hierarchical structure exists, the number of truly existing latent attribute profiles will be less than 2^K, and the corresponding model will be a nested model of the full model with all possible latent attribute profiles. Testing the latent hierarchical structure is then equivalent to testing the sparsity structure of the proportion parameter vector. A popular choice for testing a nested model is the likelihood ratio test with an asymptotic Chi-squared distribution under some regularity conditions. One commonly assumed regularity condition is that the true parameter vector is in the interior of the parameter space. However, in our testing problem, the true proportion parameter vector $\boldsymbol{\pi}$ lies on the boundary of the simplex under the null hypothesis, making the conventional Chi-squared limiting distribution no longer hold. In this section, we review the nonstandard asymptotic behaviors of the LRT statistic and provide statistical insights on the failures of such limiting distributions under practical conditions. Then we propose to use resampling-based methods to test hierarchical structures and conduct a comprehensive simulation study to compare different testing procedures.

4.3.1 Failure of Limiting Distribution of LRT

When the parameter of the null model lies on the boundary of the parameter space, the LRT statistic has been shown to often follow a mixture of χ^2 distributions asymptotically (Self and Liang, 1987). We first present some general asymptotic theories on the LRT statistic under such nonstandard conditions and discuss the application to our testing problem for latent hierarchies.

Let $f(\boldsymbol{x};\boldsymbol{\theta})$ be the probability density function of a random variable $\boldsymbol{X}$, where $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$ takes values in the parameter space $\boldsymbol{\Omega}$, a subset of $\mathbb{R}^p$. When the model is identifiable, distinct values of $\boldsymbol{\theta}$ correspond to distinct probability distributions. Let $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ be N independent observations of $\boldsymbol{X}$ and denote the log-likelihood function,

$\sum_{i=1}^{N}\log[f(\boldsymbol{x}_i;\boldsymbol{\theta})]$, by $l_N(\boldsymbol{\theta})$. Consider the hypothesis testing

$$\mathrm{H}_0 : \boldsymbol{\theta}_0 \in \boldsymbol{\Omega}_0 \quad vs \quad \mathrm{H}_1 : \boldsymbol{\theta}_0 \in \boldsymbol{\Omega} \setminus \boldsymbol{\Omega}_0,$$

where $\boldsymbol{\theta}_0$ is the true parameter and $\boldsymbol{\Omega}_0$ is a subset of $\boldsymbol{\Omega}$. When $\boldsymbol{\Omega}_0$ is an r-dimensional subset of $\boldsymbol{\Omega}$, $\boldsymbol{\theta}_0$ is a boundary point of both $\boldsymbol{\Omega}_0$ and $\boldsymbol{\Omega}\setminus\boldsymbol{\Omega}_0$ but an interior point of $\boldsymbol{\Omega}$, under some regularity conditions, by the Wilk's theorem, the asymptotic distribution of the LRT statistic, $\lambda_N := -2\big(\sup_{\boldsymbol{\theta}\in\boldsymbol{\Omega}_0} l_N(\boldsymbol{\theta}) - \sup_{\boldsymbol{\theta}\in\boldsymbol{\Omega}} l_N(\boldsymbol{\theta})\big)$, will be $\chi^2(p-r)$. However, when $\boldsymbol{\theta}_0$ is a boundary point of $\boldsymbol{\Omega}$, the regularity condition is not satisfied and the conventional Chi-squared limiting distribution does not hold either.

In Self and Liang (1987), the authors studied the nonstandard tests where the parameter of the null model is on the boundary of the parameter space. It is shown that when some of the true parameter values are on the boundary of the parameter space, under certain regularity conditions, the limiting distribution of the LRT statistic is the same as the distribution of the projection of the Gaussian random variable onto the region of admissible values for the mean. Specifically, both the whole parameter space $\boldsymbol{\Omega}$ and the null parameter space $\boldsymbol{\Omega}_0$ are assumed to be regular enough to be approximated by cones with vertices at the true parameter $\boldsymbol{\theta}_0$, which is defined as below.

Definition 4.3.1. The set $\boldsymbol{\Omega} \subset \mathbb{R}^p$ is approximated at $\boldsymbol{\theta}_0$ by a cone with vertex at $\boldsymbol{\theta}_0$, $C_{\boldsymbol{\Omega}}$, if

$$(1)\ \inf_{\boldsymbol{x}\in C_{\boldsymbol{\Omega}}} ||\boldsymbol{x}-\boldsymbol{y}|| = o(||\boldsymbol{y}-\boldsymbol{\theta}_0||), \quad \forall \boldsymbol{y} \in \boldsymbol{\Omega},$$

$$(2)\ \inf_{\boldsymbol{y}\in \boldsymbol{\Omega}} ||\boldsymbol{x}-\boldsymbol{y}|| = o(||\boldsymbol{x}-\boldsymbol{\theta}_0||), \quad \forall \boldsymbol{x} \in C_{\boldsymbol{\Omega}}.$$

When the model is identifiable, with further regularity conditions (see Section 1 in Self and Liang (1987) for details), the following asymptotic distribution of the LRT statistic is derived.

Theorem 4.3.2 (Self and Liang, 1987)**.** *Let $\boldsymbol{Z}$ be a random variable with a multivariate Gaussian distribution with mean $\boldsymbol{\theta}_0$ and covariance matrix $I^{-1}(\boldsymbol{\theta}_0)$, where $I(\boldsymbol{\theta}) = N^{-1}I_N(\boldsymbol{\theta})$ and $I_N(\boldsymbol{\theta})$ is the second derivative of the log-likelihood function $l_N(\boldsymbol{\theta})$. Let $C_{\boldsymbol{\Omega}_0}$ and $C_{\boldsymbol{\Omega}}$ be non-empty cones approximating $\boldsymbol{\Omega}_0$ and $\boldsymbol{\Omega}$ at $\boldsymbol{\theta}_0$, respectively. Then the asymptotic distribution of the likelihood ratio test statistic is the same as the distribution of the likelihood ratio test of $\boldsymbol{\theta} \in C_{\boldsymbol{\Omega}_0}$ versus the alternative $\boldsymbol{\theta} \in C_{\boldsymbol{\Omega}}$ based on a single realization $\boldsymbol{Z}$ when $\boldsymbol{\theta} = \boldsymbol{\theta}_0$.*

Following Self and Liang (1987), the asymptotic representation of the LRT statistic given by Theorem 4.3.2 can be written as

$$\sup_{\boldsymbol{\theta}\in C_{\boldsymbol{\Omega}}-\boldsymbol{\theta}_0} \{-(\boldsymbol{Z}-\boldsymbol{\theta})^\top I(\boldsymbol{\theta}_0)(\boldsymbol{Z}-\boldsymbol{\theta})\} - \sup_{\boldsymbol{\theta}\in C_{\boldsymbol{\Omega}_0}-\boldsymbol{\theta}_0} \{-(\boldsymbol{Z}-\boldsymbol{\theta})^\top I(\boldsymbol{\theta}_0)(\boldsymbol{Z}-\boldsymbol{\theta})\}, \qquad (4.2)$$

where $\boldsymbol{Z}$ has a multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $I^{-1}(\boldsymbol{\theta}_0)$. We can further rewrite it as

$$\inf_{\boldsymbol{\theta}\in\tilde{C}_0} ||\tilde{\boldsymbol{Z}}-\boldsymbol{\theta}||^2 - \inf_{\boldsymbol{\theta}\in\tilde{C}} ||\tilde{\boldsymbol{Z}}-\boldsymbol{\theta}||^2, \qquad (4.3)$$

where $\tilde{C} = \{\tilde{\boldsymbol{\theta}} : \tilde{\boldsymbol{\theta}} = \Lambda^{1/2}P^T\boldsymbol{\theta},\ \forall\ \boldsymbol{\theta} \in C_{\boldsymbol{\Omega}} - \boldsymbol{\theta}_0\}$, $\tilde{C}_0 = \{\tilde{\boldsymbol{\theta}} : \tilde{\boldsymbol{\theta}} = \Lambda^{1/2}P^T\boldsymbol{\theta},\ \forall\ \boldsymbol{\theta} \in C_{\boldsymbol{\Omega}_0} - \boldsymbol{\theta}_0\}$, $\tilde{\boldsymbol{Z}}$ follows a multivariate Gaussian distribution with mean $\mathbf{0}$ and the identity covariance matrix, and $P\Lambda P^T$ represents the spectral decomposition of $I(\boldsymbol{\theta}_0)$. Therefore, after the orthogonal transformation, the distribution in equation (4.2) can be computed using the standard Gaussian distribution.

This result provides a promising direction for the hypothesis testing in the HLAM setting, and we consider a simple example in Example IV.8.

Example IV.8. *Consider the DINA model with two latent attributes. Suppose that we want to test whether the first attribute is a prerequisite for the second attribute,* *that is, the hierarchical structure $\mathcal{E}_0 = \{1 \to 2\}$. Assume that the identifiability conditions in Proposition 4.2.1 are satisfied. The model parameters include the proportion parameters and item parameters $\{\pi_{(0,0)},\ \pi_{(0,1)},\ \pi_{(1,0)},\ \pi_{(1,1)},\ \theta_j^+,\ \theta_j^-,\ j = 1, \ldots, J\}$, so the total number of parameters is $3 + 2 \times J$, noting that the proportion parameter vector $\boldsymbol{\pi} = (\pi_{(0,0)},\ \pi_{(0,1)},\ \pi_{(1,0)},\ \pi_{(1,1)})$ lies in the 3-simplex. To test the hierarchy $\mathcal{E}_0$, it is equivalent to test*

$$H_0 : \pi_{(0,1)} = 0 \quad v.s. \quad H_1 : \pi_{(0,1)} \neq 0.$$

Therefore we have one parameter of interest that has a true value on the boundary and $2 + 2 \times J$ nuisance parameters with true values, not on the boundary. After an orthogonal transformation, we have $\tilde{C} = [0, \infty) \times \mathbb{R}^{2+2\times J}$ and $\tilde{C}_0 = \{0\} \times \mathbb{R}^{2+2\times J}$ and thus the asymptotic distribution of the LRT statistic is reduced to

$$\tilde{Z}_1^2 \cdot I(\tilde{Z}_1 > 0),$$

where $\tilde{Z}_1$ follows a standard univariate gaussian distribution. Therefore the limiting distribution of the LRT statistic is a mixture of Chi-squared distribution $\frac{1}{2}\chi_0^2 + \frac{1}{2}\chi_1^2$.

In Example IV.8, we derive the closed-form of the limiting distribution of the LRT statistic in the DINA model with two latent attributes and a linear hierarchy. In this example, we take the advantage of the fact that there is only one boundary parameter and it occurs as the parameter of interest. However, the asymptotic distribution of the LRT statistic in fact becomes considerably more complicated if there are more latent attributes and more complex hierarchical structures. Moreover, even in the simple setting as in Example IV.8, the convergence may be very slow if the number of items J is large or the guessing parameter θ_j^- and slipping parameter $1 - \theta_j^+$ are close to the boundary, as illustrated in Figure IV.1. Specifically, in Figure IV.1, we present the p-values under various settings for Example IV.8. In the titles of these plots, we use

s_j to denote the slipping parameter $1-\theta_j^+$, and g_j to denote the guessing parameter θ_j^- for ease of presentation. The observed p-values are plotted by blue points, and the p-values for the reference distribution $\frac{1}{2}\chi_0^2+\frac{1}{2}\chi_1^2$ are plotted as the red lines. The first row in Figure IV.1 contains three plots of p-values with the same sample size and item parameters but different numbers of items. It is noted that when the number of items was small, the observed p-values were very close to those of the mixture Chi-squared limiting distribution. However, as the number of items increased, the gap between the observed p-values and the reference limiting distribution became larger. The second row in Figure IV.1 contains three plots of p-values with more extreme item parameters. Compared with the plots in the first row, it is shown that when the item parameters were close to the boundary, the convergence of the LRT statistic became much slower, and such testing tended to fail even with a large sample size $N=10,000$.

As pointed out in Self and Liang (1987), even though based on Theorem 4.3.2 we can derive the asymptotic distribution of the LRT statistic for any fixed $\boldsymbol{\theta}_0\in\boldsymbol{\Omega}_0$, this distribution is generally different for different $\boldsymbol{\theta}_0$. Moreover, these distributions typically vary over $\boldsymbol{\Omega}_0$ in a discontinuous way when some of the nuisance parameters may also be on the boundary. This discontinuity can affect the quality of the asymptotic approximation much. As in our example, when the slipping parameter $1-\theta_j^+$ and the guessing parameter θ_j^- in the DINA model are close to the boundary, the distribution with a given finite sample may be far away from the weighted Chi-squared mixture as described in Example IV.8.

We provide further insights on why the convergence was slow when the number of items was large or the item parameters were close to the boundaries as shown in Figure IV.1. The average log-likelihood of LAMs is given by $l_N/N := \sum_{i=1}^N \log\big(\sum_{\boldsymbol{\alpha}}\pi_{\boldsymbol{\alpha}}\mathbb{P}(\boldsymbol{R}_i\mid\boldsymbol{\alpha})\big)/N$. Consider Example IV.8 where we are interested in the hierarchy $\{1\to 2\}$ and want to test whether $\pi_{(0,1)}=0$. If we write $\pi_{(0,0)}=1-\sum_{\boldsymbol{\alpha}\neq(0,0)}\pi_{\boldsymbol{\alpha}}$, then

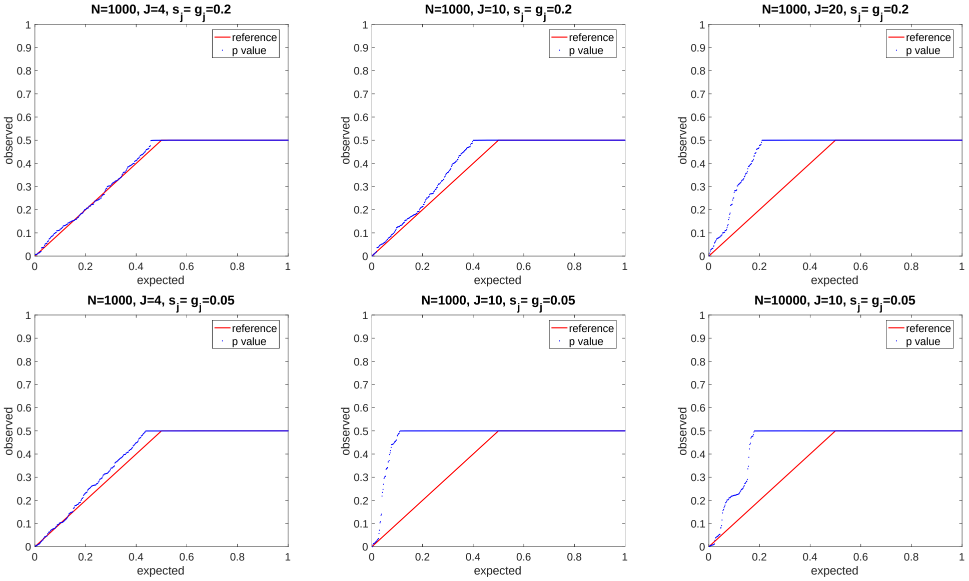


Figure IV.1: QQ-plots for Example IV.8 under various settings. s_j denotes the slipping parameter $1 - \theta_j^+$ and g_j denotes the guessing parameter θ_j^- for item j. The x-axis is the expected percentile of the p-values under the null hypothesis, and the y-axis is the percentile of the observed p-values. The observed p-values are plotted by blue points, and the p-values for the reference limiting distribution $\frac{1}{2}\chi_0^2 + \frac{1}{2}\chi_1^2$ are plotted as red lines. If the blue points are close to the red lines, it indicates that the empirical distribution of the observed p-values approximates the asymptotic distribution well.

$l_N/N = \sum_{i=1}^N \log\big((1-\sum_{\boldsymbol{\alpha}\neq(0,0)}\pi_{\boldsymbol{\alpha}})\mathbb{P}\big(\boldsymbol{R}_i \mid \boldsymbol{\alpha}=(0,0)\big) + \sum_{\boldsymbol{\alpha}\neq(0,0)}\pi_{\boldsymbol{\alpha}}\mathbb{P}(\boldsymbol{R}_i \mid \boldsymbol{\alpha})\big)/N$. The derivative of the log-likelihood w.r.t. $\pi_{(0,1)}$ becomes

$$\begin{aligned}
\frac{\partial l_N/N}{\partial \pi_{(0,1)}} &= \frac{1}{N}\sum_{i=1}^N \frac{\mathbb{P}\big(\boldsymbol{R}_i \mid \boldsymbol{\alpha}=(0,1)\big)-\mathbb{P}\big(\boldsymbol{R}_i \mid \boldsymbol{\alpha}=(0,0)\big)}{\sum_{\boldsymbol{\alpha}}\pi_{\boldsymbol{\alpha}}\mathbb{P}\big(\boldsymbol{R}_i \mid \boldsymbol{\alpha}\big)} \\
&= \frac{1}{N}\sum_{i=1}^N \frac{\mathbb{P}\big(\boldsymbol{R}_i \mid \boldsymbol{\alpha}=(0,1)\big)-\mathbb{P}\big(\boldsymbol{R}_i \mid \boldsymbol{\alpha}=(0,0)\big)}{\mathbb{P}(\boldsymbol{R}_i)} \\
&= \frac{1}{N}\sum_{i=1}^N \sum_{\boldsymbol{r}\in\{0,1\}^J} \mathbb{I}(\boldsymbol{R}_i=\boldsymbol{r})\frac{\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,1)\big)-\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,0)\big)}{\mathbb{P}(\boldsymbol{r})}. \qquad (4.4)
\end{aligned}$$

When the null hypothesis that $\pi_{(0,1)}=0$ is true, by the strong law of large number, we have

$$\begin{aligned}
\frac{\partial l_N/N}{\partial \pi_{(0,1)}}\Big|_{\pi_{(0,1)}=0} \xrightarrow{a.s.}\ & \mathbb{E}_0\Big[\sum_{\boldsymbol{r}\in\{0,1\}^J}\mathbb{I}(\boldsymbol{R}=\boldsymbol{r})\frac{\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,1)\big)-\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,0)\big)}{\mathbb{P}(\boldsymbol{r})}\Big] \\
&= \sum_{\boldsymbol{r}\in\{0,1\}^J}\Big(\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,1)\big)-\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,0)\big)\Big) \\
&= \sum_{\boldsymbol{r}\in\{0,1\}^J}\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,1)\big) - \sum_{\boldsymbol{r}\in\{0,1\}^J}\mathbb{P}\big(\boldsymbol{r} \mid \boldsymbol{\alpha}=(0,0)\big) \\
&= 0.
\end{aligned}$$

However, since the number of possible response patterns is $|\{0,1\}^J| = 2^J$ which grows exponentially with the number of items J, it requires an exponentially growing sample size to cover all the possible response patterns, and therefore the convergence can be slow when J is large.

Next, consider the case when the item parameters are close to the boundary. Note that

$$\begin{aligned}\mathbb{P}\big(\boldsymbol{R} \mid \boldsymbol{\alpha}\big) &= \prod_{j=1}^{J} \mathbb{P}\big(R_j \mid \boldsymbol{\alpha}\big) \\ &= \prod_{j=1}^{J} \Big((\theta_j^-)^{1-\Gamma_{j,\alpha}} (\theta_j^+)^{\Gamma_{j,\alpha}}\Big)^{R_j} \Big((1-\theta_j^-)^{1-\Gamma_{j,\alpha}} (1-\theta_j^+)^{\Gamma_{j,\alpha}}\Big)^{1-R_j}.\end{aligned}$$

When the item parameters are very close to the boundaries, that is, θ_j^- and $1-\theta_j^+$ are very close to 0, the model becomes near deterministic. For simplicity, let $1-\theta_j^+ = \theta_j^- = \delta$ which is very close to 0 for all $j \in [J]$. Then

$$\begin{aligned}\mathbb{P}(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha}) &= \prod_{j=1}^{J} \Big(\delta^{(1-\Gamma_{j,\boldsymbol{\alpha}})} (1-\delta)^{\Gamma_{j,\boldsymbol{\alpha}}}\Big)^{r_j} \Big((1-\delta)^{(1-\Gamma_{j,\boldsymbol{\alpha}})} \delta^{\Gamma_{j,\boldsymbol{\alpha}}}\Big)^{1-r_j} \\ &= \prod_{r_j=1} \delta^{(1-\Gamma_{j,\boldsymbol{\alpha}})} (1-\delta)^{\Gamma_{j,\boldsymbol{\alpha}}} \prod_{r_j=0} (1-\delta)^{(1-\Gamma_{j,\boldsymbol{\alpha}})} \delta^{\Gamma_{j,\boldsymbol{\alpha}}} \\ &= \delta^{\sum_{r_j=1}(1-\Gamma_{j,\boldsymbol{\alpha}}) + \sum_{r_j=0} \Gamma_{j,\boldsymbol{\alpha}}} \cdot (1-\delta)^{\sum_{r_j=1} \Gamma_{j,\boldsymbol{\alpha}} + \sum_{r_j=0}(1-\Gamma_{j,\boldsymbol{\alpha}})}.\end{aligned}$$

For $\boldsymbol{r} = \boldsymbol{\Gamma}_{\cdot,\boldsymbol{\alpha}}$, we have $\mathbb{P}(\boldsymbol{R} = \boldsymbol{\Gamma}_{\cdot,\boldsymbol{\alpha}} \mid \boldsymbol{\alpha}) = (1-\delta)^J$. And for any $\boldsymbol{r} \neq \boldsymbol{\Gamma}_{\cdot,\boldsymbol{\alpha}}$, $\delta^J \leq \mathbb{P}(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha}) \leq \delta$. Moreover, when $\pi_{(0,1)} = 0$, since $\mathbb{P}(\boldsymbol{R} = \boldsymbol{r}) = \sum_{\boldsymbol{\alpha} \neq (0,1)} \pi_{\boldsymbol{\alpha}} \mathbb{P}(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha})$, we have

$$\mathbb{P}(\boldsymbol{R} = \boldsymbol{\Gamma}_{\cdot,\boldsymbol{\alpha}}) \geq \pi_{\boldsymbol{\alpha}} (1-\delta)^J, \text{ for } \boldsymbol{\alpha} \in \mathcal{A} = \{(0,0),(1,0),(1,1)\},$$

$$\mathbb{P}\big(\boldsymbol{R} \neq \boldsymbol{\Gamma}_{\cdot,\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathcal{A}\big) \leq 1 - (1-\delta)^J \to 0 \text{ as } \delta \to 0.$$

Therefore from the above discussions, when $\pi_{(0,1)} = 0$, the probability mass is concentrated around three response patterns $\boldsymbol{\Gamma}_{\cdot,\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathcal{A} = \{(0,0),(1,0),(1,1)\}$. For the terms in the RHS of (4.4), when $\boldsymbol{r} = \Gamma_{\cdot,(0,0)}$, we have

$$\begin{aligned}&\big[\mathbb{P}\big(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha} = (0,1)\big) - \mathbb{P}\big(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha} = (0,0)\big)\big]\big(\mathbb{P}(\boldsymbol{R} = \boldsymbol{r})\big)^{-1} \\ &\in \Big[\frac{\delta^J - (1-\delta)^J}{(1-\delta)^J \pi_{(0,0)}},\ \frac{\delta - (1-\delta)^J}{(1-\delta)^J \pi_{(0,0)} + \delta(1-\pi_{(0,0)})}\Big] \longrightarrow -1/\pi_{(0,0)} \text{ as } \delta \to 0.\end{aligned}$$

When $\boldsymbol{r} = \Gamma_{\cdot,(1,0)}$ (or $\Gamma_{\cdot,(1,1)}$), we have

$$\begin{aligned}&\big[\mathbb{P}\big(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha} = (0,1)\big) - \mathbb{P}\big(\boldsymbol{R} = \boldsymbol{r} \mid \boldsymbol{\alpha} = (0,0)\big)\big]\big(\mathbb{P}(\boldsymbol{R} = \boldsymbol{r})\big)^{-1} \\ &\in \Big[\frac{\delta^J - \delta}{(1-\delta)^J \pi_{(1,0)}},\ \frac{\delta - \delta^J}{(1-\delta)^J \pi_{(1,0)} + \delta(1-\pi_{(1,0)})}\Big] \longrightarrow 0 \text{ as } \delta \to 0.\end{aligned}$$

Therefore the terms in the RHS of (4.4) also concentrate around two points, $-1/\pi_{(0,0)}$ and 0, making the convergence slow since more data points are needed to have it converge to 0.

Based on the above discussions about the asymptotic behaviors of the LRT statistic under the nonstandard conditions and in the HLAM setting, it has been shown that even in the simple setting where we could derive a closed form of the limiting distribution, the convergence can be very slow. Moreover, the asymptotic distribution of the LRT will be much more complicated if we have more latent attributes and more complex hierarchical structures. Therefore, it is not practical to use the theoretical limiting distribution of the LRT statistic to test latent hierarchical structures in LAMs, especially considering that the number of test items is usually relatively large (e.g. more than 20).

4.3.2 Bootstrap and Numerical Studies

From the discussions about the LRT in Section 4.3.1, we learn that the limiting distribution of the LRT statistic under latent hierarchical structures can be very complicated and the convergence can be slow when the number of items is large or the item parameters are close to the boundary even in simple settings. To overcome these difficulties, we propose to use the bootstrap method as an alternative to the asymptotic limiting distribution method. The bootstrap method (Efron, 1979) has been shown to be successful in many nonstandard situations. The basic idea of bootstrap is treating inference of the true probability distribution, given the original data, as being analogous to the inference of the empirical distribution, given the resampled data. If the empirical distribution is a reasonable approximation to the true distribution, then the bootstrap method will provide good inferences.

In this section, we consider two different bootstrap procedures: nonparametric bootstrap and parametric bootstrap. The idea of nonparametric bootstrap is to simulate data from the empirical distribution by directly resampling from the original data. To be specific, in nonparametric bootstrap, we draw samples of the same size from the original data with replacement. Then the statistic of interest is computed based on the resampled data set and we repeat this routine many times. The steps for nonparametric bootstrap are summarized as below:

Step 1. Initially estimate the model with the specified hierarchy under the null hypothesis, and the model under the alternative hypothesis (without the null hypothesis hierarchy constraints), and calculate the LRT statistic.

Step 2. Draw a sample of the same size with replacement from the original data and calculate the LRT statistic.

Step 3. Repeat Step 2 independently many times and estimate the distribution of the LRT statistic.

Step 4. Estimate the p-value by comparing the distribution obtained in Step 3 with the LRT statistic obtained in Step 1. Then this p-value is used to determine whether the null model with the specified null hierarchy should be rejected in favor of the model without the hierarchical constraints.

The idea of parametric bootstrap is to simulate data based on good estimates of distribution parameters, often by maximum likelihood. In parametric bootstrap, a parametric model is fitted to the original data, and samples are drawn from this fitted model. The steps for parametric bootstrap are similar to those of nonparametric bootstrap except for Step 2:

Step 2*. Based on the estimates of the model with specified hierarchy from step 1, generate a bootstrap sample from the fitted model and calculate the LRT statistic.

Next, we conduct comprehensive simulation studies to compare parametric bootstrap and nonparametric bootstrap for testing latent hierarchical structures under various settings. We considered four different hierarchical structures shown in Figure II.1. For the data generating process, we considered the DINA model and the GDINA model respectively. For both models, we included three different sample sizes ($N =$ 200, 500, or 1000) and the number of items was set to 30 ($J = 30$). In terms of uncertainty, two levels of guessing and slipping parameters in the DINA model were included ($\theta_j^- = 1 - \theta_j^+ = 0.1$ or 0.2 for $j \in [J]$). For the GDINA model, we also considered two different uncertainty levels, where the highest item parameter was 0.9 or 0.8, and the lowest item parameter was 0.1 or 0.2. The other item parameters in between were equally spaced. To satisfy testability conditions, the Q-matrix contained two identity sub-matrices and the remaining items were randomly generated. For each scenario, we performed 500 independent repetitions and in each repetition, we generated bootstrap samples 500 times. To fit the models under the null and alternative hypotheses, we used R package `CDM`.

The type I errors with significance level $\alpha = 0.05$ under different settings are plotted in Appendix B. There we also provide corresponding error bars for uncertainty quantification of the Monte Carlo errors. The naïve Chi-squared test is included for a comprehensive comparison. From the plots, we can see that the type I errors for parametric bootstrap were around 0.05 in most cases and therefore parametric bootstrap controlled the type I errors generally well. By contrast, nonparametric bootstrap was too conservative and the type I errors for nonparametric bootstrap were very close to 0. In terms of the naïve Chi-squared test, it was also very conservative in most cases even though the type I errors for the GDINA model with larger noises under the "unstructured" hierarchy were closer to the significance level of 0.05.

To further examine the behaviors of the testing procedures, the QQ plots for p-values under the null hypothesis are provided in Figure IV.2 and Figure IV.4. For presentation brevity, we only present results of four hierarchies with the same noise level and sample size. More comprehensive simulation results are presented in Appendix B. It is known that under the null hypothesis, the p-values should follow a uniform distribution on $[0, 1]$. In the QQ-plots, if the points are lying closer to the identity line, it indicates that it approximates the uniform distribution better. From Figure IV.2 and IV.4, one can see that under the null hypothesis, the p-values of parametric bootstrap approximated the uniform distribution on $[0, 1]$ very well in almost all the settings. By contrast, the p-values of the nonparametric bootstrap and the naïve Chi-squared test were far away from the uniform distribution, indicating that these testing procedures are not reliable. We also conducted power analysis where all the latent attribute profiles existed in the data generation process. The true proportion parameters were equally assigned. The QQ plots for p-values under the alternative hypothesis are shown in Figure IV.3 and Figure IV.5 respectively. To have more power, we expect the p-values to be small so that we would reject the null hypothesis. Therefore in the QQ-plots, the closer to 0 the points are, the more powerful the test is. From Figure IV.3 and IV.5, one can see that the p-values of parametric bootstrap and the naïve Chi-squared test were almost 0 and therefore the power was close to 1. However, the p-values of the nonparametric bootstrap were close to or above 0.5, which means we would not reject the null hypothesis, making the power almost 0. Taking both the type I error and power into consideration, the parametric bootstrap outperformed the other two testing procedures.

