

Learning Latent and Hierarchical Structures in Cognitive Diagnosis Models

Chenchen Ma, Jing Ouyang, and Gongjun Xu

Department of Statistics, University of Michigan

Abstract

Cognitive Diagnosis Models (CDMs) are a special family of **discrete latent variable models** that are widely used in educational and psychological measurement. A key component of CDMs is the **Q -matrix** characterizing the dependence structure between the **items and the latent attributes**. Additionally, researchers also assume in many applications certain **hierarchical structures** among the latent attributes to characterize their dependence. In most CDM applications, the attribute-attribute hierarchical structures, the item-attribute Q -matrix, the item-level diagnostic model, as well as the number of latent attributes, need to be fully or partially pre-specified, which however may be subjective and misspecified as noted by many recent studies. This paper considers the problem of **jointly learning these latent and hierarchical structures in CDMs from observed data with minimal model assumptions**. Specifically, a **penalized likelihood approach** is proposed to **select the number of attributes and estimate the latent and hierarchical structures simultaneously**. An **expectation-maximization (EM) algorithm** is developed for efficient computation, and **statistical consistency theory** is also established under mild conditions. The good performance of the proposed method is illustrated by simulation studies and real data applications in educational assessment.

Keywords— Cognitive diagnosis model, hierarchical structure, Q -matrix, Penalized EM algorithm

1 Introduction

Cognitive Diagnosis Models (CDMs) are a popular family of discrete latent variable models that have been widely used in modern educational, psychological, social and biological sciences. These models focus on multivariate discrete noisy observations, and assume the existence of discrete latent attributes which explain or govern the observed variables. In practice, the discrete latent attributes often have special scientific interpretations. For example, in educational assessments, the latent attributes are often assumed to be the mastery or deficiency of target skills (de la Torre, 2011; Junker and Sijtsma, 2001); in psychiatric diagnosis, they may be modeled as the presence or absence of certain underlying mental disorders (de la Torre et al., 2018; Templin and Henson, 2006); and in epidemiological and medical measurement studies, they can be interpreted as the existence or nonexistence of some disease pathogens (O’Brien et al., 2019; Wu et al., 2016b,c). A common feature of such latent variable models is that the probabilistic distribution of the observed responses is governed by the unobserved latent attributes. Upon observing the responses, one can infer the underlying latent attributes. As such, CDMs can provide fine-grained inference on subjects’ latent attribute profiles based on their multivariate observed responses, and the corresponding latent subgroups of a population can also be detected based on the inferred latent attribute profiles.

In the CDM 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 also 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 the Generalized DINA (GDINA) model, where the interactions among all the latent attributes were included. Other popularly used CDMs 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).

In many applications of CDMs, researchers are also interested in hierarchical structures among the latent attributes. For example, in a learning context, the possession of lower level



skills are often assumed to be the prerequisites 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 variation of Tatsuoaka’s rule-space approach (Tatsuoka, 1983), which explicitly defined the hierarchical attribute structures through an adjacency matrix. Under the CDM framework, Templin and Bradshaw (2014) proposed the Hierarchical Cognitive Diagnosis Models (HCDMs), in which a Directed Acyclic Graph (DAG) is essentially used to impose hard constraints on possible latent attribute profiles under hierarchies.

To fit hierarchical CDMs, 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 (Chen et al., 2015; Li et al., 2022; Liu et al., 2012; Xu and Shang, 2018) or Bayesian approaches (Chen et al., 2018; Chung and Johnson, 2018; Culpepper, 2019), and methods to validate the pre-specified Q -matrix (Chiu, 2013; de la Torre, 2008; de la Torre and Chiu, 2016; DeCarlo, 2012; Gu et al., 2018). Many of these Q -matrix learning or validation methods, however, do not take the hierarchical structures into consideration, 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., 2017) 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 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 hierarchical CDMs. However, their algorithm also assumed that the number of latent attributes was known and they only considered the DINA and DINO models.

In this paper, 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 paper proceeds as follows: the model setup of hierarchical CDMs is provided in Section 2. The proposed penalized likelihood approach and its theoretical properties are presented in Section 3. An efficient algorithm is developed and related computational issues are discussed in Section 4. Simulation studies are presented in Section 5. In Section 6, the model is applied to two real data sets of international educational assessment. Finally, Section 7 concludes with some discussions. The proof of the main theorem and detailed derivations for the proposed algorithm are presented in the supplementary material.

2 Model Setup

In this section, we introduce the general model setup of hierarchical CDMs and illustrate the connections between hierarchical CDMs and restricted latent class models, which motivates the proposed approach in Section 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.

2.1 Hierarchical Cognitive Diagnosis Models

In a CDM with J items which depend on K latent attributes of interest, two types of subject-specific variables are considered, including the responses $\mathbf{R} = (R_1, \dots, R_J)$, and the latent binary attribute profile $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$. In this paper, both the responses \mathbf{R} and the latent attribute profile $\boldsymbol{\alpha}$ are assumed to be binary. The J -dimensional vector $\mathbf{R} \in \{0, 1\}^J$ denotes the binary responses to the 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 $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$ 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 hierarchical CDMs is that there exist certain hierarchical structures among the latent attributes. For example, in cognitive diagnosis modeling, the possession of lower-level skills are often regarded as the prerequisites 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 $\alpha_k \rightarrow \alpha_l$ (or $k \rightarrow l$) to denote the hierarchy that attribute α_k is a prerequisite of attribute α_l . We assume such hierarchy $\alpha_k \rightarrow \alpha_l$ (or $k \rightarrow l$) 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 $\mathcal{E} = \{k \rightarrow l : \text{attribute } k \text{ is a prerequisite for attribute } l, 1 \leq k \neq l \leq K\}$, 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 1.

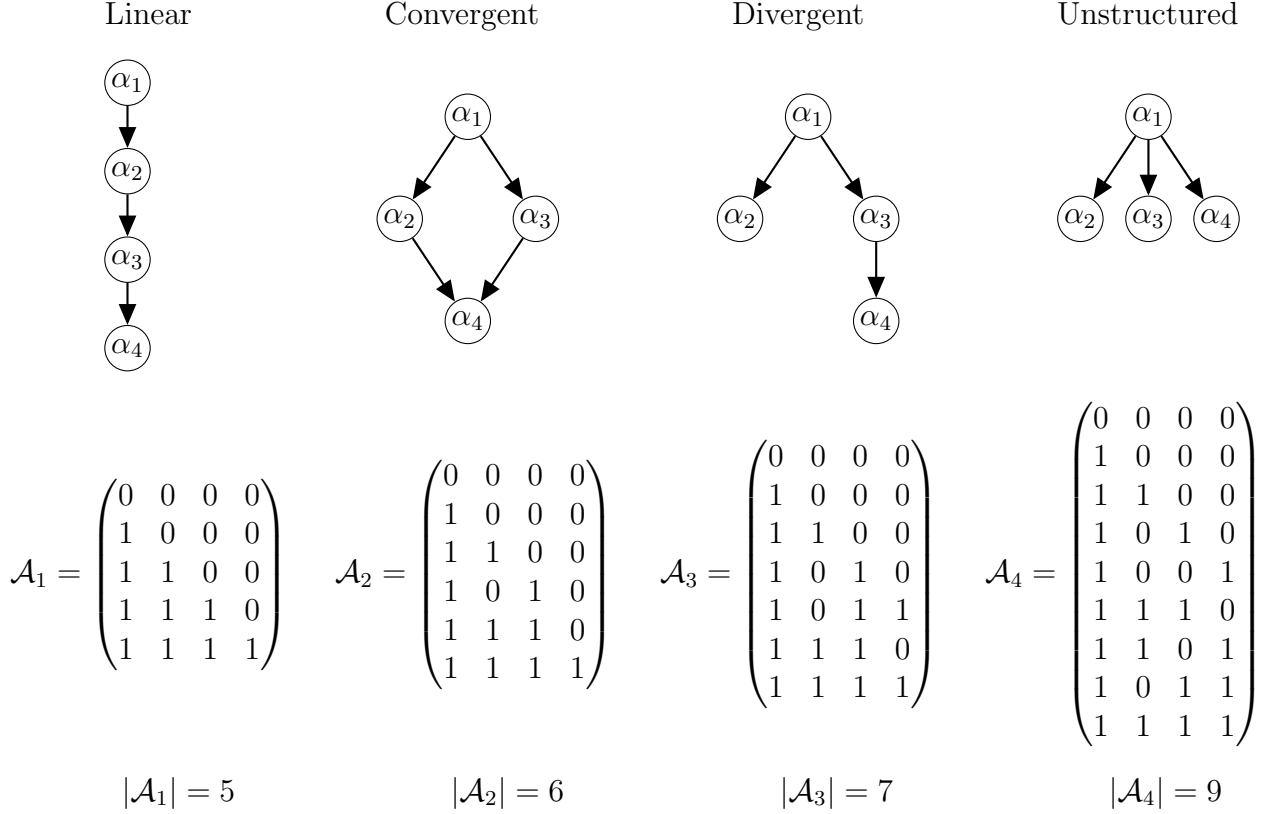


Figure 1: Examples of hierarchical structures of latent attributes. For $i = 1, \dots, 4$, each \mathcal{A}_i represents the induced set of existent attribute profiles under the hierarchical structure above it, where each row in \mathcal{A}_i represents an attribute profile $\boldsymbol{\alpha}$ with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

In CDMs, the structural matrix $\mathbf{Q} = (q_{j,k}) \in \{0, 1\}^{J \times K}$ is an important component which 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 j th row vector of \mathbf{Q} denoted by \mathbf{q}_j describes the full dependence of item j on K latent attributes. In many applications, the matrix \mathbf{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 2 for the illustration of the \mathbf{Q} -matrix and the corresponding graphical representation.

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,

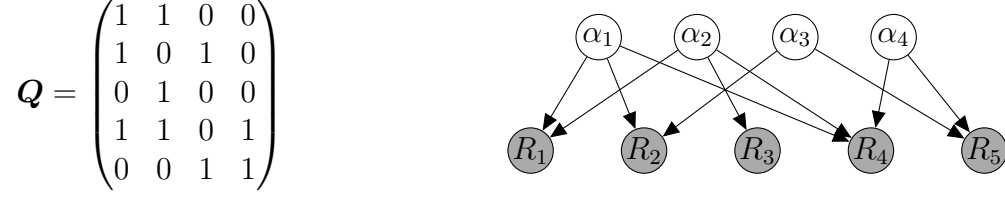


Figure 2: Illustration of Q-matrix

and follow Bernoulli distributions with parameters $\theta_{1,\alpha}, \dots, \theta_{J,\alpha}$, which are called item parameters. Specifically, we have $\theta_{j,\alpha} := \mathbb{P}(R_j = 1 \mid \alpha)$. Under the local independence assumption, the probability mass function of a subject's response vector $\mathbf{R} = (R_1, \dots, R_J) \in \{0, 1\}^J$ can be written as

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

So far we have a latent attribute profile α for a subject to indicate the subject's possession of K attributes, and a structural vector \mathbf{q} for an item to reflect the item's dependence on K latent attributes. Moreover, the structural matrix \mathbf{Q} puts constraints on item parameters to reflect the diagnostic model assumptions. One important and common assumption is that the item parameters $\theta_{j,\alpha}$ only depends on whether the latent attribute profile α 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 have the same item parameters. In this way, the structural matrix \mathbf{Q} forces some entries in the item parameter matrix Θ to be the same. The dependence of item parameters on the required attributes are modeled differently in different CDMs, as shown in Example 1 and Example 2.

