A Tensor-EM Method for Large-Scale Latent Class Analysis with Binary Responses

Zhenghao Zeng[1], Yuqi Gu[2], and Gongjun Xu[3]

[1]Carnegie Mellon University

[2]Columbia University

[3]University of Michigan, Ann Arbor

Abstract

Latent class models are powerful statistical modeling tools widely used in psychological, behavioral, and social sciences. In the modern era of data science, researchers often have access to response data collected from large-scale surveys or assessments, featuring many items (large J) and many subjects (large N). This is in contrary to the traditional regime with fixed J and large N. To analyze such large-scale data, it is important to develop methods that are both computationally efficient and theoretically valid. In terms of computation, the conventional EM algorithm for latent class models tends to have a slow algorithmic convergence rate for large-scale data and may converge to some local optima instead of the maximum likelihood estimator (MLE). Motivated by this, we introduce the tensor decomposition perspective into latent class analysis with binary responses. Methodologically, we propose to use a moment-based tensor power method in the first step, and then use the obtained estimates as initialization for the EM algorithm in the second step. Theoretically, we establish the clustering consistency of the MLE in assigning subjects into latent classes when N and J both go to infinity. Simulation studies suggest that the proposed tensor-EM pipeline enjoys both good accuracy and computational efficiency for large-scale data with binary responses. We also apply the proposed method to an educational assessment dataset as an illustration.

Keywords: Large-scale latent class analysis; Tensor decomposition; Tensor power method; EM algorithm; Clustering consistency.

1 Introduction

Latent class models (LCMs) (Lazarsfeld and Henry, 1968; Goodman, 1974) are powerful statistical modeling tools widely used in psychological, behavioral, and social sciences. LCMs use a categorical latent variable to model the unobserved heterogeneity of multivariate categorical data and identify meaningful latent subgroups of subjects. LCMs have seen broad applications in a variety of scientific fields, including psychology and psychiatry (Bucholz et al., 2000; Keel et al., 2004), sociology and organizational research (Vermunt, 2003; Wang and Hanges, 2011), and biomedical and epidemiological studies (Bandeen-Roche et al., 1997; Dean and Raftery, 2010; Kongsted and Nielsen, 2017). For instance, Bucholz et al. (2000) explored the existence of potential subtypes of Antisocial Personality Disorder via latent class analysis. Keel et al. (2004) applied LCMs to empirically define four eating disorder phenotypes and identified features that differentiate between phenotypes. Wang and Hanges (2011) summarized several areas in organizational research where LCMs are particularly useful, such as identifying unobserved subpopulations and recognizing the unobserved heterogeneity in measurement functioning. Vermunt (2003) provided an overview on applications of LCM and its extensions in social science research. Dean and Raftery (2010) considered variable selection in LCMs and identified meaningful group structures in single nucleotide polymorphism data. Kongsted and Nielsen (2017) introduced and illustrated the applications of LCMs in health research. There are also various extensions and generalizations building upon LCMs, including LCMs with covariates or distal outcomes (Vermunt, 2010; Lanza and Rhoades, 2013; Ouyang and Xu, 2022), longitudinal LCMs and latent transition analysis (Dunn et al., 2006; Collins and Lanza, 2009), factor mixture models (Lubke and Muthén, 2005; Muthén and Shedden, 1999), and also restricted LCMs known as diagnostic classification models that involve multiple categorical latent variables (Rupp and Templin, 2008; Xu, 2017; von Davier and Lee, 2019). LCMs may also serve as initial modeling step before fitting a more delicate cognitive diagnostic model (Ma et al., 2022). For general introductions and applications of LCMs, see Hagenaars and McCutcheon (2002) and Collins and Lanza (2009).

In this paper, we focus on LCMs with binary responses for large-scale data, which are typically collected in modern educational assessments (correct/wrong responses) and psychological or social science surveys (yes/no responses). Such data are characterized by many test takers with large

N and also many items with large J. This is in contrary to the traditional regime with large N and fixed J, a relatively well understood setting. Such a large scope of data poses challenges to classical statistical analysis methods and calls for new developments for LCMs. In the following, we summarize the two main questions motivating our study.

The first question of large-scale latent class analysis is how to perform computations efficiently. A conventional estimation method is the Expectation-Maximization (EM; Dempster et al., 1977) algorithm to maximize the marginal likelihood. EM algorithms have two potential drawbacks, the slow algorithmic convergence rate in high-dimensional problems and a tendency to converge to some local optima when the initial values are poorly chosen (Balakrishnan et al., 2017). In practice, researchers often run EM with many random initializations and select the one that gives the largest log-likelihood value. This procedure can be very time-consuming, especially for large-scale data. It is thus desirable to develop more efficient computational tools for large-scale latent class analysis. Motivated by this, we introduce the tensor decomposition perspective into latent class analysis.

There has been active research on tensor decompositions since they were introduced in Hitchcock (1927). The concept of tensors appeared in the literature of psychometrics dating back to 1960s-1970s (Tucker, 1964, 1966; Harshman, 1970; Kruskal, 1976). The interest of tensor decompositions has expanded to other areas, including chemometrics (Smilde et al., 2005), signal processing (De Lathauwer and De Moor, 1998), and data mining (McCullagh, 2018). In particular, tensor methods have also been used in learning latent variable models. Anandkumar et al. (2012a) derived tensor structures for low-order moments of latent Dirichlet allocation and applied tensor power method to learning the parameters. Anandkumar et al. (2012b) used the method of moments for mixture models and hidden Markov models as a viable alternative to EM algorithms. Hsu and Kakade (2013) derived similar tensor structure in mixtures of spherical Gaussian models. Anandkumar et al. (2014) summarized the common structures in several different latent variable models and used tensor power method to learn the parameters under a unified framework. Generally speaking, these tensor methods are all based on lower-order moments of observed variables rather than the entire likelihood function. As a result, an advantage of using moment-based tensor decomposition algorithms for learning latent variable models is the provable consistency guarantee;

see Anandkumar et al. (2014) for more details.

Besides the computational challenge, the second question of large-scale latent class analysis is how to ensure the estimators in the large-N and large-J regime are theoretically valid and meaningful. Traditionally, the subjects' latent class indicators in an LCM are often treated as random variables and marginalized out to obtain the marginal likelihood; we call the resulting model a random-effect LCM. On the other hand, an alternative approach is to treat the subjects' latent class indicators as fixed unknown parameters and directly incorporate them into the likelihood; we call the resulting model a fixed-effect LCM. In the classical scenario with sample size N going to infinity and the number of items J held fixed, the fixed-effect LCMs are known to be inconsistent for estimating the subject-level latent class indicators (e.g., see Neyman and Scott, 1948). However, for data featuring large N and large J, with an increasing amount of information collected per subject, an interesting theoretical question is whether we can obtain consistency in clustering the subjects into their corresponding true latent class in fixed-effect LCMs?