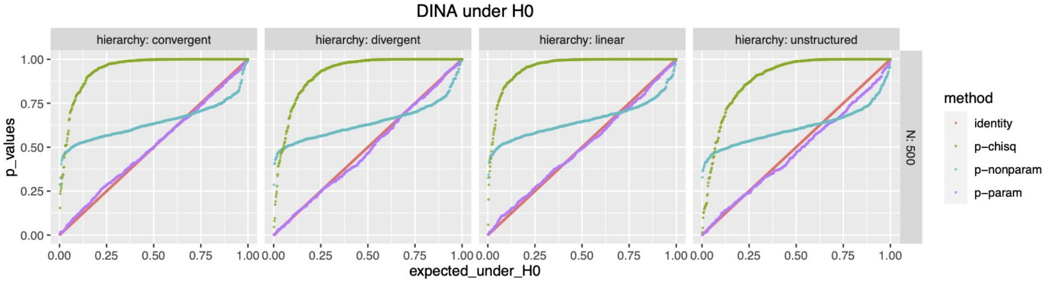


Figure IV.2: QQ-plots for the p-values of the DINA model under the null hypothesis where $\theta_j^+ = 0.8,\ \theta_j^- = 0.2$ that corresponds to the case with high noises.

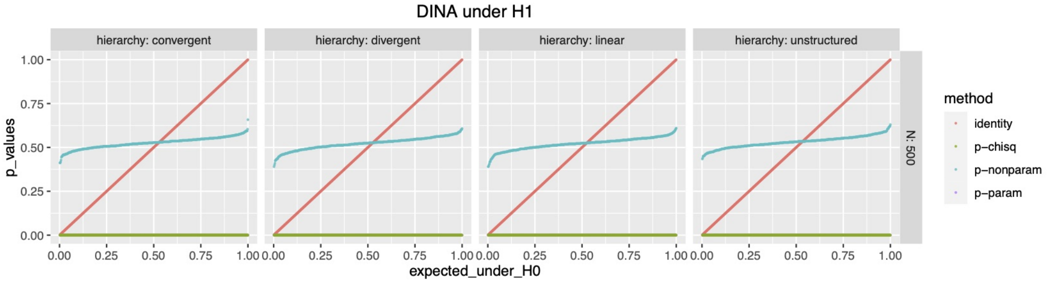


Figure IV.3: QQ-plots for the p-values of the DINA model under the alternative hypothesis where $\theta_j^+ = 0.8,\ \theta_j^- = 0.2$ corresponding to high noises. The expected quantiles are the expected quantiles of the p-values under the null hypothesis, that is, the uniform distribution on [0,1].

In order for the nonparametric bootstrap to work, the empirical distribution of the sample data should be close to the true distribution, which may not hold for the cases here especially when the number of items is relatively large. As we discussed in Section 4.3.1, the total number of possible response patterns 2^J grows exponentially with the number of items J. Therefore, in nonparametric bootstrap, we need a very large sample size to cover all the possible response patterns, which may explain the failures of nonparametric bootstrap. On the contrary, in parametric bootstrap, we first fit a model from the original data and resample data from the fitted model, which incorporates the variance in data generation better and thus makes parametric bootstrap perform better. Note that in Templin and Bradshaw (2014), using the naïve

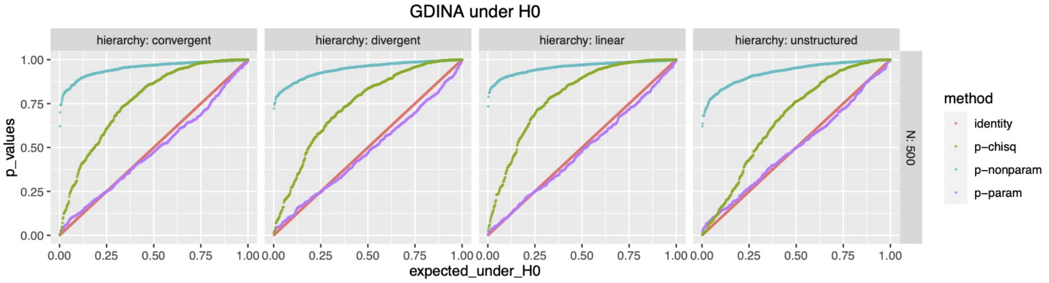


Figure IV.4: QQ-plots for the p-values of the GDINA model under the null hypothesis where $\theta_j^+ = 0.8,\ \theta_j^- = 0.2$ corresponding to the case with high noises.

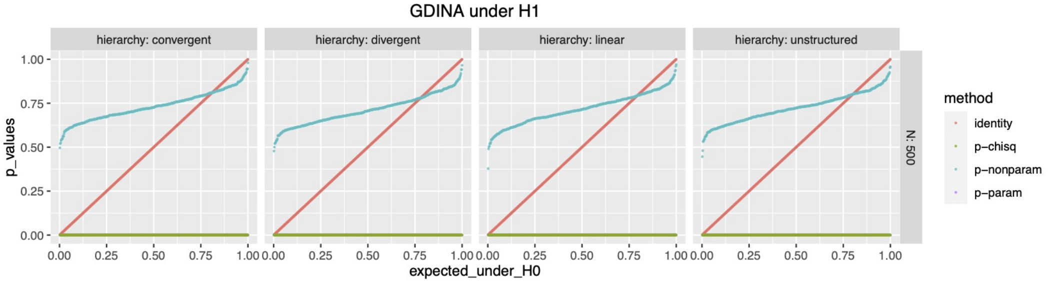


Figure IV.5: QQ-plots for the p-values of the GDINA model under the alternative hypothesis where $\theta_j^+ = 0.8,\ \theta_j^- = 0.2$ corresponding to high noises. The expected quantiles are the expected quantiles of the p-values under the null hypothesis, that is, the uniform distribution on [0,1].

Chi-squared test, the authors concluded that "*the DINA model cannot detect attribute hierarchies*". However, through our comprehensive simulation, by using parametric bootstrap, the attribute hierarchies can also be well detected in the DINA model.

In summary, based on our discussions about the failure of the limiting distribution of LRT in Section 4.3.1 and the simulation results in Section 4.3.2, we recommend using parametric bootstrap to perform hypothesis testing for latent hierarchical structures in LAMs.

4.4 Real Data Analysis

In this section, we perform hypothesis testing procedures on the Examination for the Certificate of Proficiency in English (ECPE) data, which has been studied using the proposed method in Chapter II where we learned the latent and hierarchical structures from the observed responses. Three target attributes are considered, including morphosyntactic rules (α_1), cohesive rules (α_2) and lexical rules (α_3). The Q-matrix of the ECPE data is given in Appendix B. Since the Q-matrix contains four identity submatrices, the testability conditions are satisfied. A linear hierarchical structure $\mathcal{E}_0 = \{\alpha_3 \to \alpha_2 \to \alpha_1\}$ is often considered in literature such as Templin and Bradshaw (2014).

Under the linear hierarchy $\mathcal{E}_0$, the latent attribute profile set is $\mathcal{A}_0 = \{(0,0,0),(0,0,1),(0,1,1),(1,1,1)\}$. Under the null hypothesis, we fitted a GDINA model with the profile set $\mathcal{A}_0$, and under the alternative, we fitted a saturated GDINA model with all the possible attribute profiles. We generated bootstrap samples 1000 times in parametric bootstrap and nonparametric bootstrap respectively. The p-value obtained from parametric bootstrap was 0.041, while the p-value obtained from nonparametric bootstrap was 0.952. Moreover, we also calculated the p-value corresponding to the naïve test using the conventional Chi-squared limiting distribution and got the p-value of 0.02. If we set the significance level to be 0.05, then by parametric bootstrap, we would reject the null hypothesis and conclude the linear hierarchy does not present in this data set; while if the significance level is set to be 0.01, we would not reject the null hypothesis and conclude there is such a linear attribute hierarchy. This conclusion is consistent with that in Templin and Bradshaw (2014). For both significance levels, the nonparametric bootstrap does not reject the null hypothesis.

To conduct a more comprehensive study of the linear hierarchies among the three target attributes, we further tested each linear hierarchy relationship separately and examined which one is the strongest. In particular, we considered the following various test settings:

- $\mathrm{H}_0 : \mathcal{E}_0 = \{\alpha_3 \to \alpha_2\}$ vs. H_1: no hierarchical structure $\mathcal{E}_1 = \emptyset$;

- $\mathrm{H}_0 : \mathcal{E}_0 = \{\alpha_2 \to \alpha_1\}$ vs. H_1: no hierarchical structure $\mathcal{E}_1 = \emptyset$;

- $\mathrm{H}_0 : \mathcal{E}_0 = \{\alpha_3 \to \alpha_1\}$ vs. H_1: no hierarchical structure $\mathcal{E}_1 = \emptyset$;
- $\mathrm{H}_0 : \mathcal{E}_0 = \{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\mathrm{H}_1 : \mathcal{E}_1 = \{\alpha_3 \to \alpha_2\}$;
- $\mathrm{H}_0 : \mathcal{E}_0 = \{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\mathrm{H}_1 : \mathcal{E}_1 = \{\alpha_2 \to \alpha_1\}$;
- $\mathrm{H}_0 : \mathcal{E}_0 = \{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\mathrm{H}_1 : \mathcal{E}_1 = \{\alpha_3 \to \alpha_1\}$;

The resulting p-values for parametric bootstrap, nonparametric bootstrap and the naïve Chi-squared test under different settings are presented in Table 4.1. From the table, we can see that among all the settings, the p-values of nonparametric bootstrap are very large and therefore we do not reject the null hypotheses. This is also consistent with our simulation study where we find nonparametric bootstrap is more conservative. The p-values of parametric bootstrap are much smaller than those of nonparametric bootstrap. If we set the significance level to be 0.05, parametric bootstrap does not reject the null except for the settings "$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\emptyset$" and "$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\{\alpha_3 \to \alpha_1\}$". The p-values of the naïve Chi-squared test are of the similar scales of those in parametric bootstrap, and if the significance level is set to 0.05, the naïve Chi-squared test does not reject the null except for the settings "$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\{\alpha_3 \to \alpha_2\}$" and "$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\emptyset$". However, as we have shown in our simulation results, since the true limiting distribution of the LRT statistic is no longer the conventional Chi-squared distribution, the naïve test is not reliable.

For testing a single linear hierarchy relationship versus none hierarchical structure (i.e., the second to the fourth tests in Table 4.1), since the alternative hypothesis is the same while the null hypothesis varies, a larger p-value suggests a stronger prerequisite relationship. Therefore based on the results in Table 4.1 that the p-value for "$\{\alpha_3 \to \alpha_1\}$ vs. $\emptyset$" is the largest among these three tests, we can see that the prerequisite relationship between the third attribute and the first attribute is the strongest. Similarly, for testing the whole linear hierarchical $\mathcal{E}_0$ versus a single linear

Setting	Para-boot	Nonpara-boot	Chi-squared
$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\emptyset$	0.041	0.952	0.020
$\{\alpha_3 \to \alpha_2\}$ vs. $\emptyset$	0.052	0.511	0.072
$\{\alpha_2 \to \alpha_1\}$ vs. $\emptyset$	0.057	0.722	0.064
$\{\alpha_3 \to \alpha_1\}$ vs. $\emptyset$	0.169	0.954	0.066
$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\{\alpha_3 \to \alpha_2\}$	0.098	0.962	0.047
$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\{\alpha_2 \to \alpha_1\}$	0.073	0.906	0.052
$\{\alpha_3 \to \alpha_2 \to \alpha_1\}$ vs. $\{\alpha_3 \to \alpha_1\}$	0.026	0.625	0.051

Table 4.1: p-values for different testing settings.

hierarchy (i.e., the last three tests in Table 4.1), since now the null hypothesis is the same and the alternative hypothesis varies, a smaller p-value indicates a stronger pre-requisite relationship. In this case, the prerequisite relationship between the third attribute and the first attribute seems still the strongest.

4.5 Discussion

In this chapter, we consider the hypothesis testing problem for latent hierarchical structures in latent attribute models. We first discuss the testability issues and present sufficient conditions for testability. Under the testability conditions, we study the asymptotic properties of the likelihood ratio test and show the practical difficulties of directly using the limiting distribution of the LRT statistic to test latent hierarchies. We then compare two resampling-based testing procedures including parametric bootstrap and nonparametric bootstrap through comprehensive simulations under different settings and recommend using parametric bootstrap for testing latent hierarchical structures.

We mainly focus on the hypothesis testing where the hierarchical structure is fully specified and all the latent attribute profiles that respect the hierarchy exist in the population. In many applications, the number of latent attributes K could be large, leading to a high-dimensional space for all the possible configurations of the attributes, where the number of potential attribute profiles can be even larger than the sample size. For scientific interpretability and practical use, it is often assumed that not all the possible attribute profiles exist in the population. In such cases, to test hierarchical structures, we may perform the selection of significant latent attribute profiles first and then conduct testing procedures.

This chapter proposes to use parametric bootstrap, which is a resampling-based procedure and can be computationally expensive, especially for large-scale data sets. Therefore it would be useful to develop more efficient testing procedures. Moreover, further theoretical results are needed to better characterize the asymptotic distribution of the likelihood ratio test with the presence of latent variables and complex constraint structures as in HLAMs.

CHAPTER V

Bridging Parametric and Nonparametric Methods

5.1 Introduction

As we introduced in the previous chapters, several parametric models for cognitive diagnosis have been developed and widely applied in practice. Popular examples include the deterministic input, noisy "and" gate (DINA) model (Junker and Sijtsma, 2001), the deterministic input, noisy "or" gate (DINO) model (Templin and Henson, 2006), the reduced reparameterized unified model (Reduced RUM; Hartz, 2002), the general diagnostic model (GDM; von Davier, 2005), the log-linear CDM (LCDM; Henson et al., 2009), and the generalized DINA model (GDINA; de la Torre, 2011). To estimate these parametric models, estimators maximizing the marginal likelihood or joint likelihood functions have been employed (e.g., Chiu et al., 2016; de la Torre, 2009).

Parametric LAMs, such as the DINA or DINO model, invoke certain parametric assumptions about the item response functions. As pointed out in Chiu and Douglas (2013), such assumptions may raise validity concerns about the assumed model and the underlying process. As an alternative, some researchers have explored nonparametric methods for assigning subjects to latent groups without relying on parametric model assumptions. For example, Chiu and Douglas (2013) proposed the nonparametric classification (NPC) method, where a subject is classified to its closet latent group by comparing the observed responses with ideal responses either from the DINA or DINO model. Its generalization, the general NPC (GNPC) method proposed by Chiu et al. (2018), uses the weighted average of ideal responses from the DINA and DINO models to accommodate more general settings. Consistency results for the NPC and the GNPC methods were established by Wang and Douglas (2015) and Chiu and Köhn (2019a), respectively. Simulation results show that, compared to parametric methods, nonparametric methods tend to perform better when the sample sizes are not sufficiently large to provide reliable maximum likelihood estimates.

Even though the aforementioned parametric and nonparametric methods have been used in many cognitive diagnosis applications, the relationship between these two families of methods have not been explicitly discussed in the literature. Although seemingly divergent from the surface, these frameworks are in fact closely related. In this chapter, we propose a unified estimation framework for cognitive diagnosis that subsumes both parametric and nonparametric methods. In the proposed framework, we use a general loss function to measure the distance between a subject's responses and the centroid of a latent class. By using different loss functions, the method can assume different parametric and nonparametric forms. Under the general framework, we further develop a unified iterative joint estimation algorithm, as well as establish the consistency properties of the corresponding estimators. Finally, we conduct comprehensive simulation studies to compare different parametric and nonparametric methods under a variety of settings and provide relevant practical recommendations accordingly.

The rest of the chapter is organized as follows. Section 5.2 gives a brief review of nonparametric methods in cognitive diagnosis assessment. Section 5.3 introduces the proposed general estimation framework with several illustrative examples. Section 5.4 presents the consistency results of the proposed method, and Section 5.5 presents the simulation results. Finally, Section 5.6 discusses some future extensions, whereas proofs of the main theorems are reported in the Appendix C.

5.2 Nonparametric Methods

Before introducing our proposed estimation framework, we give a brief review of nonparametric methods that are widely used in the cognitive diagnosis literature.

As the name suggests, nonparametric methods no longer depend on parametric model assumptions. Instead of modeling item response functions, nonparametric methods directly classify the subjects into latent classes by minimizing the distance between the subject's observed item responses and the centers of the latent classes. Two popular examples of nonparametric methods are the NPC and the GNPC methods, which compare the subject's observed item responses to the so-called ideal response vectors of each proficiency class. Different cognitive diagnosis models define the ideal response vectors differently. For example, as specified in equations (2.2) or (2.3), the ideal response in the DINA or DINO model will be 1 only if the subject possesses all the required attributes or one of the required attributes, respectively. In the following, we give a brief introduction to the NPC and the GNPC methods. Please refer to Chiu and Köhn (2019b), Chiu and Douglas (2013) and Chiu et al. (2018) for more details.

For the NPC method, we use $M = 2^K$ to denote the total number of proficiency latent classes (i.e., attribute profiles), and for $m = 1, \ldots, M$, we write $\boldsymbol{\eta}_m = (\eta_{1,m}, \eta_{2,m}, \ldots, \eta_{J,m})$ as the ideal response vector for the mth proficiency-class, where $\eta_{j,m}$ can be the DINA or DINO ideal response. We use $\boldsymbol{r}_i$ to denote the response vector of subject i to J items. Given the ideal response vectors for each proficiency class, a subject is classified to the closest proficiency class that minimizes the distance between the subject's observed responses and the ideal responses:

$$\hat{\boldsymbol{\alpha}}_i = \underset{m \in \{1,2,\ldots,M\}}{\arg\min}\; d(\boldsymbol{r}_i, \boldsymbol{\eta}_m),$$

where $d(\cdot)$ is a distance function. For example, in Chiu and Douglas (2013), they used the Hamming distance:

$$d_H(\boldsymbol{r}, \boldsymbol{\eta}) = \sum_{j=1}^{J} |r_j - \eta_j|.$$

In the NPC method, the ideal responses are either the DINA ideal responses or the DINO ideal responses, which are all binary; thus, the absolute difference will be 0 if the observed response is equal to the ideal response, and 1 otherwise. Moreover, because the observed and the ideal responses are all binary, the L_2 distance will lead to the same results as the Hamming distance in the NPC method.

Due to its dependence on the DINA or DINO model assumptions, which define two extreme relations between $\boldsymbol{q}$ and $\boldsymbol{\alpha}$, the NPC method may not be sufficiently flexible. The GNPC method addresses this issue by considering a more general ideal response that represents a weighted average of the ideal responses of the DINA and DINO models, as in:

$$\eta_{j,m}^{(w)} = w_{j,m}\eta_{j,m}^{\text{DINA}} + (1 - w_{j,m})\eta_{j,m}^{\text{DINO}},$$

where $w_{j,m}$ is the weight for the jth item and the mth proficiency class. We use $\boldsymbol{\eta}_m^{(w)} = (\eta_{1,m}^{(w)}, \ldots, \eta_{J,m}^{(w)})$ to denote the weighted ideal response vector for the mth proficiency class in the GNPC method. To get the estimates of the weights, Chiu et al. (2018) proposed to minimize the L_2 distance between the responses to item j and the weighted ideal responses $\eta_{j,m}^{(w)}$:

$$d_{jm} = \sum_{i \in C_m} \left(r_{ij} - \eta_{j,m}^{(w)}\right)^2, \tag{5.1}$$

where $\{C_m\}_{m=1}^{M}$ is the partition of the subjects into M proficiency classes. Minimizing

(5.1) leads to

$$\hat{w}_{j,m} = 1 - \bar{r}_{j,C_m}, \quad \hat{\eta}^{(w)}_{j,m} = \bar{r}_{j,C_m},$$

where $\bar{r}_{j,C_m} = |C_m|^{-1} \sum_{i \in C_m} r_{ij}$, the mean of the jth item responses for subjects in the mth proficiency class, and $|C_m|$ is the number of subjects in C_m. Because the true memberships are unknown, they proposed to iteratively estimate the memberships and the ideal response vectors. Specifically, starting with an initial partition of the subjects, the ideal response vectors are chosen to minimize the L_2 distance $\sum_{m=1}^{M} \sum_{i \in C_m} \sum_{j=1}^{J} (r_{ij} - \eta^{(w)}_{j,m})^2$. The memberships of the subjects are then determined by minimizing the L_2 distance between the observed responses of a subject and the ideal response vectors estimated from the former step, as in, $\hat{\boldsymbol{\alpha}}_i = \arg\min_{m \in \{1,2,\ldots,M\}} d\big(\boldsymbol{r}_i, \hat{\boldsymbol{\eta}}^{(w)}_m\big)$.

To implement the GNPC method, start with some initial values at $t = 0$ step. At the $(t+1)$th step, update the estimates as follows:

$$\hat{\boldsymbol{\alpha}}_i^{(t+1)} = \underset{m \in \{1,2,\ldots,M\}}{\arg\min}\, d\big(\boldsymbol{r}_i, \hat{\boldsymbol{\eta}}^{(w)(t)}_m\big), \quad \hat{\eta}^{(w)(t+1)}_{j,m} = \bar{\boldsymbol{r}}_{j,\hat{C}_m^{(t+1)}},$$

where $\hat{\boldsymbol{\eta}}^{(w)(t)}_m$ is the estimated centroids obtained in step t, and $\hat{C}_m^{(t+1)}$ is the partition of the subjects based on $\big\{\hat{\boldsymbol{\alpha}}_i^{(t+1)}\big\}_{i=1}^{N}$. Chiu et al. (2018) demonstrated through simulation studies that, compared to parametric methods, the nonparametric methods generally performed better in small-scale test settings.

5.3 A General Estimation Framework

In this section, we propose a unified estimation framework that subsumes both the parametric and nonparametric models considered in Section 5.2. This approach would facilitate a better statistical understanding of the relationship between the two families of cognitive diagnosis estimations.

For the parametric methods, we shall focus on the joint estimation of the subjects' latent classes $(\boldsymbol{\alpha}_i)_{i=1}^n$ and the model parameters. Considering the joint maximum likelihood estimation for parametric LAMs and the nonparametric estimation approaches as introduced in Section 5.2, we can see that the item parameters $\boldsymbol{\theta}$ in the parametric models and the ideal response vectors $\boldsymbol{\eta}$ in the nonparametric methods are closely related, both denoting a certain "centroid" of the responses of the latent classes under different model assumptions. For instance, $\theta_{j,\boldsymbol{\alpha}} = \mathbb{P}(r_j = 1 \mid \boldsymbol{\alpha})$ can be viewed as the statistical population average (center) of the responses to item j of those subjects with attribute profile $\boldsymbol{\alpha}$, whereas $\eta_{j,\boldsymbol{\alpha}}$ corresponds to the nonparametric clustering center of the responses to item j of those in cluster $\boldsymbol{\alpha}$. Therefore, similarly to the nonparametric clustering methods, the joint maximum likelihood estimation of the parametric model can be viewed as the minimization of some "distance" function, introduced by the negative log-likelihood, between the observed responses and the "centroid" responses $\boldsymbol{\theta}$.

Motivated by this observation, we propose a unified estimation framework for both the parametric and nonparametric methods. Specifically, we let $\boldsymbol{A} = \big(\boldsymbol{\alpha}_i\big)_{i=1}^N$ denote a class membership matrix for N subjects. Based on the membership matrix $\boldsymbol{A}$, we can obtain a partition of N subjects into 2^K proficiency classes, denoted by $\boldsymbol{C}(\boldsymbol{A}) = \big\{C_{\boldsymbol{\alpha}}(\boldsymbol{A}) : \boldsymbol{\alpha} \in \{0,1\}^K\big\}$, where $C_{\boldsymbol{\alpha}}(\boldsymbol{A})$ denotes the set of subjects whose latent patterns are specified as $\boldsymbol{\alpha}$ by $\boldsymbol{A}$. For each latent class $\boldsymbol{\alpha} \in \{0,1\}^K$, we use $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ to denote the "centroid" parameters for both parametric and nonparametric methods. Our proposed estimators for the latent attributes and centroid parameters are obtained by minimizing a loss function of $(\boldsymbol{A}, \boldsymbol{\mu})$ as follows:

$$L(\boldsymbol{A}, \boldsymbol{\mu}) := \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \sum_{i \in C_{\boldsymbol{\alpha}}(\boldsymbol{A})} l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}), \tag{5.2}$$

and the corresponding estimators are $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{\mu}}) = \underset{(\boldsymbol{A},\boldsymbol{\mu})}{\arg\min} L(\boldsymbol{A}, \boldsymbol{\mu})$. In (5.2), $l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}})$ is a loss function that measures the distance between the ith subject's response vector $\boldsymbol{r}_i$ and the centroid of latent class $\boldsymbol{\alpha}$. Specifically, the loss function takes the additive form $l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) = \sum_{j=1}^{J} l(r_{ij}, \mu_{j,\boldsymbol{\alpha}})$, where we abuse the notation $l(\cdot, \cdot)$ a little, and when the loss function takes two vectors, we use it to denote the summation of the element-wise losses. In this work, we also assume that $l(r_{ij}, \mu_{j,\boldsymbol{\alpha}})$ is continuous in $\mu_{j,\boldsymbol{\alpha}}$. Note that (5.2) can also be expressed as

$$L(\boldsymbol{A}, \boldsymbol{\mu}) = \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \sum_{i \in C_{\boldsymbol{\alpha}}(\boldsymbol{A})} l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) = \sum_{i=1}^{N} \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \mathbb{I}\{\boldsymbol{\alpha}_i = \boldsymbol{\alpha}\} \cdot l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) = \sum_{i=1}^{N} l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}_i}), \tag{5.3}$$

which corresponds to a joint estimation of $(\boldsymbol{A}, \boldsymbol{\mu})$ under the loss function $l(\cdot, \cdot)$. From the joint estimation perspective, we can show that, with appropriate loss functions (e.g., L_1, L_2, cross-entropy) and constraints on the centroids (e.g., centroids based on the ideal responses, weighted ideal responses, or specific cognitive diagnosis assumptions), the proposed framework can provide estimates for all the parametric models introduced in Section 2.1 and nonparametric methods discussed in Section 5.2. The examples below demonstrate how the NPC method, the GNPC method, and parametric estimation of the DINA and GDINA models can be derived from the proposed framework using various loss functions and centroid constraints.