Example 1 (DINA and DINO Models) We first consider 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 $\Gamma = (\Gamma_{j,\alpha} : j \in [J], \alpha \in \{0, 1\}^K) \in \{0, 1\}^{J \times 2^K}$, 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

$$\Gamma_{j,\alpha}^{DINA} = \prod_k^K \alpha_k^{q_{j,k}} = \prod_{k \in \mathcal{K}_j} \alpha_k. \quad (2)$$

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

$$\Gamma_{j,\alpha}^{DINO} = \mathbb{I}(\exists k \in [K], q_{j,k} = \alpha_k = 1). \quad (3)$$

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 the item parameters are defined as:

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

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

Example 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

$$\theta_{j,\alpha}^{GDINA} = \beta_{j,0} + \sum_{k=1}^K \beta_{j,k} \alpha_k q_{j,k} + \sum_{k=1}^K \sum_{k'=k+1}^K \beta_{j,k,k'} \alpha_k \alpha_{k'} q_{j,k} q_{j,k'} + \cdots + \beta_{j,1,2,\dots,K} \prod_{k=1}^K \alpha_k q_{j,k} \quad (4)$$

$$= \beta_{j,0} + \sum_{k \in \mathcal{K}_j} \beta_{j,k} \alpha_k + \sum_{k,k' \in \mathcal{K}_j, k \neq k'} \beta_{j,k,k'} \alpha_k \alpha_{k'} + \cdots + \beta_{j,\mathcal{K}_j} \prod_{k \in \mathcal{K}_j} \alpha_k. \quad (5)$$

The coefficients in the GDINA model can be interpreted as following: $\beta_{j,0}$ is the probability of a positive response for the most incapable subjects with none 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 CDMs.

2.2 CDMs as Restricted Latent Class Models

CDMs 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 give a brief description of the general model setup of LCMs ([Goodman, 1974](#)). In an LCM, we assume that each subject belongs to one of M latent classes, $m = 1, \dots, M$. 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 $\boldsymbol{\Theta} = (\theta_{jk}) \in [0, 1]^{J \times M}$ to denote the item parameter matrix. To be specific, for a subject's response $\mathbf{R} = (R_1, R_2, \dots, R_J)$, we have $\theta_{jm} = \mathbb{P}(R_j = 1 \mid m)$. Then the probability mass function of an LCM can be written as

$$\mathbb{P}(\mathbf{R} \mid \boldsymbol{\Theta}, \boldsymbol{\pi}) = \sum_{m=1}^M \pi_m \prod_{j=1}^J \theta_{jm}^{R_j} (1 - \theta_{jm})^{1-R_j}. \quad (6)$$

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

CDMs 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 CDMs with K latent attributes, each latent attribute profile $\boldsymbol{\alpha}$ is a K -dimensional binary vector and has a proportion parameter $\pi_{\boldsymbol{\alpha}}$. We can use a one to one correspondence from $\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \{0, 1\}^K\}$ to $\{m : m = 1, \dots, 2^K\}$ such as $m = \sum_{k=1}^K \alpha_k \cdot 2^{k-1} + 1$. Then in a CDM with K latent attributes and no hierarchical structure, we have $M = 2^K$ latent classes. In hierarchical CDMs, the number of allowed latent attribute profiles is smaller than 2^K and we have $M = |\mathcal{A}|$, as discussed in [Section 2.1](#). Moreover, in CDMs, there are additional restrictions on the item parameter matrix $\boldsymbol{\Theta}$ 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 CDM with or without any hierarchical structure can be viewed as a submodel of a saturated LCM.

3 Regularized Estimation Method

3.1 Motivation and Proposed Method

To fit hierarchical CDMs, 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 learn 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, we propose to learn a hierarchical CDM with minimal model assumptions starting with an unrestricted LCM. We use the following discussion and examples to first illustrate the key idea.

- As discussed in Section 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, in Figure 1, 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 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 cognitive diagnosis 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 $\mathbf{Q}_1 \stackrel{\mathcal{E}}{\sim} \mathbf{Q}_2$, if they give the same item parameter matrices, that is, $\Theta(\mathbf{Q}_1, \mathcal{A}_{\mathcal{E}}) = \Theta(\mathbf{Q}_2, \mathcal{A}_{\mathcal{E}})$, where $\mathcal{A}_{\mathcal{E}}$ is the induced latent attribute profile set under hierarchy \mathcal{E} . As we introduced in Example 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 linear hierarchy, that is, $\mathcal{E} = \{1 \rightarrow 2 \rightarrow 3\}$. We have

$$\mathbf{Q}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \stackrel{\mathcal{E}}{\sim} \mathbf{Q}^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \stackrel{\mathcal{E}}{\sim} \mathbf{Q}^{(*)} = \begin{pmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{pmatrix}, \quad (7)$$

$$\theta_{j,\alpha}^{GDINA} = \beta_{j,0} + \sum_{k=1}^K \beta_{j,k} \alpha_k q_{j,k} + \sum_{k=1}^K \sum_{k'=k+1}^K \beta_{j,k,k'} \alpha_k \alpha_{k'} q_{j,k,k'} + \cdots + \beta_{j,1,2,\dots,K} \prod_{k=1}^K \alpha_k q_{j,k} \quad (4)$$

$$= \beta_{j,0} + \sum_{k \in \mathcal{K}_j} \beta_{j,k} \alpha_k + \sum_{k,k' \in \mathcal{K}_j, k \neq k'} \beta_{j,k,k'} \alpha_k \alpha_{k'} + \cdots + \beta_{j,\mathcal{K}_j} \prod_{k \in \mathcal{K}_j} \alpha_k. \quad (5)$$

When the underlying model is the GDINA model, all attributes are considered, there would not exist the second item of the Q -matrices in (7),

for $\mathbf{Q}^{(1)}$, $\mathbf{q}_2^{(1)} = (0, 1, 0)$ 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 $\mathbf{Q}^{(2)}$, $\mathbf{q}_2^{(2)} = (1, 1, 0)$ 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 $\mathbf{Q}^{(1)}$, and thus, the equivalence no longer holds under the GDINA model. Therefore, to learn the Q -matrix and infer the item models in hierarchical CDMs, it is also necessary to learn the item parameter matrix and investigate the inner constraint structure of it. Moreover, after learning the item parameter matrix, we can get the partial orders among the selected latent classes, which 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 4.2.

Motivated by the above discussions and the fact that CDMs 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 j th item's item 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

$$L_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R}) = \prod_{i=1}^N \left(\sum_{k=1}^M (\pi_k \prod_{j=1}^J \theta_{jk}^{R_{ij}} (1 - \theta_{jk})^{1-R_{ij}}) \right). \quad (8)$$

And the log-likelihood is

$$l_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R}) = \sum_{i=1}^N \log \left(\sum_{k=1}^M (\pi_k \prod_{j=1}^J \theta_{jk}^{R_{ij}} (1 - \theta_{jk})^{1-R_{ij}}) \right) \quad (9)$$

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

$$l_N(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{R}) - \lambda_1 \sum_{k=1}^M \log_{[\rho_N]} \pi_k - \lambda_2 \sum_{j=1}^J \mathcal{J}(\boldsymbol{\theta}_j), \quad (10)$$

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 $\log_{[\rho_N]} \pi_k = \log \pi_k \cdot \mathbb{I}(\pi_k > \rho_N) + \log \rho_N \cdot \mathbb{I}(\pi_k \leq \rho_N)$, 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 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 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 2.1, due to the restrictions of the Q -matrix and item model assumptions, for each

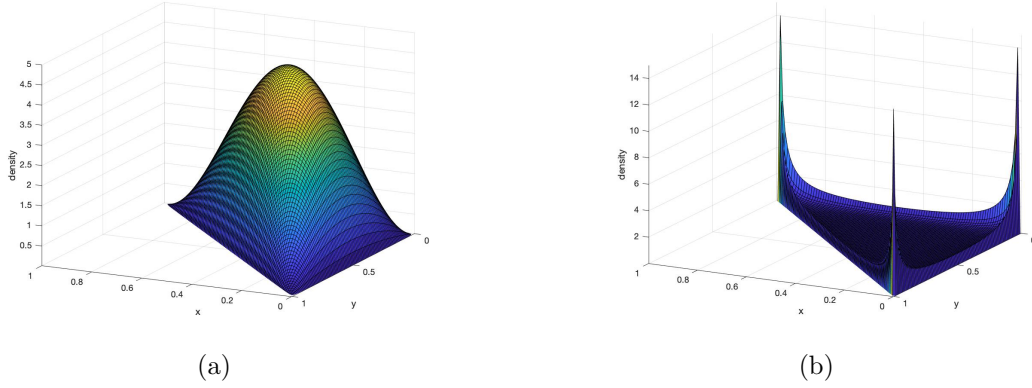


Figure 3: (a): probability density function of 3-dimensional Dirichlet distribution with parameters all equal to 1 – $\lambda_1 = 1$; (b): (improper) probability density function of 3-dimensional Dirichlet distribution with parameters all equal to 1 – $\lambda_1 = -0.9$.

item, some subset of latent attribute profiles have the same item parameters; and such constraint structure of the item parameter matrix can be used to further estimate the hierarchical structure and Q-matrix. Therefore, in order 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),

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

where $\text{TLP}(x; \tau) = \min(|x|, \tau)$, and τ here is a positive tuning parameter. Figure 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

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

A key feature for the truncated Lasso penalty is that it can be regarded as L_1 penalty for a small $|x| \leq \tau$, while it does not put further penalization for a large $|x| > \tau$. In this way, it corrects the Lasso bias through adaptive shrinkage combining with thresholding. It discriminates small from large differences through thresholding and consequently is capable of handling low-resolution

difference through tuning τ .

In [Chen et al. \(2017\)](#), the authors used the Smoothly Clipped Absolute Deviation (SCAD) penalty ([Fan and Li, 2001](#)), which can also be used to merge similar item parameters. 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

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

Figure 4(c) plots the SCAD penalty with $\lambda = 0.5$ and $a = 2$. Here, we want to mention several additional advantages for 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 4(a) and Figure 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 literatures ([Shen et al., 2012](#)).

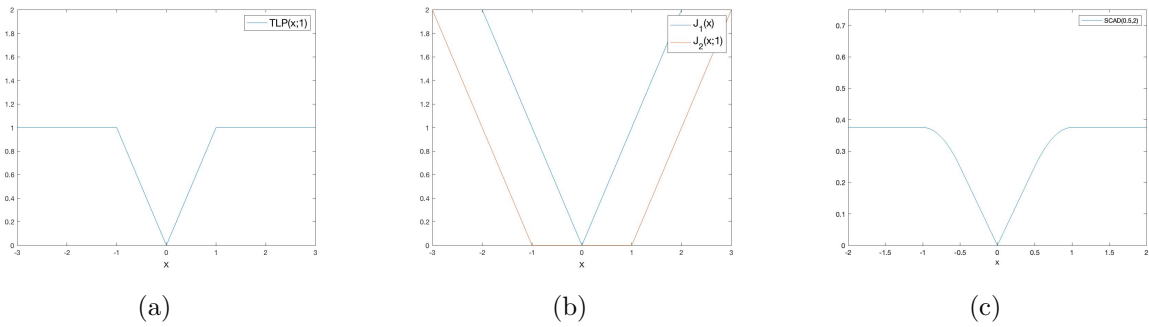


Figure 4: (a): truncated Lasso function $\text{TLP}(x; \tau)$ with $\tau = 1$; (b): the DC decomposition into a difference of two convex functions $J_1(x)$ and $J_2(x; 1)$; (c): SCAD penalty with $\lambda = 0.5$ and $a = 2$.