In this paper, in the regime where both N and J go to infinity, we propose an efficient computational pipeline and develop the theory of clustering consistency for LCMs with binary responses. It is known that the method of moments can be viewed as good complementary to the maximum likelihood approach (Chaganty and Liang, 2013; Zhang et al., 2014; Balakrishnan et al., 2017). Balakrishnan et al. (2017) theoretically examined the properties of two-stage estimators where a suitable initial estimator is refined with the EM algorithm. Chaganty and Liang (2013) and Zhang et al. (2014) considered mixed linear models and multi-class crowd labeling problem, respectively. They showed that in both problems the tensor estimator based on moments serves as an effective initialization for EM algorithm. Inspired by their work, we introduce the two-step estimator for LCM. On the computational side, we propose an efficient two-step estimation pipeline integrating the moment-based tensor decomposition method and the EM algorithm. In the first step, we apply the tensor power method in Anandkumar et al. (2014) for LCMs to quickly and reliably find roughly accurate parameter estimates. In the second step, we propose to use the tensor estimates as initialization for the EM algorithm to refine the parameter estimation. With good initialization, EM algorithms typically converge in very few iterations. Therefore, such an estimation

pipeline combines the advantages of both the tensor decomposition algorithm and the EM algorithm for latent class analysis. Our extensive simulation studies empirically show that such an estimation pipeline enjoys both computational efficiency and estimation accuracy. Further, on the theoretical side, we prove the clustering consistency of the joint maximum likelihood estimator (joint MLE) for fixed-effect LCMs. That is, we prove that the joint MLE is consistent in estimating the subjects' latent class memberships under certain mild assumptions when N and J both go to infinity. We also derive a bound on the rate of convergence of the joint MLE's clustering performance. The consistency of item parameters is established as a corollary of clustering consistency.

The rest of this paper is organized as follows. The setups of random-effect and fixed-effect LCMs are introduced in Section 2. The proposed estimation procedures of large-scale LCM are presented in Section 3. Some preliminaries about tensor are also provided in Section 3 to make this section self-contained. Section 4 presents our theoretical results on clustering consistency of joint MLE. Section 5 presents simulation studies that evaluate the proposed estimation procedures and assess the empirical behavior of clustering consistency. A real data example is shown in Section 6, and we conclude this paper with some discussion in Section 7. All the proofs and additional simulation results are presented in Supplementary Material.

2 Latent Class Models with Binary Responses

In this section we introduce two perspectives of LCM. In random-effect LCMs, the latent class indicators are random variables; while in fixed-effect LCM, the latent class indicators are fixed and treated as unknown parameters. These two models share common assumptions on how the observed variables depend on the latent ones.

2.1 Latent Variables as Random Effects

We first introduce random-effect LCM. Consider a binary-outcome latent class model with J items and L classes. Throughout this paper we will use boldface type to denote vectors, matrices and tensors while standard type is used to denote scalars. There are two types of individual-specific variables in the model, that is a binary response vector $\mathbf{R}_i \in \{0,1\}^J$ and a latent variable $z_i \in [L]$.

Here $[L] = \{1, 2, \ldots, L\}$ is the set of positive integers smaller than or equal to L. The response vector $\mathbf{R}_i = (R_{i,1}, \ldots, R_{i,J})$ contains the observed responses to the J items of i-th subject. The j-th component of $\mathbf{R}_i$ will be 1 if this subject gives a positive response to the j-th item and will be 0 otherwise. For instance, in a test with J items, if a student answers the j-th item correctly, then $R_{i,j}$, the j-th component of $\mathbf{R}_i$, will be 1. If the student fails to give a right answer then $R_{i,j} = 0$. The latent variable z_i is introduced to categorize different observations and explain the dependence among items.

The generative process for a response vector $\mathbf{R}_i$ of an observation is as follows: first the class of this observation z_i is drawn from a discrete distribution specified by the probability vector $\mathbf{p} = (p_1, p_2, \ldots, p_L)$, where $p_k \geq 0$ and $\sum_{k=1}^{L} p_k = 1$. So we have

$$P(z_i = \ell) = p_\ell, \ \ell \in [L],$$

where p_ℓ is the proportion of subjects belonging to ℓ-th class in the population. Then given the latent class $z_i = \ell$, the responses to J items are drawn conditionally independently from a Bernoulli distribution with parameter $\theta_{j,\ell}$ for each item j. That is

$$P(R_{i,j} = 1 | z_i = \ell) = \theta_{j,\ell}.$$

So $\theta_{j,\ell}$ measures the ability of subjects from ℓ-th class to give a positive response on item j and is also known as item parameters. Like many other latent variables, local independence is assumed here, implying the dependence of item responses is fully explained by the latent classes. We collect all the item parameters for the L classes in the matrix $\boldsymbol{\theta} = (\theta_{j,\ell}) \in [0,1]^{J \times L}$ whose rows are indexed by the J items and columns indexed by the L classes. All the response vectors are collected in a $N \times J$ matrix $\mathbf{R}$, and the corresponding log-likelihood function under the random-effect LCM is

$$\ell(\mathbf{R};\, \mathbf{p},\, \boldsymbol{\theta}) = \log \left\{ \prod_{i=1}^{N} \left[\sum_{\ell=1}^{L} p_\ell \prod_{j=1}^{J} \theta_{j,\ell}^{R_{i,j}} (1 - \theta_{j,\ell})^{1 - R_{i,j}} \right] \right\},$$

with $(\mathbf{p},\, \boldsymbol{\theta})$ the parameters to be estimated.

2.2 Latent Variables as Fixed Effects

Another way to model latent classes is to view latent class assignment as fixed unknown parameters. For a fixed-effect LCM, denote the i-th subject's latent class membership by a vector of binary

entries $\mathbf{Z}_{i,\cdot} = (Z_{i,1}, \ldots, Z_{i,L})$, with $Z_{i,\ell} = 1$ if subject i belongs to the latent class ℓ. We also introduce another notation for the latent class membership $z_i \in \{1, 2, \ldots, L\}$ and $z_i = \ell$ corresponds to $Z_{i,l} = 1$. Given a sample of size N, collect all the $\mathbf{Z}_{1,\cdot}, \ldots, \mathbf{Z}_{N,\cdot}$ in a $N \times L$ matrix $\mathbf{Z}$, then each row of $\mathbf{Z}$ contains only one entry of "1" and the remaining entries are zeros. We will use the two equivalent notations $\mathbf{Z}$ and $\mathbf{z} = (z_1, \ldots, z_N)$ interchangeably. The components of response vector $\mathbf{R}_i$ are independent Bernoulli variables with parameters specified by $\boldsymbol{\theta}$. So we have $P(R_{i,j} = 1) = \theta_{j,z_i}$. The log-likelihood for $(\mathbf{Z}, \boldsymbol{\theta})$ takes the following form

$$
\begin{aligned}
\ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta}) = & \log \left\{ \prod_{i=1}^{N} \prod_{l=1}^{L} \left[\prod_{j=1}^{J} (\theta_{j,\ell})^{R_{i,j}} (1 - \theta_{j,\ell})^{1 - R_{i,j}} \right]^{Z_{i,\ell}} \right\} \\
= & \sum_i \sum_j \left\{ R_{i,j} \Big[\sum_{\ell=1}^{L} Z_{i,\ell} \log(\theta_{j,\ell}) \Big] + (1 - R_{i,j}) \Big[\sum_{\ell=1}^{L} Z_{i,\ell} \log(1 - \theta_{j,\ell}) \Big] \right\} \\
= & \sum_i \sum_j \Big\{ R_{i,j} \log(\theta_{j,z_i}) + (1 - R_{i,j}) \log(1 - \theta_{j,z_i}) \Big\}.
\end{aligned}
\tag{1}
$$