Example V.1 (NPC). *In the proposed framework, let the ideal responses under the NPC method be the centroids, that is, $\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \boldsymbol{\eta}_{\boldsymbol{\alpha}}$. If we use the L_1 loss function $l(r_{ij}, \eta_{j,\boldsymbol{\alpha}}) = |r_{ij} - \eta_{j,\boldsymbol{\alpha}}|$, then our proposed framework will become exactly the NPC method. Recall that in the NPC method, the ideal response vectors $\boldsymbol{\eta}_{\boldsymbol{\alpha}}$ are determined by pre-specified model assumptions (either the DINA or the DINO); thus, we only need to classify each subject to the closest proficiency class.*

Example V.2 (GNPC). *Recall that in the GNPC method, the ideal response is defined as $\eta_{j,m}^{(w)} = w_{j,m}\eta_{j,m}^{DINA} + (1 - w_{j,m})\eta_{j,m}^{DINO}$, a weighted average of the DINA ideal response and the DINO ideal response. Note that for proficiency classes and items* *such that* $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$*, we have* $\eta^{DINA}_{\boldsymbol{\alpha},j} = \eta^{DINO}_{\boldsymbol{\alpha},j} = 1$*, and for* $\boldsymbol{\alpha} \odot \boldsymbol{q}_j = \mathbf{0}$*, where* $\odot$ *denotes the elementwise multiplication of vectors, we have* $\eta^{DINA}_{\boldsymbol{\alpha},j} = \eta^{DINO}_{\boldsymbol{\alpha},j} = 0$*. In such cases, the weights in fact do not affect the weighted ideal responses since the DINA and the DINO models have the same ideal responses. Therefore, if we constrain* $\boldsymbol{\mu}_{\boldsymbol{\alpha}} = (\mu_{j,\boldsymbol{\alpha}}, j = 1, \ldots, J)$ *in* (5.2)*, such that* $\mu_{j,\boldsymbol{\alpha}} = 1$ *if* $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$*,* $\mu_{j,\boldsymbol{\alpha}} = 0$ *if* $\boldsymbol{\alpha} \odot \boldsymbol{q}_j = \mathbf{0}$*, and* $\mu_{j,\boldsymbol{\alpha}} = \eta^{(w)}_{j,m}$ *as defined in the GNPC for the rest of the items, while at the same time use the* L_2 *loss function* $l(r_{ij}, \eta_{j,\boldsymbol{\alpha}}) = (r_{ij} - \eta_{j,\boldsymbol{\alpha}})^2$*, then the criterion in* (5.2) *is equivalent to the GNPC method.*

Example V.3 (DINA)**.** *Let's consider the cross-entropy loss (i.e., the negative log-likelihood function),*

$$l(r_{ij}, \mu_{j,\boldsymbol{\alpha}}) = -\big(r_{ij} \log \mu_{j,\boldsymbol{\alpha}} + (1 - r_{ij}) \log(1 - \mu_{j,\boldsymbol{\alpha}})\big). \tag{5.4}$$

In addition, if we constrain the centroids to satisfy the following conditions:

$$\max_{\boldsymbol{\alpha}:\boldsymbol{\alpha}\succeq \boldsymbol{q}_j} \mu_{j,\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}:\boldsymbol{\alpha}\succeq \boldsymbol{q}_j} \mu_{j,\boldsymbol{\alpha}} \;\geq\; \max_{\boldsymbol{\alpha}:\boldsymbol{\alpha}\nsucceq \boldsymbol{q}_j} \mu_{j,\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}:\boldsymbol{\alpha}\nsucceq \boldsymbol{q}_j} \mu_{j,\boldsymbol{\alpha}},$$

that is, all the capable subjects share the same higher item parameters, whereas all the incapable subjects share the same lower item parameters, then the proposed criterion (5.2) *becomes the Joint Maximum Likelihood Estimation (JMLE, Chiu et al., 2016) criterion for the DINA model. Moreover, the centroids here correspond to item response parameters* $\boldsymbol{\theta}$ *for each latent class in the DINA model.*

Example V.4 (GDINA)**.** *In Example V.3, we can put the following constraints on the centroids:* $\mu_{j,\boldsymbol{\alpha}} = \mu_{j,\boldsymbol{\alpha}'}$*, if* $\boldsymbol{\alpha}_{\mathcal{K}_j} = \boldsymbol{\alpha}'_{\mathcal{K}_j}$*, where* $\boldsymbol{\alpha}_{\mathcal{K}_j} = (\alpha_k)_{k\in\mathcal{K}_j}$ *is the sub-vector of* $\boldsymbol{\alpha}$ *on the set* $\mathcal{K}_j$*, and* $\mathcal{K}_j = \{k \in [K] : q_{j,k} = 1\}$ *is the set of required attributes by item* j*. Equivalently, these constraints will result in the same centroid parameters for any two latent patterns sharing the same values on the required attributes of item* j*, which* *is a GDINA model assumption. Furthermore, if we take the same loss functions as in Example V.3, it will result in the JMLE criterion for the GDINA model. Again, the centroids correspond to item response parameters $\boldsymbol{\theta}$ for each proficiency class.*

As demonstrated in the above examples, by taking different loss functions and different constraints on the centroid of each latent class, our proposal (5.2) provides a general estimation framework bridging both the parametric and nonparametric methods in the literature. The parametric estimation approaches mostly use the cross-entropy loss (negative log-likelihood) function, whereas the nonparametric approaches use the L_1 or L_2 distance measures. The analogous roles of negative log-likelihood for a parametric LAM and the distance function for a nonparametric LAM were also noted in Chiu et al. (2018). It can be noted that the proposed estimation criterion (5.2) does not directly use the information pertaining to the population distribution of the latent attribute profiles, which differentiates it from marginal likelihood estimation. As the population proportion of each latent class of attribute profiles may also provide useful information for the model estimation, we propose to further generalize (5.2) by including the proportion parameters in the loss function as follows:

$$L(\boldsymbol{A}, \boldsymbol{\mu}, \boldsymbol{\pi}) := \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \sum_{i \in C_{\boldsymbol{\alpha}}(\boldsymbol{A})} \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) + h(\pi_{\boldsymbol{\alpha}}) \Big), \tag{5.5}$$

where $l(\cdot,\cdot)$ is the loss function as in (5.2), and $h(\cdot)$ is a continuous nonincreasing regularization function of the proportion parameter $\pi_{\boldsymbol{\alpha}}$, which denotes the population proportion of latent class $\boldsymbol{\alpha}$. As can be seen from (5.5), the loss function L depends on both the centroids and the class proportions, with one part measuring the distance between a subject's response $\boldsymbol{r}_i$ and the centroid of a latent class $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$, and the other part involving a regularization of class proportions.

Implicitly, Examples V.1–V.4 take $h(\pi_{\boldsymbol{\alpha}}) = 0$. When we take the loss function $l(r_{ij}, \mu_{j,\boldsymbol{\alpha}})$ to be the cross-entropy loss function as in (5.4), and let $h(\pi_{\boldsymbol{\alpha}}) = -\log \pi_{\boldsymbol{\alpha}}$, then (5.5) becomes

$$L(\boldsymbol{A}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \sum_{i \in C_{\boldsymbol{\alpha}}(\boldsymbol{A})} \Big(l(r_{ij}, \mu_{j,\boldsymbol{\alpha}}) - \log \pi_{\boldsymbol{\alpha}} \Big) = - \sum_{i=1}^{N} \log \Big\{ \pi_{\boldsymbol{\alpha}_i} \times Lik(\boldsymbol{r}_i; \boldsymbol{\mu}_{\boldsymbol{\alpha}_i}) \Big\}, \tag{5.6}$$

where $Lik(\boldsymbol{r}; \boldsymbol{\mu}_{\boldsymbol{\alpha}}) = \mathbb{P}(\boldsymbol{r} \mid \boldsymbol{\mu}_{\boldsymbol{\alpha}})$ is the likelihood function for latent class $\boldsymbol{\alpha}$ and observation $\boldsymbol{r}$, and $\boldsymbol{\mu}_{\boldsymbol{\alpha}} = (\mu_{j,\boldsymbol{\alpha}}, j = 1, \ldots, J)$ is the corresponding model parameters with $\mu_{j,\boldsymbol{\alpha}} = \theta_{j,\boldsymbol{\alpha}} = \mathbb{P}(r_{ij} = 1 \mid \boldsymbol{\alpha})$. Note that $\pi_{\boldsymbol{\alpha}_i} \times Lik(\boldsymbol{r}_i; \boldsymbol{\mu}_{\boldsymbol{\alpha}_i})$ in the RHS of (5.6) corresponds to the *complete-data* likelihood of $(\boldsymbol{\alpha}_i, \boldsymbol{r}_i)$; therefore, the loss function (5.6) is in fact the complete-data log-likelihood of $(\boldsymbol{A}, \boldsymbol{R})$, where $\boldsymbol{R} = (r_{ij})$.

The loss function (5.6) also corresponds to the extension of the classification maximum likelihood (CML) criterion (Celeux and Govaert, 1992) applied to the cognitive diagnosis setting. In Examples V.3 and V.4, using the loss function as in (5.6) corresponds to the CML criterion for the DINA or GDINA model respectively. It can be noted that the CML differs from the JMLE in that the former has an additional term $\log \pi_{\boldsymbol{\alpha}}$ in the loss function to make use of the information in the proportion parameters. The CML is also closely related to the EM estimation for the marginal MLE in that the CML directly maximizes the complete-data log-likelihood whereas the EM algorithm maximizes the expected complete-data log-likelihood with respect to the posterior distribution of the latent variables. Finally, it can also be underscored that, by incorporating a wide range of loss functions, the proposed criterion (5.5) is a generalization of the CML criterion (5.6).

To implement the unified estimation framework, we develop an algorithm to minimize (5.5). The algorithm is a general iterative algorithm to classify each subject to the closet proficiency class. Starting from initial values, the current loss for each subject's responses and the centroid of each latent class is first computed, after which the subject is assigned to the closest latent class that minimizes the loss. Based on the assigned memberships, the estimates for the centroids and class proportions are updated. The details of the steps are shown in Algorithm V.1.

```
Algorithm V.1: General Iterative Classification Algorithm
Input  : Binary response matrix $\boldsymbol{R} \in \{0,1\}^{N\times J}$ and structural $Q$-matrix
         $\boldsymbol{Q} \in \{0,1\}^{J\times K}$
Initialize $\hat{\boldsymbol{A}}^{(0)}$, $\hat{\boldsymbol{\mu}}^{(0)}$ and $\hat{\boldsymbol{\pi}}^{(0)}$.
while convergence not reached do
    At the $(t+1)^{th}$ iteration,
    Step 1: Compute the current loss between $\boldsymbol{r}_i$ and the centroid of each
      proficiency class,
          $l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}^{(t)}_{\boldsymbol{\alpha}}) + h(\hat{\pi}^{(t)}_{\boldsymbol{\alpha}}),\ i = 1, \dots, N,\ \boldsymbol{\alpha} \in \{0,1\}^K.$
    Step 2: Assign each $\boldsymbol{r}_i$ to the closest proficiency class, as in,
          $\hat{\boldsymbol{\alpha}}^{(t)}_i = \arg\min_{\boldsymbol{\alpha}} l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}^{(t)}_{\boldsymbol{\alpha}}) + h(\hat{\pi}^{(t)}_{\boldsymbol{\alpha}}),\ i = 1, \dots, N.$
        and obtain the resulting partition $\hat{\boldsymbol{C}}^{(t)} := \boldsymbol{C}(\hat{\boldsymbol{A}}^{(t)})$.
    Step 3: Compute the centroid and proportion of each proficiency class,
          $(\hat{\boldsymbol{\mu}}^{(t+1)}_{\boldsymbol{\alpha}}, \hat{\boldsymbol{\pi}}^{(t+1)}_{\boldsymbol{\alpha}}) = \arg\min_{(\boldsymbol{\mu},\boldsymbol{\pi})} \sum_{i\in\hat{C}^{(t)}_{\boldsymbol{\alpha}}} \Big(l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}^{(t)}_{\boldsymbol{\alpha}}) + h(\hat{\pi}^{(t)}_{\boldsymbol{\alpha}})\Big),\ \boldsymbol{\alpha} \in \{0,1\}^K.$
end
Output: $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{\mu}}$, and $\hat{\boldsymbol{\pi}}$.
```

In the cognitive diagnosis modeling context, certain proficiency classes share the same item response parameters for each item given a particular Q-matrix. For example, for all LAMs, any $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$, has the same item parameter; for the DINA model, there are only two levels of item parameter for each item, and the capable classes with $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$ share the same item parameter θ^+_j, and the incapable classes with $\boldsymbol{\alpha} \not\succeq \boldsymbol{q}_j$ share the same item parameter θ^-_j. Based on this observation, under certain model assumptions, the proficiency classes can be partitioned into some equivalence classes for each item according to the Q-matrix. Specifically, for item j, let $\tilde{A}_j = \big\{\tilde{A}_{j,\boldsymbol{\alpha}} = \{\boldsymbol{\alpha}' : \mu_{j,\boldsymbol{\alpha}} = \mu_{j,\boldsymbol{\alpha}'}\}\big\}$. Under this partitioning, the proficiency classes in the same equivalence class will have the same item response probability for this item. For example, in a DINA model with two latent attributes, if $\boldsymbol{q}_j = (1,0)$, then the proficiency classes can be partitioned into $\Big\{\big\{(0,0),(0,1)\big\}, \big\{(1,0),(1,1)\big\}\Big\}$, where $\boldsymbol{\alpha} \in \big\{(0,0),(0,1)\big\}$ will have the same item parameter, θ_j^-, and $\boldsymbol{\alpha} \in \big\{(1,0),(1,1)\big\}$ will also share the same item parameter, θ_j^+. Therefore, by incorporating information of the Q-matrix and certain model assumptions, we can develop an iterative classification algorithm tailored for LAMs that updates the centroids associated with equivalence classes together.

To illustrate, if we let the negative log-likelihood function be the loss function as specified in (5.4), then Step 3 in Algorithm V.1 simplifies to

$$\hat{\pi}_{\boldsymbol{\alpha}}^{(t+1)} = \frac{\sum_{i=1}^{N} I\{\hat{\boldsymbol{\alpha}}_i^{(t)} = \boldsymbol{\alpha}\}}{N}, \quad \hat{\mu}_{j,\boldsymbol{\alpha}}^{(t+1)} = \frac{\sum_{\boldsymbol{\alpha}' \in \tilde{A}_{j,\boldsymbol{\alpha}}} \sum_{i \in \hat{C}_{\boldsymbol{\alpha}'}^{(t)}} r_{ij}}{\sum_{\boldsymbol{\alpha}' \in \tilde{A}_{j,\boldsymbol{\alpha}}} |\hat{C}_{\boldsymbol{\alpha}'}^{(t)}|},$$

where $|\cdot|$ is the cardinality of a set. Based on this simplification, the estimated proportion parameters are the sample proportions based on the estimated partition of the subjects, and the estimated centroids are the corresponding sample means of the equivalence classes also based on the estimated partition. Moreover, if fixed and equal proportions, together with L_2 loss $l(r_{ij}, \mu_{j,\alpha}) = (r_{ij} - \mu_{j,\alpha})^2$ are used, the algorithm becomes the iterative algorithm for the GNPC method outlined in Chiu et al. (2018).

5.4 Analysis of the Proposed Framework

In this section, we provide a theoretical analysis of the proposed framework. We show that, under certain conditions, the proposed estimation framework will give consistent estimates. The consistency results can be regarded as extensions of those for the NPC and the GNPC methods developed in Wang and Douglas (2015) and Chiu and Köhn (2019a). In addition to the asymptotic results, we also provide an analysis of the proposed algorithm in the finite sample situations.

As we introduced in Section 2.3, all the parametric LAMs belong to the family of latent class models. Hence, in our following analysis, we assume a general latent class model as the underlying model. Our results below are also easily adapted to the Q-matrix restricted latent class models. We use $\theta^0_{j,\boldsymbol{\alpha}}$ to denote the true item parameter for the jth item and latent pattern $\boldsymbol{\alpha}$, as in, $\theta^0_{j,\boldsymbol{\alpha}} = \mathbb{P}(r_j = 1 \mid \boldsymbol{\alpha})$, and we use $\boldsymbol{\theta}^0_{\boldsymbol{\alpha}} = (\theta^0_{1,\boldsymbol{\alpha}}, \ldots, \theta^0_{J,\boldsymbol{\alpha}})$ to denote item parameter vector for latent pattern $\boldsymbol{\alpha}$, and $\boldsymbol{\theta} = \big(\theta_{j,\boldsymbol{\alpha}} : j \in [J], \boldsymbol{\alpha} \in \{0,1\}^K\big)$ to denote the item parameter matrix. We let $\boldsymbol{A}^0 = (\boldsymbol{\alpha}^0_i)_{i=1}^N$ denote the true latent pattern matrix of the N subjects to be classified. Before we establish the consistency results, we first make some mild assumptions.

Assumption 5.4.1. *The loss function $l(x, \mu)$ is Hölder continuous in μ on $[\tau, 1-\tau]$ for any $\tau \in (0, 0.5)$, and the total loss* (5.5) *is minimized at class means given the subjects' membership, as in, $\hat{\mu}_{j,\boldsymbol{\alpha}} = \sum_{i\in C_{\boldsymbol{\alpha}}} r_{ij}/|C_{\boldsymbol{\alpha}}|$.*

Assumption 5.4.2. *$h(\cdot)$ in* (5.5) *is a continuous nonincreasing function of the proportion parameters.*

Assumption 5.4.3. *There exist constants $\delta_1, \delta_2 > 0$ such that $\lim\limits_{J\to\infty}\{\min\limits_{\boldsymbol{\alpha}\neq\boldsymbol{\alpha}'} J^{-1}\|\boldsymbol{\theta}^0_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}'}\|_1\} \geq \delta_1$, and $\delta_2 \leq \min\limits_{j,\boldsymbol{\alpha}} \theta^0_{j,\boldsymbol{\alpha}} < \max\limits_{j,\boldsymbol{\alpha}} \theta^0_{j,\boldsymbol{\alpha}} \leq 1-\delta_2$, where $\|\cdot\|_1$ denotes the L_1 norm.*

Assumption 5.4.4. *There exists $\delta \geq 1$ such that*

$$\Big|E[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})] - E[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}^0_i})]\Big| \geq \big|\theta^0_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}^0_i}\big|^{\delta}, \ \forall\ \boldsymbol{\alpha} \neq \boldsymbol{\alpha}^0_i. \tag{5.7}$$

One can easily check that the L_2 and cross-entropy (negative log-likelihood) loss functions given in Section 5.3 satisfy Assumption 5.4.1. Note that the second part of Assumption 5.4.1 is a natural requirement for the consistent estimation of $\theta^0_{j,\boldsymbol{\alpha}}$, as $\theta^0_{j,\boldsymbol{\alpha}}$ represents the population average of the responses of subjects in latent class $\boldsymbol{\alpha}$, that is, $\theta^0_{j,\boldsymbol{\alpha}} = \mathbb{P}(r_j = 1 \mid \boldsymbol{\alpha})$. Given the true memberships of the subjects, for an estimator $\hat{\mu}_{j,\boldsymbol{\alpha}}$ that is consistent for $\theta^0_{j,\boldsymbol{\alpha}}$, it must satisfy $\big|\hat{\mu}_{j,\boldsymbol{\alpha}} - \sum_{i\in C_{\boldsymbol{\alpha}}} r_{ij}/|C_{\boldsymbol{\alpha}}|\big| \to 0$

in probability by the law of large number. An interesting counterexample is the L_1 loss function, which does not satisfy this assumption because given the memberships, $\hat{\mu}_{j,\boldsymbol{\alpha}}$ that minimizes the L_1 loss function is the sample median instead of the sample mean. Since in the cognitive diagnosis setting the responses are binary, the sample median would be either 0 or 1, which makes $\hat{\mu}_{j,\boldsymbol{\alpha}}$ under the L_1 loss not a consistent estimator of $\theta^0_{j,\boldsymbol{\alpha}}$ even when the true memberships are known. In other words, the L_1 loss cannot provide a consistent estimation of the centroid parameters while the L_2 and cross-entropy losses can, as to be shown in the following theorems. More generally, following the M-estimation theory (van der Vaart, 2000), the second part of Assumption 5.4.1 can be further relaxed to requiring $E_{\theta^0_{j,\boldsymbol{\alpha}}}[l(r_{ij}, \mu_{j,\boldsymbol{\alpha}})]$ has a unique minima at $\theta^0_{j,\boldsymbol{\alpha}}$ and some additional technical conditions. For the presentation brevity, here we shall use the current assumption, which is already broad enough for practical use.

Assumptions 5.4.2 and 5.4.3 ensure the identifiability of the model, and also keep the true parameters away from the boundaries of the parameter space. Particularly, the assumption $\lim_{J\to\infty}\{\min_{\boldsymbol{\alpha}\neq\boldsymbol{\alpha}'} J^{-1}\|\boldsymbol{\theta}^0_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}'}\|_1\} \geq \delta_1$ implies that there is sufficient information to distinguish any two different classes $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$, thus ensuring the completeness (Chiu et al., 2009) and identifiability conditions (Gu and Xu, 2020). It is also similar to Condition (b) in Wang and Douglas (2015):

Condition(b). *Define the set $\mathcal{A}_{m,m'} = \{j \mid \eta_{mj} \neq \eta_{m'j}\}$, where m and m' index the attribute profiles of different proficiency classes among all the $M = 2^K$ realizable proficiency classes; Card$(\mathcal{A}_{m,m'}) \to \infty$ as $J \to \infty$.*

Condition (b) in Wang and Douglas (2015) and Assumption 5.4.3 in our work are essentially stating that for any two different proficiency classes, there are infinitely many items such that the item parameters for these two proficiency classes are different.

The condition (5.7) in Assumption 5.4.4 also holds for the aforementioned loss functions in Section 5.3. For example, it is easy to check the condition (5.7) for the L_2 loss and the cross-entropy loss. For the L_2 loss, we have $E\big[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})\big] - E\big[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}^0_i})\big] = (\theta^0_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}^0_i})^2$. For the cross-entropy loss, we have

$$
\begin{aligned}
& E\big[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})\big] - E\big[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}^0_i})\big] \\
= & -\theta^0_{j,\boldsymbol{\alpha}_0} \log(\theta^0_{j,\boldsymbol{\alpha}}) - (1 - \theta^0_{j,\boldsymbol{\alpha}^0_i}) \log(1 - \theta^0_{j,\boldsymbol{\alpha}}) + \theta^0_{j,\boldsymbol{\alpha}^0_i} \log(\theta^0_{j,\boldsymbol{\alpha}^0_i}) + (1 - \theta^0_{j,\boldsymbol{\alpha}^0_i}) \log(1 - \theta^0_{j,\boldsymbol{\alpha}^0_i}) \\
= & D_{KL}\Big(p(\theta^0_{j,\boldsymbol{\alpha}}) \;||\; p(\theta^0_{j,\boldsymbol{\alpha}^0_i})\Big) \geq \; \frac{1}{2}\Big(\big|\theta^0_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}^0_i}\big| + \big|(1 - \theta^0_{j,\boldsymbol{\alpha}}) - (1 - \theta^0_{j,\boldsymbol{\alpha}^0_i})\big|\Big)^2 \\
= & \, 2(\theta^0_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}^0_i})^2,
\end{aligned}
$$

where $D_{KL}(\cdot \;||\; \cdot)$ is the Kullback-Leibler divergence, $p(\cdot)$ is the mass function of a Bernoulli distribution, and the last inequality follows from Theorem 1.3 in Popescu et al. (2016).

Similar to the analysis of the joint maximum likelihood estimation in Chiu et al. (2016), we assume that there is a calibration dataset that would give a statistically consistent estimator of the calibration subjects' latent class membership $\hat{\boldsymbol{A}}_c$, in the sense that $\mathbb{P}(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \to 0$ as $J \to \infty$. We use N_c and $\boldsymbol{A}^0_c$ to denote the sample size and the true membership matrix of the calibration dataset, respectively. Here the subscript c denotes the calibration dataset. A similar assumption is also made in the consistency theories of the GNPC method in and Chiu and Köhn (2019a). In the next theorem, we show that the consistent membership estimator will give consistent estimators for the centroids of the latent classes.

Theorem 5.4.5. *Suppose the data conforms to LAMs that can be expressed in terms of general latent class models, and Assumptions 1-3 hold. Further assume that* $J \exp\big(-N_c\epsilon\big) \to 0$ *as* $J, N_c \to \infty$ *for any* $\epsilon > 0$*. If* $\hat{\boldsymbol{A}}_c$ *is a consistent estimator of* $\boldsymbol{A}^0_c$*, then* $\hat{\boldsymbol{\mu}}$ *is also consistent for* $\boldsymbol{\theta}^0$ *as* $J, N_c \to \infty$*, that is,* $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\theta}^0\|_\infty \xrightarrow{P} 0$ *as* $J, N_c \to \infty$*, where* $\|\cdot\|_\infty$ *is the supremum norm.*

Theorem 5.4.5 states that if we could get a consistent estimate of the calibration subjects' membership $\hat{\boldsymbol{A}}_c$, then the estimated centroids $\hat{\boldsymbol{\mu}}$ are also consistent for the true item parameters $\boldsymbol{\theta}^0$ in a uniform sense that all item parameters can be uniformly consistently estimated. The detailed proof is in Appendix C.1. This result is similar to Lemmas 1 and 2 in Chiu and Köhn (2019a) under the GNPC framework, and Theorem 2 in Chiu et al. (2016) under the JMLE framework. Note that for the GNPC method, the centroids are weighted averages of the ideal responses from the DINA and DINO models. As discussed in Example V.2, if the DINA and DINO models have the same ideal responses (i.e., $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$ or $\boldsymbol{\alpha} \odot \boldsymbol{q}_j = \mathbf{0}$), then the corresponding centroid will be fixed to be 0 or 1, which thus does not lead to a consistent estimation of the corresponding item parameter $\theta^0_{j,\boldsymbol{\alpha}}$; however, note that for the nonparametric GNPC method, such a fixed centroid does not necessarily lead to inconsistency of $\hat{\boldsymbol{\alpha}}$. Here we allow all the centroid parameters to be free, and the consistency estimation is ensured as in Theorem 5.4.5.

The next theorem shows that if we start with a consistent membership $\hat{\boldsymbol{A}}_c$ obtained from the calibration dataset, and use the estimated centroids to classify the subjects, then the resulting classifications are also consistent for each subject.

Theorem 5.4.6. *Suppose Assumptions 1–4 and the assumptions of Theorem 5.4.5 hold. If we start with a consistent $\hat{\boldsymbol{A}}_c$ obtained from a calibration dataset to estimate the centroid $\hat{\boldsymbol{\mu}}$, then $\hat{\boldsymbol{\alpha}}_i$ obtained by Algorithm V.1 is also a consistent estimator of $\boldsymbol{\alpha}_i$ for each $i = 1, \ldots, N$.*

To establish the consistency in Theorem 5.4.6, the following two lemmas are needed.

Lemma 5.4.7. *Suppose Assumptions in Theorem 5.4.6 hold. For each subject i, the true attribute pattern minimizes $E[l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}) + h(\hat{\pi}_{\boldsymbol{\alpha}})]$ with probability approaching 1 as $J \to \infty$, as in,*

$$P\Big(\boldsymbol{\alpha}_i^0 = \arg\min_{\boldsymbol{\alpha}} E\big[l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}) + h(\hat{\pi}_{\boldsymbol{\alpha}})\big]\Big) \to 1 \qquad as\ J \to \infty.$$

Lemma 5.4.8. *Suppose Assumptions in Theorem 5.4.6 hold, then we have*

$$P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}})-E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})]\big)\Big|\geq\epsilon\Big)\to 0,\ \ \textit{as } J\to\infty.$$

Lemma 5.4.7 extends Proposition 1 in Wang and Douglas (2015) and Lemma 3 in Chiu and Köhn (2019a) to more general loss functions. Lemma 5.4.8 generalizes Proposition 3 in Wang and Douglas (2015) and Lemma 4 in Chiu and Köhn (2019a). The detailed proofs of Lemma 5.4.7, Lemma 5.4.8, and Theorem 5.4.6 are given in Appendices C.2 – C.4. Note that Theorem 5.4.6 only gives the consistency for each $\boldsymbol{\alpha}_i$; however, we can further establish uniform consistency for all $\boldsymbol{\alpha}_i$, $i = 1, \ldots, N$, as shown in Theorem 5.4.9.

Theorem 5.4.9. *Suppose all the assumptions of Theorem 5.4.6 hold. Further assume that $N > J$, $N_c > J$ and for any $\epsilon > 0$, $N\exp(-J\epsilon) \to 0$. If we start with a consistent $\hat{\boldsymbol{A}}_c$ obtained from a calibration dataset, then $\hat{\boldsymbol{\alpha}}_i$ obtained from Algorithm V.1 is uniformly consistent for $\boldsymbol{\alpha}_i$, for all $i = 1, \ldots, N$.*

Uniform consistency has also been established for specific nonparametric methods, such as Theorem 2 in Wang and Douglas (2015) and Theorem 2 in Chiu and Köhn (2019a). Our uniformly consistent result in Theorem 5.4.9 can be regarded as their generalization. Specifically, in Wang and Douglas (2015), they showed the uniform consistency for the NPC method, where the loss function is taken to be L_1 loss and the centroids are fixed, to be the ideal responses of the DINA or DINO model. In Chiu and Köhn (2019a), the authors generalize the uniform consistency for the NPC method to the GNPC method, where the loss function is L_2 loss and the centroids are weighted averages of ideal responses from the DINA and the DINO models.

The above analysis pertains to the asymptotic properties of our framework. For finite-sample situations, we have the following theoretical properties for the proposed iterative algorithms in Section 5.3, which are established following the theory in

Celeux and Govaert (1992).

Proposition 5.4.1. *Any sequence* $(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)})$ *obtained by Algorithm V.1 decreases the criterion (5.5) and the sequence* $L(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)})$ *converges to a stationary value. Moreover, if for any fixed* $\boldsymbol{A}$*, the minima of the loss function* $L(\boldsymbol{A}, \boldsymbol{\mu}, \boldsymbol{\pi})$ *is well-defined, then the sequence* $(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)})$ *also converges to a stationary point.*

Proposition 5.4.1 indicates that the update sequence $(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)})$ from the proposed algorithm converges to a stationary point of the proposed criterion (5.5) with finite samples. Additionally, all the loss functions in the examples in Section 5.3 satisfy the condition that the minima are well-defined. Now, consider a smoothed version of $L(\boldsymbol{A}, \boldsymbol{\mu}, \boldsymbol{\pi})$,

$$L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \sum_{i=1}^{n} u_{i,\boldsymbol{\alpha}} \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) + h(\pi_{\boldsymbol{\alpha}}) \Big),$$

where $\boldsymbol{U} = \{u_{i,m}\} \in [0,1]^{n \times 2^K}$ is a matrix with nonnegative entries and each column sums to one, which is called a standard classification matrix in Celeux and Govaert (1992). Recall that $L(\boldsymbol{A}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{\boldsymbol{\alpha}} \sum_{i \in C_{\boldsymbol{\alpha}}(\boldsymbol{A})} \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) + h(\pi_{\boldsymbol{\alpha}}) \Big) = \sum_{\boldsymbol{\alpha}} \sum_{i=1}^{n} I(\boldsymbol{\alpha}_i = \boldsymbol{\alpha}) \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) + h(\pi_{\boldsymbol{\alpha}}) \Big)$. Therefore, $L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi})$ can be regarded as a smoothed version, where the hard membership matrix $\boldsymbol{A}$ is replaced by $\boldsymbol{U}$. Note that the minimum of $L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi})$ is attained when $\boldsymbol{U}$ is equal to some hard membership matrix $\boldsymbol{A}$.