Remark 1 *Our approach shares some similarities with the regularized LCM approach in [Chen et al. \(2017\)](#) that both use exploratory LCM to estimate the latent structures. However, in*

Chen et al. (2017), the number of latent classes is pre-specified or selected in a way that all the possible values should be considered. This 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, we also develop in Section 4 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 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 plays 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.

3.2 Theoretical Properties

In this section, we present statistical properties of the regularized estimator obtained from (10). We first present some identifiability results of hierarchical cognitive diagnosis model 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 \mathbf{a} and \mathbf{b} of dimension n , we say $\mathbf{a} \preceq \mathbf{b}$ if $a_i \leq b_i$ for $i = 1, \dots, n$, and $\mathbf{a} \succeq \mathbf{b}$ if $a_i \geq b_i$ for $i = 1, \dots, n$.

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 $\mathbf{\Gamma} := (\mathbb{I}\{\theta_{j,m} = \max_{m' \in [M]} \theta_{j,m'}\}, j \in [J], m \in [M]) \in \{0, 1\}^{J \times M}$, indicating whether the latent classes possess the highest level of each item's parameters. Let $\mathbf{\Gamma}_{\cdot, m}$ denote the m th column vector of the $\mathbf{\Gamma}$ 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 $\mathbf{\Gamma}$ if $\mathbf{\Gamma}_{\cdot, m_1} \succeq \mathbf{\Gamma}_{\cdot, m_2}$. See Figure 5 for an illustrative example, where

we use a Directed Acyclic Graph (DAG) to represent partial orders, where Γ_{\cdot, m_1} points to Γ_{\cdot, m_2} if $\Gamma_{\cdot, m_1} \preceq \Gamma_{\cdot, m_2}$.

$$\Theta = \begin{pmatrix} 0.2 & 0.8 & 0.8 \\ 0.2 & 0.2 & 0.8 \\ 0.2 & 0.2 & 0.8 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{Diagram: } \Gamma_{\cdot, 1} \rightarrow \Gamma_{\cdot, 2} \rightarrow \Gamma_{\cdot, 3}$$

Figure 5: Indicator matrix and partial orders

As in Gu and Xu (2019b), for CDMs, we define the indicator matrix for a set of latent attribute profiles \mathcal{A} as $\Gamma^{\mathcal{A}} := (\mathbb{I}\{\theta_{j, \alpha} = \max_{\alpha' \in \mathcal{A}} \theta_{j, \alpha'}\} : j \in [J], \alpha \in \mathcal{A}) \in \{0, 1\}^{J \times |\mathcal{A}|}$. 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 CDM. Similarly, we define the proportion parameter vector and item parameter matrix for a set of latent attribute profiles \mathcal{A} as $\pi^{\mathcal{A}} = (\pi_{\alpha} : \alpha \in \mathcal{A})$ and $\Theta^{\mathcal{A}} = (\theta_{j, \alpha} : j \in [J], \alpha \in \mathcal{A})$. Following Gu and Xu (2019b), for any subset of items $S \subset [J]$, we define a partial order among the latent attribute profiles. For $\alpha, \alpha' \in \mathcal{A}$, we say $\alpha \succeq_S \alpha'$ under $\Gamma^{\mathcal{A}}$ if $\Gamma_{j, \alpha}^{\mathcal{A}} \geq \Gamma_{j, \alpha'}^{\mathcal{A}}$ 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 $\alpha, \alpha' \in \mathcal{A}$, we have $\alpha \succeq_{S_1} \alpha'$ if and only if $\alpha \succeq_{S_2} \alpha'$. 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 3.1. For a subset of items $S \subset [J]$ and a set of attribute profiles \mathcal{A} , we define the corresponding indicator matrix $\Gamma^{(S, \mathcal{A})} = (\Gamma_{j, \alpha} : j \in S, \alpha \in \mathcal{A})$.

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

Definition 1 (strict identifiability, Gu and Xu (2019b))

Consider a CDM 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 $\Gamma^{\mathcal{A}}$ of size $J \times |\mathcal{A}|$ with $|\mathcal{A}| \leq |\mathcal{A}_0|$, any proportion parameter vector $\pi^{\mathcal{A}}$ and any valid item parameter matrix $\Theta^{\mathcal{A}}$ respecting constraints given by $\Gamma^{\mathcal{A}}$, the following equality

$$\mathbb{P}(\mathbf{R} \mid \pi^{\mathcal{A}}, \Theta^{\mathcal{A}}) = \mathbb{P}(\mathbf{R} \mid \pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0}) \quad (11)$$

implies $\mathcal{A} = \mathcal{A}_0$. Moreover, if (11) implies $(\pi^{\mathcal{A}}, \Theta^{\mathcal{A}}) = (\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0})$, then we say the model parameters $(\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0})$ are (strictly) identifiable.

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

Theorem 1 (strict identifiability, Gu and Xu (2019b))

Consider a CDM 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 $\alpha, \alpha' \in \mathcal{A}_0$ where $\alpha' \succeq_{S_i} \alpha$ under $\mathbf{\Gamma}^{\mathcal{A}_0}$ for $i = 1$ or 2 , there exists some $j \in (S_1 \cup S_2)^c$ such that $\Gamma_{j,\alpha}^{\mathcal{A}_0} \neq \Gamma_{j,\alpha'}^{\mathcal{A}_0}$.
- (3) Any column vector of $\mathbf{\Gamma}^{\mathcal{A}_0}$ is different from any column vector of $\mathbf{\Gamma}^{\mathcal{A}_0^c}$, where $\mathcal{A}_0^c = \{0, 1\}^K \setminus \mathcal{A}_0$.

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

Theorem 1 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 (generic identifiability, Gu and Xu (2019b))

Consider a CDM with a hierarchy \mathcal{E}_0 and the induced latent attribute profile set \mathcal{A}_0 . Denote the parameter space of $(\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0})$ constrained by $\mathbf{\Gamma}^{\mathcal{A}_0}$ 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 $(\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0}) \in \Omega \setminus \mathcal{V}$, Equation (11) implies $\mathcal{A} = \mathcal{A}_0$. Moreover, for any $(\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0}) \in \Omega \setminus \mathcal{V}$, if (11) implies $(\pi^{\mathcal{A}}, \Theta^{\mathcal{A}}) = (\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0})$, then we say $(\pi^{\mathcal{A}_0}, \Theta^{\mathcal{A}_0})$ are generically identifiable.

The next theorem presents the generic identifiability results of latent hierarchies and model parameters.

Theorem 2 (generic identifiability, Gu and Xu (2019b)) Consider a CDM with a hierarchy \mathcal{E}_0 . The hierarchy is generically 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:

- (A) There exist two disjoint item sets S_1 and S_2 , such that altering some entries from 0 to 1 in $\mathbf{\Gamma}^{(S_1 \cup S_2, \mathcal{A}_0)}$ yields a $\tilde{\mathbf{\Gamma}}^{(S_1 \cup S_2, \mathcal{A}_0)}$ satisfying that $\tilde{\mathbf{\Gamma}}^{(S_i, \mathcal{A}_0)}$ has distinct column vectors for $i = 1, 2$ and " \succeq_{S_1} " = " \succeq_{S_2} " under $\tilde{\mathbf{\Gamma}}^{\mathcal{A}_0}$.
- (B) For any $\alpha, \alpha' \in \mathcal{A}_0$ where $\alpha' \succeq_{S_i} \alpha$ under $\mathbf{\Gamma}^{\mathcal{A}_0}$ for $i = 1$ or 2 , there exists some $j \in (S_1 \cup S_2)^c$ such that $\tilde{\mathbf{\Gamma}}_{j,\alpha}^{\mathcal{A}_0} \neq \tilde{\mathbf{\Gamma}}_{j,\alpha'}^{\mathcal{A}_0}$.
- (C) Any column vector of $\mathbf{\Gamma}^{\mathcal{A}_0}$ is different from any column vector of $\mathbf{\Gamma}^{\mathcal{A}_0^c}$, where $\mathcal{A}_0^c = \{0, 1\}^K \setminus \mathcal{A}_0$.

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

Theorem 2 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 1 $[l_N(\hat{\pi}^*, \hat{\Theta}^*) - l_N(\hat{\pi}_0, \hat{\Theta}_0)]/N = O_p(N^{-\delta})$, for some $1/2 < \delta \leq 1$, where $(\hat{\pi}^*, \hat{\Theta}^*)$ is the maximum likelihood estimator (MLE) directly obtained from (9), and $(\hat{\pi}_0, \hat{\Theta}_0)$ 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 (π^0, Θ^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 $(\hat{\pi}, \hat{\Theta})$ obtained from optimizing (10), we define the selected latent classes as $\{m : \hat{\pi}_m > \rho_N, m \in [M]\}$, and the number of selected latent classes $\hat{M} := |\{m : \hat{\pi}_m > \rho_N, m \in [M]\}|$. For the true item parameter matrix Θ^0 , we defined the set $S^0 = \{(j, k_1, k_2) : \theta_{j,k_1}^0 = \theta_{j,k_2}^0, 1 \leq k_1 < k_2 \leq M_0, 1 \leq j \leq J\}$ to indicate the constraint structure of the item parameter matrix. Similarly, for $(\hat{\pi}, \hat{\Theta})$, we define $\hat{S} = \{(j, k_1, k_2) : \hat{\theta}_{j,k_1} = \hat{\theta}_{j,k_2}, 1 \leq k_1 < k_2 \leq M, \hat{\pi}_{k_1} > \rho_N, \hat{\pi}_{k_2} > \rho_N\}$. We say $\hat{S} \sim S^0$ if there exists a column

permutation σ of $\hat{\Theta}$ such that $\hat{S}_\sigma = S^0$. Given the above assumptions, we have the following consistency results.

Theorem 3 (consistency) *Suppose the identifiability conditions in Theorem 1 are satisfied and Assumption 1 holds. For $\lambda_1, \lambda_2, \tau$ 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,*

$$\mathbb{P}(\hat{M} \neq M_0) \rightarrow 0, \text{ as } N \rightarrow \infty. \quad (12)$$

Moreover, the estimated parameter $(\hat{\pi}, \hat{\Theta})$ is also consistent of (π^0, Θ^0) . If we further assume $\lambda_1 = o(N^{1/2})$, $\lambda_2\tau = o(N^{1/2})$, $\lambda_2N^{-1/2} \rightarrow \infty$ and $\tau N^{1/2} \rightarrow \infty$, up to a column permutation, the identical item parameter pair set S^0 is also consistently estimated,

$$\mathbb{P}(\hat{S} \not\approx S^0) \rightarrow 0, \text{ as } N \rightarrow \infty. \quad (13)$$

Theorem 3 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 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, 1978), to help select the tuning parameters, which will be further discussed in Section 4.1. The proof of the theorem is presented in the supplementary material.

Based on the learned latent classes and estimated item parameter matrix, we develop a latent structure recovery algorithm outlined in Algorithm 2 in Section 4. Specifically, we recover the number of latent attributes, the 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 $\hat{\Theta}$ 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 4.