The parameters to estimate are $(\mathbf{Z}, \boldsymbol{\theta})$. The above display is also called the complete data likelihood in the literature. In the next section we will discuss how to apply tensor method to efficiently estimate the parameters in these two types of latent class models.

3 Estimation Procedures

The EM algorithm is a popular method to maximize likelihood and estimate parameters in LCM by iterating between E-step and M-step. In E-step the probability of each subject belonging to each class is updated by current estimates of item parameters, and in M-step item parameters are updated given the probabilities of each subject's latent class membership. However, the likelihood function under LCM is nonconcave due to the mixture model formulation. Hence, EM algorithm may suffer from convergence to local optima and slow convergence rate under poor initializations. Good initial values are critical to the success of the EM algorithm. In this section we introduce tensor method in Anandkumar et al. (2014) to find good initializations and hence improve the performance of EM algorithm. We first introduce some basics about tensor in Section 3.1 and show the tensor structure in random-effect LCM in Section 3.2. In Section 3.3, we introduce the tensor

power method, which is central to recovering the parameters $(\boldsymbol{\theta}, \boldsymbol{p})$ from the tensor structure. The tensor-EM method, which uses the tensor estimates of $(\boldsymbol{\theta}, \boldsymbol{p})$ as initializations for EM algorithm, is given in Section 3.4. In Section 3.5 we discuss how to select the number of latent classes L.

3.1 Preliminaries about Tensor

We will follow the discussions of Anandkumar et al. (2014) and be succinct and self-contained. First we introduce some notations borrowed from Anandkumar et al. (2014). A real p-th order tensor $\mathbf{T} \in \otimes_{i=1}^{p} \mathbb{R}^{n_i}$ is a p-way array of real numbers where $[\mathbf{T}]_{i_1,\ldots,i_p}$ is the $(i_1, \ldots, i_p)$-th entry in the array. We will mostly consider the case where $n_i = n$ for all $i \in [p]$. Vectors and matrices are special cases of tensors where $p = 1$ and $p = 2$, respectively. Another view of tensor is that it is a multilinear map from a set of matrices $\{\mathbf{V}_i \in \mathbb{R}^{n \times m_i} : i \in [p]\}$ to a p-th order tensor $\mathbf{T}(\mathbf{V}_1, \ldots, \mathbf{V}_p) \in \mathbb{R}^{m_1 \times \cdots \times m_p}$, where $m_1, \ldots, m_p$ are positive integers, defined as

$$[\mathbf{T}(\mathbf{V}_1, \ldots, \mathbf{V}_p)]_{i_1,\ldots,i_p} := \sum_{j_1,j_2,\ldots,j_p} [\mathbf{T}]_{j_1,j_2,\ldots,j_p} [\mathbf{V}_1]_{j_1,i_1} \ldots [\mathbf{V}_p]_{j_p,i_p}. \tag{2}$$

In this paper we will mainly consider three-way tensor and third-order case of this multilinear map. For a third-order tensor $\mathbf{T} \in \otimes^3 \mathbb{R}^d$ and a vector $\boldsymbol{u} \in \mathbb{R}^d$, we will make use of the following vector-valued map in the iteration of tensor power methods

$$\mathbf{T}(\mathbf{I}, \boldsymbol{u}, \boldsymbol{u}) = \sum_{i=1}^{d} \sum_{1 \le j,l \le d} [\mathbf{T}]_{i,j,l} (\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{u})(\boldsymbol{e}_l^{\mathrm{T}} \boldsymbol{u}) \boldsymbol{e}_i, \tag{3}$$

where $\mathbf{I}$ is the d-dimensional identity matrix and $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ are the canonical basis vectors of $\mathbb{R}^d$; that is, each $\boldsymbol{e}_k$ is a d-dimensional vector with only the k-th entry being one and the other entries being zero. To obtain (3) from (2), we note $\mathbf{T}(\mathbf{I}, \boldsymbol{u}, \boldsymbol{u})$ is a d-dimensional vector and

$$[\mathbf{T}(\mathbf{I}, \boldsymbol{u}, \boldsymbol{u})]_k = \sum_{i=1}^{d} \sum_{1 \le j,l \le d} [\mathbf{T}]_{i,j,l} \mathbf{I}_{i,k} u_j u_l = \sum_{i=1}^{d} \sum_{1 \le j,l \le d} [\mathbf{T}]_{i,j,l} (\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{u})(\boldsymbol{e}_l^{\mathrm{T}} \boldsymbol{u}) \boldsymbol{e}_{i,k}.$$

We will also use the following map in the iteration

$$\mathbf{T}(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u}) = \sum_{i,j,k} [\mathbf{T}]_{i,j,k} (\boldsymbol{e}_i^{\mathrm{T}} \boldsymbol{u})(\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{u})(\boldsymbol{e}_k^{\mathrm{T}} \boldsymbol{u}). \tag{4}$$

These maps are all special cases of (2).

Most tensors we consider in this paper are symmetric tensors, which means that an element of a tensor is invariant to permutations of its coordinates. If $\mathbf{T} \in \otimes^p \mathbb{R}^d$ is a symmetric tensor, then we have $[\mathbf{T}]_{i_1,\ldots,i_p} = [\mathbf{T}]_{i_{\pi(1)},\ldots,i_{\pi(p)}}$ for all permutations π on $[p]$. This concept is a generalization of symmetric matrices.

A simple case of a tensor is called rank-one tensor. A rank-one tensor $\mathbf{T} \in \otimes^p \mathbb{R}^d$ can be expressed as tensor product of p vectors: $\mathbf{T} = \boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \cdots \otimes \boldsymbol{v}_p$ for some vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p \in \mathbb{R}^d$, where $[\mathbf{T}]_{i_1,\ldots,i_p} = \prod_{k=1}^p [\boldsymbol{v}_k]_{i_k}$ and $[\boldsymbol{v}_k]_{i_k}$ is the i_kth component of $\boldsymbol{v}_k$. When $\boldsymbol{v}_k = \boldsymbol{v}$ for all k, we can get a symmetric tensor. More detailed discussion and introductions about tensor can be found in Kolda and Bader (2009).