Proposition 5.4.2. *Assume that* $L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi})$ *has a local minimum at* $(\boldsymbol{U}^*, \boldsymbol{\mu}^*, \boldsymbol{\pi}^*)$ *and that the Hessian of* $L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi})$ *exists and is positive definite at* $(\boldsymbol{U}^*, \boldsymbol{\mu}^*, \boldsymbol{\pi}^*)$*. Then there is a neighborhood of* $(\boldsymbol{U}^*, \boldsymbol{\mu}^*, \boldsymbol{\pi}^*)$ *such that starting with any* $(\boldsymbol{U}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\pi}^{(0)})$ *in that neighborhood, the resulting sequence* $(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)})$ *of the Algorithm V.1 converges to* $(\boldsymbol{U}^*, \boldsymbol{\mu}^*, \boldsymbol{\pi}^*)$ *at a linear rate.*

Proposition 5.4.2 states that if we start with a good initial value that is close enough to the optimal point, then the update sequence will also converge to the optimal point. These two propositions give good finite-sample properties of our proposed estimation framework. The detailed proofs of Proposition 5.4.1 and Proposition 5.4.2 are given in Appendix C.6 and C.7, respectively.

5.5 Simulation Studies

We conducted comprehensive simulations under a variety of settings to compare the performance of different methods. The methods compared were:

- NPC: the baseline method, where the centers are the ideal responses from the DINA model, and the loss function is the L_1 loss;
- GNPC: the centers are weighted averages of the ideal responses from the DINA and DINO models, and the loss function is the L_2 loss;
- GNPC + log penalty: add log penalties on the proportion parameters to the GNPC method, where the loss function is L_2 loss for the centroids plus the summation of the log functions of the proportion parameters;
- JMLE: the Joint Maximum Likelihood Estimate, where the centroid parameters are to be estimated, and the loss function is the negative log-likelihood;
- CMLE: the Classification Maximum Likelihood Estimate, where the centers and the loss function are the same as JMLE but with an additional term of class proportions as specified in (5.6);
- MMLE: the Marginal Maximum Likelihood Estimate obtained from the traditional EM algorithm under the DINA or GDINA model assumption.

MMLE, as one of, if not the most commonly used estimation algorithm in the cognitive diagnosis literature, was included in the comparison to provide a comprehensive understanding of how different cognitive diagnosis estimation methods perform.

For the underlying true models, we considered two different settings: all items conformed to the DINA, or all items conformed to the GDINA model. Following the simulation design in Chiu et al. (2018), the subjects' true latent attribute patterns were either drawn from a uniform distribution or derived from the multivariate normal threshold model. More specifically, for the uniform setting, each latent pattern $\boldsymbol{\alpha}$ had the same probability $1/2^K$ of being drawn. For the multivariate normal setting, each subject's attribute profile was linked to a latent continuous ability vector $\boldsymbol{z} = (z_1, \ldots, z_K)' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with values along the main diagonal of $\boldsymbol{\Sigma}$ setting to 1 and the off-diagonal entries setting to either $r = 0.40$ or 0.80 for different levels of correlation. The latent continuous ability vector $\boldsymbol{z}$ was randomly sampled, and the kth entry of the attribute pattern was determined by

$$\alpha_{ik} = \begin{cases} 1, & z_{ik} \geq \Phi^{-1}\left(\frac{k}{K+1}\right) \\ 0, & \text{otherwise}, \end{cases}$$

where Φ is the inverse cumulative distribution function of standard normal distribution.

We considered different numbers of latent attributes ($K = 3$ or 5), different sample sizes ($N = 30$, 50, 200 or 500) and different number of items ($J = 30$ or 50). To ensure identifiability, we set the first two $K \times K$ submatrices of the Q-matrix to be identity matrices. The remaining items were randomly generated from all the possible latent patterns. When $K = 5$, the Q-matrix contained items that measured up to three attributes and was constructed the same way as that for $K = 3$. When the underlying model was the DINA or DINO model, different signal strengths were considered. Specifically we set $1 - \theta_j^+ = \theta_j^- = 0.1$ or 0.3. When the true model was the GDINA model, we also considered two different signal strength levels. One had the same item parameters as those specified in Chiu et al. (2018), which are listed in Table 5.1; the other setting contained larger noise as listed in Table 5.2.

$P(\boldsymbol{\alpha}_1)$	$P(\boldsymbol{\alpha}_2)$	$P(\boldsymbol{\alpha}_3)$	$P(\boldsymbol{\alpha}_4)$	$P(\boldsymbol{\alpha}_5)$	$P(\boldsymbol{\alpha}_6)$	$P(\boldsymbol{\alpha}_7)$	$P(\boldsymbol{\alpha}_8)$
0.2	0.9						
0.1	0.8						
0.1	0.9						
0.2	0.5	0.4	0.9				
0.1	0.3	0.5	0.9				
0.1	0.2	0.6	0.8				
0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.9

Table 5.1: Item response parameters for GDINA with small noises.

$P(\boldsymbol{\alpha}_1)$	$P(\boldsymbol{\alpha}_2)$	$P(\boldsymbol{\alpha}_3)$	$P(\boldsymbol{\alpha}_4)$	$P(\boldsymbol{\alpha}_5)$	$P(\boldsymbol{\alpha}_6)$	$P(\boldsymbol{\alpha}_7)$	$P(\boldsymbol{\alpha}_8)$
0.3	0.7						
0.3	0.8						
0.3	0.4	0.7	0.8				
0.3	0.4	0.6	0.7				
0.2	0.3	0.6	0.7				
0.2	0.3	0.3	0.4	0.4	0.5	0.6	0.7

Table 5.2: Item response parameters for GDINA with large noises.

To evaluate the performance of different methods, two metrics were used: the pattern-wise agreement rate (PAR) and the attribute-wise agreement rate (AAR), as defined below,

$$\text{PAR} = \frac{\sum_{i=1}^{N} I\{\hat{\boldsymbol{\alpha}}_i = \boldsymbol{\alpha}_i\}}{N}, \quad \text{AAR} = \frac{\sum_{i=1}^{N} \sum_{k=1}^{K} I\{\hat{\alpha}_{ik} = \alpha_{ik}\}}{NK}.$$

For parametric methods including JMLE, CMLE, and MMLE of the DINA and the GDINA models, we also calculate the Mean Squared Errors (MSEs) for item parameters of each latent class. For each setting, we repeated 100 times and reported the obtained means of PAR, AAR, and MSE. Note that the aforementioned methods are iterative, hence, would be affected by how they are initialized. For comparability purposes, we treated the NPC method as the baseline in this work and used its results to initialize all the other methods. Using the NPC to perform the initialization is a reasonable choice given its non-iterative nature. In the following result plots, we use DINA or GDINA to stand for the results of MMLE obtained by the EM algorithm under the corresponding model assumptions.

Result I: DINA

Figures V.1 and V.2 present the PARs and AARs when the underlying process followed the DINA model. The same scale of the coordinates was used across different simulation settings for comparison. We provide zoomed-in versions in the appendix. Under the independent attribute (i.e., uniform) setting, the NPC performed the best, as expected, in almost all the cases. The JMLE performed similarly to the CMLE in most cases, and slightly better than CMLE when there were more latent attributes ($K = 5$) - this was so because the JMLE method correctly assumed that the true latent patterns were uniformly distributed. The GNPC produced similar results to the JMLE and the CMLE in most cases, but much worse results with large noises ($1 - \theta_j^+ = \theta_j^- = 0.3$) and more items ($J = 50$). Adding log penalty to the GNPC method degraded the results under the uniform setting especially when the sample size was large, which is also expected since the GNPC method implicitly assumes a uniform penalty on the latent classes. In comparison, the MMLE of the DINA and GDINA models did not perform as well as the others. This was particularly true when the noise was large and the sample size was small.

Under the dependent attribute (i.e., multivariate normal) settings, although the NPC still performed the best in almost all the cases with moderate correlation ($r = 0.4$), it performed poorly with larger correlation ($r = 0.8$) and sample size ($N = 200$ or 500) as a consequence of more unequal latent patterns proportions. The MMLE of the DINA provided the best results when the sample size was larger ($N = 200/500$) and the correlation was large ($r = 0.8$), but did not perform well with smaller sample sizes. The GNPC and JMLE performed similarly when the noise was small, but the GNPC was much worse than the JMLE when the noise was large. Adding log penalty on the proportions improved the performance of the GNPC method under the correlated settings, though still not as good as the CMLE method. In contrast, the CMLE performed uniformly well in almost all cases, and its advantages became more apparent when there were more latent attributes, and the correlation and the noise were large. Specifically, the CMLE performed similarly to the NPC when the sample sizes were small, and the MMLE of the DINA when the sample sizes were large. In almost all the conditions, the MMLE of the GDINA did not perform as well as the other methods, which was not unexpected as the DINA was the true model.

Mean Squared Errors (MSEs) for the item parameters using parametric methods including JMLE, CMLE, and MMLE for the DINA and GDINA models are plotted in Figure V.3. From the results, we can see that across different settings the MMLE for the DINA model gave the best item parameter estimates, while the MMLE for the GDINA model performed the worst. The JMLE and CMLE methods provided similar results. It is actually not surprising that the MMLE for the DINA performed the best for item parameter estimation since it correctly assumed a two-level DINA model and directly estimated the corresponding guessing and slipping parameters, while other methods did not have such prior knowledge about the underlying model.

Result II: GDINA

Figures V.4 and V.5 show the PARs and the AARs when the data conformed to the GDINA model under different settings. The zoomed-in versions are provided in the appendix. Based on the results, when the latent attributes were independent, the GNPC performed generally the best across the settings, whereas the JMLE, the CMLE, and the MMLE of the GDINA model improved with increasing sample size. As in the DINA cases, log penalty on the proportions degraded the performance of the GNPC method under the independent setting. The JMLE provided comparable or slightly better results than the CMLE, particularly when K was larger. As mentioned earlier, this is because the JMLE correctly assumed a uniform prior distribution for the latent attributes, whereas the CMLE, although made no assumptions, needed to estimate additional parameters.

Under the correlated latent attributes settings, adding log penalties on the proportions to the GNPC method greatly improved the performance especially when the noise was large or the correlation was high. GNPC+log penalty tended to provide the best results with small sample sizes, and the CMLE and the MMLE of the GDINA gave the best results with larger samples. Moreover, with larger noises, the CMLE method provided better results than the MMLE of the GDINA model, particularly when there were more latent attributes. As the correlation became larger, with large sample sizes, the performance of the CMLE method became more similar to that of the MMLE of the GDINA, and better than the JMLE method, due to the proportions of latent attribute patterns no longer being equal. Based on the above analysis, one can note that the CMLE method was more robust to large noise.

The MSEs for the parametric methods (JMLE, CMLE, and MMLE) under the GDINA settings are given in Figure V.6. These three methods gave similar results in most cases, while JMLE and CMLE performed better than the MMLE of the GDINA settings especially when the number of attributes was large or noises were large.

Summary and Recommendations

Based on the above simulation results, we can see that there is no dominating method that performed uniformly better than other methods across all the simulation settings. Hence, the choice of the method should be based on the specific setting and other information we have at hand. In the following, we provide recommendations in practice under different circumstances.

If we can safely assume that the true underlying model is the DINA model, then the NPC method would give good results if the latent attributes are independent. When the latent attributes are moderately correlated, either the NPC or the CMLE method is recommended. When the correlations are high among the latent attributes, the NPC and the CMLE would perform well with small sample sizes, whereas the CMLE and the MMLE of the DINA model would give better results with sufficiently large data sizes.

In situations where the true model is the GDINA model, the GNPC method will perform generally well if the latent attributes are independent. When the latent attributes are correlated and the sample size is small, the GNPC augmented by log penalties on the proportion parameters is preferred. However, when the sample is sufficiently large, the CMLE method is more robust. The CMLE method also performs well with small sample sizes when the noise is large.

Finally, if the true data-generating models are unknown, the CMLE method is recommended when the latent attributes are correlated. If the latent attributes are independent, the GNPC method is preferred. Moreover, when the sample size is large enough, the MMLE method for the GDINA model is also recommended. If the noise is small, the GNPC method will also perform well when the sample size is small, and augmenting the GNPC method with log penalties on the proportion parameters will improve its performance under the correlated setting.