4 Learning Algorithms

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

$$l_C(\mathcal{R}, \mathbf{z}; \boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{i=1}^N \sum_{k=1}^M z_{ik} \log(\pi_k \varphi(\mathbf{R}_i; \boldsymbol{\theta}_k)), \quad (14)$$

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

$$\max_{\boldsymbol{\pi}, \boldsymbol{\Theta}} Q(\boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)}) = \sum_{i=1}^N \sum_{k=1}^M s_{ik}^{(c)} (\log \pi_k + \log \varphi_k(\mathbf{R}_i; \boldsymbol{\theta}_k)), \quad (15)$$

where

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

With the additional penalty terms in (10), the new objective function denoted as $G(\boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)})$ becomes:

$$\min_{\boldsymbol{\pi}, \boldsymbol{\Theta}} G(\boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)}) = -\frac{1}{N} Q(\boldsymbol{\pi}, \boldsymbol{\Theta} \mid \boldsymbol{\pi}^{(c)}, \boldsymbol{\Theta}^{(c)}) + \tilde{\lambda}_1 \sum_{k=1}^M \log_{[\rho_N]} \pi_k + \tilde{\lambda}_2 \sum_{j=1}^J \mathcal{J}_{\tau, \rho_N}(\boldsymbol{\theta}_j), \quad (16)$$

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

As we mentioned in Section 3.1, the truncated Lasso penalty can be decomposed into a difference of 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 1 and the derivations of the algorithm is presented in the supplementary material.

Algorithm 1: PEM: Penalized EM with log-penalty and truncated Lasso penalty

Data: Binary response matrix $\mathcal{R} = (R_{i,j})_{N \times J}$.

Set hyperparameters $\tilde{\lambda}_1, \tilde{\lambda}_2, \tau, \gamma$ and ρ .

Set an upper bound for the number of latent classes M .

Initialize parameters $\boldsymbol{\pi}, \boldsymbol{\Theta}$, and the conditional expectations \mathbf{s} .

while not converged do

 In the $(c+1)$ th iteration,

for $(i, k) \in [N] \times [M]$ **do**

$$\left[s_{ik}^{(c+1)} = \frac{\pi_k^{(c)} \varphi_k(\mathbf{R}_i; \boldsymbol{\theta}_k^{(c)})}{\sum_{k'} \pi_{k'}^{(c)} \varphi_{k'}(\mathbf{R}_i; \boldsymbol{\theta}_{k'}^{(c)})}, \text{ where } \varphi(\mathbf{R}_i; \boldsymbol{\theta}_k) = \prod_{j=1}^J \theta_{jk}^{R_{ij}} (1 - \theta_{jk})^{1-R_{ij}}. \right.$$

for $k \in [M]$ and $\pi_k^{(c)} > \rho$ **do**

$$\left[\pi_k^{(c+1)} = \frac{\sum_{i=1}^N s_{ik}^{(c+1)} / N - \tilde{\lambda}_1}{1 - M\tilde{\lambda}_1}. \right.$$

for $(j, k) \in [J] \times [M]$ and $\pi_k^{(c+1)} > \rho$ **do**

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

for $j \in [J], 1 \leq k < l \leq M$ and $\pi_k^{(c+1)} > \rho, \pi_l^{(c+1)} > \rho$ **do**

$$\begin{aligned} \hat{d}_{jkl}^{(c+1)} = \begin{cases} (\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}) \cdot I(|\hat{d}_{jkl}^{(c)}| \geq \tau) \\ \text{ST}(\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)} - \hat{\mu}_{jkl}^{(c)}; \tilde{\lambda}_2/\gamma) \cdot I(|\hat{d}_{jkl}^{(c)}| < \tau), \end{cases} \\ \text{where } \text{ST}(x; \gamma) = (|x| - \gamma)_+ x / |x|. \\ \hat{\mu}_{jkl}^{(c+1)} = \hat{\mu}_{jkl}^{(c)} + \hat{d}_{jkl}^{(c+1)} - (\hat{\theta}_{jk}^{(c+1)} - \hat{\theta}_{jl}^{(c+1)}) \end{aligned}$$

Output: $\{\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}, \hat{\mathbf{s}}\}$

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}_{obs} and the missing part $\mathcal{R}_{\text{miss}}$, then after marginalization over the missing values, the initial likelihood $l(\mathcal{R}; \boldsymbol{\pi}, \boldsymbol{\Theta})$ is

simplified 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 with missing values is summarized in the supplementary material.

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 $(\hat{\boldsymbol{\pi}}^{(c+1/2)}, \hat{\boldsymbol{\Theta}}^{(c+1/2)})$. Then we use a weighted average of $(\hat{\boldsymbol{\pi}}^{(c)}, \hat{\boldsymbol{\Theta}}^{(c)})$ and $(\hat{\boldsymbol{\pi}}^{(c+1/2)}, \hat{\boldsymbol{\Theta}}^{(c+1/2)})$ 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:

$$\text{BIC}(\boldsymbol{\pi}, \boldsymbol{\Theta}) = -2l_N(\boldsymbol{\pi}, \boldsymbol{\Theta}) + \log N(M_{\rho_N} - 1 + \sum_{j=1}^J \dim(\boldsymbol{\theta}_j)) \quad (17)$$

where l_N is the log-likelihood, $M_{\rho_N} := |\{m : \pi_m > \rho_N, m \in [M]\}|$ 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 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.

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 $\hat{\boldsymbol{\Theta}}$, 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 CDMs is that more capable subjects have higher item parameters and thus larger indicator vectors, that is, $\mathbf{\Gamma}_{\cdot, \alpha} \succeq \mathbf{\Gamma}_{\cdot, \alpha^*}$, if $\alpha \succeq \alpha^*$. 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 $\mathbf{P} \in \{0, 1\}^{\hat{M} \times \hat{M}}$ 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 $(\mathbf{P}^2)_{m_1, m_2} > 0$, 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 \mathbf{P} . From \mathbf{P} , we can get a partial order set $\{m_1 \rightarrow m_2 : P_{m_1, m_2} = 1\}$, 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 \mathbf{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 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.

$$\mathbf{\Gamma} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \text{Diagram: } \textcircled{\Gamma_{\cdot,1}} \rightarrow \textcircled{\Gamma_{\cdot,2}} \rightarrow \textcircled{\Gamma_{\cdot,3}}$$

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

After we plot the DAG, we then recover the 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 7 and Figure 8 to illustrate the procedures of recovering binary representations. In the upper plot of Figure 7, $\Gamma_{.,2}$ only has one parent node $\Gamma_{.,1}$, then we need to add a dimension in the binary representations. In the middle plot of Figure 7, $\Gamma_{.,3}$ has two parent nodes $\Gamma_{.,1}$ and $\Gamma_{.,2}$. Since there is no partial order between $\Gamma_{.,1}$ and $\Gamma_{.,2}$, then there are at least two dimensions in which $\Gamma_{.,1}$ and $\Gamma_{.,2}$ have different values. Then we should set $\Gamma_{.,3}$ to be the union of $\Gamma_{.,1}$ and $\Gamma_{.,2}$, which will be larger than $\Gamma_{.,1}$ and $\Gamma_{.,2}$. A more general case is shown in the lower plot of Figure 7. In Figure 8, we provide a more complicated example. Since $\Gamma_{.,2}$ only has one parent node $\Gamma_{.,1}$, we need one binary digit for $\Gamma_{.,2}$ and set $\Gamma_{.,1} = (0)$ and $\Gamma_{.,2} = (1)$. Since $\Gamma_{.,3}$ and $\Gamma_{.,4}$ also have only one parent node, we need two additional dimensions, and set $\Gamma_{.,3} = (1, 1, 0)$ and $\Gamma_{.,4} = (1, 0, 1)$. Next because $\Gamma_{.,5}$ has two parent nodes $\Gamma_{.,3}$ and $\Gamma_{.,4}$, we set $\Gamma_{.,5} = (1, 1, 1)$. Lastly since $\Gamma_{.,6}$ has one parent node $\Gamma_{.,5}$, we need one more dimension and set $\Gamma_{.,6} = (1, 1, 1, 1)$. Therefore, in total we have four latent attributes and the reconstruction process is highlighted in blue in Figure 8. We want to point out that when we recover the latent structures using Algorithm 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.

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 8, our reconstructed latent attribute profile matrix \mathcal{A} is shown in Figure 9, where rows of \mathcal{A} are the binary representations of the latent classes. We can see that $\mathcal{A}_{.,1} \succeq \mathcal{A}_{.,k}$ 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

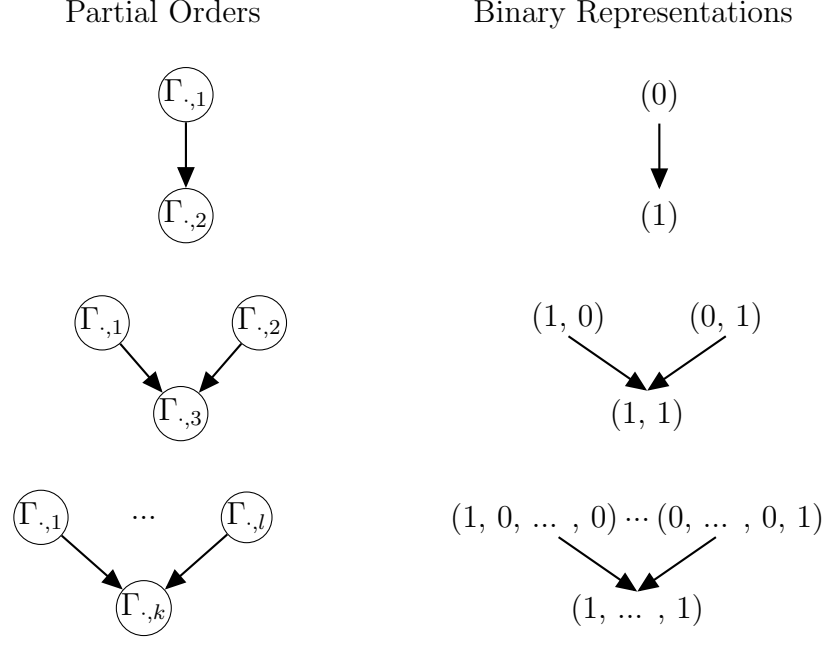


Figure 7: Examples of binary representations from partial orders

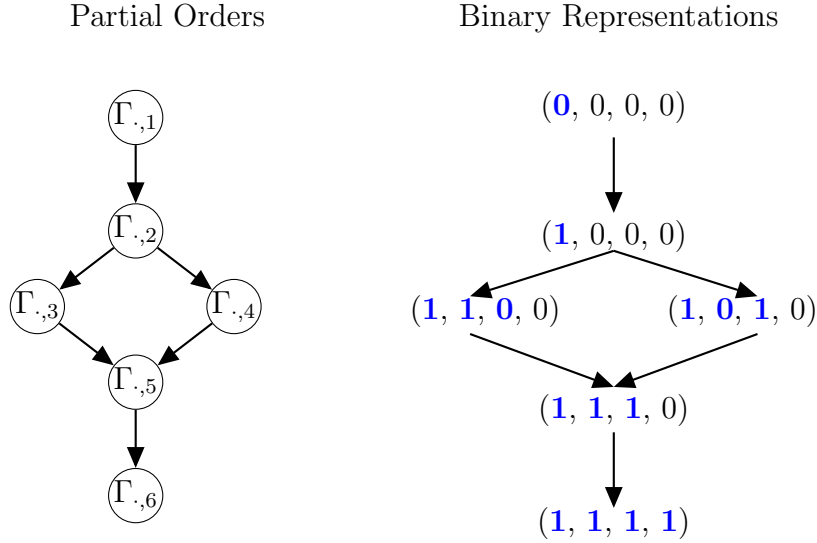


Figure 8: A more complicated example of binary representations from partial orders

is 1 only if all the other attributes are 1, indicating that the fourth attribute is the highest and requires all the other attributes as prerequisites. Formally, we can use $\mathcal{E} = \{k \rightarrow l : \text{attribute } k \text{ is a prerequisite for attribute } l\}$ introduced in Section 2.1 to denote the prerequisite relationship set, where $k \rightarrow l$ if $\mathcal{A}_{.,k} \succeq \mathcal{A}_{.,l}$. For latent attribute profile matrix \mathcal{A} in Figure 9, we have $\mathcal{E} = \{1 \rightarrow 2, 1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 4\}$. We can also plot a DAG according to the prerequisite

relationship set \mathcal{E} as shown in the right plot of Figure 9.