3.2 Tensor Structure in Random-Effect LCM

Anandkumar et al. (2014) showed that for some latent variable models, their low-order moments can be expressed as a sum of rank-one tensors. Once this structure of cross moments is obtained for a particular model, one can apply the orthogonal tensor decomposition to learn the parameters of the model. For random-effect LCM, we show that there is also useful tensor structure in low-order moments by examining Theorem 3.6 in Anandkumar et al. (2014), which studies the multi-view models. Although LCM can be viewed as a special case of multi-view models there, we believe it is still inspiring to introduce tensor method to estimate the parameters in LCM from many perspectives. First, the tensor method is based on lower-order moments of responses (2nd and 3rd orders) and has consistency guarantees. It is also computationally efficient based on our simulations. Moreover, in psychometrics, researchers usually use likelihood to study the identification and estimation of parameters. And we will show that with appropriate manipulations on second and third order cross-moments, we can also uniquely recover the parameters in a random-effect LCM. Hence, tensor method provides a new insight into identifying and estimating parameters in latent variable models widely used in psychometrics. On the other hand, we would also like to clarify that the EM algorithm employed in our estimation method is not considered in Anandkumar et al. (2014). To our best knowledge, the proposed tensor-EM method of combining tensor decomposition and the EM algorithm for latent class analysis is new, and moreover, the established consistency

theory of our final estimators in Section 4 is not previously studied.

Recall that we use $\mathbf{R}_i$ to generally denote subject i's response vector of length J. We consider response vector on a population level and divide the items into three disjoint parts (so we assume $J \geq 3$) $\mathbf{R}_i^1, \mathbf{R}_i^2, \mathbf{R}_i^3$ with each $\mathbf{R}_i^t \in \mathbb{R}^{J_t}$ and $J_1 + J_2 + J_3 = J$. The goal is to relate the cross-moments of these three parts with the parameters we want to estimate. The item paramters of $\mathbf{R}_i^t$ are denoted by $\boldsymbol{\theta}_t \in \mathbb{R}^{J_t \times L}$, which is a sub-matrix of $\boldsymbol{\theta}$, with rows corresponding to rows in $\mathbf{R}_i^t$. We need the following assumption to derive the tensor structure.

Condition 1. *Each $\boldsymbol{\theta}_t$ has full column rank L for t = 1,2,3.*

Note that the partition of items can be arbitrary as long as the item parameters for each part satisfy Condition 1. So we can try different partitions to estimate the parameters and take average to obtain the final estimates.

We denote the i-th column of $\boldsymbol{\theta}_t$ to be $\boldsymbol{\theta}_{t,i}$. The following theorem restates Theorem 3.6 in Anandkumar et al. (2014) in our setting and characterizes the tensor structure in random-effect LCM.

Theorem 1. *Assume that Condition 1 holds and $p_1, \ldots, p_L > 0$. Define*

$$\begin{aligned}
\widetilde{\mathbf{R}}_i^2 &:= \mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^3]\mathbb{E}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]^+ \mathbf{R}_i^2 \\
\widetilde{\mathbf{R}}_i^3 &:= \mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^2]\mathbb{E}[\mathbf{R}_i^3 \otimes \mathbf{R}_i^2]^+ \mathbf{R}_i^3 \\
\mathbf{M}_2 &:= \mathbb{E}[\mathbf{R}_i^1 \otimes \widetilde{\mathbf{R}}_i^2] \\
\mathbf{M}_3 &:= \mathbb{E}[\mathbf{R}_i^1 \otimes \widetilde{\mathbf{R}}_i^2 \otimes \widetilde{\mathbf{R}}_i^3]
\end{aligned} \tag{5}$$

where $\mathbf{A}^+$ denotes the Moore-Penrose pseudoinverse of matrix $\mathbf{A}$. Then we have

$$\begin{aligned}
\mathbf{M}_2 &= \sum_{k=1}^{L} p_k \, \boldsymbol{\theta}_{1,k} \otimes \boldsymbol{\theta}_{1,k}, \\
\mathbf{M}_3 &= \sum_{k=1}^{L} p_k \, \boldsymbol{\theta}_{1,k} \otimes \boldsymbol{\theta}_{1,k} \otimes \boldsymbol{\theta}_{1,k}.
\end{aligned} \tag{6}$$

Proof. First we compute the cross moment. For $t \neq t'$, $\mathbf{R}_i^t$ and $\mathbf{R}_i^{t'}$ are conditionally independent, we have

$$\mathbb{E}[\mathbf{R}_i^t \otimes \mathbf{R}_i^{t'}] = \sum_{k=1}^{L} p_k \mathbb{E}[\mathbf{R}_k^t \otimes \mathbf{R}_k^{t'} | z_i = k] = \sum_{k=1}^{L} p_k \mathbb{E}[\mathbf{R}_i^t | z_i = k] \otimes \mathbb{E}[\mathbf{R}_i^{t'} | z_i = k] = \sum_{k=1}^{L} p_k \boldsymbol{\theta}_{t,k} \otimes \boldsymbol{\theta}_{t',k}.$$

If we denote $\mathbf{D} = \mathrm{diag}\{p_1, \ldots, p_L\}$, then we have $\mathbb{E}[\mathbf{R}_i^t \otimes \mathbf{R}_i^{t'}] = \boldsymbol{\theta}_t \mathbf{D} \boldsymbol{\theta}_{t'}^{\mathrm{T}}$. In the following calculations we need to use the Moore–Penrose inverse of $\mathbb{E}[\mathbf{R}_i^t \otimes \mathbf{R}_i^{t'}]$ and we first compute it. The following fact is useful: To compute the Moore-Penrose inverse of AB, if A has linearly independent columns and B has linearly independent rows, then $(AB)^+ = B^+ A^+$. Now by condition 1, $\boldsymbol{\theta}_t \mathbf{D}$ has linearly independent columns and $\boldsymbol{\theta}_{t'}^{\mathrm{T}}$ has linearly independent rows. So we can write $(\boldsymbol{\theta}_t \mathbf{D} \boldsymbol{\theta}_{t'}^{\mathrm{T}})^+ = (\boldsymbol{\theta}_{t'}^{\mathrm{T}})^+ (\boldsymbol{\theta}_t \mathbf{D})^+$. Apply the fact again on $(\boldsymbol{\theta}_t \mathbf{D})^+$ we have $(\boldsymbol{\theta}_t \mathbf{D} \boldsymbol{\theta}_{t'}^{\mathrm{T}})^+ = (\boldsymbol{\theta}_{t'}^{\mathrm{T}})^+ \mathbf{D}^{-1} \boldsymbol{\theta}_t^+$.