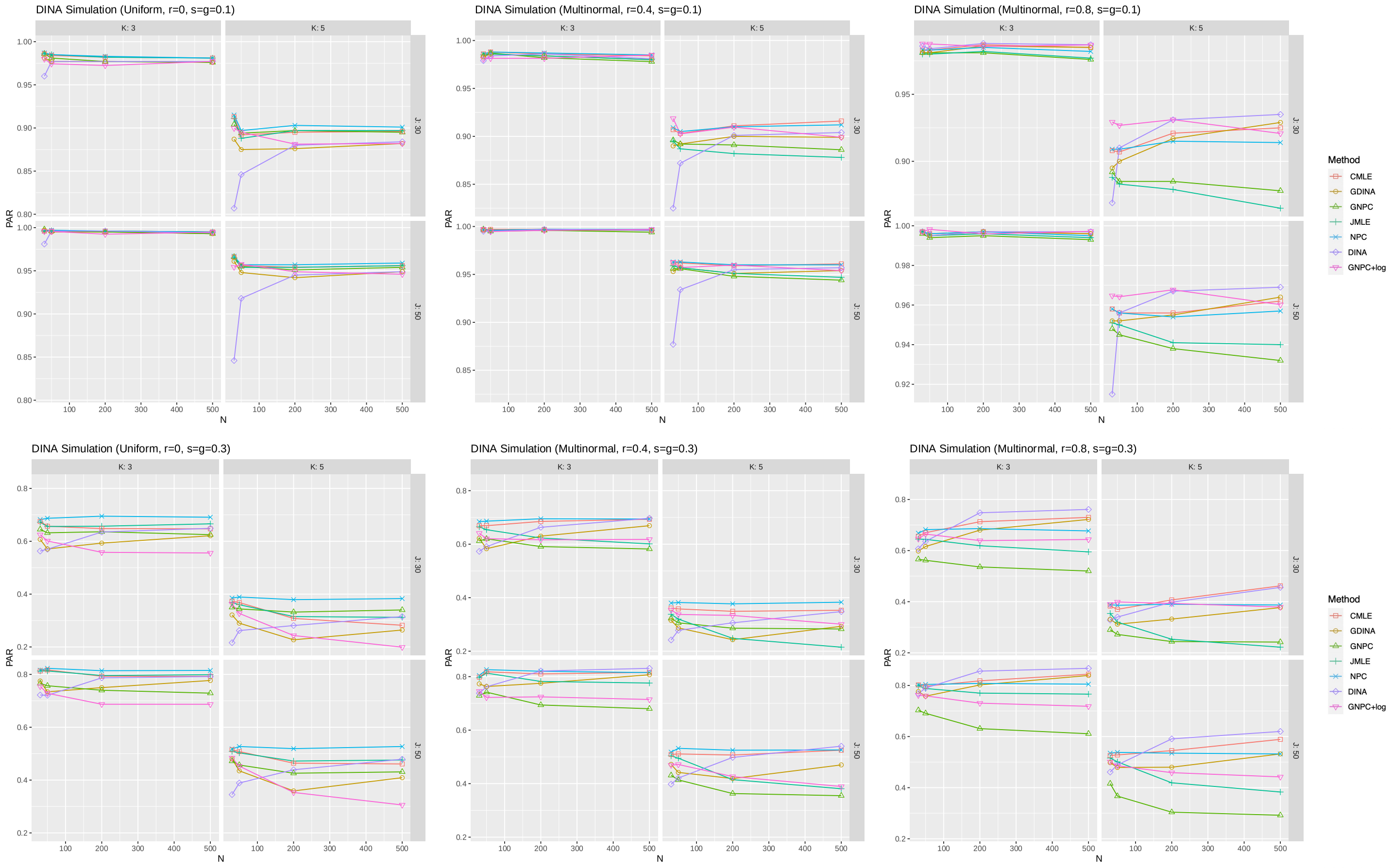


Figure V.1: PARs when the data conformed to the DINA model

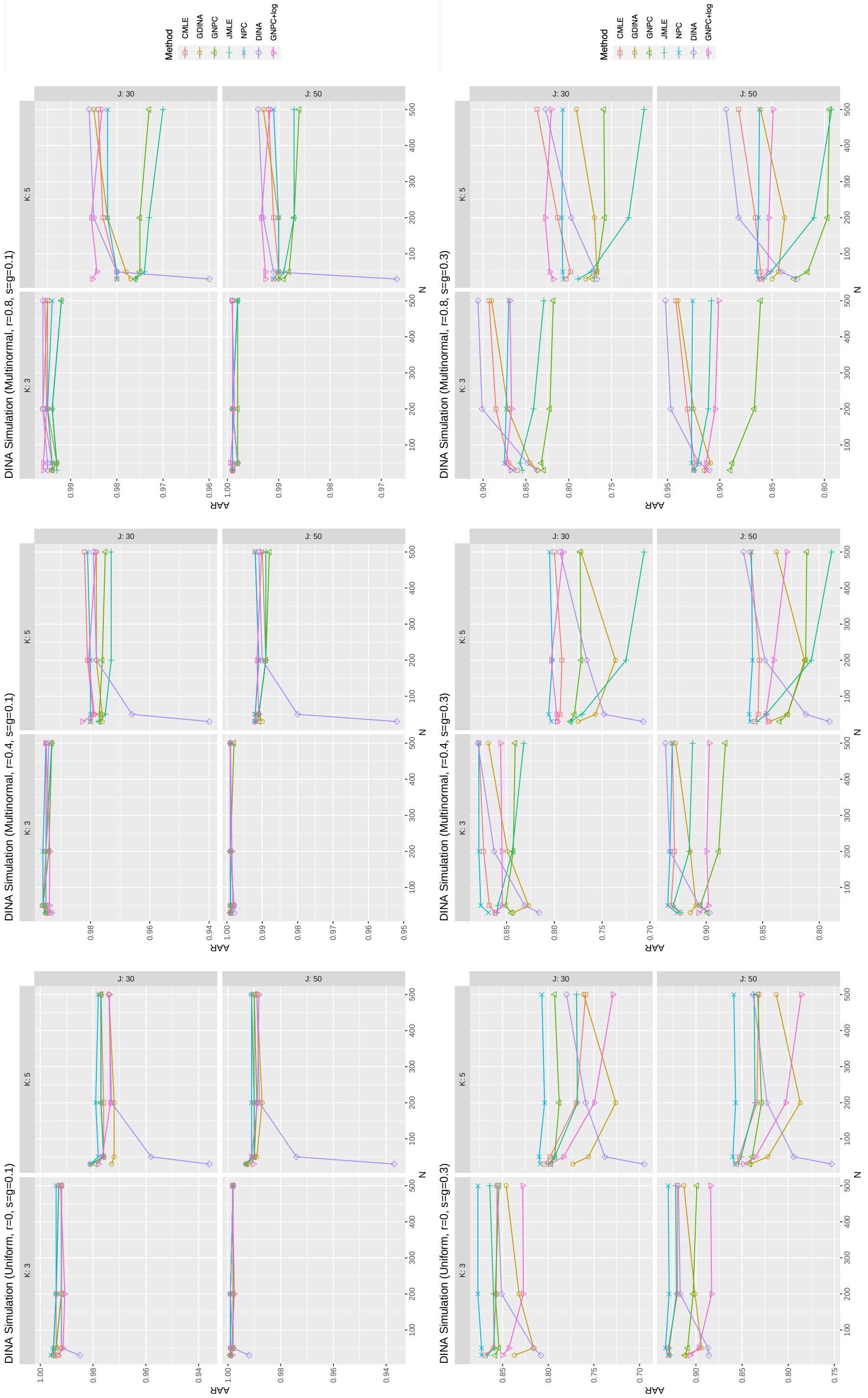


Figure V.2: AARs when the data conformed to the DINA model

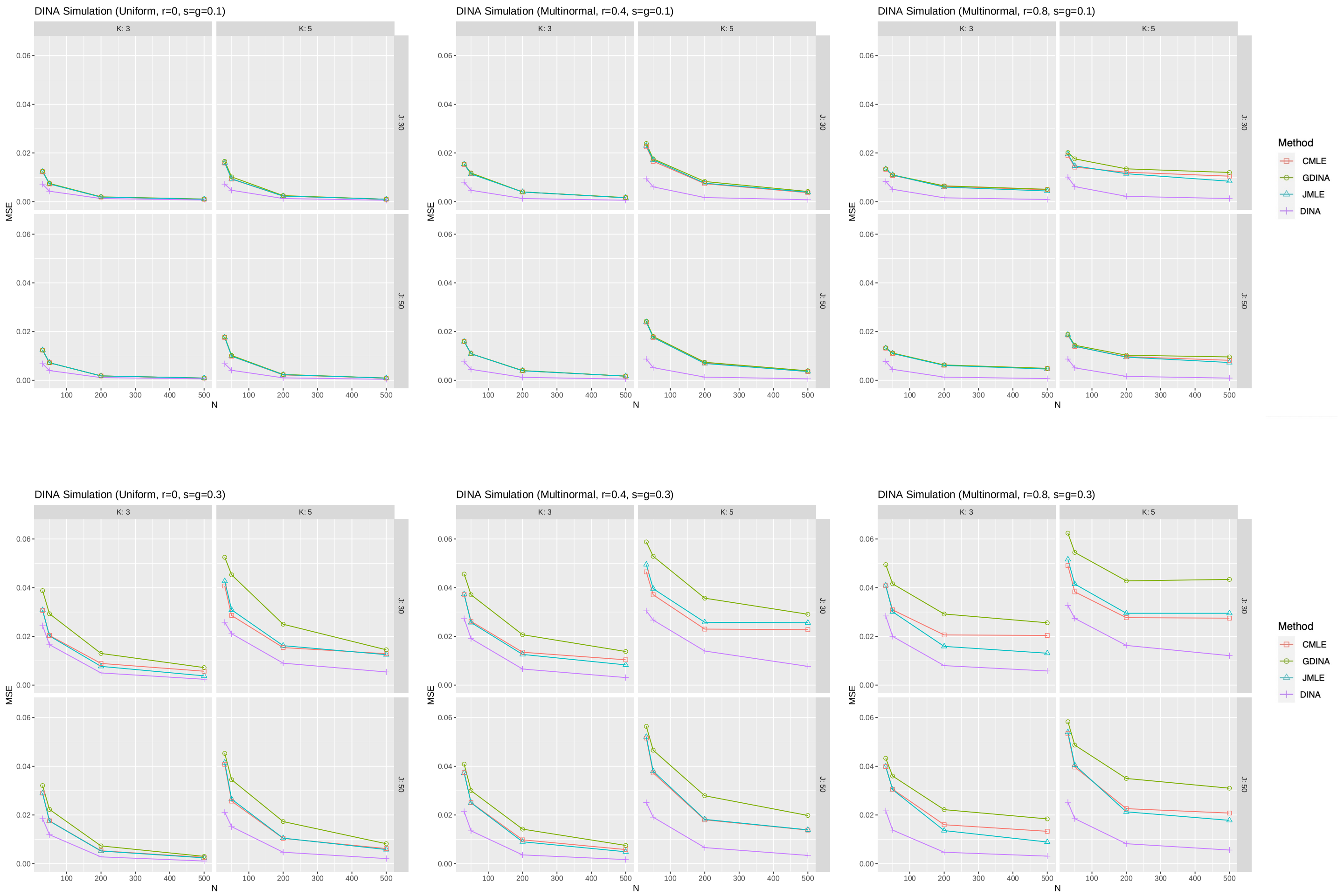


Figure V.3: MSEs when the data conformed to the DINA model

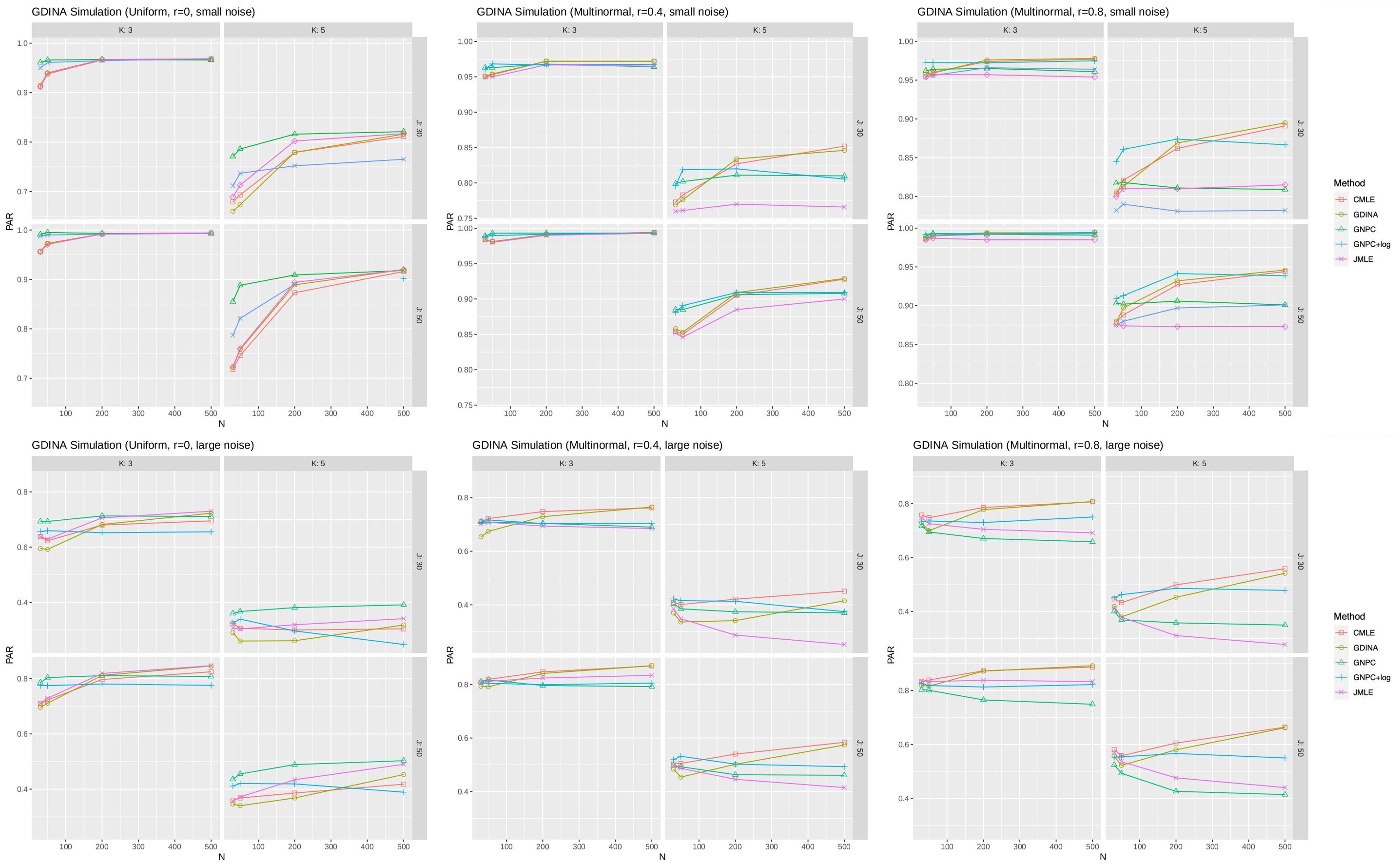


Figure V.4: PARs when the data conformed to the GDINA model

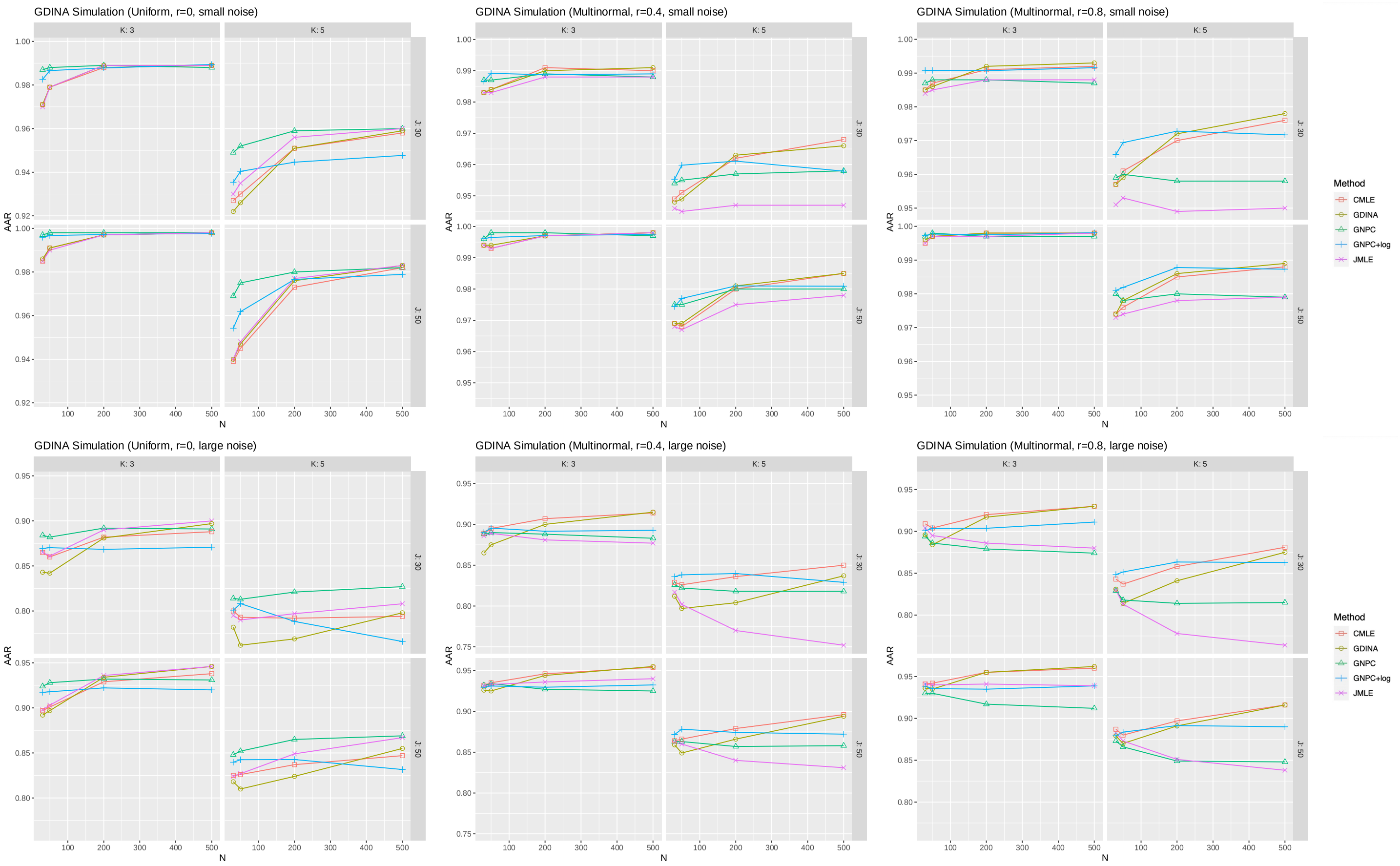


Figure V.5: AARs when the data conformed to the GDINA model

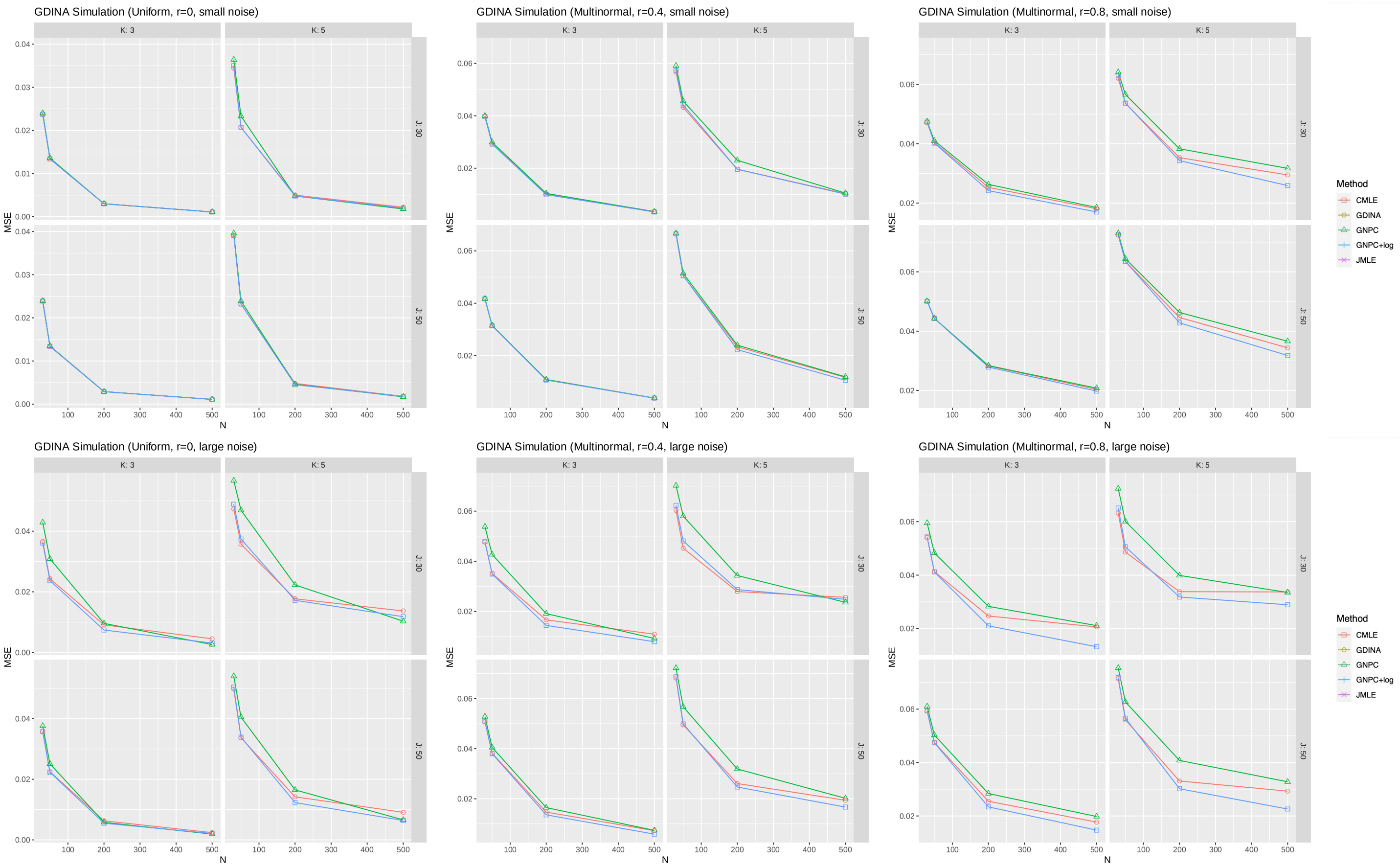


Figure V.6: MSEs when the data conformed to the GDINA model

5.6 Discussion

In this chapter, a unified estimation framework is proposed to bridge the parametric and nonparametric methods of cognitive diagnosis, and corresponding computational algorithms are developed. Specifically, by choosing different loss functions and potentially imposing additional constraints on the centroids of the proficiency classes, the proposed framework essentially provides estimations for both parametric cognitive diagnosis models and nonparametric methods for classifying subjects into proficiency classes. Moreover, we also provide theoretical analysis and establish consistency theories of the proposed framework. The simulation studies under various settings demonstrate the advantages and disadvantages of different methods

In our proposed framework (5.5), we decompose the loss function into two additive parts. In addition to the losses between the responses and class centroids, we also put a regularization term on the class proportions. The regularization term can also play a role in selecting significant latent classes in the population. For instance, similar to the CML in Examples V.3 and V.4, a log-type penalty $h(\pi_{\boldsymbol{\alpha}}) = -\lambda \log(\pi_{\boldsymbol{\alpha}})$, where $\lambda > 0$ is a tuning parameter and $\pi_{\boldsymbol{\alpha}}$ is the proportion parameter for the latent pattern $\boldsymbol{\alpha}$, can be used. Such a log-type penalty penalizes smaller proportions more heavily, and as shown in Chapter II and Chapter III, can effectively select significant latent classes in the population. Alternatively, to perform such latent class selection, the use of Lasso or elastic-net type penalty can be explored in the future.

Another interesting problem is the uncertainty quantification of the latent pattern classification. Since in the proposed framework we directly assign the latent patterns by minimizing a loss function, the subjects' latent patterns are treated as fixed effects instead of random variables. Based on the clustering literature, it is theoretically challenging to quantify the uncertainty of clustering accuracy. One practical approach is to use bootstrap, where we resample the data multiple times and use the bootstrapped samples to quantify the estimation and classification uncertainty. It is also possible to further model latent pattern probabilities and use large deviation theory to approximate the misclassification errors. For instance, Liu et al. (2015) studied the asymptotic misclassification error rate for LAMs under the assumption that the item parameters are pre-calibrated. However, in the proposed framework, the item parameters and the latent patterns are unknown and jointly estimated, and we focus on a more complicated double asymptotic regime, where the sample size N and the number of items J both go to infinity, making uncertainty quantification even more challenging. This interesting problem will be explored further in the future.

One constraint of all the methods discussed in this paper pertain to the assumption that the Q-matrix is known and accurately specified. In practice, the Q-matrix may not be given or subjectively specified by domain experts, with possible misspecifications. There are some existing methods for estimating the Q-matrix in the literature (Chen, Culpepper, Chen, and Douglas, 2018; Chen, Liu, Xu, and Ying, 2015; Chung and Johnson, 2018; Culpepper, 2019; Liu, Xu, and Ying, 2012; Li, Ma, and Xu, 2022). Developing computational methods and theories for estimating LAMs with unknown Q-matrix under our proposed general framework is a natural next step that is left for future work. Another possible extension is to consider hierarchical structures among the latent attributes as we did in Chapter II and Chapter IV, which may exclude some latent patterns in the subjects' population. Our proposed framework and computational algorithms should be easily adapted if the latent hierarchical structure is given. Our theoretical analysis will also be readily carried over to the hierarchical setting.

APPENDIX A

Appendix of Chapter II

In this appendix, we provide the proof of the main theorem in chapter II, the derivations for the penalized EM algorithm and a sensitivity analysis of our algorithm with variation of the upper bound for the number of latent classes.

A.1 Proof for Theorem 2.3.6

In this section, we provide the proof of Theorem 2.3.6.

Proof. We first introduce some notations. For two sequences $\{a_N\}$ and $\{b_N\}$, we denote $a_N \lesssim b_N$ if $a_N = O(b_N)$, and $a_N \asymp b_N$ if $a_N \lesssim b_N$ and $b_N \lesssim a_N$. We use $(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$ to denote the true model parameter and use $(\hat{\boldsymbol{\pi}}^0, \hat{\boldsymbol{\Theta}}^0)$ to denote the oracle MLE obtained by assuming the number of latent attributes, the hierarchical structure, the Q-matrix and the item-level diagnostic models are known. Let $(\hat{\boldsymbol{\pi}}^*, \hat{\boldsymbol{\Theta}}^*)$ be the MLE obtained by directly optimizing log-likelihood (2.7) and $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$ be the estimator obtained by optimizing the regularized log-likelihood (2.8). We define $\hat{\boldsymbol{\pi}}_{\rho_N} := \{\hat{\pi}_m : \hat{\pi}_m > \rho_N,\ m \in [M]\}$ and $\hat{\boldsymbol{\Theta}}_{\rho_N} := \{\hat{\theta}_{j,m} : \hat{\pi}_m > \rho_N,\ j \in [J],\ m \in [M]\}$, the model parameters corresponding to the selected latent classes. Let M be the upper bound for the number of latent classes, M_0 be the true number of latent classes, and

$\hat{M} = \big|\{m : \hat{\pi}_m > \rho_N,\ m \in [M]\}\big|$ be the estimated number of latent classes. Without loss of generality, let $\hat{\boldsymbol{\pi}}^0_{\text{full}} = (\hat{\boldsymbol{\pi}}^0, \mathbf{0}_{M-M_0})$. For the true item parameter matrix $\boldsymbol{\Theta}^0$, we defined the set of identical item parameter pairs $S^0 = \big\{(j, k_1, k_2) : \theta^0_{j,k_1} = \theta^0_{j,k_2}, 1 \leq k_1 < k_2 \leq M_0\big\}$. Similarly, for $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$ we define $\hat{S} = \big\{(j, k_1, k_2) : \hat{\theta}_{j,k_1} = \hat{\theta}_{j,k_2},\ 1 \leq k_1 < k_2 \leq M,\ \hat{\pi}_{k_1} > \rho_N,\ \hat{\pi}_{k_2} > \rho_N\big\}$. We say $\hat{S} \sim S^0$ if there exists a column permutation σ of $\hat{\boldsymbol{\Theta}}$ such that $\hat{S}_\sigma = S^0$.

The probability $\mathbb{P}\big(\hat{M} \neq M_0\big)$ can be decomposed into two parts:

$$\mathbb{P}\big(\hat{M} \neq M_0\big) = \mathbb{P}\big(\hat{M} < M_0\big) + \big(\hat{M} > M_0\big). \tag{A.1}$$

Similarly, the probability $\mathbb{P}\big(\hat{S} \neq S^0\big)$ can be decomposed into three parts:

$$\mathbb{P}\big(\hat{S} \nsim S^0\big) = \mathbb{P}\big(\hat{M} < M_0\big) + \big(\hat{M} > M_0\big) + \mathbb{P}\big(\hat{S} \nsim S^0, \hat{M} = M_0\big). \tag{A.2}$$

In the following ,we will bound each part in (A.1) and (A.2) respectively. Therefore, we will consider three cases below:

1. overfitted case: $\hat{M} > M_0$,
2. underfitted case: $\hat{M} < M_0$,
3. $\hat{M} = M_0$ but $\hat{S} \nsim S^0$.

The objective function is

$$G_N(\boldsymbol{\pi}, \boldsymbol{\Theta}) = \frac{l_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R})}{N} - \frac{\lambda^{(1)}_N}{N} \sum_{k=1}^{M} \log_{[\rho_N]} \pi_k - \frac{\lambda^{(2)}_N}{N} \sum_{j=1}^{J} \mathcal{J}_{\tau,\rho_N}(\boldsymbol{\theta}_j), \tag{A.3}$$

where $\log_{[\rho_N]} \pi_k = \log \pi_k \cdot \mathbb{I}\big(\pi_k > \rho_N\big) + \log \rho_N \cdot \mathbb{I}\big(\pi_k \leq \rho_N\big)$. Let $\log_{[\rho_N]}(\boldsymbol{\pi}) = \sum_{k=1}^{M} \log_{[\rho_N]} \pi_k$.

First consider the overfitted case where $\hat{M} > M_0$. The event $\big\{G_N\big(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}\big) >$

$G_N\big(\hat{\boldsymbol{\pi}}^0, \hat{\boldsymbol{\Theta}}^0\big)\big\}$ implies that

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\log\Big[\frac{\sum_{k=1}^{M}\hat{\pi}_k\prod_{j=1}^{J}\hat{\theta}_{j,k}^{R_{ij}}(1-\hat{\theta}_{j,k})^{1-R_{ij}}}{\sum_{k=1}^{M}\hat{\pi}_k^0\prod_{j=1}^{J}(\hat{\theta}_{j,k}^0)^{R_{ij}}(1-\hat{\theta}_{j,k}^0)^{1-R_{ij}}}\Big] \\
&> \frac{\lambda_N^{(1)}}{N}\big\{\log_{[\rho_N]}(\hat{\boldsymbol{\pi}}) - \log_{[\rho_N]}(\hat{\boldsymbol{\pi}}_{full}^0)\big\} + \frac{\lambda_N^{(2)}}{N}\Big\{\sum_{j=1}^{J}\mathcal{J}_{\tau,\rho_N}(\hat{\boldsymbol{\theta}}_j) - \sum_{j=1}^{J}\mathcal{J}_{\tau,\rho_N}(\hat{\boldsymbol{\theta}}_j^0)\Big\} \qquad \text{(A.4)} \\
&:= J_1 + J_2.
\end{aligned}
$$

For the RHS of (A.4), we have $J_1 \gtrsim N^{-1}\lambda_N^{(1)}|\log\rho_N|$ and $J_2 \gtrsim -N^{-1}\lambda_N^{(2)}\tau JM^2$. Since $\lambda_N^{(2)}\tau = o(\lambda_N^{(1)}|\log\rho_N|)$, we have RHS $\gtrsim N^{-1}\lambda_N^{(1)}|\log\rho_N|$.

For the LHS of (A.4), we have

$$
\begin{aligned}
\text{LHS of (A.4)} &= \frac{1}{N}\log\big[\sum_{k=1}^{M}\hat{\pi}_k\prod_{j=1}^{J}\hat{\theta}_{j,k}^{R_{ij}}(1-\hat{\theta}_{j,k})^{1-R_{ij}}\big] \\
&\quad - \frac{1}{N}\log\big[\sum_{k=1}^{M}\hat{\pi}_k^0\prod_{j=1}^{J}(\hat{\theta}_{j,k}^0)^{R_{ij}}(1-\hat{\theta}_{j,k}^0)^{1-R_{ij}}\big] \\
&\le \frac{1}{N}\log\big[\sum_{k=1}^{M}\hat{\pi}_k^*\prod_{j=1}^{J}(\hat{\theta}_{j,k}^*)^{R_{ij}}(1-(\hat{\theta}_{j,k}^*))^{1-R_{ij}}\big] \\
&\quad - \frac{1}{N}\log\big[\sum_{k=1}^{M}\hat{\pi}_k^0\prod_{j=1}^{J}(\hat{\theta}_{j,k}^0)^{R_{ij}}(1-\hat{\theta}_{j,k}^0)^{1-R_{ij}}\big] \\
&\lesssim N^{-\delta},
\end{aligned}
$$

where the last inequality follows from Assumption 2.3.5. When $N^{1-\delta}/|\log(\rho_N)| = o\big(\lambda_N^{(1)}\big)$, we have $N^{-\delta} = o\big(N^{-1}\lambda_N^{(1)}|\log\rho_N|\big)$, which implies that the event described in (A.4) will happen with probability tending to zero. Therefore we have $\mathbb{P}\big(\hat{M} > M_0\big) \to 0$ as $N \to \infty$. That is to say, with the appropriate choice of tuning parameters, the extent that the log-penalty part favors a smaller model would dominate the extent that the likelihood part favors a larger model in the overfitted case.

Now consider the under-fitted case where $\hat{M} < M_0$. We need to bound

$$\mathbb{P}\Big(\sup_{\hat{M}<M_0} \big[G_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}) - G_N(\hat{\boldsymbol{\pi}}^0, \hat{\boldsymbol{\Theta}}^0)\big] > 0\Big). \tag{A.5}$$

We follow a similar argument to Shen et al. (2012). More specifically, since

$$\begin{aligned} &\mathbb{P}\Big(\sup_{\hat{M}<M_0} \big[G_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}) - G_N(\hat{\boldsymbol{\pi}}^0, \hat{\boldsymbol{\Theta}}^0)\big] > 0\Big) \\ \leq &\sum_{m=1}^{M_0-1} \mathbb{P}\Big(\sup_{\hat{M}=m} \big[G_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}) - G_N(\hat{\boldsymbol{\pi}}^0, \hat{\boldsymbol{\Theta}}^0)\big] > 0\Big), \end{aligned} \tag{A.6}$$

we will bound each term in the RHS of (A.6). By the large deviation inequality in Theorem 1 of Wong et al. (1995), we have

$$\begin{aligned} &\mathbb{P}\Big(\sup_{h^2\big((\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}), (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\big) \geq \epsilon_N^2} \big[\frac{1}{N} l_N\big(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}\big) - \frac{1}{N} l_N\big(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0\big)\big] > -\epsilon_N^2\Big) \\ \leq\ &\mathbb{P}\Big(\sup_{h^2\big((\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}), (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\big) \geq \epsilon_N^2} \big[\frac{1}{N} l_N\big(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}\big) - \frac{1}{N} l_N\big(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0\big)\big] > -\epsilon_N^2\Big) \leq \exp(-N\epsilon_N^2), \end{aligned} \tag{A.7}$$

where $h^2\big((\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}), (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\big) = \sum_{\boldsymbol{R}\in\{0,1\}^J} \big[\mathbb{P}(\boldsymbol{R} \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})^{1/2} - \mathbb{P}(\boldsymbol{R} \mid \boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)^{1/2}\big]$ is the Hellinger distance. From the remark in Wong et al. (1995), the inequality (A.7) holds for any $t > \epsilon_N$.

To use this large deviation inequality, we need to introduce the notion of bracketing Hellinger metric entropy $H(t, \mathcal{B}_m)$, which characterizes the size of the local parameter space. Consider the local parameter space $\mathcal{B}_m = \big\{\big(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}\big) : \hat{M} = m \leq M_0,\ h^2\big((\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}), \big(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0\big)\big) \leq 2\epsilon_N^2\big\}$, then $H(t, \mathcal{B}_m)$ is defined as the logarithm of the cardinality of the t-bracketing of $\mathcal{B}_m$ of the smallest size. Specifically, following the definition in Shen et al. (2012), consider a bracket covering $S(t, m) = \{f_1^l, f_1^u, \cdots, f_m^l, f_m^u\}$ such that $\max_{1\leq j\leq m} ||f_j^u - f_j^l||_2 \leq t$ and for any $f \in \mathcal{B}_m$, there is some j such that

$f_j^l \leq f \leq f_j^u$ almost surely. Then $H(t, \mathcal{B}_m)$ is defined as $\log\big(\min\{m : S(t,m)\}\big)$. Following Lemma 3 in Gu and Xu (2019b), for any $2^{-4}\epsilon < t < \epsilon$, there is

$$H(t, \mathcal{B}_m) \lesssim M_0 \log M \log(2\epsilon/t). \tag{A.8}$$

Next we need to verify the conditions in Wong et al. (1995). Let's take $\epsilon_N = \sqrt{M_0 \log M/N}$ and verify the entropy integral condition in Theorem 1 of Wong et al. (1995) for such ϵ_N. The integral of bracketing Hellinger metric entropy on the interval $[2^{-8}\epsilon_N^2, \sqrt{2}\epsilon_N]$ satisfies the following inequality

$$\begin{aligned}
\int_{2^{-8}\epsilon_N^2}^{\sqrt{2}\epsilon_N} H^{1/2}(t, \mathcal{B}_m)dt &\leq \int_{2^{-8}\epsilon_N^2}^{\sqrt{2}\epsilon_N} \sqrt{M_0 \log M \log(2\epsilon_N/t)}dt \\
&= \sqrt{M_0 \log M} \int_{\sqrt{\log \sqrt{2}}}^{\sqrt{\log \frac{2^9}{\epsilon_N}}} 4\epsilon_N u^2 e^{-u^2} du \\
&= \sqrt{M_0 \log M} \cdot 2\epsilon_N \int_{\log \sqrt{2}}^{\log \frac{2^9}{\epsilon_N}} \sqrt{u} e^{-u} du \\
&\lesssim \sqrt{N}\epsilon_N^2.
\end{aligned}$$

Note that $\epsilon_N = o(1)$ as $N \to \infty$.

Following the proof in Gu and Xu (2019b), there exists a constant c_0, for some small constant $t > \epsilon_N$, we have

$$C_{\min}(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0) := \inf_{(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}): \hat{M} \leq M_0} \left\{ \frac{h^2\big((\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}), (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\big)}{\max\big(M_0 - \hat{M}, 1\big)} \right\} \geq c_0 \gtrsim t^2 > \epsilon_N^2.$$

Moreover, for $\hat{M} = m < M_0$, there is $h^2((\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}), (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)) \geq \big(M_0 - m\big) C_{\min}(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$. In order to have the probability of the event (A.4) go to zero in the under-fitted case, the log-penalty term should not be too large such that the likelihood part is dominated by the log-penalty term that favors a smaller model. Here we take $\lambda_N^{(1)} = o(N \log \rho_N|^{-1})$. Then for (A.6) we have

$$
\begin{aligned}
&\text{RHS of (A.6)} \\
&\leq \sum_{m=1}^{M_0-1} \mathbb{P}\Big(\sup_{h^2((\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}),(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)) \geq (M_0-m)C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0),\hat{M}=m} \big[G_N(\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}) - G_N(\hat{\boldsymbol{\pi}}^0,\hat{\boldsymbol{\Theta}}^0)\big] > 0\Big) \\
&\leq \sum_{m=1}^{M_0-1} \mathbb{P}\Big(\sup_{h^2((\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}),(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)) \geq (M_0-m)C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0),\hat{M}=m} \big[l_N(\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}) - l_N(\hat{\boldsymbol{\pi}}^0,\hat{\boldsymbol{\Theta}}^0)\big] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad > -\frac{\lambda_N^{(1)} M_0 |\log \rho_N|}{N}\Big) \\
&\leq \sum_{m=1}^{M_0-1} \mathbb{P}\Big(\sup_{h^2((\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}),(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)) \geq (M_0-m)C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0),\hat{M}=m} \big[l_N(\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}) - l_N(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)\big] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad > -\frac{\lambda_N^{(1)} M_0 |\log \rho_N|}{N}\Big) \\
&\leq \sum_{m=1}^{M_0-1} \mathbb{P}\Big(\sup_{h^2((\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}),(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)) \geq (M_0-m)C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0),\hat{M}=m} \big[l_N(\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}}) - l_N(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)\big] > \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad - (M_0-m)C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)\Big) \\
&\leq \sum_{m=1}^{M_0-1} \exp\big(-c_2 N (M_0-m) C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)\big) \\
&\leq c_3 \exp\big(-c_2 N C_{\min}(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)\big).
\end{aligned}
$$

Therefore we have $\mathbb{P}\big(\hat{M} < M_0\big) \to 0$ as $N \to \infty$. So far we have proved (2.10) in Theorem 2.3.6,

$$
\mathbb{P}\big(\hat{M} \neq M_0\big) = \mathbb{P}\big(\hat{M} < M_0\big) + \mathbb{P}\big(\hat{M} > M_0\big) \longrightarrow 0.
$$

Finally, we consider the third case where $\hat{M} = M_0$ but $\hat{S} \nsim S^0$. The argument is similar to the proof of Proposition 2 in Xu and Shang (2018). We first show $(\hat{\boldsymbol{\pi}}_{\rho_N}, \hat{\boldsymbol{\Theta}}_{\rho_N})$ converge to $(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$ with rate $N^{-1/2}$. For $(\boldsymbol{\pi}, \boldsymbol{\Theta})$ with $(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N})$ in a small neighborhood of $(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$,

$$\begin{aligned} G'_N(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}) :=& \frac{l_N(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}; \mathcal{R})}{N} - \frac{\lambda_N^{(1)}}{N} \sum_{k:\pi_k > \rho_N} \log \pi_k - \frac{\lambda_N^{(2)}}{N} \sum_{j=1}^{J} \mathcal{J}_{\tau,\rho_N}(\boldsymbol{\theta}_j) \\ =& \frac{l_N(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}; \mathcal{R})}{N} - O(\lambda_N^{(1)} N^{-1} |\log \rho_N|) - O(\lambda_N^{(2)} \tau N^{-1}), \end{aligned}$$

converges uniformly to the same limit of $l_N(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}; \mathcal{R})/N$ by the uniform law of large number, since $\lambda_N^{(1)} N^{-1} |\log \rho_N| \to 0$ and $\lambda_N^{(2)} \tau N^{-1} \to 0$. We use $G_0(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N})$ to denote the limit process, which is the expectation of the negative log-likelihood of a single observation. By Taylor's expansion, we have $G_0(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}) - G_0(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0) = O(\big|\big|(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N})) - (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\big|\big|^2)$.

For the log-likelihood function $l_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}; \mathcal{R}) = \sum_{i=1}^{N} \log \big(\sum_{k=1}^{M} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big)$, we have

$$\begin{aligned} & \frac{1}{N} \big| l_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}; \mathcal{R}) - l_N(\hat{\boldsymbol{\pi}}_{\rho_N}, \hat{\boldsymbol{\Theta}}_{\rho_N}; \mathcal{R}) \big| \\ \leq & \frac{1}{N} \sum_{i=1}^{N} \Big| \log \big(\sum_{k=1}^{M} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big) - \log \big(\sum_{k:\hat{\pi}_k > \rho_N} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big) \Big| \\ \leq & \frac{1}{N} \sum_{i=1}^{N} \frac{\big| \big(\sum_{k=1}^{M} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big) - \big(\sum_{k:\hat{\pi}_k > \rho_N} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big) \big|}{\sqrt{\big(\sum_{k=1}^{M} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big) \times \big(\sum_{k:\hat{\pi}_k > \rho_N} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}}) \big)}} \qquad \text{(A.9)} \\ \leq & \frac{1}{N} \sum_{i=1}^{N} \frac{(M - \hat{M}) \rho_N}{\sum_{k:\hat{\pi}_k > \rho_N} \hat{\pi}_k \prod_{j=1}^{J} \hat{\theta}_{j,k}^{R_{ij}} (1 - \hat{\theta}_{j,k}^{1-R_{ij}})} \\ = & O(\rho_N) = O(N^{-d}), \ d \geq 1, \qquad \text{(A.10)} \end{aligned}$$

where inequality (A.9) follows from an upper bound for log function. Specifically, for $x \geq 1$, we know $\log x \leq (x-1)/\sqrt{x}$, and thus for $0 < x \leq y$, we have $\log y - \log x \leq (y-x)/\sqrt{xy}$. From (A.10), $G'_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}) = G'_N(\hat{\boldsymbol{\pi}}_{\rho_N}, \hat{\boldsymbol{\Theta}}_{\rho_N}) + O(N^{-d}) \geq G'_N(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)$ and thus $G'_N(\hat{\boldsymbol{\pi}}_{\rho_N}, \hat{\boldsymbol{\Theta}}_{\rho_N}) > G'_N(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0) - O(N^{-d}) \geq G'_N(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0) - O(N^{-1})$. Since $N^{-1/2} \lambda_N^{(1)} \to$

0 and $N^{-1/2}\lambda_N^{(2)}\tau \to 0$, then for sufficiently small ζ, by Taylor's expansion,

$$\begin{aligned}
&\mathbb{E}\Big(\sup_{||(\boldsymbol{\pi}_{\rho_N},\boldsymbol{\Theta}_{\rho_N})-(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)||\le\zeta}\big[G_N'(\boldsymbol{\pi}_{\rho_N},\boldsymbol{\Theta}_{\rho_N};\mathcal{R}) - G_0(\boldsymbol{\pi}_{\rho_N},\boldsymbol{\Theta}_{\rho_N}) \\
&\qquad\qquad - G_N'(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0;\mathcal{R}) + G_0(\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0)\big]\Big) \\
=&O(\zeta N^{-1/2}).
\end{aligned}$$

By Theorem 3.2.5 in Vaart and Wellner (1996), $(\hat{\boldsymbol{\pi}}_{\rho_N},\hat{\boldsymbol{\Theta}}_{\rho_N}) - (\boldsymbol{\pi}^0,\boldsymbol{\Theta}^0) = O_p(N^{-1/2})$.