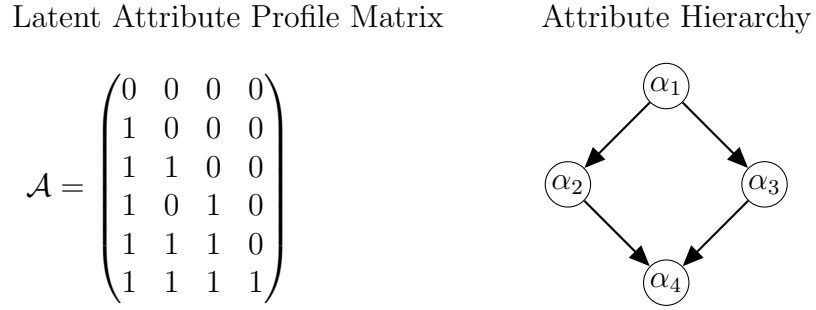


Figure 9: Latent attribute profile matrix and attribute hierarchy; rows of \mathcal{A} are the binary representations of the select latent classes in Figure 8.

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 to the smallest latent attribute profile such that the corresponding indicator is 1. To be more formal, let \mathbf{q}_j be the j th row of the Q -matrix, we have

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

where $\mathcal{A}_{m,\cdot}$ denotes the m th row vector of the latent attribute profile matrix \mathcal{A} , i.e., the binary representation of the m th latent class in $\mathbf{\Gamma}$. The procedures are summarized in Algorithm 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 (Gu and Xu, 2019b,c; Xu and Zhang, 2016), 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 models and the remains

Algorithm 2: Recover Latent Attribute Profiles, Hierarchical Structure and Q -matrix

Input : Item parameter Matrix Θ

Step 1 : Construct the indicator matrix $\Gamma = (\mathbb{I}\{\theta_{j,m} = \max_{l \in [M]} \theta_{j,l}\})$.

Step 2 : Construct \mathbf{P} based on the partial orders among the columns of Γ ;
plot a DAG based on \mathbf{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

if *the node has more than one parent node* **then**

 └ set the binary representation to be the union of all of its parent nodes

Step 4 : Construct prerequisite relationship set \mathcal{E} and thus recover latent hierarchy.

Step 5 : Reconstruct the Q -matrix $\mathbf{Q} = (\mathbf{q}_j)_{j=1}^J$:

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

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

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 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 parameter was 0.9 or 0.8, and the lowest item parameters 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 hyperparameters 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 penalty 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 $\tilde{\lambda}_1 \in \{0.01, 0.015, \dots, 0.05\}$ 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 penalty 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 penalty would facilitate the shrinkage of the proportion parameters, while a large TLP penalty 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 penalty 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. \(2016a\)](#), 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 of all the other 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 hyperparameters, but our simulation results below showed that the aforementioned candidate sets were enough to provide good results.

Following [Chen et al. \(2017\)](#) 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. \(2017\)](#), 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 Section 4 of Supplementary Material. For the penalty term, we selected $\lambda \in \{0.01, 0.02, \dots, 0.1\}$ 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 $\text{Acc}(\hat{M})$. Based on the learned item parameters, we reconstruct the indicator matrix $\hat{\Gamma} = (\mathbb{I}\{\theta_{j,m} = \max_{l \in [\hat{M}]} \theta_{j,l}\}) \in \{0, 1\}^{J \times \hat{M}}$ and the corresponding partial order matrix $\hat{\mathbf{P}}$. It's worth noting that when we extract the partial orders among the latent classes, a 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 $\Gamma_{\cdot,k} \succeq \Gamma_{\cdot,j}$, 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. 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 $\text{Acc}(\hat{\mathbf{P}})$ 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 $\text{Acc}(\hat{\mathcal{E}})$ 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 $\text{MSE}(\hat{\boldsymbol{\Theta}})$. Finally, if the hierarchical structure is correctly recovered, we compute the accuracy of the estimated Q -matrix, denoted by $\text{Acc}(\hat{\mathbf{Q}})$. In summary, we have five evaluation metrics : $\text{Acc}(\hat{M})$, $\text{Acc}(\hat{\mathbf{P}})$, $\text{Acc}(\hat{\mathcal{E}})$, $\text{MSE}(\hat{\boldsymbol{\Theta}})$ and $\text{Acc}(\hat{\mathbf{Q}})$.

The simulation results of the DINA, and DINA + DINO settings are presented in Table 1 and Table 2. The results of the GDINA model are shown in Table 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 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 sufficiently large data size. Among the four hierarchical structures, the

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

Table 1: DINA Results; $\text{Acc}(\hat{M})$, $\text{Acc}(\hat{P})$ and $\text{Acc}(\hat{E})$ are calculated for all the cases; $\text{MSE}(\hat{\Theta})$ is calculated for the cases when the number of latent classes are correctly selected; $\text{Acc}(\hat{Q})$ is calculated for the cases when the hierarchical structure is successfully recovered.

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,

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

Table 2: DINA+DINO Results; $\text{Acc}(\hat{M})$, $\text{Acc}(\hat{P})$ and $\text{Acc}(\hat{E})$ are calculated for all the cases; $\text{MSE}(\hat{\Theta})$ is calculated for the cases when the number of latent classes are correctly selected; $\text{Acc}(\hat{Q})$ is calculated for the cases when the hierarchical structure is successfully recovered.

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 $\alpha = (1, 0, 0, 0)$ and $\alpha' = (1, 1, 0, 0)$. Under the DINA model, only the items with the q-vector $\mathbf{q}_j = (0, 1, 0, 0)$ or $(1, 1, 0, 0)$ can distinguish these two latent classes. By contrast, under the DINO model, the

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

Table 3: GDINA Results; $\text{Acc}(\hat{M})$, $\text{Acc}(\hat{P})$ and $\text{Acc}(\hat{E})$ are calculated for all the cases; $\text{MSE}(\hat{\Theta})$ is calculated for the cases when the number of latent classes are correctly selected; $\text{Acc}(\hat{Q})$ is calculated for the cases when the hierarchical structure is successfully recovered.

items with $\mathbf{q}_j = (0, 1, *, *)$ 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 $\mathbf{q}_j = (*, 1, *, *)$. Therefore, if the underlying data generating model is the DINA model, it requires larger sample size to achieve good performance. It is also noted that for the Q -matrix estimation, $\text{Acc}(\hat{Q})$ for both 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 much better overall Q -matrix estimation.

6 Real Data Analysis

6.1 Analysis of ECPE Data

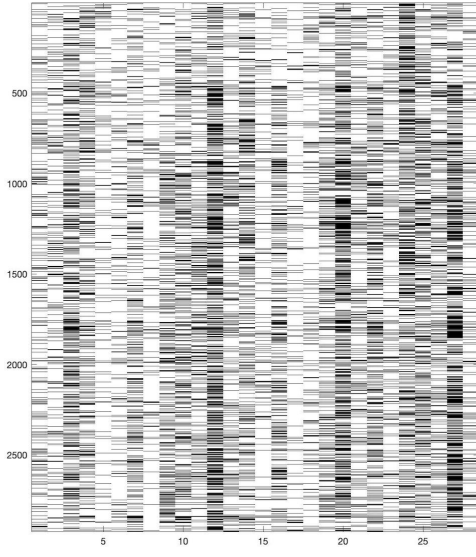
We applied the proposed approach to the Examination for the Certificate of Proficiency in English (ECPE) data to learn the latent hierarchical structure. The ECPE data was collected by the English Language Institute of the University of Michigan, and we used the data from R package CDM. The dataset includes 2,922 examinees and 28 ECPE items with three target attributes including lexical rules, cohesive rules and morphosyntactic rules. In the literature of the analysis of the ECPE data, [Templin and Bradshaw \(2014\)](#) fitted an HCDM with the corresponding 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 CDM 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 of the third one. Though estimations of the ECPE data have been widely studied under the CDM setting, [von Davier and Haberman \(2014\)](#) pointed out that ECPE data appear to have mainly a unidimensional structure, which may not be suitable for CDM modeling.

Our proposed method uses a penalized exploratory latent class analysis approach, which does not depend on the CDM 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 number of latent classes M . Here we took $M = 8$, and used spectral clustering to initialize the 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 $D = \text{diag}\{\sum_j R_{1j}, \sum_j R_{2j}, \dots, \sum_j R_{Nj}\}$. Then we took the first M eigenvectors of L and performed

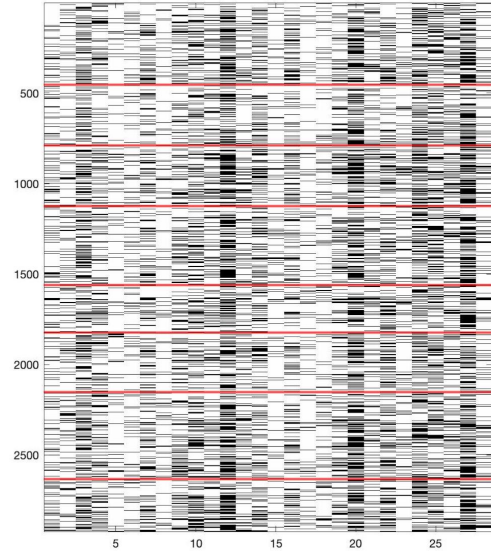
k -means clustering on the eigenvectors. Based on the clustering results, we had an initialization of the partition of the subjects to M classes and then used class proportions and mean responses to the items as the model initializations. The clustered data from spectral initialization is shown in Figure 10(b) and the final estimation results with spectral initialization is in Figure 10(d). In the plots, 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.

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 10(c). We found that the GDINA model initialization using the pre-specified Q -matrix resulted in the same learned model as the spectral initialization, which does not require the pre-specified Q -matrix. 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 10(c) and Figure 10(d), we can also see that the clustered data based on our method showed a much clearer ordered structure among the latent classes. Specifically, using the proposed method, we obtained four significant latent classes, as shown in Figure 10(d). From the clustered results, there seemed to be an ordered structure: the subjects in the first cluster were more likely to give positive responses than those in the second clusters, and the second cluster tended 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 $\hat{\Theta}$ and the reconstructed indicator matrix $\hat{\Gamma}$ are shown in Figure 11. It is easy to see $\hat{\Gamma}_{\cdot,1} \prec \hat{\Gamma}_{\cdot,2} \prec \hat{\Gamma}_{\cdot,3} \prec \hat{\Gamma}_{\cdot,4}$, which indicated 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).