Then we calculate the conditional mean

$$\mathbb{E}[\widetilde{\mathbf{R}}_i^2 | z_i = k] = \mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^3] \mathbb{E}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]^+ \mathbb{E}[\mathbf{R}_i^2 | z_i = k].$$

According to the model setting $\mathbb{E}[\mathbf{R}_i^2 | z_i = k] = \boldsymbol{\theta}_2 \boldsymbol{e}_k$, then we have

$$\mathbb{E}[\widetilde{\mathbf{R}}_i^2 | z_i = k] = \boldsymbol{\theta}_1 \mathbf{D} \boldsymbol{\theta}_3^{\mathrm{T}} (\boldsymbol{\theta}_2 \mathbf{D} \boldsymbol{\theta}_3^{\mathrm{T}})^+ \boldsymbol{\theta}_2 \boldsymbol{e}_k = \boldsymbol{\theta}_1 \mathbf{D} (\boldsymbol{\theta}_3^+ \boldsymbol{\theta}_3)^{\mathrm{T}} \mathbf{D}^{-1} \boldsymbol{\theta}_2^+ \boldsymbol{\theta}_2 \boldsymbol{e}_k.$$

By condition 1, $\boldsymbol{\theta}_t^+ \boldsymbol{\theta}_t = I_L$ for all t, thus $\mathbb{E}[\widetilde{\mathbf{R}}_i^2 | z_i = k] = \boldsymbol{\theta}_{1,k}$. Similarly, $\mathbb{E}[\widetilde{\mathbf{R}}_i^3 | z_i = k] = \boldsymbol{\theta}_{1,k}$. So we have

$$\mathbf{M}_2 = \sum_{k=1}^{L} p_k \mathbb{E}[\mathbf{R}_i^1 \otimes \widetilde{\mathbf{R}}_i^2 | z_i = k] = \sum_{k=1}^{L} p_k \mathbb{E}[\mathbf{R}_i^1 | z_i = k] \otimes \mathbb{E}[\widetilde{\mathbf{R}}_i^2 | z_i = k] = \sum_{k=1}^{L} p_k \, \boldsymbol{\theta}_{1,k} \otimes \boldsymbol{\theta}_{1,k}.$$

Similarly one can get the decomposition for $\mathbf{M}_3$ in (6). □

In applications we only have finite samples and the moments in Theorem 1 should be approximated by empirical moments. In particular, once we have samples $\mathbf{R}_1, \ldots, \mathbf{R}_N \in \mathbb{R}^J$, we partition each sample and obtain $\mathbf{R}_i^t \in \mathbb{R}^{J_t}$ corresponding to the partition on population level. Then the transformed response and estimated moments can be computed by

$$\begin{aligned}
&\widehat{\mathbb{E}}[\mathbf{R}_i^t \otimes \mathbf{R}_i^{t'}] := \tfrac{1}{N} \textstyle\sum_{j=1}^{N} \mathbf{R}_j^t \otimes \mathbf{R}_j^{t'} \\
&\widetilde{\mathbf{R}}_{i,e}^2 := \widehat{\mathbb{E}}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^3] \left(\widehat{\mathbb{E}}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3] \right)^+ \mathbf{R}_i^2 \\
&\widetilde{\mathbf{R}}_{i,e}^3 := \widehat{\mathbb{E}}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^2] \left(\widehat{\mathbb{E}}[\mathbf{R}_i^3 \otimes \mathbf{R}_i^2] \right)^+ \mathbf{R}_i^3 \\
&\widehat{\mathbf{M}}_2 := \tfrac{1}{N} \textstyle\sum_{j=1}^{N} \mathbf{R}_j^1 \otimes \widetilde{\mathbf{R}}_{j,e}^2, \\
&\widehat{\mathbf{M}}_3 := \tfrac{1}{N} \textstyle\sum_{j=1}^{N} \mathbf{R}_j^1 \otimes \widetilde{\mathbf{R}}_{j,e}^2 \otimes \widetilde{\mathbf{R}}_{j,e}^3.
\end{aligned} \tag{7}$$

Due to the randomness of sample, it is possible that $r =: \mathrm{rank}(\widehat{\mathbb{E}}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]) > L$ and $\widehat{\mathbb{E}}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]$ has $(r - L)$ extra non-zero singular values. These singular values will be small since they equal to

0 in $\widehat{\mathbb{E}}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]$'s population counterpart $\mathbb{E}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]$. In this case we should discard these extra singular values and only use first L sigular values when calculating $\widehat{\mathbb{E}}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]^+$, otherwise one has to compute the inverse of these small singular values, which will incur large error.

After learning $\boldsymbol{\theta}_1$ from data by the tensor power method to be introduced in Section 3.3, we can obtain $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$ by setting $\boldsymbol{\theta}_2 = \mathbb{E}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]\mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^3]^+\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3 = \mathbb{E}[\mathbf{R}_i^3 \otimes \mathbf{R}_i^2]\mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^2]^+\boldsymbol{\theta}_1$. This can be derived in a same way as Theorem 1. So the main problem is to estimate $\boldsymbol{\theta}_1$ from moments $\mathbf{M}_2$ and $\mathbf{M}_3$.

Although this structure only holds for random-effect LCMs, in fixed-effect LCMs we can view $z_1, \ldots, z_N$ as random with some prior distribution. For instance, they are sampled from some discrete distributions on $[L]$ independently from $\mathbf{R}$. Then the data generation process of random-effect and fixed-effect LCMs are the same and the estimation procedures for random-effect LCMs also apply to fixed-effect LCMs.

3.3 Tensor Method to Learn the Parameters

In this section we briefly describe the procedures in Anandkumar et al. (2014) to recover the parameters in (6). That is, given

$$\begin{aligned} \mathbf{M}_2 &= \sum_{i=1}^{L} w_i \, \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i, \\ \mathbf{M}_3 &= \sum_{i=1}^{L} w_i \, \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \otimes \boldsymbol{\mu}_i \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^d$, we want to obtain the elements of decomposition $(w_i, \boldsymbol{\mu}_i)$'s from $\mathbf{M}_2$ and $\mathbf{M}_3$. Condition 1 now becomes $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L\}$ are linearly independent. First we introduce the orthogonal decomposition of a tensor. Then we see how we can use tensor power method to recover the orthogonal decomposition of a tensor and estimate the parameters.

3.3.1 Orthogonal Decomposition

Since the moments structures in (8) are about at most a third-order tensor, we only consider the case $p = 3$ (third-order tensor).

A symmetric tensor $\mathbf{T} \in \otimes^3 \mathbb{R}^d$ has an orthogonal decomposition if there exists a collection of orthonormal unit vectors $\{\boldsymbol{v}_1, \ldots \boldsymbol{v}_L\}$ and positive scalers $\lambda_i > 0$ such that

$$\mathbf{T} = \sum_{i=1}^{L} \lambda_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i \otimes \boldsymbol{v}_i. \tag{9}$$

Without loss of generality we assume $\lambda_i > 0$ because for third-order tensor we have $-\lambda_i \boldsymbol{v}_i^{\otimes 3} = \lambda_i(-\boldsymbol{v}_i)^{\otimes 3}$. However we do not assume λ_i's are ordered. In fact, according to Theorem 2 in Section 3.3.3, the eigenvector that tensor power method converges to depends on the magnitude of elements in $\{|\lambda_i \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{u}_0|, 1 \le i \le L\}$ instead of the magnitude of λ_i's. Here $\boldsymbol{u}_0$ is the initial point for tensor power method. This definition is a generalization of spectral decomposition for a symmetric matrix. We can also generalize the concept of eigenvalue and eigenvectors.