We next show the selection consistency of S^0. If true item parameters $\theta^0_{j,k_1} \neq \theta^0_{j,k_2}$, then from the above convergence result, we know $\hat{\theta}_{j,k_1} \to \theta^0_{j,k_1}$ and $\hat{\theta}_{j,k_2} \to \theta^0_{j,k_2}$, and thus $\hat{\theta}_{j,k_1} \neq \hat{\theta}_{j,k_2}$ in probability. If true item parameters $\theta^0_{j,k_1} = \theta^0_{j,k_2}$ but $\hat{\theta}_{j,k_1} \neq \hat{\theta}_{j,k_2}$, by the Karush-Kuhn-Tucker (KKT) conditions, we have $N^{-1/2}\partial l_N(\boldsymbol{\pi},\boldsymbol{\Theta};\mathcal{R})/\partial\theta_{j,k_1}|_{(\boldsymbol{\pi},\boldsymbol{\Theta})=(\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}})}$ $= N^{-1/2}\lambda_N^{(2)} \to \infty$ in probability. However $N^{-1/2}\partial l_N(\boldsymbol{\pi},\boldsymbol{\Theta};\mathcal{R})/\partial\theta_{j,k_1}|_{(\boldsymbol{\pi},\boldsymbol{\Theta})=(\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\Theta}})} = O_p(1)$. Therefore, if $\theta^0_{j,k_1} = \theta^0_{j,k_2}$, we have $\hat{\theta}_{j,k_1} = \hat{\theta}_{j,k_2}$ in probability, which proved the selection consistency that $\mathbb{P}(\hat{S} \nsim S^0) \to 0$ as $N \to \infty$. □

A.2 Derivations of PEM Algorithm

In this section, we give detailed derivations of the penalized EM algorithm in Section 2.4.1. First let's introduce a new variable $\boldsymbol{d} = (d_{jkl}, j = 1,\ldots,J, 1 \le k < l \le M)$ to be the differences of the item parameters for each item. Then our problem becomes

$$\begin{aligned}
\min_{\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d}} \quad & G(\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{d}) \\
\text{s.t.} \quad & d_{jkl} = \theta_{jk} - \theta_{jl} \\
& j = 1,\ldots,J,\ 1 \le k < l \le M.
\end{aligned} \tag{A.11}$$

By using the difference convex property of the truncated Lasso penalty, we can decompose the objective function into two parts:

$$G(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d}) = G_1(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d}) - G_2(\boldsymbol{d}), \tag{A.12}$$

where

$$G_1(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d}) = -\frac{1}{N} Q(\boldsymbol{\pi}, \boldsymbol{\Theta} | \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)}) + \tilde{\lambda}_1 \sum_{k=1}^{M} \log \pi_k + \tilde{\lambda}_2 \sum_{j=1}^{J} \sum_{1 \le k < l \le M} |d_{jkl}|, \tag{A.13}$$

$$G_2(\boldsymbol{d}) = \tilde{\lambda}_2 \sum_{j=1}^{J} \sum_{1 \le k < l \le M} \big(|d_{jkl} - \tau|\big)_+. \tag{A.14}$$

Then we construct a sequence of upper approximation of $G(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d})$ iteratively by replacing $G_2(\boldsymbol{d})$ at iteration $c+1$ with its piecewise affine minorization:

$$G_2^{(c)}(\boldsymbol{d}) = G_2(\hat{\boldsymbol{d}}^{(c)}) + \tilde{\lambda}_2 \sum_{j=1}^{J} \sum_{1 \le k < l \le M} \big(|d_{jkl}| - |\hat{d}_{jkl}^{(c)}|\big) \cdot \mathbb{I}\big(|\hat{d}_{jkl}^{(c)}| \ge \tau\big), \tag{A.15}$$

at the current estimate $\hat{\boldsymbol{d}}^{(c)}$, which lead to an upper convex approximation:

$$\begin{aligned} G^{(c+1)}(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d}) = & -\frac{1}{N} Q(\boldsymbol{\pi}, \boldsymbol{\Theta} | \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)}) + \tilde{\lambda}_1 \sum_{k=1}^{M} \log \pi_k \\ & + \tilde{\lambda}_2 \sum_{j=1}^{J} \sum_{1 \le k < l \le M} |d_{jkl}| \cdot \mathbb{I}\big(|\hat{d}_{jkl}^{(c)}| < \tau\big) \\ & + \tilde{\lambda}_2 \tau \sum_{j=1}^{J} \sum_{1 \le k < l \le M} \mathbb{I}\big(|\hat{d}_{jkl}^{(c)}| \ge \tau\big). \end{aligned}$$

Now we can apply ADMM. At iteration $c+1$, the augmented Lagrangian is

$$\begin{aligned} L_\gamma(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d}, \boldsymbol{y}) = G^{(c+1)}(\boldsymbol{\pi}, \boldsymbol{\Theta}, \boldsymbol{d}) & + \sum_{j=1}^{J} \sum_{1 \le k < l \le M} y_{jkl} \cdot \big(d_{jkl} - (\theta_{jk} - \theta_{jl})\big) \\ & + \frac{\gamma}{2} \sum_{j=1}^{J} \sum_{1 \le k < l \le M} \big|d_{jkl} - (\theta_{jk} - \theta_{jl})\big|^2, \end{aligned}$$

where y_{jkl}'s are the dual variables and γ is a nonnegative penalty parameter. Then ADMM (Boyd et al., 2011) consists of the following iterations:

$$
\begin{aligned}
\boldsymbol{\pi}^{(c+1)} &= \operatorname*{argmin}_{\boldsymbol{\pi}} L_{\gamma}(\boldsymbol{\pi}, \boldsymbol{\Theta}^{(c)}, \boldsymbol{d}^{(c)}, \boldsymbol{y}^{(c)}),\\
\boldsymbol{\Theta}^{(c+1)} &= \operatorname*{argmin}_{\boldsymbol{\Theta}} L_{\gamma}(\boldsymbol{\pi}^{(c+1)}, \boldsymbol{\Theta}, \boldsymbol{d}^{(c)}, \boldsymbol{y}^{(c)}),\\
\boldsymbol{d}^{(c+1)} &= \operatorname*{argmin}_{\boldsymbol{d}} L_{\gamma}(\boldsymbol{\pi}^{(c+1)}, \boldsymbol{\Theta}^{(c+1)}, \boldsymbol{d}, \boldsymbol{y}^{(c)}),\\
y_{jkl}^{(c+1)} &= y_{jkl}^{(c)} + \gamma(d_{jkl}^{(c+1)} - \big(\theta_{jk}^{(c+1)} - \theta_{jl}^{(c+1)})\big),\ j = 1, ..., J, 1 \le k < l \le M.
\end{aligned}
$$

Using the scaled Lagrangian multiplier $\mu_{jkl} = y_{jkl}/\gamma$ and defining the residual $r_{jkl} = d_{jkl} - (\theta_{jk} - \theta_{jl})$, we have:

$$
\begin{aligned}
&y_{jkl} \cdot \big(d_{jkl} - (\theta_{jk} - \theta_{jl})\big) + \frac{\gamma}{2}\big|d_{jkl} - (\theta_{jk} - \theta_{jl})\big|^2\\
=&y_{jkl} \cdot r_{jkl} + \frac{\gamma}{2} r_{jkl}^2\\
=&\frac{\gamma}{2}\big(r_{jkl} + (1/\gamma)y_{jkl}\big)^2 - \frac{1}{2\gamma}\mu_{jkl}^2\\
=&\frac{\gamma}{2}\big(r_{jkl} + \mu_{jkl}\big)^2 - \frac{1}{2\gamma}\mu_{jkl}^2.
\end{aligned}
$$

Then using the scaled dual variable, we can express ADMM as:

$$
\begin{aligned}
\boldsymbol{\pi}^{(c+1)} &= \operatorname*{argmin}_{\boldsymbol{\pi}} G^{(c+1)}(\boldsymbol{\pi}, \boldsymbol{\Theta}^{(c)}, \boldsymbol{d}^{(c)}),\\
\boldsymbol{\Theta}^{(c+1)} &= \operatorname*{argmin}_{\boldsymbol{\Theta}} \Big\{ G^{(c+1)}(\boldsymbol{\pi}^{(c+1)}, \boldsymbol{\Theta}, \boldsymbol{d}^{(c)})\\
&\qquad + \frac{\gamma}{2} \sum_{j=1}^{J} \sum_{1 \le k < l \le M} \big(d_{jkl}^{(c)} - (\theta_{jk}^{(c)} - \theta_{jl}^{(c)}) + \mu_{jkl}^{(c)}\big)\Big\},\\
\boldsymbol{d}^{(c+1)} &= \operatorname*{argmin}_{\boldsymbol{d}} \Big\{ G^{(c+1)}(\boldsymbol{\pi}^{(c+1)}, \boldsymbol{\Theta}^{(c+1)}, \boldsymbol{d})\\
&\qquad + \frac{\gamma}{2} \sum_{j=1}^{J} \sum_{1 \le k < l \le M} \big(d_{jkl} - (\theta_{jk}^{(c+1)} - \theta_{jl}^{(c+1)}) + \mu_{jkl}^{(c)}\big)\Big\},\\
\mu_{jkl}^{(c+1)} &= \mu_{jkl}^{(c)} + d_{jkl}^{(c+1)} - \big(\theta_{jk}^{(c+1)} - \theta_{jl}^{(c+1)}\big),\ j = 1, \ldots, J,\ 1 \le k < l \le M.
\end{aligned}
$$

Specifically, we get the following updates:

(1)

$$\pi_k^{(c+1)} = \frac{\sum_{i=1}^N s_{ik}^{(c+1)}/N - \tilde{\lambda}_1}{1 - M\tilde{\lambda}_1}, \quad \text{where } s_{ik}^{(c+1)} = \frac{\pi_k^{(c)} \varphi_k(\boldsymbol{R}_i; \boldsymbol{\Theta}_k^{(c)})}{\sum_{k'}^{(c)} \pi_{k'}^{(c)} \varphi_{k'}^{(c)}(\boldsymbol{R}_i; \boldsymbol{\theta}_{k'}^{(c)})}.$$

(2)

$$\begin{aligned}\hat{\theta}_{jk}^{(c+1)} = \underset{\theta_{jk}}{\operatorname{argmin}} \Big\{ & - \frac{\sum_{i=1}^N s_{ik}^{(c)} R_{ij}}{N} \log \theta_{jk} - \frac{\sum_{i=1}^N s_{ik}^{(c)} (1 - R_{ij})}{N} \log(1 - \theta_{jk}) \\ & + \frac{\gamma}{2} \sum_{l>k} \big(\hat{d}_{jkl}^{(c)} - (\theta_{jk} - \hat{\theta}_{jl}^{(c)}) + \hat{\mu}_{jkl}^{(c)} \big)^2 \\ & + \frac{\gamma}{2} \sum_{l<k} \big(\hat{d}_{jlk}^{(c)} - (\hat{\theta_{jl}}^{(c+1)} - \theta_{jk}) + \hat{\mu}_{jlk}^{(c)} \big)^2 \Big\}.\end{aligned}$$

(3)

$$\hat{d}_{jkl}^{(c+1)} = \begin{cases} \hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}, & \text{if } |\hat{d}_{jkl}^{(c)}| \geq \tau \\ \mathrm{ST}\big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}; \tilde{\lambda}_2/\gamma\big), & \text{if } |\hat{d}_{jkl}^{(c)}| < \tau \end{cases},$$

where $\mathrm{ST}(x; \gamma) = (|x| - \gamma)_+ x/|x|$.

(4)

$$\hat{\mu}_{jkl}^{(c+1)} = \hat{\mu}_{jkl}^{(c)} + \hat{d}_{jkl}^{(c+1)} - \big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)}\big).$$

Note that the objective in step (2) is convex in θ_{jk}, therefore we use gradient descent to perform the minimization.

A.3 PEM Algorithm with Missing Values

In this section, we present the penalized EM algorithm with missing values. Here we use a mask matrix $M \in \{0, 1\}^{N \times J}$ to indicate the locations of the missing values, where $M_{i,j} = 0$ means the ith subject's response to the jth item is missing. The details of the algorithm is summarized in Algorithm A.1.

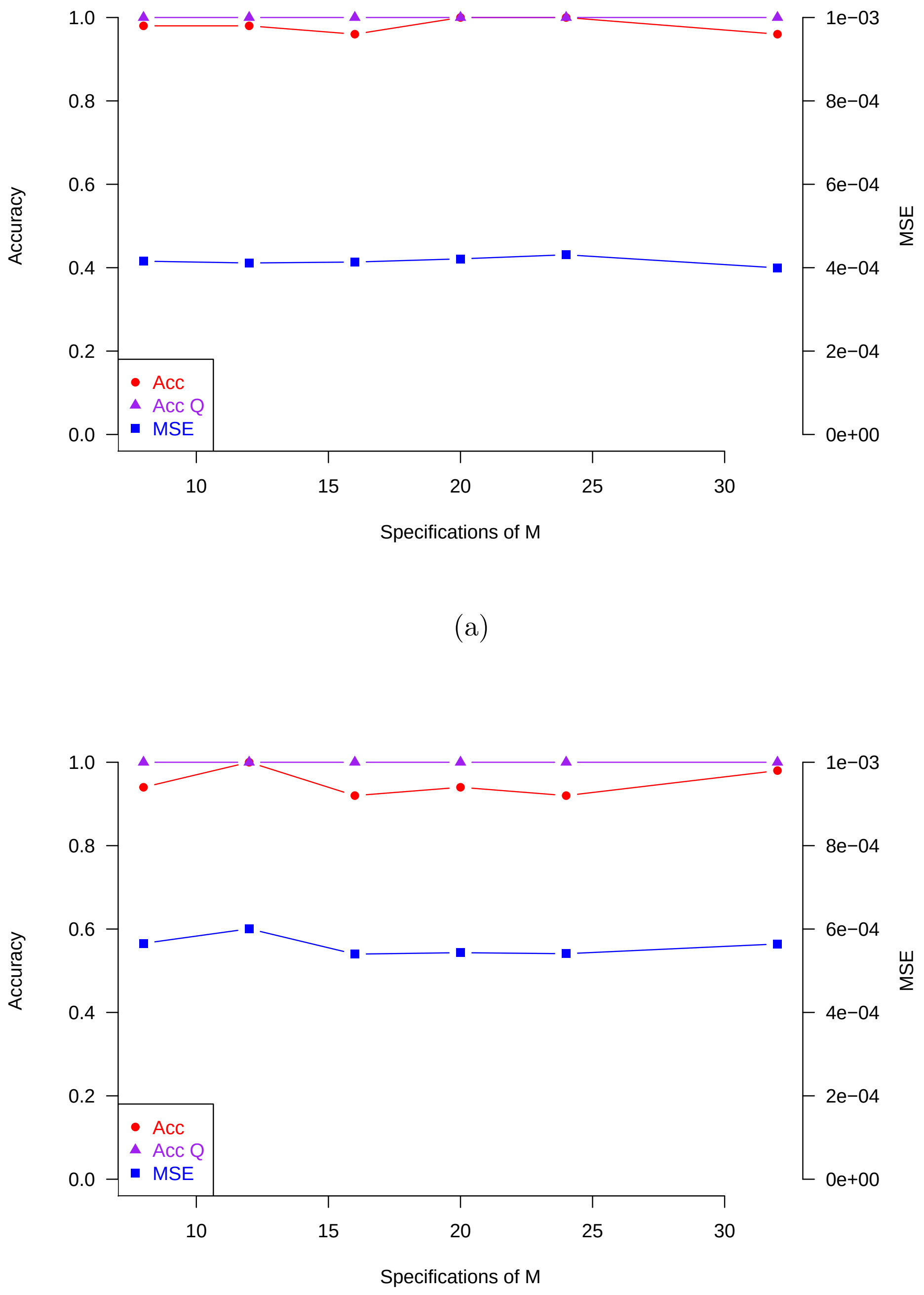
Algorithm A.1: Penalized EM with missing data

```
Data: Binary response matrix $\mathcal{R} = (R_{i,j})_{N \times J}$ and the mask matrix
      $\boldsymbol{M} = (M_{ij})_{N \times J}$ indicating missing values.
Set hyperparameters $\tilde{\lambda}_1,\ \tilde{\lambda}_2,\ \tau,\ \gamma$ and $\rho$.
Set an upper bound of the number of latent classes $L$.
Initialize parameters $\boldsymbol{\pi}$, $\boldsymbol{\Theta}$, and the conditional expectations $\boldsymbol{s}$.
while not converged do
    In the $(c+1)$th iteration,
    for $(i,k) \in [N] \times [L]$ do
        $s_{ik}^{(c+1)} = \frac{\pi_k^{(c)} \varphi_k(\boldsymbol{R}_i; \boldsymbol{\theta}_k^{(c)})}{\sum_{k'}^{(c)} \pi_{k'}^{(c)} \varphi_{k'}^{(c)}(\boldsymbol{R}_i; \boldsymbol{\theta}_{k'}^{(c)})}, \quad \varphi(\boldsymbol{r}_i; \boldsymbol{\theta}_k) = \prod_{j=1}^{J} \big(\theta_{jk}^{R_{ij}} (1-\theta_{kj})^{1-R_{ij}}\big)^{m_{ij}}$
    end
    for $k \in [L]$ and $\pi_k^{(c)} > \rho$ do
        $\pi_k^{(c+1)} = \frac{\sum_{i=1}^{N} s_{ik}^{(c+1)}/N - \tilde{\lambda}_1}{1 - L\tilde{\lambda}_1}$.
    end
    for $(j,k) \in [J] \times [L]$ and $\pi_k^{(c+1)} > \rho$ do
        $$\theta_{jk}^{(c+1)} = \underset{\theta_{jk}}{\mathrm{argmin}} \Big\{ - \frac{\sum_{i=1}^{N} s_{ik}^{(c)} R_{ij} m_{ij}}{\sum_{i=1}^{N} m_{ij}} \log \theta_{jk}
            - \frac{\sum_{i=1}^{N} s_{ik}^{(c)} (1-_{ij}) m_{ij}}{\sum_{i=1}^{N} m_{ij}} \log(1-\theta_{jk})
            + \frac{\gamma}{2} \sum_{l>k} \big(\hat{d}_{jkl}^{(c)} - (\theta_{jk} - \hat{\theta}_{jl}^{(c)}) + \hat{\mu}_{jkl}^{(c)}\big)^2
            + \frac{\gamma}{2} \sum_{l<k} \big(\hat{d}_{jlk}^{(c)} - (\hat{\theta}_{jl}^{(c+1)} - \theta_{jk}) + \hat{\mu}_{jlk}^{(c)}\big)^2 \Big\}$$
    end
    for $j \in [J], k, l \in [L],\ k < l$ and $\pi_k^{(c+1)} > \rho$, $\pi_l^{(c+1)} > \rho$ do
        $$\hat{d}_{jkl}^{(c+1)} = \begin{cases} \hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}, & \text{if } |\hat{d}_{jkl}^{(c)}| \geq \tau \\ \mathrm{ST}\big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}; \tilde{\lambda}_2/\gamma\big), & \text{if } |\hat{d}_{jkl}^{(c)}| < \tau \end{cases},$$
        $$\hat{\mu}_{jkl}^{(c+1)} = \hat{\mu}_{jkl}^{(c)} + \hat{d}_{jkl}^{(c+1)} - \big(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)}\big).$$
    end
end
Output: $\big\{\hat{\boldsymbol{\pi}},\ \hat{\boldsymbol{\Theta}},\ \hat{\boldsymbol{s}}\big\}$
```

A.4 Sensitivity Analysis

In this section, we conduct the sensitivity analysis of our algorithm by investigating the effects of different inputs of M, the upper bound of the number of latent classes, on the simulation results. In particular, we focus on two simulation settings: (1) DINA model with linear hierarchical structure, $N = 500$ and $r = 0.1$; (2) GDINA model with linear hierarchical structure, $N = 1000$ and $r = 0.1$. Both two settings have $K = 4$ latent attributes and $J = 30$ test items, and run for 50 repetitions. We keep the parameter generation process and the hyperparameter tuning strategy consistent with the simulation studies in the main article. In this sensitivity analysis, we fit our proposed method with various $M = \{8, 12, 16, 20, 24, 32\}$ in the two simulations settings. The evaluation results in DINA and GDINA settings are based on metrics Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$), Acc($\hat{\mathcal{E}}$), MSE($\hat{\boldsymbol{\Theta}}$) and Acc($\hat{\boldsymbol{Q}}$). Consistent with the simulation studies in the main article, the Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$) and Acc($\hat{\mathcal{E}}$) are calculated for all the cases; MSE($\hat{\boldsymbol{\Theta}}$) is calculated for the cases when the number of latent classes is correctly selected; Acc($\hat{\boldsymbol{Q}}$) is calculated for the cases when the hierarchical structure is successfully recovered. The results are plotted in Figure A.1.

From the simulation results in Figure A.1, we see our proposed method is robust to the different specifications of M, in terms of all metrics. Among cases with different M, our method achieves a high accuracy in estimating the number of latent classes, and in recovering the partial orders, the hierarchical structures, the item parameter matrix, and the Q-matrix. In terms of computation time, the average running time is 0.36 seconds and 1.12 seconds for DINA and GDINA, respectively, per repetition per set of tuning hyperparameters.



(a)

(b)

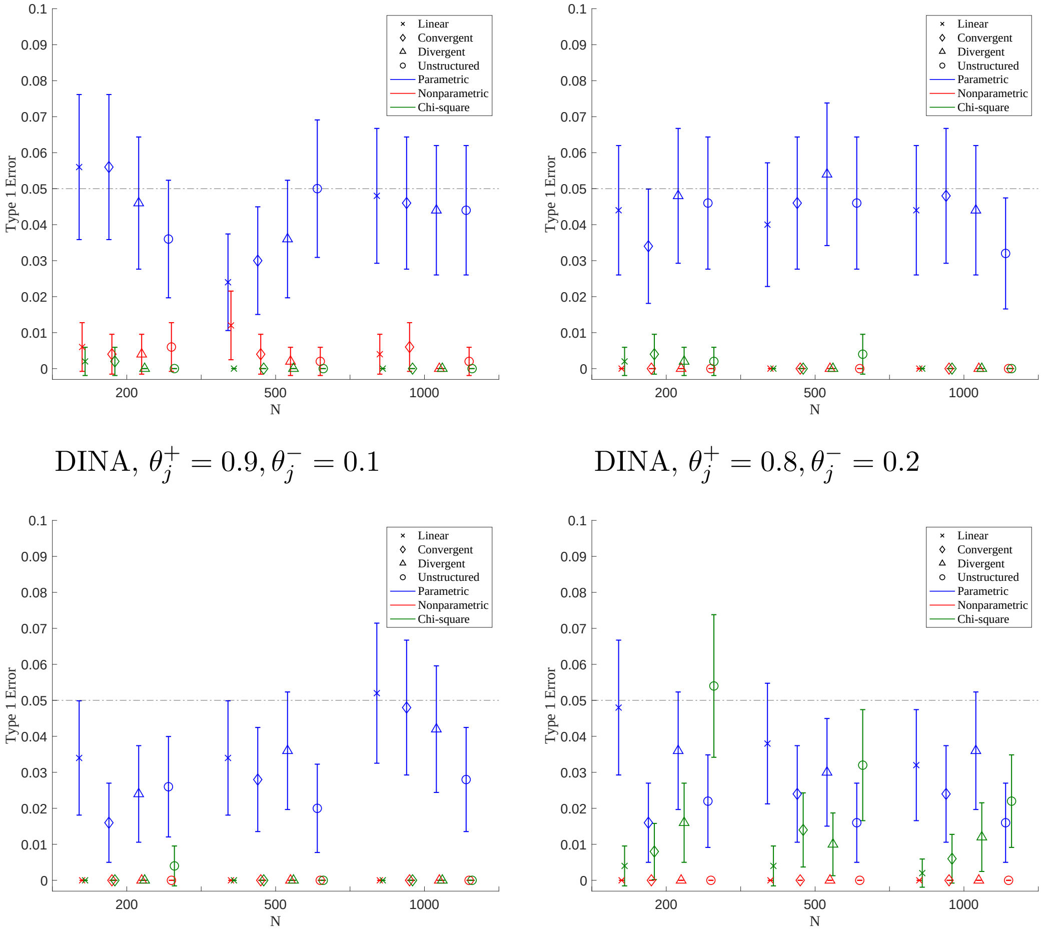
Figure A.1: Sensitivity analysis results. (a) DINA results; (b) GDINA results. The red curve captures the Acc($\hat{M}$), Acc($\hat{\boldsymbol{P}}$), Acc($\hat{\mathcal{E}}$), the blue curve captures MSE($\hat{\boldsymbol{\Theta}}$) and the purple curve captures the Acc($\hat{\boldsymbol{Q}}$) for various M.

APPENDIX B

Appendix of Chapter IV

This appendix includes Type 1 Errors in Section 4.3.2, the Q-matrix for ECPE data in Section 4.4 and additional simulation results. Specifically, bootstrap results for DINA and GDINA models under both null hypothesis and alternative hypothesis with different sample sizes and noise levels are presented in Figures B.2 – B.5.

B.1 Type 1 Errors



DINA, $\theta_j^+ = 0.9, \theta_j^- = 0.1$

DINA, $\theta_j^+ = 0.8, \theta_j^- = 0.2$

GDINA, $\theta_j^+ = 0.9, \theta_j^- = 0.1$

GDINA, $\theta_j^+ = 0.8, \theta_j^- = 0.2$

Figure B.1: Type 1 Errors. Different colors indicate different testing procedures. Different marker shapes stand for different hierarchical structures. The middle points are the means of the type I errors and the vertical errors bars with ±2 s.e. are constructed based on 500 replications. $\theta_j^+ = 0.9$, $\theta_j^- = 0.1$ corresponds to the case with low noises, and $\theta_j^+ = 0.8$, $\theta_j^- = 0.2$ corresponds to the case with high noises.

B.2 Q-matrix for ECPE data

Item	Attributes		
	Mor.rules (α_1)	Coh.rules (α_2)	Lex.rules (α_3)
1	1	1	0
2	0	1	0
3	1	0	1
4	0	0	1
5	0	0	1
6	0	0	1
7	1	0	1
8	0	1	0
9	0	0	1
10	1	0	0
11	1	0	1
12	1	0	1
13	1	0	0
14	1	0	0
15	0	0	1
16	1	0	1
17	0	1	1
18	0	0	1
19	0	0	1
20	1	0	1
21	1	0	1
22	0	0	1
23	0	1	0
24	0	1	0
25	1	0	0
26	0	0	1
27	1	0	0
28	0	0	1

Table B.1: The Q-matrix for ECPE data. “Mor.” is short for “morphosyntactic”, “Coh.” is short for “cohesive”, and “Lex.” is short for “lexical”.