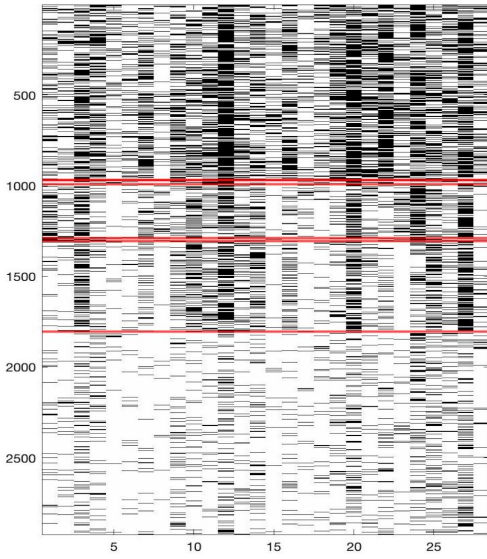
To present the latent class structure under the HCDM framework, we can apply the proposed Algorithm 2. Since there are four latent classes and $\hat{\Gamma}_{\cdot,1} \prec \hat{\Gamma}_{\cdot,2} \prec \hat{\Gamma}_{\cdot,3} \prec \hat{\Gamma}_{\cdot,4}$, the smallest K will be 3 and the corresponding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



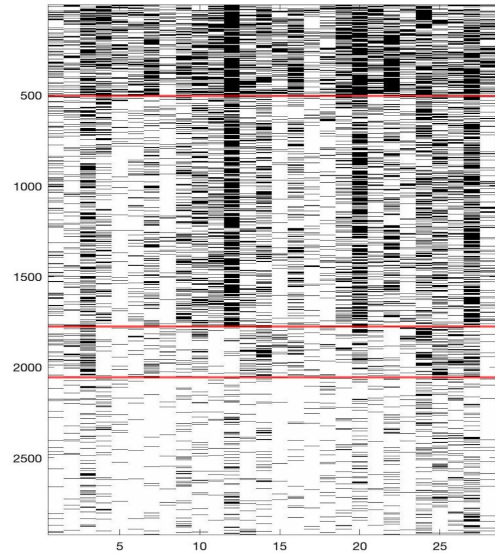
(a)



(b)



(c)



(d)

Figure 10: (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.

shaw (2014) under the CDM framework. Moreover, we also fitted the 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 obtained from our method and the original Q -matrix are shown in Figure 11(b) and 11(c). From the fitted GDINA models, we found that the BIC for the original Q -matrix was 86,117, while the BIC for our learned Q -matrix was 86,000, indicating that our learned Q fits the data better in terms of BIC.

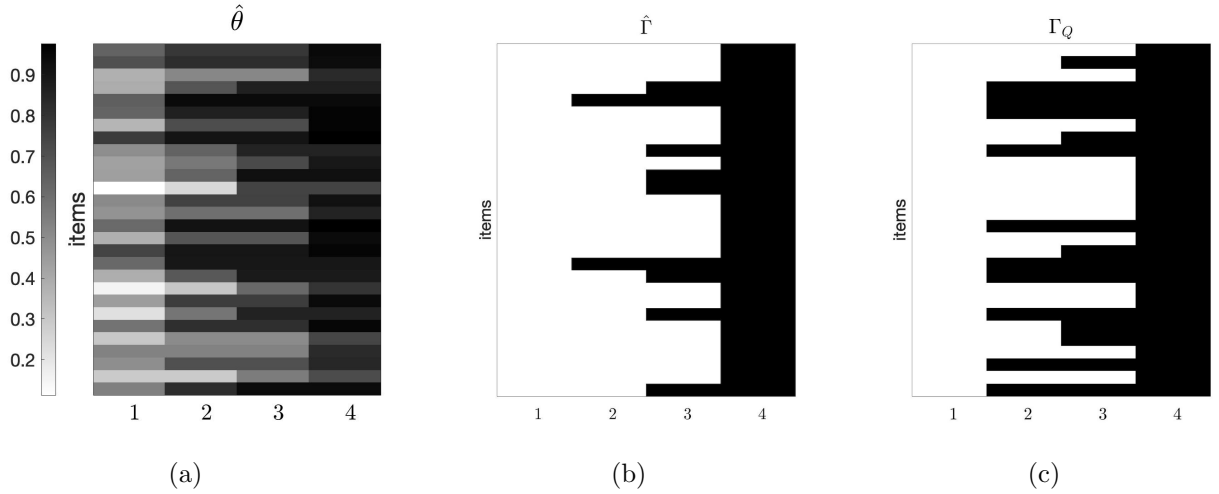


Figure 11: (a): estimated $\hat{\Theta}$ matrix; (b): reconstructed indicator matrix $\hat{\Gamma}$; (c): the indicator matrix based on the pre-specified Q -matrix. Black blocks indicate value 1, and white blocks indicate value 0.

6.2 Analysis of PISA Data

To test on 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 examinee’s reading abilities. Most of 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 models 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 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 $\hat{\Theta}$ and the reconstructed indicator matrix $\hat{\Gamma}$ are shown in Figure 12(a) and Figure 12(b), respectively. Based on the indicator matrix, we recovered the partial orders of these 10 latent classes in Figure 12(c), which suggests a multi-dimensional latent structure. With partial orders recovered, we applied Algorithm 2 and recovered six latent attributes with a hierarchical structure as shown in Figure 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 12(d), α_1 can be viewed as a basic prerequisite for other attributes, which makes sense because examinees need to first understand the item and correctly identify the key information in the article before forming understanding or evaluating the text, while developing a logical interpretation (α_6) can be interpreted as a more advanced skill.

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

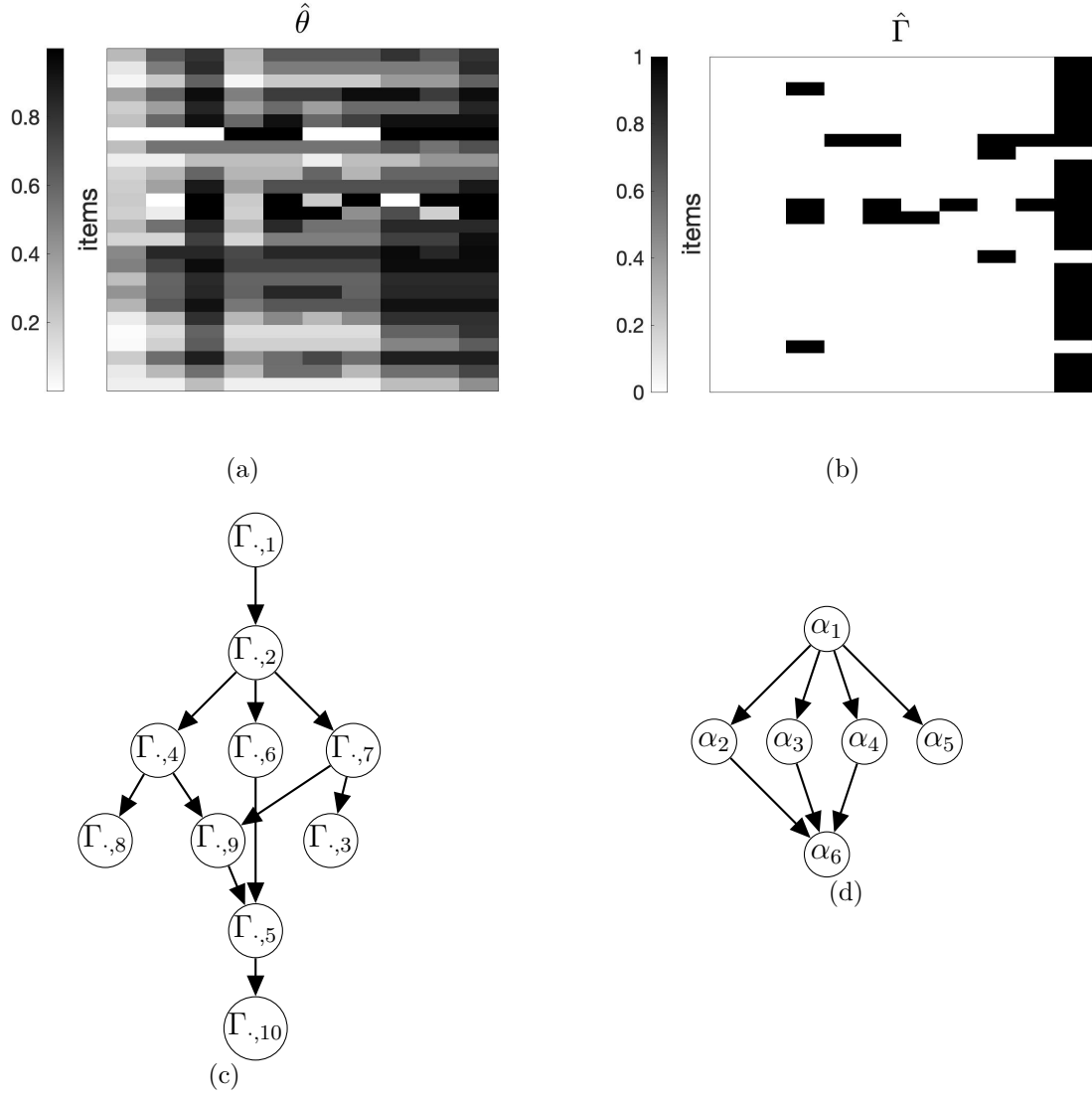


Figure 12: (a): Estimated $\hat{\Theta}$ matrix; (b): Reconstructed indicator matrix $\hat{\Gamma}$. (c): Partial orders; (d): Constructed hierarchical structure of latent attributes.

7 Discussion

In this paper 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 hierarchical CDMs**. 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 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 good performance of the proposed method.

In most existing works of learning CDMs, 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, 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 estimation 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 CDM 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.

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 learning trajectories of the subjects. Moreover, taking the hierarchical structures into consideration, we can generate recommendations for learning materials or test items by formulating a sequential decision problem. We leave these interesting directions for future work.

Acknowledgments

The authors are grateful to the Editor-in-Chief Professor Matthias von Davier, an Associate Editor, and three referees for their valuable comments and suggestions. This research is partially supported by NSF CAREER SES-1846747 and Institute of Education Sciences R305D200015.

References

- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.
- Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.
- Chen, J. and de la Torre, J. (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., Culpepper, S. A., Chen, Y., and Douglas, J. (2018). Bayesian estimation of the DINA Q-matrix. *Psychometrika*, 83(1):89–108.
- Chen, Y., Li, X., Liu, J., and Ying, Z. (2017). Regularized latent class analysis with application in cognitive diagnosis. *Psychometrika*, 82(3):660–692.
- Chen, Y., Liu, J., Xu, G., and Ying, Z. (2015). Statistical analysis of Q-matrix based diagnostic classification models. *Journal of the American Statistical Association*, 110(510):850–866.