Recall the definition of the multilinear map induced by a tensor in (2). A unit vector $\boldsymbol{u} \in \mathbb{R}^d$ is an eigenvector of $\mathbf{T}$ with corresponding eigenvalue $\lambda \in \mathbb{R}$ if $\mathbf{T}(\mathbf{I}, \boldsymbol{u}, \boldsymbol{u}) = \lambda \boldsymbol{u}$. For an orthogonally decomposable tensor $\mathbf{T} = \sum_{i=1}^{L} \lambda_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i \otimes \boldsymbol{v}_i$, one can check operation (3) is

$$\mathbf{T}(\mathbf{I}, \boldsymbol{u}, \boldsymbol{u}) = \sum_{i=1}^{L} \lambda_i (\boldsymbol{u}^{\mathrm{T}} \boldsymbol{v}_i)^2 \boldsymbol{v}_i$$

and operation (4) reduces to

$$\mathbf{T}(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u}) = \sum_{i=1}^{L} \lambda_i (\boldsymbol{u}^{\mathrm{T}} \boldsymbol{v}_i)^3.$$

By the orthogonality of $\boldsymbol{v}_i$'s, $\mathbf{T}(\mathbf{I}, \boldsymbol{v}_i, \boldsymbol{v}_i) = \lambda_i \boldsymbol{v}_i$ and $\mathbf{T}(\boldsymbol{v}_i, \boldsymbol{v}_i, \boldsymbol{v}_i) = \lambda_i$ for all $i \in [L]$. Thus $(\lambda_i, \boldsymbol{v}_i)$ is an eigenvector/eigenvalue pair of $\mathbf{T}$.

The eigenvalues and eigenvectors of a tensor are more complicated than those of a matrix, and there are some subtle points. For example, unlike matrices, orthogonal decompositions do not necessarily exist for each symmetric tensor. Moreover, if a tensor $\mathbf{T}$ admits an orthogonal decomposition in (9), then this is the unique orthogonal decomposition of $\mathbf{T}$. This is very different from the spectral decomposition of a matrix. In fact, $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L\}$ are the set of robust eigenvectors for $\mathbf{T}$. See Anandkumar et al. (2014) for more discussion.

3.3.2 Whitening Process

Comparing the third-order moment structure $\mathbf{M}_3$ in (8) with the orthogonal decomposition form (9), we find they have almost the same form except that the vectors $\boldsymbol{\mu}_i$'s in (8) may not necessarily

be orthogonal to each other. So we need to whiten the tensor $\mathbf{M}_3$ to $\widetilde{\mathbf{M}}_3$, which has an orthogonal decomposition. In the whitening process we will make use of $\mathbf{M}_2$ in (8).

Let $\mathbf{W} \in \mathbb{R}^{d\times L}$ satisfy $\mathbf{M}_2(\mathbf{W},\mathbf{W}) = \mathbf{W}^{\mathrm{T}}\mathbf{M}_2\mathbf{W} = \mathbf{I}_L$. We can take $\mathbf{W} = \mathbf{U}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagnoal matrix containing all positive eigenvalues of $\mathbf{M}_2$ and $\mathbf{U} \in \mathbb{R}^{d\times L}$ is the matrix of corresponding orthogonal eigenvectors of $\mathbf{M}_2$. $\mathbf{D}^{-1/2}$ is well-defined since we assume $\boldsymbol{\mu}_i$'s are linearly independent and thus $\mathbf{M}_2$ is of rank L.

Suppose $\mathbf{M}_2$ and $\mathbf{M}_3$ admit the decomposition as in (8), define $\widetilde{\boldsymbol{\mu}}_i := \sqrt{\omega_i}\,\mathbf{W}^{\mathrm{T}}\boldsymbol{\mu}_i$ and observe that

$$\mathbf{I}_L = \mathbf{M}_2(\mathbf{W},\mathbf{W}) = \sum_{i=1}^{L} \mathbf{W}^{\mathrm{T}}(\sqrt{\omega_i}\boldsymbol{\mu}_i)(\sqrt{\omega_i}\boldsymbol{\mu}_i)^{\mathrm{T}}\mathbf{W} = \sum_{i=1}^{L} \widetilde{\boldsymbol{\mu}}_i\widetilde{\boldsymbol{\mu}}_i^{\mathrm{T}}.$$

So $\widetilde{\boldsymbol{\mu}}_i{}'s$ are orthonormal vectors.

Define

$$\widetilde{\mathbf{M}}_3 := \mathbf{M}_3(\mathbf{W},\mathbf{W},\mathbf{W}) = \sum_{i=1}^{L} \omega_i(\mathbf{W}^{\mathrm{T}}\boldsymbol{\mu}_i)^{\otimes 3} = \sum_{i=1}^{L} \frac{1}{\sqrt{\omega_i}}\widetilde{\boldsymbol{\mu}}_i^{\otimes 3}.$$

Since $\widetilde{\boldsymbol{\mu}}_i{}'s$ are orthonormal vectors, this is the orthogonal decomposition of $\widetilde{\mathbf{M}}_3$. We can use tensor power method described in Section 3.3.3 to obtain the eigenvalue/eigenvector pairs $(\lambda_i, \boldsymbol{v}_i) = (1/\sqrt{\omega_i}, \widetilde{\boldsymbol{\mu}}_i)$. Then we can recover the parameters ω_i's and $\boldsymbol{\mu}_i$'s as $(\omega_i, \boldsymbol{\mu}_i) = (\frac{1}{\lambda_i^2}, \lambda_i(\mathbf{W}^{\mathrm{T}})^{+}\widetilde{\boldsymbol{\mu}}_i)$, where $(\mathbf{W}^{\mathrm{T}})^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{W}^{\mathrm{T}}$.

3.3.3 Tensor Power Method

Now we show how to recover the parameters $(\lambda_i, \boldsymbol{v}_i)$'s in (9) from a tensor $\mathbf{T}$. In analogy to matrix power method, here we use the tensor power method of De et al. (2000) to obtain the eigenvalue/eigenvector pairs $(\lambda_i, \boldsymbol{v}_i)$ in (9). First suppose a third-order tensor has an exact orthogonal decomposition. We have the following result on the algorithmic convergence of tensor power method (Lemma 5.1 in Anandkumar et al. (2014)).