B.3 Bootstrap Results under the DINA model

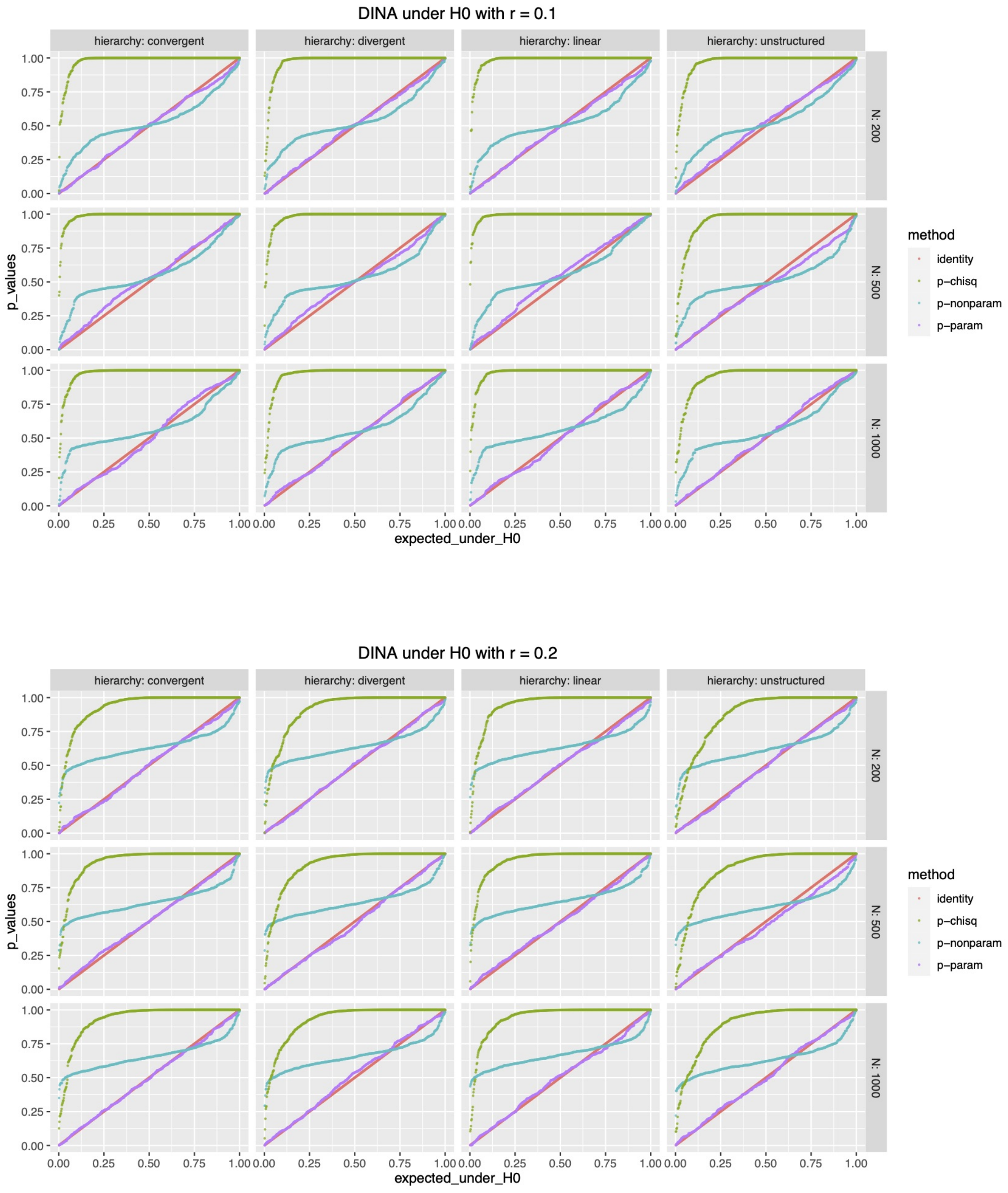


Figure B.2: Bootstrap results for DINA under null hypothesis

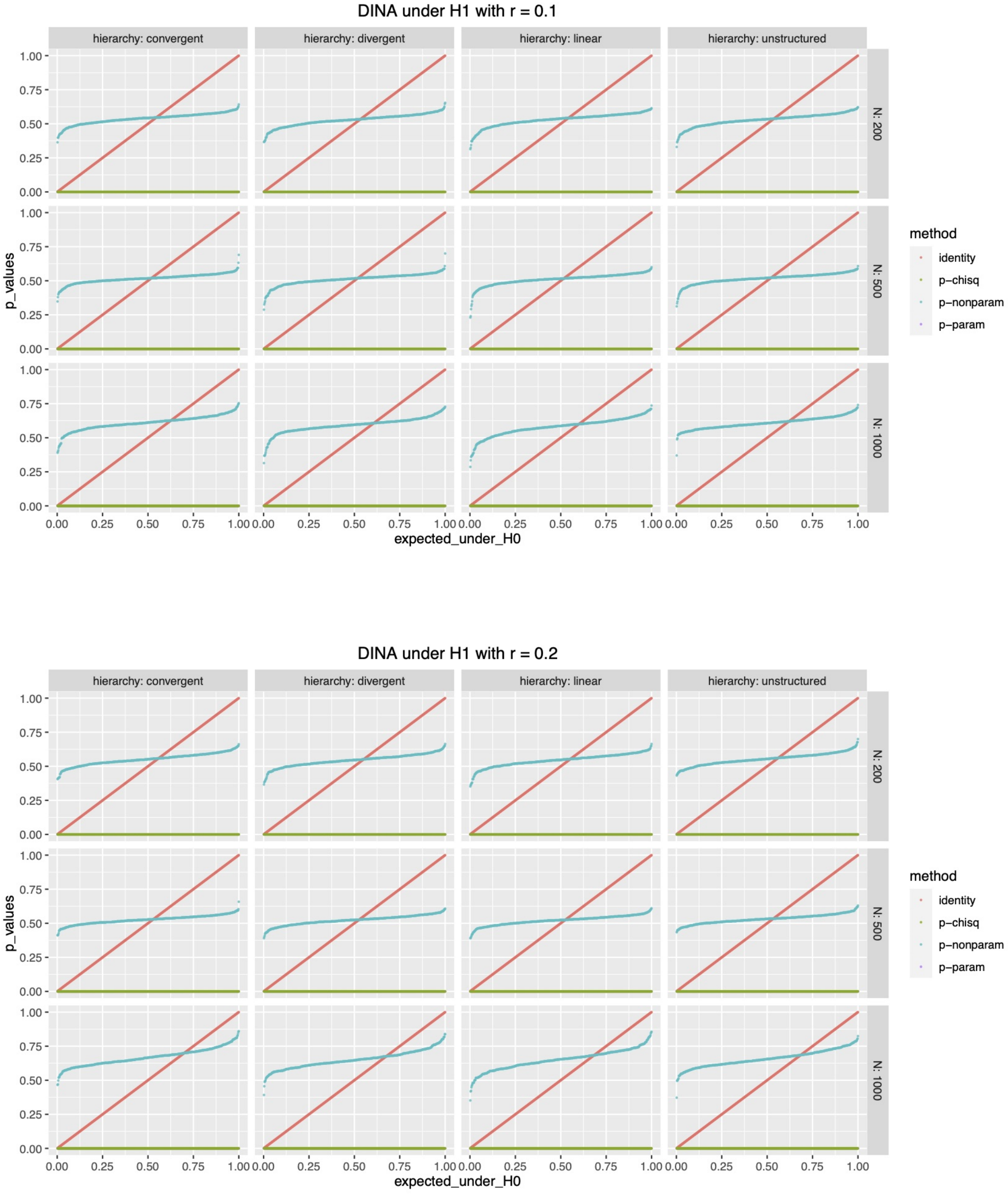
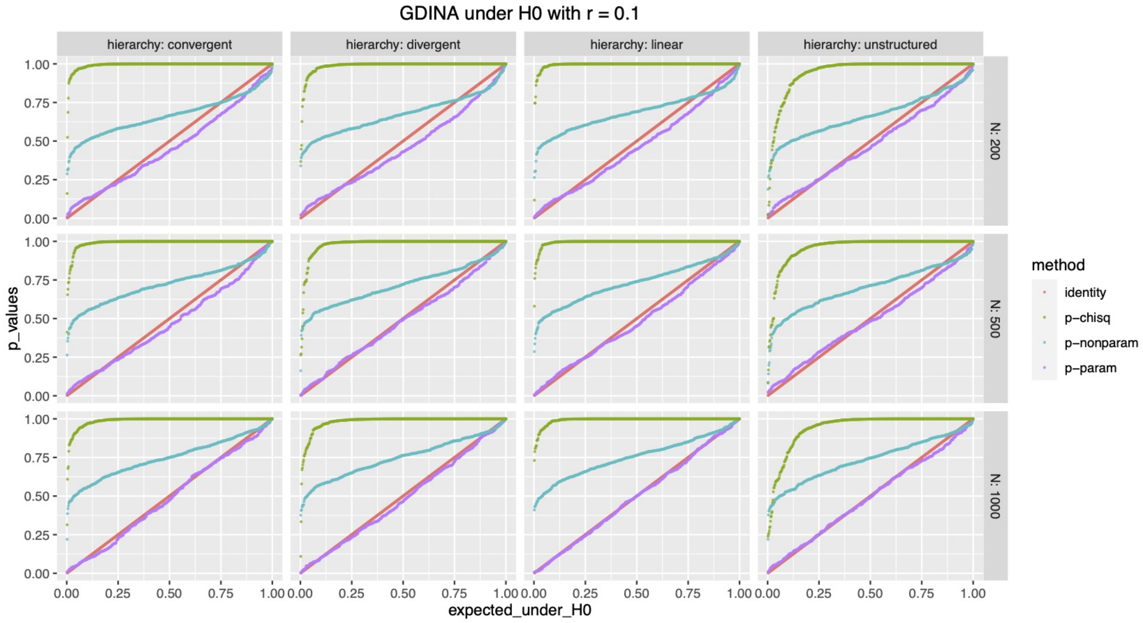


Figure B.3: Bootstrap results for DINA under alternative hypothesis

B.4 Bootstrap Results under the GDINA model



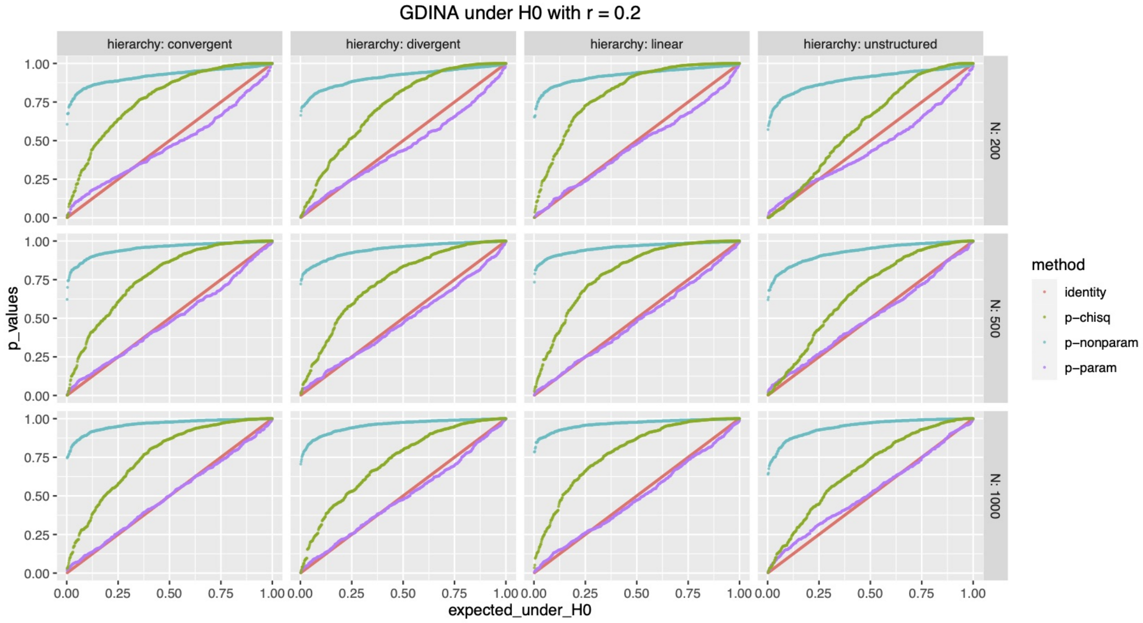


Figure B.4: Bootstrap results for GDINA under null hypothesis

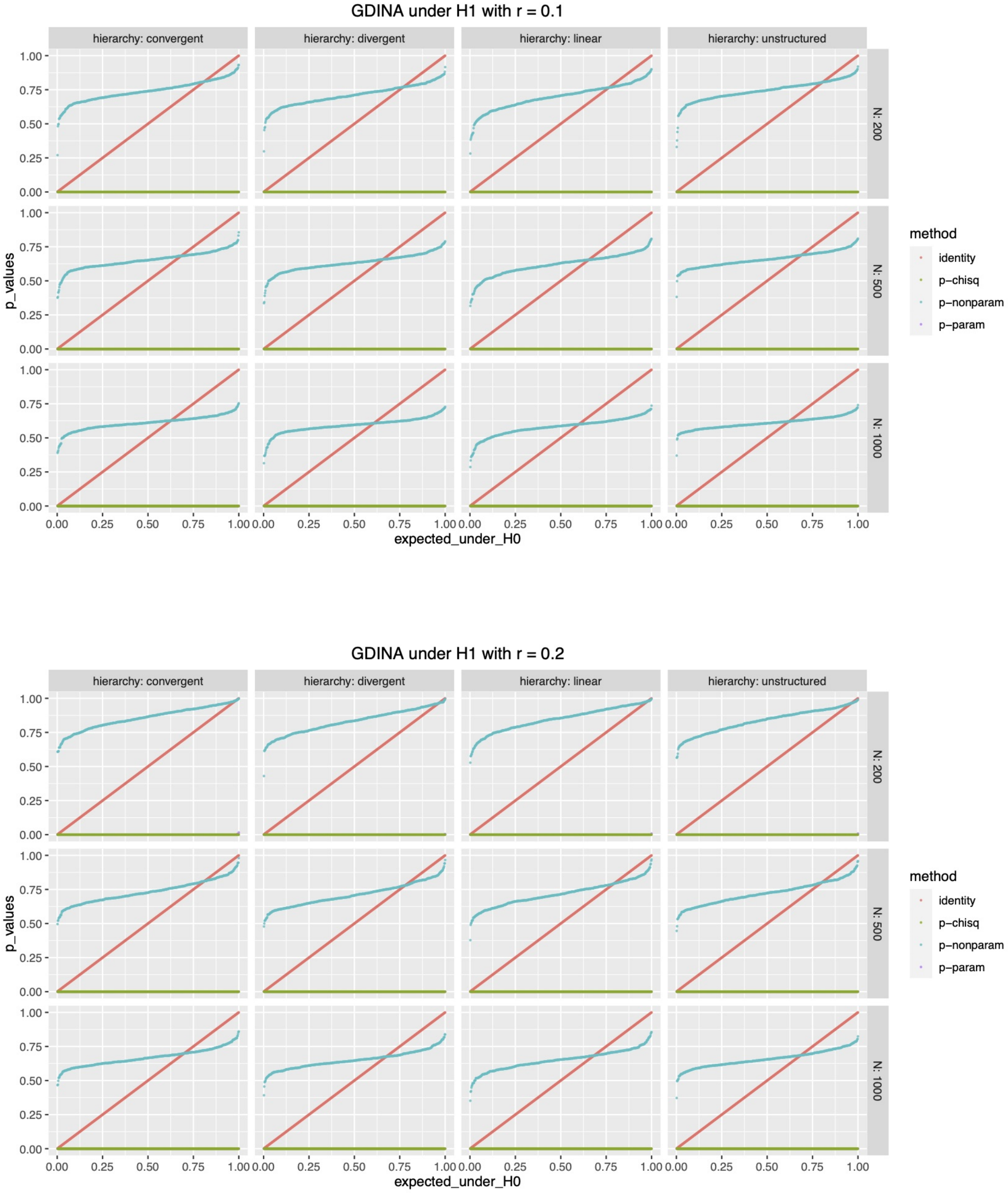


Figure B.5: Bootstrap results for GDINA under alternative hypothesis

APPENDIX C

Appendix of Chapter V

In this appendix section, we provide detailed proofs of the Lemmas and Theorems in Section 5.4.

C.1 Proof of Theorem 5.4.5

Proof. Our proof uses similar arguments as in Chiu et al. (2016). First consider the case when the true membership $\boldsymbol{A}_c^0$ is known. Since $\hat{\mu}_{j,\boldsymbol{\alpha}} = \sum_{i\in C_{\boldsymbol{\alpha}}} r_{ij}/|C_{\boldsymbol{\alpha}}| := \bar{r}_{j,\boldsymbol{\alpha}}$, by Hoeffding's inequality (Hoeffding, 1994), for any $\epsilon > 0$,

$$\begin{aligned} P\big(\|\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}_{\boldsymbol{\alpha}}^0\|_\infty \geq \epsilon \bigm| \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\big) &= P\Big(\max_j |\bar{r}_{j,\boldsymbol{\alpha}} - \theta_{j,\boldsymbol{\alpha}}^0| \geq \epsilon \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\ &\leq \sum_{j=1}^J P\Big(|\bar{r}_{j,\boldsymbol{\alpha}} - \theta_{j,\boldsymbol{\alpha}}^0| \geq \epsilon \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\ &\leq\ 2J\exp\big(-2|C_{\boldsymbol{\alpha}}|\cdot\epsilon^2\big). \end{aligned}$$

Since $\lim_{n\to\infty}|C_{\boldsymbol{\alpha}}|/N_c \to \pi_{\boldsymbol{\alpha}}$ almost surely and $J\exp\big(-N_c\epsilon\big) \to 0$ for any $\epsilon > 0$, we have $J\exp\big(-2|C_{\boldsymbol{\alpha}}|\cdot\epsilon^2\big) = J\exp\Big(-2\big(1+o(1)\big)N_c\cdot\pi_{\boldsymbol{\alpha}}\cdot\epsilon^2\Big) \to 0$ almost surely.

Now consider the case when $\hat{\boldsymbol{A}}_c$ is consistent for $\boldsymbol{A}_c^0$, that is, $P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0) \to 0$.

Then for any $\epsilon > 0$, we have

$$\begin{aligned}
& P\big(\|\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}}\|_\infty \geq \epsilon\big) \\
\leq & P\big(\|\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}}\|_\infty \geq \epsilon \bigm| \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\big) \cdot P\big(\hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\big) \\
& + P\big(\|\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}}\|_\infty \geq \epsilon \bigm| \hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c\big) \cdot P\big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c\big) \\
\leq & P\big(\|\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}}\|_\infty \geq \epsilon \bigm| \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\big) + P\big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c\big) \\
\xrightarrow{P} & \, 0, \quad \text{as } J \to \infty.
\end{aligned}$$

Therefore we have $\|\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}}\|_\infty \xrightarrow{P} 0$. Since there are finitely many $\boldsymbol{\alpha}$'s, we have $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\theta}^0\|_\infty \xrightarrow{P} 0$. □

C.2 Proof of Lemma 5.4.7

Proof. Let $\tilde{\boldsymbol{\alpha}}_i$ denote the latent attribute pattern that minimizes $E[l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}) + h(\hat{\pi}_{\boldsymbol{\alpha}})]$, that is,

$$\begin{aligned}
\tilde{\boldsymbol{\alpha}}_i := & \arg\min_{\boldsymbol{\alpha}} \big\{E\big[l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}) + h(\hat{\pi}_{\boldsymbol{\alpha}})\big]\big\} \\
= & \arg\min_{\boldsymbol{\alpha}} \, E\Big[\sum_{j=1}^{J} l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}}) + h(\hat{\pi}_{\boldsymbol{\alpha}})\Big] \\
= & \arg\min_{\boldsymbol{\alpha}} \Big\{\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}})\big] + \frac{1}{J} h(\hat{\pi}_{\boldsymbol{\alpha}})\Big\}.
\end{aligned}$$

For the second term, under the Assumption 2, since $\hat{\pi}_{\boldsymbol{\alpha}}$ is asymptotically bounded and $h(\cdot)$ is continuous, hence $h(\hat{\pi}_{\boldsymbol{\alpha}})$ is also bounded, and we have $h(\hat{\pi}_{\boldsymbol{\alpha}})/J \to 0$ as $J \to \infty$, which is asymptotically negligible. For the first term, we need to compare $\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}})\big]$ and $\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}^0_i})\big]$ for any $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}^0_i$.

$$\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}})\big] - \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}^0_i})\big]$$

$$
\begin{aligned}
&= \Big(\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}})\big] - \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big]\Big) \\
&\quad + \Big(\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big] - \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}^0_i})\big]\Big) \qquad \text{(C.1)} \\
&\quad + \Big(\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}^0_i})\big] - \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}^0_i})\big]\Big) \\
&:= E_1 + E_2 + E_3. \qquad \text{(C.2)}
\end{aligned}
$$

Since $\hat{\boldsymbol{\mu}}$ is consistent for $\boldsymbol{\theta}^0$, by Assumption 1, we have $E_1 \xrightarrow{P} 0$ and $E_3 \xrightarrow{P} 0$. Specifically, first consider the case when $\hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c$. By Assumption 5.4.3, we know that the true item parameters are bounded. There exists $\delta_2 \in (0, 0.5)$ such that $\delta_2 \le \min_{j,\boldsymbol{\alpha}} \theta^0_{j,\boldsymbol{\alpha}} < \max_{j,\boldsymbol{\alpha}} \theta^0_{j,\boldsymbol{\alpha}} \le 1-\delta_2, \forall 1 \le j \le J, \boldsymbol{\alpha} \in \{0,1\}^K$. Let's now look at the probability that $\hat{\mu}_{j,\boldsymbol{\alpha}}$ is also bounded. Specifically, we consider $P(\hat{\mu}_{j,\boldsymbol{\alpha}} \ge 1-\delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c)$ and $P(\hat{\mu}_{j,\boldsymbol{\alpha}} \le \delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c)$ respectively. Since $\hat{\mu}_{j,\boldsymbol{\alpha}} = \sum_{i \in C_{\boldsymbol{\alpha}}} r_{ij}/|C_{\boldsymbol{\alpha}}| := \bar{r}_{j,\boldsymbol{\alpha}}$, we have

$$
\begin{aligned}
P(\hat{\mu}_{j,\boldsymbol{\alpha}} \ge 1-\delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c) = {} & P(\bar{r}_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}} \ge 1-\delta_2/2-\theta^0_{j,\boldsymbol{\alpha}} \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c) \\
\le {} & \exp\big(-2|C_{\boldsymbol{\alpha}}|(1-\delta_2/2-\theta^0_{j,\boldsymbol{\alpha}})^2\big) \\
\le {} & \exp\big(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2\big).
\end{aligned}
$$

Similarly, we also have $P(\hat{\mu}_{j,\boldsymbol{\alpha}} \le \delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c) \le \exp\big(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2\big)$. Therefore,

$$
P\big(\min_j \hat{\mu}_{j,\boldsymbol{\alpha}} \le \delta_2/2 \text{ or } \max_j \hat{\mu}_{j,\boldsymbol{\alpha}} \ge 1-\delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\big) \le 2J\exp\big(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2\big).
$$

Moreover, since under the Assumption 5.4.1, the loss function is assumed to be Hölder continuous, that is, there exist $c > 0$ and $\beta > 0$, such that for any $\mu_1, \mu_2 \in (\delta_2/2, 1-$

$\delta_2/2)$, we have $|l(x,\mu_1) - l(x,\mu_2)| \leq c|\mu_1 - \mu_2|^\beta$ for $x = 0$ or 1. Then

$$\begin{aligned}
|E_1| &= \Big|\frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}})\big] - \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big]\Big| \\
&\leq \frac{1}{J}\sum_{j=1}^{J} E\Big[\big|l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big|\Big] \\
&\leq \frac{1}{J}\sum_{j=1}^{J} E\big[c|\hat{\mu}_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}}|^\beta\big] \\
&\leq c\max_j\{E\big[|\hat{\mu}_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}}|^\beta\big]\}
\end{aligned}$$

Therefore for any $\epsilon > 0$,

$$\begin{align*}
&P(|E_1| > \epsilon) \\
\leq\ & P(|E_1| > \epsilon \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c) + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \\
\leq\ & P(E_1| > \epsilon \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c, \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} < 1 - \delta_2/2, j = 1,\ldots,J) \\
& + P\big(\min_j \hat{\mu}_{j,\boldsymbol{\alpha}} \leq \delta_2/2 \text{ or } \max_j \hat{\mu}_{j,\boldsymbol{\alpha}} \geq 1 - \delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\big) + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \\
\leq\ & P(||\hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}}||_\infty > (\epsilon/c)^{1/\beta}) + 2J\exp(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2) + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \\
\leq\ & 2J\exp(-2|C_{\boldsymbol{\alpha}}|(\epsilon/c)^{2/\beta}) + 2J\exp(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2) + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \tag{C.3} \\
=\ & 2J\exp\Big(-2\big(1+o(1)\big)N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot (\epsilon/c)^{2/\beta}\Big) + 2J\exp\Big(-\big(1+o(1)\big)N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot \delta_2^2/2\Big) \\
& \quad + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \tag{C.4} \\
\longrightarrow\ & 0,
\end{align*}$$

where (C.3) follows from Theorem 1. Similarly we can show that $E_3 \xrightarrow{P} 0$ as well.

For the second term, by Assumption 5.4.4, we have

$$E_2 = \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big] - \frac{1}{J}\sum_{j=1}^{J} E\big[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}^0_i})\big] \geq \frac{1}{J}\sum_{j=1}^{J} |\theta^0_{j,\boldsymbol{\alpha}^0_i} - \theta^0_{j,\boldsymbol{\alpha}}|^\delta, \tag{C.5}$$

for any $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}_i^0$. Since in Assumption 5.4.3, there exists $\delta_1 > 0$ such that $\lim\limits_{J\to\infty} \min\limits_{\boldsymbol{\alpha}\neq\boldsymbol{\alpha}'} ||\boldsymbol{\theta}^0_{\boldsymbol{\alpha}} - \boldsymbol{\theta}^0_{\boldsymbol{\alpha}'}||_1/J > \delta_1$, then for a small enough $c_0 > 0$, there exists $c_1 > 0$ such that $\big|\{j : |\theta^0_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}'}| \geq c_0\}\big| \geq c_1 J$ for any $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}'$ and large enough J. That is, there should be as many items as of order J that can differentiate two different classes. Otherwise, $\big|\{j : |\theta^0_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}'}| \geq c_0\}\big|/J \to 0$, which contradicts with Assumption 5.4.3 for a small enough c_0. Then in (C.5), we have $E_2 \geq c_1 c_0^\delta$ as $J \to \infty$. Therefore, the true attribute pattern minimizes $E[l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}; \hat{\pi}_{\boldsymbol{\alpha}})]$ with probability approaching 1. □

C.3 Proof of Lemma 5.4.8

Proof. We first decompose the probability in Lemma 5.4.8 into two parts:

$$\begin{aligned}
&P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})]\big)\Big| \geq \epsilon\Big)\\
\leq\ &P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2\Big)\\
&+ P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}}) - E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})]\big)\Big| \geq \epsilon/2\Big).
\end{aligned} \tag{C.6}$$

The first term in (C.6) goes to zero since $\hat{\boldsymbol{\theta}}$ is uniform consistent for $\boldsymbol{\theta}^0$. Specifically, from Lemma 1, we have $P(\hat{\mu}_{j,\boldsymbol{\alpha}} \leq \delta_2/2$ or $\hat{\mu}_{j,\boldsymbol{\alpha}} \geq 1 - \delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0) \leq 2\exp\big(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2\big)$. Moreover, due to the Hölder continuity of the loss function, we have $|l(x,\mu_1) - l(x,\mu_2)| \leq c|\mu_1 - \mu_2|^\beta$ for $x = 0$ or 1. Then

$$\begin{aligned}
&P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2 \;\Big|\; \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} \leq 1-\delta_2/2, \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big)\\
\leq &\sum_{\boldsymbol{\alpha}} P\Big(\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2 \;\Big|\; \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} \leq 1-\delta_2/2, \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big)\\
\leq &\, 2^K \sum_{j=1}^{J} P\Big(\Big|l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\Big| \geq \epsilon/2 \;\Big|\; \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} \leq 1-\delta_2/2, \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big)
\end{aligned}$$

$$
\begin{aligned}
&\leq 2^K \sum_{j=1}^{J} P\Big(\Big|\hat{\mu}_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}}\Big|^{\beta} \geq \epsilon/2c \;\Big|\; \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} \leq 1-\delta_2/2, \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\Big) \\
&= 2^K \sum_{j=1}^{J} P\Big(\Big|\bar{r}_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}}\Big| \geq (\epsilon/2c)^{1/\beta} \;\Big|\; \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} \leq 1-\delta_2/2, \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\Big) \\
&\leq 2^{K+1} J \exp\big(-2|C_{\boldsymbol{\alpha}}|(\epsilon/2c)^{2/\beta}\big).
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2 \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\Big) \\
&\leq \sum_{\boldsymbol{\alpha}}\sum_{j=1}^{J}\Big[P(\hat{\mu}_{j,\boldsymbol{\alpha}} < \delta_2/2 \text{ or } \hat{\mu}_{j,\boldsymbol{\alpha}} > 1-\delta_2/2 \mid \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c) \\
&\qquad\qquad + P\big(\big|\hat{\mu}_{j,\boldsymbol{\alpha}} - \theta^0_{j,\boldsymbol{\alpha}}\big| \geq (\epsilon/2c)^{1/\beta} \mid \delta_2/2 < \hat{\mu}_{j,\boldsymbol{\alpha}} \leq 1-\delta_2/2,\ \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\big)\Big] \\
&\leq 2^{K+1} J \exp(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2) + 2^{K+1} J \exp(-2|C_{\boldsymbol{\alpha}}|(\epsilon/2c)^{2/\beta}) \\
&= 2^{K+1} J \exp\Big(-\big(1+o(1)\big)N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot \delta_2^2/2\Big) \\
&\quad + 2^{K+1} J \exp\Big(-2\big(1+o(1)\big)N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot (\epsilon/2c)^{2/\beta}\Big) \\
&\longrightarrow 0, \text{ as } J \to \infty.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
&P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2\Big) \\
&\leq P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2 \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\Big) \cdot P(\hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c) \\
&\quad + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \\
&\leq P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})\big)\Big| \geq \epsilon/2 \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}^0_c\Big) + P(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}^0_c) \\
&\longrightarrow 0, \text{ as } J \longrightarrow \infty.
\end{aligned}
$$

Next we need to bound the second term. By Assumption 5.4.3, $\theta^0_{j,\boldsymbol{\alpha}}$'s are uniformly bounded and thus $l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})$'s are also uniformly bounded. There exists $M > 0$ such that $\left|l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})\right| \leq M$ for any j and $\boldsymbol{\alpha}$. Then by Hoeffding's inequality (Hoeffding, 1994), we have

$$P\Big(\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}}) - E[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})]\big)\Big| \geq \epsilon/2\Big) \leq 2\exp\big(-J\epsilon^2/2M^2\big),$$

and therefore

$$\begin{aligned} & P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}}) - E[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})]\big)\Big| \geq \epsilon/2\Big) \\ \leq\ & \sum_{\boldsymbol{\alpha}} P\Big(\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}}) - E[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}})]\big)\Big| \geq \epsilon/2\Big) \\ \leq\ & 2^{K+1}\exp\big(-J\epsilon^2/2M^2\big) \longrightarrow 0, \text{ as } J \to \infty. \end{aligned}$$

□

C.4 Proof of Theorem 5.4.6

Proof. Since $\hat{\boldsymbol{A}}_c$ is consistent for $\boldsymbol{A}^0_c$, by Theorem 5.4.5, $\hat{\boldsymbol{\mu}}$ is consistent for $\boldsymbol{\theta}^0$. Note that $\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}^0_i$ is equivalent to that

$$\frac{1}{J}\sum_{j=1}^{J} l(r_{ij}, \hat{\mu}_{j,\boldsymbol{\alpha}^0_i}) + \frac{1}{J}h(\hat{\pi}_{\boldsymbol{\alpha}^0_i}) > \frac{1}{J}\sum_{j=1}^{J} l(r_{ij}, \hat{\mu}_{j,\hat{\boldsymbol{\alpha}}_i}) + \frac{1}{J}h(\hat{\pi}_{\hat{\boldsymbol{\alpha}}_i}). \tag{C.7}$$