- Chiu, C.-Y. (2013). Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, 37(8):598–618.
- Chung, M. and Johnson, M. S. (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., Hult, H., Dahlgren, L. O., af Segerstad, H. H., and Johansson, K. (2006). From senior student to novice worker: Learning trajectories in political science, psychology and mechanical engineering. *Studies in Higher Education*, 31(5):569–586.
- 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. (2011). The generalized DINA model framework. *Psychometrika*, 76(2):179–199.
- de la Torre, J. and Chiu, C.-Y. (2016). A general method of empirical Q-matrix validation. *Psychometrika*, 81(2):253–273.
- de la Torre, J., van der Ark, L. A., and Rossi, G. (2018). Analysis of clinical data from a cognitive diagnosis modeling framework. *Measurement and Evaluation in Counseling and Development*, 51(4):281–296.
- 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., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128.
- DiBello, L. V., Stout, W. F., and Roussos, L. A. (1995). Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques. *Cognitively Diagnostic Assessment*, 361389.
- Fan, J. and Li, R. (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 Robitzsch, A. (2015). Cognitive diagnosis models in R: A didactic. *The Quantitative Methods for Psychology*, 11(3):189–205.
- Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61(2):215–231.
- Gu, Y., Liu, J., Xu, G., and Ying, Z. (2018). Hypothesis testing of the Q-matrix. *Psychometrika*, 83(3):515–537.
- Gu, Y. and Xu, G. (2019a). Identification and estimation of hierarchical latent attribute models. *arXiv preprint arXiv:1906.07869*.
- Gu, Y. and Xu, G. (2019b). Learning attribute patterns in high-dimensional structured latent attribute models. *Journal of Machine Learning Research*, 20(2019).
- Gu, Y. and Xu, G. (2019c). The sufficient and necessary condition for the identifiability and estimability of the DINA model. *Psychometrika*, 84(2):468–483.
- 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.
- Henson, R. A., Templin, J. L., and Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2):191.
- Jimoyiannis, A. and Komis, V. (2001). Computer simulations in physics teaching and learning: a case study on students’ understanding of trajectory motion. *Computers & Education*, 36(2):183–204.
- Junker, B. W. and Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272.
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.
- Leighton, J. P., Gierl, M. J., and Hunka, S. M. (2004). The attribute hierarchy method for cognitive assessment: A variation on Tatsuoaka’s rule-space approach. *Journal of Educational Measurement*, 41(3):205–237.

- Li, C., Ma, C., and Xu, G. (2022). Learning large Q-matrix by restricted Boltzmann machines. *Psychometrika*, pages 1–32, DOI 10.1007/s11336-021-09828-4.
- Liu, J., Xu, G., and Ying, Z. (2012). Data-driven learning of Q-matrix. *Applied Psychological Measurement*, 36(7):548–564.
- Ma, C. and Xu, G. (2021). Hypothesis testing for hierarchical structures in cognitive diagnosis models. *Journal of Data Science*, pages 1–24, DOI 10.6339/21-JDS1024.
- O’Brien, K. L., Baggett, H. C., Brooks, W. A., Feikin, D. R., Hammitt, L. L., Higdon, M. M., Howie, S. R., Knoll, M. D., Kotloff, K. L., Levine, O. S., 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.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- Shen, X., Pan, W., and Zhu, Y. (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232.
- Simon, M. A. and Tzur, R. (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.
- Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, pages 345–354.
- Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnosis. *Diagnostic Monitoring of Skill and Knowledge Acquisition*, pages 453–488.
- Templin, J. and Bradshaw, L. (2014). Hierarchical diagnostic classification models: A family of models for estimating and testing attribute hierarchies. *Psychometrika*, 79(2):317–339.
- Templin, J. L. and Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods*, 11(3):287.
- Tuy, H. (1995). DC optimization: theory, methods and algorithms. In *Handbook of Global Optimization*, pages 149–216. Springer.

- von Davier, M. (2005). A general diagnostic model applied to language testing data. *ETS Research Report Series*, 2005(2):i–35.
- von Davier, M. (2019). The general diagnostic model. In *Handbook of Diagnostic Classification Models*, pages 133–153. Springer.
- von Davier, M. and Haberman, S. J. (2014). Hierarchical diagnostic classification models morphing into unidimensional ‘diagnostic’ classification models—a commentary. *Psychometrika*, 79(2):340–346.
- Wang, C. and Gierl, M. J. (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.
- Wang, C. and Lu, J. (2021). Learning attribute hierarchies from data: Two exploratory approaches. *Journal of Educational and Behavioral Statistics*, 46(1):58–84.
- Wang, S. and Liao, L. (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.
- Wu, C., Kwon, S., Shen, X., and Pan, W. (2016a). A new algorithm and theory for penalized regression-based clustering. *Journal of Machine Learning Research*, 17(1):6479–6503.
- Wu, Z., Deloria-Knoll, M., Hammitt, L. L., Zeger, S. L., and for Child Health Core Team, P. E. R. (2016b). Partially latent class models for case–control studies of childhood pneumonia aetiology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(1):97–114.
- Wu, Z., Deloria-Knoll, M., and Zeger, S. L. (2016c). Nested partially latent class models for dependent binary data; estimating disease etiology. *Biostatistics*, 18(2):200–213.
- Xu, G. (2017). Identifiability of restricted latent class models with binary responses. *The Annals of Statistics*, 45(2):675–707.
- Xu, G. and Shang, Z. (2018). Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association*, 113(523):1284–1295.

Xu, G. and Zhang, S. (2016). Identifiability of diagnostic classification models. *Psychometrika*, 81(3):625–649.

Supplementary Material for “Learning Latent and Hierarchical Structures in Cognitive Diagnosis Models”

Chenchen Ma, Jing Ouyang, and Gongjun Xu

Department of Statistics, University of Michigan

In the supplementary material, we provide the proof of the main theorem, the derivations for the penalized EM algorithm and a sensitivity analysis of our algorithm with varying upper bounds for the number of latent classes.

1 Proof for Theorem 3

In this section, we provide the proof of Theorem 3.

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 (??) and $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$ be the estimator obtained by optimizing the regularized log-likelihood (??). 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 $\hat{M} = |\{m : \hat{\pi}_m > \rho_N, m \in [M]\}|$ be the estimated number of latent classes. Without loss of generality, let $\hat{\boldsymbol{\pi}}_{\text{full}}^0 = (\hat{\boldsymbol{\pi}}^0, \mathbf{0}_{M-M_0})$. For the true item parameter matrix $\boldsymbol{\Theta}^0$, we defined the set of identical item parameter pairs $S^0 = \{(j, k_1, k_2) : \theta_{j,k_1}^0 = \theta_{j,k_2}^0, 1 \leq k_1 < k_2 \leq M_0\}$. Similarly, for $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}})$ we define $\hat{S} = \{(j, k_1, k_2) : \hat{\theta}_{j,k_1} = \hat{\theta}_{j,k_2}, 1 \leq k_1 < k_2 \leq M, \hat{\pi}_{k_1} > \rho_N, \hat{\pi}_{k_2} > \rho_N\}$. 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 $\mathbb{P}(\hat{M} \neq M_0)$ can be decomposed into two parts:

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

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

$$\mathbb{P}(\hat{S} \neq S^0) = \mathbb{P}(\hat{M} < M_0) + \mathbb{P}(\hat{M} > M_0) + \mathbb{P}(\hat{S} \neq S^0, \hat{M} = M_0). \quad (2)$$

In the following, we will bound each part in (1) and (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 $\hat{S} \neq S^0$.

The objective function is

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

where $\log_{[\rho_N]} \pi_k = \log \pi_k \cdot \mathbb{I}(\pi_k > \rho_N) + \log \rho_N \cdot \mathbb{I}(\pi_k \leq \rho_N)$. 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 $\{G_N(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}) > G_N(\hat{\boldsymbol{\pi}}^0, \hat{\boldsymbol{\Theta}}^0)\}$ implies that

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

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

For the LHS of (4), we have

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

where the last inequality follows from Assumption ?? . When $N^{1-\delta}/|\log(\rho_N)| = o(\lambda_N^{(1)})$, we have $N^{-\delta} = o(N^{-1}\lambda_N^{(1)}|\log \rho_N|)$, which implies that the event described in (4) will happen with probability tending to zero. Therefore we have $\mathbb{P}(\hat{M} > M_0) \rightarrow 0$ as $N \rightarrow \infty$. 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

$$\mathbb{P}\left(\sup_{\hat{M} < M_0} [G_N(\hat{\pi}, \hat{\Theta}) - G_N(\hat{\pi}^0, \hat{\Theta}^0)] > 0\right). \quad (5)$$

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

$$\mathbb{P}\left(\sup_{\hat{M} < M_0} [G_N(\hat{\pi}, \hat{\Theta}) - G_N(\hat{\pi}^0, \hat{\Theta}^0)] > 0\right) \leq \sum_{m=1}^{M_0-1} \mathbb{P}\left(\sup_{\hat{M}=m} [G_N(\hat{\pi}, \hat{\Theta}) - G_N(\hat{\pi}^0, \hat{\Theta}^0)] > 0\right), \quad (6)$$

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

$$\begin{aligned}
&\mathbb{P}\left(\sup_{h^2((\hat{\pi}, \hat{\Theta}), (\pi^0, \Theta^0)) \geq \epsilon_N^2} \left[\frac{1}{N} l_N(\hat{\pi}, \hat{\Theta}) - \frac{1}{N} l_N(\pi^0, \Theta^0)\right] > -\epsilon_N^2\right) \\
&\leq \mathbb{P}\left(\sup_{h^2((\hat{\pi}, \hat{\Theta}), (\pi^0, \Theta^0)) \geq \epsilon_N^2} \left[\frac{1}{N} l_N(\hat{\pi}, \hat{\Theta}) - \frac{1}{N} l_N(\pi^0, \Theta^0)\right] > -\epsilon_N^2\right) \leq \exp(-N\epsilon_N^2),
\end{aligned} \quad (7)$$

where $h^2((\hat{\pi}, \hat{\Theta}), (\pi^0, \Theta^0)) = \sum_{\mathbf{R} \in \{0,1\}^J} [\mathbb{P}(\mathbf{R} \mid \hat{\pi}, \hat{\Theta})^{1/2} - \mathbb{P}(\mathbf{R} \mid \pi^0, \Theta^0)^{1/2}]^2$ is the Hellinger distance. From the remark in Wong and Shen (1995), the inequality (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 $\mathcal{B}_m = \{(\hat{\pi}, \hat{\Theta}) : \hat{M} = m \leq M_0, h^2((\hat{\pi}, \hat{\Theta}), (\pi^0, \Theta^0)) \leq 2\epsilon_N^2\}$, 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 $S(t, m) = \{f_1^l, f_1^u, \dots, f_m^l, f_m^u\}$ 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 $\log(\min\{m : S(t, m)\})$. Following Lemma 3 in Gu and Xu (2019), for any $2^{-4}\epsilon < t < \epsilon$, there is

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

Next we need to verify the conditions in Wong and Shen (1995). Let's take $\epsilon_N = \sqrt{M_0 \log M/N}$ and verify the entropy integral condition in Theorem 1 of Wong and Shen (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 $N \rightarrow \infty$.

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

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

Moreover, for $\hat{M} = m < M_0$, there is $h^2((\hat{\pi}, \hat{\Theta}), (\pi^0, \Theta^0)) \geq (M_0 - m)C_{\min}(\pi^0, \Theta^0)$. In order

to have the probability of the event (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 (6) we have

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

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

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

Finally we consider the third case where $\hat{M} = M_0$ but $\hat{S} \not\approx S^0$. The argument is similar to the proof of Proposition 2 in [Xu and Shang \(2018\)](#). We first show $(\hat{\pi}_{\rho_N}, \hat{\Theta}_{\rho_N})$ converge to (π^0, Θ^0) with rate $N^{-1/2}$. For (π, Θ) with $(\pi_{\rho_N}, \Theta_{\rho_N})$ in a small neighborhood of (π^0, Θ^0) ,

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

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

expansion, we have $G_0(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}) - G_0(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0) = O(\|(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}) - (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\|^2)$.