Theorem 2. *Let $\mathbf{T} \in \otimes^3\mathbb{R}^d$ have an orthogonal decomposition as given in (9). For a vector $\boldsymbol{u}_0 \in \mathbb{R}^d$, suppose that the set of numbers $\{|\lambda_i\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{u}_0|, 1 \le i \le L\}$ has a unique largest value. Without loss of generality, say $|\lambda_1\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{u}_0|$ is this largest value and $|\lambda_2\boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{u}_0|$ is the second largest value. For*

$t = 1, 2 \ldots,$ *let*

$$\boldsymbol{u}_t := \frac{\mathbf{T}(\mathbf{I}, \boldsymbol{u}_{t-1}, \boldsymbol{u}_{t-1})}{||\mathbf{T}(\mathbf{I}, \boldsymbol{u}_{t-1}, \boldsymbol{u}_{t-1})||}.$$

Then

$$||\boldsymbol{v}_1 - \boldsymbol{u}_t||^2 \leq (2\lambda_1^2 \sum_{i=2}^{K} \lambda_i^{-2}) \left| \frac{\lambda_2 \boldsymbol{v}_2^{\mathrm{T}} \boldsymbol{u}_0}{\lambda_1 \boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{u}_0} \right|^{2^{t+1}},$$

where $\|\cdot\|$ *is the* ℓ_2 *norm.*

The result shows that the repeated iteration starting from $\boldsymbol{u}_0$ converges to $\boldsymbol{v}_1$ at a quadratic rate. The reason why the tensor power method enjoys a quadratic convergence rate in Theorem 2 while the usual matrix power method has a (relatively) slower linear convergence rate is that the iteration step in the tensor case is quadratic while that step is linear in the matrix case. Specifically, the (unnormalized) iteration in the tensor power method is

$$\bar{\boldsymbol{u}}_{t+1} = \mathbf{T}\left(\mathbf{I}, \bar{\boldsymbol{u}}_t, \bar{\boldsymbol{u}}_t\right) = \sum_{i=1}^{k} \lambda_i \left(\boldsymbol{v}_i^\top \bar{\boldsymbol{u}}_t\right)^2 \boldsymbol{v}_i,$$

and the (unnormalized) iteration in the matrix power method is

$$\bar{\boldsymbol{u}}_{t+1} = \mathbf{T}\bar{\boldsymbol{u}}_t = \sum_{i=1}^{k} \lambda_i \left(\boldsymbol{v}_i^\top \bar{\boldsymbol{u}}_t\right) \boldsymbol{v}_i,$$

where $(\lambda_i, \boldsymbol{v}_i)$'s are the eigenvalues and eigenvectors of the tensor/matrix $\mathbf{T}$. In the tensor case, $\bar{\boldsymbol{u}}_{t+1}$ depends on $\boldsymbol{v}_i^\top \bar{\boldsymbol{u}}_t$ via $\left(\boldsymbol{v}_i^\top \bar{\boldsymbol{u}}_t\right)^2$, and by induction, one can show $\bar{\boldsymbol{u}}_t = \sum_{i=1}^k \lambda_i^{2^t-1} c_i^{2^t} \boldsymbol{v}_i$ for $c_i = \boldsymbol{v}_i^\top \bar{\boldsymbol{u}}_0$. Then the quadratic rate $O\left(\rho^{2^t}\right)$ for $\rho = \left|\frac{\lambda_2 c_2}{\lambda_1 c_1}\right|$ arises from the normalization of $\bar{\boldsymbol{u}}_t$. In the matrix case, $\bar{\boldsymbol{u}}_t = \sum_{i=1}^k \lambda_i^t c_i \boldsymbol{v}_i$, which is different from the form in the tensor case, and after normalization, the convergence rate is $O(\rho^t)$ for $\rho = \left|\frac{\lambda_2}{\lambda_1}\right|$. Therefore, the tensor power method enjoys a faster convergence rate than matrix power method. To obtain all the eigenvalue/eigenvector pairs, we use "deflation" after getting an eigenvalue/eigenvector pair $(\lambda_i, \boldsymbol{v}_i)$. That is, to obtain the j-th eigenvalue/eigenvector pair, we subtract the previous $j-1$ rank-one structures from $\mathbf{T}$ and then execute the power method on $\mathbf{T} - \sum_{i=1}^{j-1} \lambda_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i \otimes \boldsymbol{v}_i$.

However, in practice we plug-in the empirical estimate of $\mathbf{M}_2$ and $\mathbf{M}_3$ in (6) and the estimate of whitened tensor $\widetilde{\mathbf{M}}_3$ may not have an exact orthogonal decomposition. So we should consider the case where we only have an approximation $\widehat{\mathbf{T}}$ of $\mathbf{T}$ and need a more robust algorithm to use an orthogonal decomposition to approximate $\widehat{\mathbf{T}}$. Following Anandkumar et al. (2014), we present

Algorithm 1 as a more robust method. Multiple starting points are used in Algorithm 1 to ensure approximate convergence at first stage. Intuitively, by restarting from different points we can start from a point from which the initial n iterations from it dominates the error $\widehat{\mathbf{T}} - \mathbf{T}$. To be concrete, define operator norm for a tensor $\mathbf{T}$ as follows

$$\|\mathbf{T}\|_{\text{op}} := \sup_{\|\boldsymbol{u}\|=1} |\mathbf{T}(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u})|.$$

Perturbation analysis (Theorem 5.1 in Anandkumar et al. (2014)) shows that under some conditions if $\|\widehat{\mathbf{T}} - \mathbf{T}\|_{\text{op}}$ is small (in our setting, this requires the cross moments are estimated accurately), then the estimated eigenvalue/eigenvector pairs returned by Algorithm 1 are close to the true eigenvalue/eigenvector pairs. Note that in contrast to Davis-Kahan's theorem (which holds for all symmetric matrix), Theorem 5.1 in Anandkumar et al. (2014) is an algorithm-dependent perturbation analysis and only applies to Algorithm 1 since in general $\widehat{\mathbf{T}}$ may not even have an orthogonal decomposition. Furthermore, Theorem 5.1 in Anandkumar et al. (2014) shows we can obtain good estimates of eigenvalue/eigenvector pairs with high probability with $K = \text{poly}(L)$ trials. When L is large the required number of initial trials K is close to linear in L. Hence, the robust tensor power method (Algorithm 1) can recover the eigenvalue/eigenvector pairs of $\mathbf{T}$ efficiently from an estimator $\widehat{\mathbf{T}}$.

In conclusion, the procedures to learn $(\omega_i, \boldsymbol{\mu}_i)$ from $\mathbf{M}_2$ and $\mathbf{M}_3$ in (8) are: (1) Use the information of $\mathbf{M}_2$ to whiten $\mathbf{M}_3$ and get $\widetilde{\mathbf{M}}_3$; (2) Apply tensor power method to $\widetilde{\mathbf{M}}_3$ and learn the orthogonal decomposition of it; (3) Obtain the original parameters with an inverse transformation of whitening. We then elaborate on how to use the tensor method to estimate paprameters in LCM.