From Assumptions 5.4.1 and 5.4.4 and the proof of Lemma 5.4.7, we know

$$\frac{1}{J}\sum_{j=1}^{J} E[l(r_{ij}, \theta^0_{j,\boldsymbol{\alpha}^0_i})] < \frac{1}{J}\sum_{j=1}^{J} E[l(r_{ij}, \theta^0_{j,\hat{\boldsymbol{\alpha}}_i})] - c_1 c_0^{\delta} \tag{C.8}$$

Let $c_2 = c_1 c_0^{\delta}$ and take $\epsilon = c_2/4$ in Lemma 5.4.8, and consider the event

$$B_{\epsilon}(J) := \big\{\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}})]\big)\Big| < \epsilon\big\}.$$

Since $h(\hat{\pi}_{\alpha})$ is bounded, there exists some J_0 such that for any $J \geq J_0$, we have $\big|\frac{1}{J}h(\hat{\pi}_{\boldsymbol{\alpha}_i^0}) - \frac{1}{J}h(\hat{\pi}_{\hat{\boldsymbol{\alpha}}_i})\big| < c_2/4$. When $B_{c_2/4}(J)$ occurs, it implies that

$$\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}_i^0}) - E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}_i^0})]\big)\Big| < c_2/4,$$

and

$$\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\hat{\boldsymbol{\alpha}}_i}) - E[l(r_{ij},\theta^0_{j,\hat{\boldsymbol{\alpha}}_i})]\big)\Big| < c_2/4.$$

Then in equation (C.7),

$$\text{LHS} < \frac{1}{J}\sum_{j=1}^{J}E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}_i^0})] + c_2/4 + \frac{1}{J}h(\hat{\pi}_{\boldsymbol{\alpha}_i^0}),$$

and

$$\text{RHS} > \frac{1}{J}\sum_{j=1}^{J}E[l(r_{ij},\theta^0_{j,\hat{\boldsymbol{\alpha}}_i})] - c_2/4 + \frac{1}{J}h(\hat{\pi}_{\hat{\boldsymbol{\alpha}}_i}),$$

which implies that

$$\begin{aligned}\frac{1}{J}\sum_{j=1}^{J}E[l(r_{ij},\theta^0_{j,\hat{\boldsymbol{\alpha}}_i})] &< \frac{1}{J}\sum_{j=1}^{J}E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}_i^0})] + c_2/2 + \frac{1}{J}h(\pi_{\boldsymbol{\alpha}_i^0}) - \frac{1}{J}h(\pi_{\hat{\boldsymbol{\alpha}}_i})\\ &< \frac{1}{J}\sum_{j=1}^{J}E[l(r_{ij},\theta^0_{j,\boldsymbol{\alpha}_i^0})] + 3c_2/4\\ &< \frac{1}{J}\sum_{j=1}^{J}E[l(r_{ij},\theta^0_{j,\hat{\boldsymbol{\alpha}}_i})],\end{aligned}$$

where the last inequality is from equation (C.7) and results in a contradiction. It indicates that $\{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\} \subset B_{c_2/4}(J)^c$ for J large enough. And therefore we have

$$
\begin{aligned}
P\Big(\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\Big) \leq & P\big(B_{c_2/4}(J)^c\big) \\
& \leq P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - E[l(r_{ij},\theta_{j,\boldsymbol{\alpha}}^0)]\big)\Big| \geq c_2/4\Big) \\
& \longrightarrow 0, \text{ as } J \to \infty. \quad \text{(by Lamma 5.4.8)}
\end{aligned}
$$

□

C.5 Proof of Theorem 5.4.9

Proof. Following the proof of Theorem 5.4.6, we have

$$
\begin{aligned}
& P\Big(\bigcup_i \{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\} \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\
\leq & \sum_i P\Big(\{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\} \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\
\leq & N \cdot P\Big(B_{c_2/4}(J)^c \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\
\leq & N \cdot P\Big(\max_{\boldsymbol{\alpha}}\Big|\frac{1}{J}\sum_{j=1}^{J}\Big(l(r_{ij},\hat{\mu}_{j,\boldsymbol{\alpha}}) - E\big[l(r_{ij},\theta_{j,\boldsymbol{\alpha}}^0)\big]\Big)\Big| \geq c_2/4 \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\
\leq & 2^{K+1}NJ\exp(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2) + 2^{K+1}NJ\exp(-2|C_{\boldsymbol{\alpha}}|(c_2/8c)^{2/\beta}) \\
& + 2^{K+1}N\exp\big(-Jc_2^2/32M^2\big) \\
\leq & 2^{K+1}N^2\exp(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2) + 2^{K+1}N^2\exp(-2|C_{\boldsymbol{\alpha}}|(c_2/8c)^{2/\beta}) \\
& + 2^{K+1}N\exp\big(-Jc_2^2/32M^2\big).
\end{aligned}
$$

Under the Assumption 2, we have $\lim_{n\to\infty}|C_{\boldsymbol{\alpha}}|/N_c \to \pi_{\boldsymbol{\alpha}}$ almost surely; therefore $N^2\exp(-|C_{\boldsymbol{\alpha}}|\delta_2^2/2) = N^2\exp\Big(-\big(1+o(1)\big)N_c\cdot\pi_{\boldsymbol{\alpha}}\cdot\delta_2^2/2\Big)$ and $N^2\exp(-2|C_{\boldsymbol{\alpha}}|(c_2/8c)^{2/\beta})$

$= N^2 \exp\Big(-2\big(1+o(1)\big)N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot (c_2/8c)^{2/\beta}\Big)$. Then we have

$$
\begin{aligned}
&P\Big(\bigcup_i \{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\}\Big) \\
&\leq P\Big(\bigcup_i \{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\} \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) P\Big(\hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) \\
&\quad + P\Big(\bigcup_i \{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\} \;\Big|\; \hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0\Big) P\Big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0\Big) \\
&\leq P\Big(\bigcup_i \{\hat{\boldsymbol{\alpha}}_i \neq \boldsymbol{\alpha}_i^0\} \;\Big|\; \hat{\boldsymbol{A}}_c = \boldsymbol{A}_c^0\Big) + P\Big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0\Big) \\
&\leq 2^{K+1} N^2 \exp\Big(-\big(1+o(1)\big)N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot \delta_2^2/2\Big) \\
&\quad + 2^{K+1} N^2 \exp\Big(-\big(1+o(1)\big)2N_c \cdot \pi_{\boldsymbol{\alpha}} \cdot (c_2/8c)^{2/\beta}\Big) + 2^{K+1} N \exp\big(-Jc_2^2/32M^2\big) \\
&\quad + P\Big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0\Big) \\
&\leq 2^{K+1} N^2 \exp\Big(-\big(1+o(1)\big)J \cdot \pi_{\boldsymbol{\alpha}} \cdot \delta_2^2/2\Big) \\
&\quad + 2^{K+1} N^2 \exp\Big(-2\big(1+o(1)\big)J \cdot \pi_{\boldsymbol{\alpha}} \cdot (c_2/8c)^{2/\beta}\Big) + 2^{K+1} N \exp\big(-Jc_2^2/32M^2\big) \\
&\quad + P\Big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0\Big) \\
&= 2^{K+1}\Big[N \exp\Big(-\big(1+o(1)\big)J \cdot \pi_{\boldsymbol{\alpha}} \cdot \delta_2^2/4\Big)\Big]^2 \\
&\quad + 2^{K+1}\Big[N \exp\Big(-\big(1+o(1)\big)J \cdot \pi_{\boldsymbol{\alpha}} \cdot (c_2/8c)^{2/\beta}\Big)\Big]^2 + 2^{K+1} N \exp\big(-Jc_2^2/32M^2\big) \\
&\quad + P\Big(\hat{\boldsymbol{A}}_c \neq \boldsymbol{A}_c^0\Big) \\
&\longrightarrow\ 0, \ \text{as } J \to \infty.
\end{aligned}
$$

Therefore, $\hat{\boldsymbol{\alpha}}_i$'s are uniformly consistent for $\boldsymbol{\alpha}_i$'s for all $i = 1, \dots, N$. □

C.6 Proof of Proposition 5.4.1

Proof. Our proof uses similar arguments as in Celeux and Govaert (1992). In Step 3 of Algorithm V.1, we have

$$L(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\pi}^{(t+1)}) \leq L(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)}).$$

Moreover, since $\hat{\boldsymbol{\alpha}}_i^{(t+1)} = \arg\min_{\boldsymbol{\alpha}} l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}^{(t+1)}) + h(\hat{\pi}_{\boldsymbol{\alpha}}^{(t+1)})$, which is equivalent to that $l\big(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}^{(t+1)}_{\hat{\boldsymbol{\alpha}}_i^{(t+1)}}\big) + h\big(\hat{\pi}^{(t+1)}_{\hat{\boldsymbol{\alpha}}_i^{(t+1)}}\big) \leq l(\boldsymbol{r}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{\alpha}}^{(t+1)}) + h(\hat{\pi}_{\boldsymbol{\alpha}}^{(t+1)})$ for any $\boldsymbol{\alpha} \neq \hat{\boldsymbol{\alpha}}_i^{(t+1)}$, we have

$$L(\boldsymbol{A}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\pi}^{(t+1)}) \leq L(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)}). \tag{C.9}$$

Therefore the criterion (5.5) is decreasing.

In the finite sample setting, since there is finite number of partitions into 2^K classes, the decreasing sequence $L(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\pi}^{(t)})$ also takes a finite number of values, which makes it converge to a stationary value. Moreover, since the minima of the loss function is well-defined, the sequence $(\boldsymbol{A}^{(t)}, \boldsymbol{\mu}^{(t)}f, \boldsymbol{\pi}^{(t)})$ also converges. □

C.7 Proof of Proposition 5.4.2

Proof. Our proof directly follows that in Celeux and Govaert (1992). Since

$$\begin{aligned} L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi}) &= \sum_{\boldsymbol{\alpha}\in\{0,1\}^K} \sum_{i=1}^{n} u_{i\boldsymbol{\alpha}} \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}}) + h(\pi_{\boldsymbol{\alpha}}) \Big) \\ &\geq \sum_{\boldsymbol{\alpha}\in\{0,1\}^K} \sum_{i=1}^{n} u_{i\boldsymbol{\alpha}} \min_{\boldsymbol{\alpha}'} \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}'}) + h(\pi_{\boldsymbol{\alpha}'}) \Big) \\ &\geq \sum_{\boldsymbol{\alpha}\in\{0,1\}^K} \sum_{i=1}^{n} \min_{\boldsymbol{\alpha}'} \Big(l(\boldsymbol{r}_i, \boldsymbol{\mu}_{\boldsymbol{\alpha}'}) + h(\pi_{\boldsymbol{\alpha}'}) \Big), \end{aligned}$$

where the RHS is attained when $\boldsymbol{U}$ is equivalent to some partition, the Algorithm V.1 can be regarded as an alternating optimization algorithm to minimize $L(\boldsymbol{U}, \boldsymbol{\mu}, \boldsymbol{\pi})$.

Specifically, the Algorithm V.1 is in fact a grouped coordinate descent method. Following the Theorem 2.2 of Bezdek, Hathaway, Howard, Wilson, and Windham (1987), the Proposition 5.4.2 is proved. □

BIBLIOGRAPHY

Azran, A., and Z. Ghahramani (2006), Spectral methods for automatic multiscale data clustering, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, pp. 190–197, IEEE.

Barber, M. J. (2007), Modularity and community detection in bipartite networks, *Physical Review E*, *76*(6), 066,102.

Bengio, Y., A. Courville, and P. Vincent (2013), Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Bezdek, J., R. Hathaway, R. Howard, C. Wilson, and M. Windham (1987), Local convergence analysis of a grouped variable version of coordinate descent, *Journal of Optimization Theory and Applications*, *54*(3), 471–477.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017), Variational inference: A review for statisticians, *Journal of the American Statistical Association*, *112*(518), 859–877.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011), Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends® in Machine learning*, *3*(1), 1–122.

Brault, V., and M. Mariadassou (2015), Co-clustering through latent block model: A review, *Journal de la Société Française de Statistique*, *156*(3), 120–139.

Brusco, M., and D. Steinley (2006), Inducing a blockmodel structure of two-mode binary data using seriation procedures, *Journal of Mathematical Psychology*, *50*(5), 468–477.

Brusco, M., and D. Steinley (2011), A tabu-search heuristic for deterministic two-mode blockmodeling of binary network matrices, *Psychometrika*, *76*(4), 612–633.

Cattell, H. E., and A. D. Mead (2008), The sixteen personality factor questionnaire (16PF), *The Sage Handbook of Personality Theory and Assessment*, p. 135–159.

Celeux, G., and G. Govaert (1992), A classification EM algorithm for clustering and two stochastic versions, *Computational Statistics & Data Analysis*, *14*(3), 315–332.

Chen, J. (2017), On finite mixture models, *Statistical Theory and Related Fields*, *1*(1), 15–27, doi:10.1080/24754269.2017.1321883.

Chen, J., and Z. Chen (2008), Extended Bayesian information criteria for model selection with large model spaces, *Biometrika*, *95*(3), 759–771.

Chen, J., and J. de la Torre (2014), A procedure for diagnostically modeling extant large-scale assessment data: The case of the programme for international student assessment in reading, *Psychology*, *5*(18), 1967–1978.

Chen, Y., and X. Li (2019), Exploratory data analysis for cognitive diagnosis: Stochastic co-blockmodel and spectral co-clustering, in *Handbook of Diagnostic Classification Models*, pp. 287–306, Springer.

Chen, Y., J. Liu, G. Xu, and Z. Ying (2015), Statistical analysis of Q-matrix based diagnostic classification models, *Journal of the American Statistical Association*, *110*(510), 850–866.

Chen, Y., X. Li, J. Liu, G. Xu, and Z. Ying (2017a), Exploratory item classification via spectral graph clustering, *Applied Psychological Measurement*, *41*(8), 579–599.

Chen, Y., X. Li, J. Liu, and Z. Ying (2017b), Regularized latent class analysis with application in cognitive diagnosis, *Psychometrika*, *82*(3), 660–692.

Chen, Y., S. A. Culpepper, Y. Chen, and J. Douglas (2018), Bayesian estimation of the DINA Q-matrix, *Psychometrika*, *83*(1), 89–108.

Chen, Y., I. Moustaki, and H. Zhang (2020), A note on likelihood ratio tests for models with latent variables, *Psychometrika*, *85*, 1–17, doi:10.1007/s11336-020-09735-0.

Cheng, Y., and G. M. Church (2000), Biclustering of expression data, in *Ismb*, vol. 8, pp. 93–103.

Chiu, C.-Y. (2013), Statistical refinement of the Q-matrix in cognitive diagnosis, *Applied Psychological Measurement*, *37*(8), 598–618.

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

Chiu, C.-Y., and H.-F. Köhn (2019a), Consistency theory for the general nonparametric classification method, *Psychometrika*, *84*(3), 830–845.

Chiu, C.-Y., and H.-F. Köhn (2019b), Nonparametric methods in cognitively diagnostic assessment, *Handbook of Diagnostic Classification Models*, pp. 107–132.

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

Chiu, C.-Y., H.-F. Köhn, Y. Zheng, and R. Henson (2016), Joint maximum likelihood estimation for diagnostic classification models, *Psychometrika*, *81*(4), 1069–1092.

Chiu, C.-Y., Y. Sun, and Y. Bian (2018), Cognitive diagnosis for small educational programs: The general nonparametric classification method, *Psychometrika*, *83*(2), 355–375.

Cho, H., I. S. Dhillon, Y. Guan, and S. Sra (2004), Minimum sum-squared residue coclustering of gene expression data, in *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp. 114–125, SIAM.

Chung, M., and M. S. Johnson (2018), An MCMC algorithm for estimating the Q-matrix in a Bayesian framework, *arXiv preprint arXiv:1802.02286*.

Culpepper, S. A. (2019), Estimating the cognitive diagnosis Q-matrix with expert knowledge: Application to the fraction-subtraction dataset, *Psychometrika*, *84*(2), 333–357.

Dahlgren, M. A., H. Hult, L. O. Dahlgren, H. H. af Segerstad, and K. Johansson (2006), From senior student to novice worker: Learning trajectories in political science, psychology and mechanical engineering, *Studies in Higher Education*, *31*(5), 569–586, doi:10.1080/03075070600923400.

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

de la Torre, J. (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, doi:10.1007/s11336-011-9207-7.

de la Torre, J., and C.-Y. Chiu (2016), A general method of empirical Q-matrix validation, *Psychometrika*, *81*(2), 253–273.

de la Torre, J., L. A. van der Ark, and G. Rossi (2018), Analysis of clinical data from a cognitive diagnosis modeling framework, *Measurement and Evaluation in Counseling and Development*, *51*(4), 281–296, doi:10.1177/0748175615569110.

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

Delyon, B., M. Lavielle, and E. Moulines (1999), Convergence of a stochastic approximation version of the EM algorithm, *The Annals of Statistics*, *27*(1), 94–128.

DiBello, L. V., W. F. Stout, and L. A. Roussos (1995), Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques, *Cognitively Diagnostic Assessment*, *361389*.

Doreian, P., V. Batagelj, and A. Ferligoj (2004), Generalized blockmodeling of two-mode network data, *Social Networks*, *26*(1), 29–53.

Doreian, P., P. Lloyd, and A. Mrvar (2013), Partitioning large signed two-mode networks: Problems and prospects, *Social Networks*, *35*(2), 178–203.

Dunn, J. C. (1974), Well-separated clusters and optimal fuzzy partitions, *Journal of Cybernetics*, *4*(1), 95–104.

Efron, B. (1979), Bootstrap methods: Another Look at the Jackknife, *The Annals of Statistics*, *7*(1), 1 – 26, doi:10.1214/aos/1176344552.

Fan, J., and R. Li (2001), Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association*, *96*(456), 1348–1360.

George, A. C., and A. Robitzsch (2015), Cognitive diagnosis models in R: A didactic, *The Quantitative Methods for Psychology*, *11*(3), 189–205, doi:10.20982/tqmp.11.3.p189.

George, T., and S. Merugu (2005), A scalable collaborative filtering framework based on co-clustering, in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 4–pp, IEEE.

Goodman, L. A. (1974), Exploratory latent structure analysis using both identifiable and unidentifiable models, *Biometrika*, *61*(2), 215–231.

Govaert, G., and M. Nadif (2003), Clustering with block mixture models, *Pattern Recognition*, *36*(2), 463–473.

Govaert, G., and M. Nadif (2005), An EM algorithm for the block mixture model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 643–647.

Govaert, G., and M. Nadif (2008), Block clustering with bernoulli mixture models: Comparison of different approaches, *Computational Statistics & Data Analysis*, *52*(6), 3233–3245.

Govaert, G., and M. Nadif (2010), Latent block model for contingency table, *Communications in Statistics—Theory and Methods*, *39*(3), 416–425.

Gu, Y., and G. Xu (2019a), Identification and estimation of hierarchical latent attribute models, *arXiv preprint arXiv:1906.07869*.

Gu, Y., and G. Xu (2019b), Learning attribute patterns in high-dimensional structured latent attribute models, *Journal of Machine Learning Research*, *20*(2019).

Gu, Y., and G. Xu (2019c), The sufficient and necessary condition for the identifiability and estimability of the DINA model, *Psychometrika*, *84*(2), 468–483.

Gu, Y., and G. Xu (2020), Partial identifiability of restricted latent class models, *The Annals of Statistics*, *48*(4), 2082–2107.

Gu, Y., and G. Xu (2022), Identifiability of hierarchical latent attribute models, *Statistica Sinica, to appear.*

Gu, Y., J. Liu, G. Xu, and Z. Ying (2018), Hypothesis testing of the Q-matrix, *Psychometrika*, *83*(3), 515–537.

Haertel, E. H. (1989), Using restricted latent class models to map the skill structure of achievement items, *Journal of Educational Measurement*, *26*(4), 301–321, doi: 10.1111/j.1745-3984.1989.tb00336.x.

Hartigan, J. A. (1972), Direct clustering of a data matrix, *Journal of the American Statistical Association*, *67*(337), 123–129.

Hartz, S. M. (2002), A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality., Ph.D. thesis, ProQuest Information & Learning.

Henson, R. A., J. L. Templin, and J. T. Willse (2009), Defining a family of cognitive diagnosis models using log-linear models with latent variables, *Psychometrika*, *74*(2), 191, doi:10.1007/s11336-008-9089-5.

Hoeffding, W. (1994), Probability inequalities for sums of bounded random variables, in *The Collected Works of Wassily Hoeffding*, pp. 409–426, Springer.

Jimoyiannis, A., and V. Komis (2001), Computer simulations in physics teaching and learning: a case study on students' understanding of trajectory motion, *Computers & Education*, *36*(2), 183–204, doi:10.1016/S0360-1315(00)00059-2.

Junker, B. W., and K. Sijtsma (2001), Cognitive assessment models with few assumptions, and connections with nonparametric item response theory, *Applied Psychological Measurement*, *25*(3), 258–272.

Keribin, C., V. Brault, G. Celeux, G. Govaert, et al. (2012), Model selection for the binary latent block model, in *Proceedings of COMPSTAT*, vol. 2012.

Keribin, C., V. Brault, G. Celeux, and G. Govaert (2015), Estimation and selection for the latent block model on categorical data, *Statistics and Computing*, *25*(6), 1201–1216.

Khoshneshin, M., and W. N. Street (2010), Incremental collaborative filtering via evolutionary co-clustering, in *Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 325–328.

Kuhn, H. W. (1955), The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly*, *2*(1-2), 83–97.

Leighton, J. P., M. J. Gierl, and S. M. Hunka (2004), The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach, *Journal of Educational Measurement*, *41*(3), 205–237, doi:10.1111/j.1745-3984.2004.tb01163.x.

Li, C., C. Ma, and G. Xu (2022), Learning large Q-matrix by restricted Boltzmann machines, *Psychometrika, to appear*, doi:10.1007/s11336-021-09828-4.

Liu, J., G. Xu, and Z. Ying (2012), Data-driven learning of Q-matrix, *Applied Psychological Measurement*, *36*(7), 548–564, doi:10.1177/0146621612456591.

Liu, J., Z. Ying, and S. Zhang (2015), A rate function approach to computerized adaptive testing for cognitive diagnosis, *Psychometrika*, *80*(2), 468–490.

Ma, C., and G. Xu (2021), Hypothesis testing for hierarchical structures in cognitive diagnosis models, *Journal of Data Science, to appear*, doi:10.6339/21-JDS1024.

Ma, C., J. de la Torre, and G. Xu (2022a), Bridging parametric and nonparametric methods in cognitive diagnosis, *arXiv preprint arXiv:2006.15409*.

Ma, C., J. Ouyang, and G. Xu (2022b), Learning latent and hierarchical structures in cognitive diagnosis models, *Psychometrika, to appear*, doi:10.1007/s11336-022-09867-5.

Neal, R. M., and G. E. Hinton (1998), A view of the EM algorithm that justifies incremental, sparse, and other variants, in *Learning in Graphical Models*, pp. 355–368, Springer.

Nowicki, K., and T. A. B. Snijders (2001), Estimation and prediction for stochastic blockstructures, *Journal of the American Statistical Association*, *96*(455), 1077–1087.

Nylund, K. L., T. Asparouhov, and B. O. Muthén (2007), Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study, *Structural Equation Modeling: A Multidisciplinary Journal*, *14*(4), 535–569, doi:10.1080/10705510701575396.

O'Brien, K. L., et al. (2019), Causes of severe pneumonia requiring hospital admission in children without HIV infection from Africa and Asia: the PERCH multi-country case-control study, *The Lancet*, *394*(10200), 757–779.

Popescu, P. G., S. S. Dragomir, E. I. Sluşanschi, and O. N. Stănăşilă (2016), Bounds for Kullback-Leibler divergence, *Electronic Journal of Differential Equations*, *2016*.

Reynolds, D. A. (2009), Gaussian mixture models., *Encyclopedia of Biometrics*, *741*(659-663).

Rohe, K., T. Qin, and B. Yu (2012), Co-clustering for directed graphs: the stochastic co-blockmodel and spectral algorithm Di-Sim, *arXiv preprint arXiv:1204.2296*.

Schwarz, G., et al. (1978), Estimating the dimension of a model, *The Annals of Statistics*, *6*(2), 461–464.

Self, S. G., and K.-Y. Liang (1987), Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions, *Journal of the American Statistical Association*, *82*(398), 605–610, doi:10.1080/01621459.1987.10478472.

Shen, X., W. Pan, and Y. Zhu (2012), Likelihood-based selection and sharp parameter estimation, *Journal of the American Statistical Association*, *107*(497), 223–232.

Simon, M. A., and R. Tzur (2004), Explicating the role of mathematical tasks in conceptual learning: An elaboration of the hypothetical learning trajectory, *Mathematical Thinking and Learning*, *6*(2), 91–104, doi:10.1207/s15327833mtl0602_2.

Tatsuoka, K. K. (1983), Rule space: An approach for dealing with misconceptions based on item response theory, *Journal of Educational Measurement*, *20*, 345–354, doi:10.4324/9780203056899-22.

Tatsuoka, K. K. (1990), Toward an integration of item-response theory and cognitive error diagnosis, in *Diagnostic Monitoring of Skill and Knowledge Acquisition*, pp. 453–488, Routledge, doi:10.4324/9780203056899-22.

Templin, J., and L. Bradshaw (2014), Hierarchical diagnostic classification models: A family of models for estimating and testing attribute hierarchies, *Psychometrika*, *79*(2), 317–339, doi:10.1007/s11336-013-9362-0.

Templin, J., R. A. Henson, et al. (2010), *Diagnostic measurement: Theory, methods, and applications*, Guilford Press.

Templin, J. L., and R. A. Henson (2006), Measurement of psychological disorders using cognitive diagnosis models., *Psychological Methods*, *11*(3), 287, doi:10.1037/1082-989X.11.3.287.

Tipping, M. E., and C. M. Bishop (1999), Probabilistic principal component analysis, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *61*(3), 611–622.

Tschannen, M., O. Bachem, and M. Lucic (2018), Recent advances in autoencoder-based representation learning, *arXiv preprint arXiv:1812.05069*.

Tuy, H. (1995), DC optimization: theory, methods and algorithms, in *Handbook of Global Optimization*, pp. 149–216, Springer.

Vaart, A. W., and J. A. Wellner (1996), Weak convergence, in *Weak Convergence and Empirical Processes*, pp. 16–28, Springer.

Van Den Oord, A., O. Vinyals, et al. (2017), Neural discrete representation learning, *Advances in Neural Information Processing Systems*, *30*.

van der Vaart, A. W. (2000), *Asymptotic statistics*, vol. 3, Cambridge university press.

von Davier, M. (2005), A general diagnostic model applied to language testing data, *ETS Research Report Series*, *2005*(2), 1–35, doi:10.1002/j.2333-8504.2005.tb01993.x.

von Davier, M. (2019), The general diagnostic model, in *Handbook of Diagnostic Classification Models*, pp. 133–153, Springer.

von Davier, M., and S. J. Haberman (2014), Hierarchical diagnostic classification models morphing into unidimensional 'diagnostic' classification models — a commentary, *Psychometrika*, *79*(2), 340–346.

Von Luxburg, U. (2007), A tutorial on spectral clustering, *Statistics and Computing*, *17*(4), 395–416.

Wang, C., and M. J. Gierl (2011), Using the attribute hierarchy method to make diagnostic inferences about examinees' cognitive skills in critical reading, *Journal of Educational Measurement*, *48*(2), 165–187, doi:10.1111/j.1745-3984.2011.00142.x.

Wang, C., and J. Lu (2021), Learning attribute hierarchies from data: Two exploratory approaches, *Journal of Educational and Behavioral Statistics*, *46*(1), 58–84.

Wang, S., and J. Douglas (2015), Consistency of nonparametric classification in cognitive diagnosis, *Psychometrika*, *80*(1), 85–100.

Wang, S., and L. Liao (2001), Decomposition method with a variable parameter for a class of monotone variational inequality problems, *Journal of Optimization Theory and Applications*, *109*(2), 415–429.

Wong, W. H., X. Shen, et al. (1995), Probability inequalities for likelihood ratios and convergence rates of sieve MLEs, *The Annals of Statistics*, *23*(2), 339–362.

Wu, C., S. Kwon, X. Shen, and W. Pan (2016), A new algorithm and theory for penalized regression-based clustering, *The Journal of Machine Learning Research*, *17*(1), 6479–6503.

Wyse, J., and N. Friel (2012), Block clustering with collapsed latent block models, *Statistics and Computing*, *22*(2), 415–428.

Wyse, J., N. Friel, and P. Latouche (2017), Inferring structure in bipartite networks using the latent block model and exact ICL, *Network Science*, *5*(1), 45–69.

Xu, G. (2017), Identifiability of restricted latent class models with binary responses, *The Annals of Statistics*, *45*(2), 675–707, doi:10.1214/16-AOS1464.

Xu, G., and Z. Shang (2018), Identifying latent structures in restricted latent class models, *Journal of the American Statistical Association*, *113*(523), 1284–1295, doi:10.1080/01621459.2017.1340889.

Xu, G., and S. Zhang (2016), Identifiability of diagnostic classification models, *Psychometrika*, *81*(3), 625–649, doi:10.1007/s11336-015-9471-z.