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

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

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

$$\mathbb{E} \left(\sup_{\|(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}) - (\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0)\| \leq \zeta} G'_N(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}; \mathcal{R}) - G_0(\boldsymbol{\pi}_{\rho_N}, \boldsymbol{\Theta}_{\rho_N}) - G'_N(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0; \mathcal{R}) + G_0(\boldsymbol{\pi}^0, \boldsymbol{\Theta}^0) \right) = O(\zeta N^{-1/2}).$$

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

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

2 Derivations of PEM Algorithm

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

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

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

$$G(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{d}) = G_1(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{d}) - G_2(\mathbf{d}), \quad (12)$$

where

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

$$G_2(\mathbf{d}) = \tilde{\lambda}_2 \sum_{j=1}^J \sum_{1 \leq k < l \leq M} (|d_{jkl} - \tau|)_+. \quad (14)$$

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

$$G_2^{(c)}(\mathbf{d}) = G_2(\hat{\mathbf{d}}^{(c)}) + \tilde{\lambda}_2 \sum_{j=1}^J \sum_{1 \leq k < l \leq M} (|d_{jkl}| - |\hat{d}_{jkl}^{(c)}|) \cdot \mathbb{I}(|\hat{d}_{jkl}^{(c)}| \geq \tau), \quad (15)$$

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

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

$$+ \tilde{\lambda}_2 \tau \sum_{j=1}^J \sum_{1 \leq k < l \leq M} \mathbb{I}(|\hat{d}_{jkl}^{(c)}| \geq \tau).$$

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

$$L_\gamma(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{d}, \mathbf{y}) = G^{(c+1)}(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{d}) + \sum_{j=1}^J \sum_{1 \leq k < l \leq M} y_{jkl} \cdot (d_{jkl} - (\theta_{jk} - \theta_{jl})) + \frac{\gamma}{2} \sum_{j=1}^J \sum_{1 \leq k < l \leq M} |d_{jkl} - (\theta_{jk} - \theta_{jl})|^2, \quad (16)$$

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:

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

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

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

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

$$\begin{aligned} \boldsymbol{\pi}^{(c+1)} &= \underset{\boldsymbol{\pi}}{\operatorname{argmin}} G^{(c+1)}(\boldsymbol{\pi}, \boldsymbol{\Theta}^{(c)}, \mathbf{d}^{(c)}), \\ \boldsymbol{\Theta}^{(c+1)} &= \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} G^{(c+1)}(\boldsymbol{\pi}^{(c+1)}, \boldsymbol{\Theta}, \mathbf{d}^{(c)}) + \frac{\gamma}{2} \sum_{j=1}^J \sum_{1 \leq k < l \leq M} (d_{jkl}^{(c)} - (\theta_{jk}^{(c)} - \theta_{jl}^{(c)}) + \mu_{jkl}^{(c)}), \\ \mathbf{d}^{(c+1)} &= \underset{\mathbf{d}}{\operatorname{argmin}} G^{(c+1)}(\boldsymbol{\pi}^{(c+1)}, \boldsymbol{\Theta}^{(c+1)}, \mathbf{d}) + \frac{\gamma}{2} \sum_{j=1}^J \sum_{1 \leq k < l \leq M} (d_{jkl} - (\theta_{jk}^{(c+1)} - \theta_{jl}^{(c+1)}) + \mu_{jkl}^{(c)}), \end{aligned}$$

$$\mu_{jkl}^{(c+1)} = \mu_{jkl}^{(c)} + d_{jkl}^{(c+1)} - (\theta_{jk}^{(c+1)} - \theta_{jl}^{(c+1)}), \quad j = 1, \dots, J, 1 \leq k < l \leq M.$$

Specifically, we get the following updates:

(1)

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

(2)

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

(3)

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

(4)

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

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

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 i th subject's response to the j th item is missing. The details of the algorithm is summarized in Algorithm 1.

Algorithm 1: Penalized EM with missing data

Data: Binary response matrix $\mathcal{R} = (R_{i,j})_{N \times J}$ and the mask matrix $\mathbf{M} = (M_{i,j})_{N \times J}$ indicating missing values.

Set hyperparameters $\tilde{\lambda}_1, \tilde{\lambda}_2, \tau, \gamma$ and ρ .

Set an upper bound of the number of latent classes L .

Initialize parameters $\boldsymbol{\pi}, \boldsymbol{\Theta}$, and the conditional expectations \mathbf{s} .

while *not converged* **do**

 In the $(c+1)$ th iteration,

for $(i, k) \in [N] \times [L]$ **do**

$$\left[s_{ik}^{(c+1)} = \frac{\pi_k^{(c)} \varphi_k(\mathbf{R}_i; \boldsymbol{\theta}_k^{(c)})}{\sum_{k'}^{(c)} \pi_{k'}^{(c)} \varphi_{k'}(\mathbf{R}_i; \boldsymbol{\theta}_{k'}^{(c)})}, \quad \varphi(\mathbf{r}_i; \boldsymbol{\theta}_k) = \prod_{j=1}^J (\theta_{jk}^{R_{ij}} (1 - \theta_{jk})^{1-R_{ij}})^{m_{ij}} \right]$$

for $k \in [L]$ *and* $\pi_k^{(c)} > \rho$ **do**

$$\left[\pi_k^{(c+1)} = \frac{\sum_{i=1}^N s_{ik}^{(c+1)} / N - \tilde{\lambda}_1}{1 - L \tilde{\lambda}_1}. \right]$$

for $(j, k) \in [J] \times [L]$ *and* $\pi_k^{(c+1)} > \rho$ **do**

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

for $j \in [J], k, l \in [L], k < l$ *and* $\pi_k^{(c+1)} > \rho, \pi_l^{(c+1)} > \rho$ **do**

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

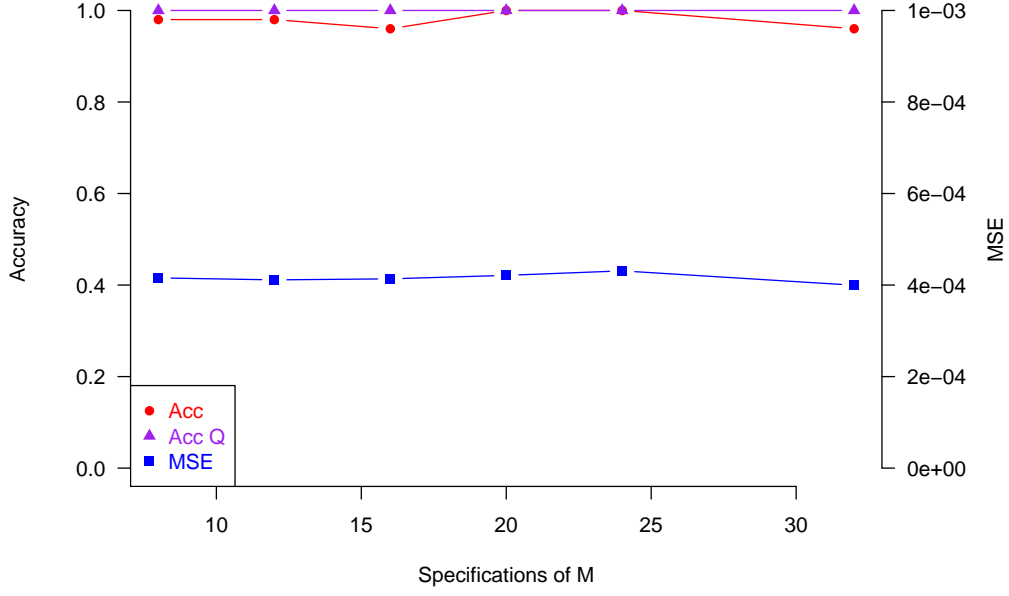
Output: $\{\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\Theta}}, \hat{\mathbf{s}}\}$

4 Sensitivity Analysis

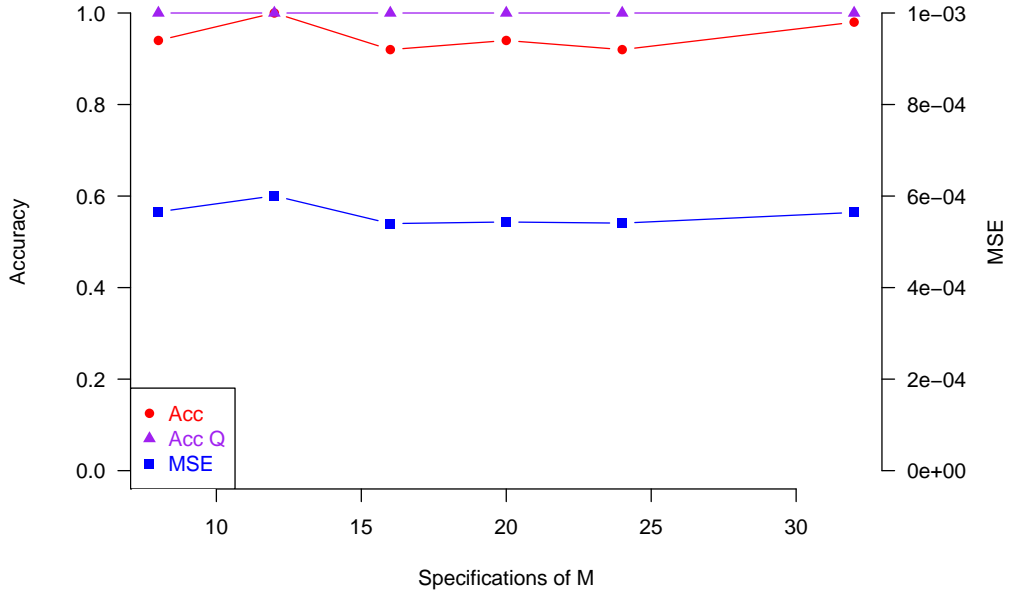
In this section, we conduct the sensitivity analysis of our algorithm by investigating the effects of different inputs of M , the upper bound of the number of latent classes, on the simulation results. In particular, we focus on two simulation settings: (1) DINA model with linear hierarchical structure, $N = 500$ 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 $\text{Acc}(\hat{M})$, $\text{Acc}(\hat{\mathbf{P}})$, $\text{Acc}(\hat{\mathcal{E}})$, $\text{MSE}(\hat{\boldsymbol{\Theta}})$ and $\text{Acc}(\hat{\mathbf{Q}})$. Consistent with the simulation studies in the main article, the $\text{Acc}(\hat{M})$, $\text{Acc}(\hat{\mathbf{P}})$ and $\text{Acc}(\hat{\mathcal{E}})$ are calculated for all the cases; $\text{MSE}(\hat{\boldsymbol{\Theta}})$ is calculated for the cases when the number of latent classes is correctly selected; $\text{Acc}(\hat{\mathbf{Q}})$ is calculated for the cases when the hierarchical structure is successfully recovered. The results are plotted in Figure 1.

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

References

- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.
- Gu, Y. and Xu, G. (2019). Learning attribute patterns in high-dimensional structured latent attribute models. *Journal of Machine Learning Research*, 20(2019).
- Shen, X., Pan, W., and Zhu, Y. (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232.
- Van Der Vaart, A. W. and Wellner, J. A. (1996). Weak convergence. In *Weak convergence and empirical processes*, pages 16–28. Springer.
- Wong, W. H. and Shen, X. (1995). Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *The Annals of Statistics*, 23(2):339–362.
- Xu, G. and Shang, Z. (2018). Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association*, 113(523):1284–1295.