3.4 Tensor-EM Method

In Section 3.3 we introduce how to recover parameters $(w_i, \boldsymbol{\mu}_i)$'s from $\mathbf{M}_2$ and $\mathbf{M}_3$ in (8). Since the tensor structure in random-effect LCM in Theorem 1 is in the exact form of (8), we can apply the methods in Section 3.3 to the empirical estimates of $\mathbf{M}_2$ and $\mathbf{M}_3$ shown in (7) and obtain the tensor estimator for $\boldsymbol{\theta}_1$ and $\boldsymbol{p}$. The relations $\boldsymbol{\theta}_2 = \mathbb{E}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]\mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^3]^+\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3 = \mathbb{E}[\mathbf{R}_i^3 \otimes \mathbf{R}_i^2]\mathbb{E}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^2]^+\boldsymbol{\theta}_1$ then give us the tensor estimates of $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$. We denote the tensor estimates as $\widehat{\boldsymbol{\theta}}_T$ and $\widehat{\boldsymbol{p}}_T$. When the sample size is large enough, this method alone can yield

Algorithm 1 Robust tensor power method

Input: symmetric tensor $\widetilde{\mathbf{T}} \in \mathbb{R}^{d\times d\times d}$, number of iterations K, n.

Output: estimates of one of eigenvalue/eigenvector pairs; the deflated tensor

1: **for** $\tau = 1$ to K **do**

2: Draw $\boldsymbol{u}_0^{(\tau)}$ uniformly from unit sphere in $\mathbb{R}^d$.

3: **for** $t = 1$ to n **do**

4: Compute power iteration and re-normalization

$$\boldsymbol{u}_t^{(\tau)} = \frac{\widetilde{\mathbf{T}}(\mathbf{I}, \boldsymbol{u}_{t-1}^{(\tau)}, \boldsymbol{u}_{t-1}^{(\tau)})}{||\widetilde{\mathbf{T}}(\mathbf{I}, \boldsymbol{u}_{t-1}^{(\tau)}, \boldsymbol{u}_{t-1}^{(\tau)})||}$$

5: **end for**

6: **end for**

7: Let $\tau^* = \text{argmax}_{\tau\in[K]}\{\widetilde{\mathbf{T}}(\boldsymbol{u}_{t-1}^{(\tau)}, \boldsymbol{u}_{t-1}^{(\tau)}, \boldsymbol{u}_{t-1}^{(\tau)})\}$

8: Do n power iteration updates further starting from $\boldsymbol{u}_n^{(\tau^*)}$ to obtain $\widehat{\boldsymbol{u}}$, and set $\widehat{\lambda} = \widetilde{\mathbf{T}}(\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{u}}, \widehat{\boldsymbol{u}})$.

9: **return** the estimated eigenvalue/eigenvector pair $(\widehat{\lambda}, \widehat{\boldsymbol{u}})$; the deflated tensor $\mathbf{T} - \widehat{\lambda}\widehat{\boldsymbol{u}} \otimes \widehat{\boldsymbol{u}} \otimes \widehat{\boldsymbol{u}}$.

an estimator close to the true parameters. However if we do not have so many samples, estimates based on tensor are not so accurate since in tensor method we only take advantage of low-order moments and ignore other information of sampling distributions. So after obtaining the tensor estimates, we use them as initial values for EM algorithms to improve the accuracy. We call this two-step estimation procedure the *tensor-EM method*. We further derive the EM algorithm for random-effect LCM here. Consider the complete log-likelihood

$$\ell_{comp}(\boldsymbol{\theta}, \boldsymbol{p}|\mathbf{R}, \mathbf{Z}) = \sum_{i=1}^{N}\sum_{l=1}^{L} Z_{i,l} \log p_l + \sum_{i=1}^{N}\sum_{l=1}^{L}\sum_{j=1}^{J} Z_{i,l}[R_{i,j}\theta_{j,l} + (1 - R_{i,j})(1 - \theta_{j,l})],$$

where we use the same notation $Z_{i,l}$ as in the fixed-effect LCM to denote the indicator $I(\text{subject } i \text{ is in class } l)$. Given $(\boldsymbol{\theta}^{(t)}, \boldsymbol{p}^{(t)})$, in the E-step we compute $\varphi_{i,l}^{(t+1)} = \mathbb{E}[Z_{i,l} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{p}^{(t)}, \mathbf{R}]$ by the posterior probability,

$$\varphi_{i,l}^{(t+1)} = \frac{p_l^{(t)} \prod_j \theta_{j,l}^{(t)R_{i,j}} (1 - \theta_{j,l}^{(t)})^{1-R_{i,j}}}{\sum_l p_l^{(t)} \prod_j \theta_{j,l}^{(t)R_{i,j}} (1 - \theta_{j,l}^{(t)})^{1-R_{i,j}}} \tag{10}$$

In the M-step, we replace $Z_{i,l}$ with $\varphi_{i,l}^{(t+1)}$ and obtain

$$Q(\boldsymbol{\theta}, \boldsymbol{p}|\boldsymbol{\theta}^{(t)}, \boldsymbol{p}^{(t)}) = \sum_{i=1}^{N}\sum_{l=1}^{L}\varphi_{i,l}^{(t+1)}\log p_l + \sum_{i=1}^{N}\sum_{l=1}^{L}\sum_{j=1}^{J}\varphi_{i,l}^{(t+1)}[R_{i,j}\theta_{j,l} + (1-R_{i,j})(1-\theta_{j,l})].$$

After maximizing over $\boldsymbol{\theta}, \boldsymbol{p}$ in $Q(\boldsymbol{\theta}, \boldsymbol{p}|\boldsymbol{\theta}^{(t)}, \boldsymbol{p}^{(t)})$ we arrive at the updated parameters

$$\theta_{j,l}^{(t+1)} = \frac{\sum_i \varphi_{i,l}^{(t+1)} R_{i,j}}{\sum_i \varphi_{i,l}^{(t+1)}}, \quad p_l^{(t+1)} = \frac{\sum_{i=1} \varphi_{i,l}^{(t+1)}}{\sum_l \sum_{i=1} \varphi_{i,l}^{t+1}}.$$

We keep iterating until some convergence criterion is met (e.g. the log-likelihood improves little after one iteration). When no prior knowledge is available, people may try many random initial values $(\boldsymbol{\theta}^0, \boldsymbol{p}^0)$ and take the one that has the maximum log-likelihood after the algorithm converges. In tensor-EM method we set the initial values $(\boldsymbol{\theta}^0, \boldsymbol{p}^0)$ to be $(\widehat{\boldsymbol{\theta}}_T, \widehat{\boldsymbol{p}}_T)$.

For fixed-effect LCM, we first learn p_i's and $\boldsymbol{\theta}$ from (6) as a initialization step and apply Classification-EM (CEM) algorithm proposed in Celeux and Govaert (1992) to obtain the final estimator. The main difference between CEM and EM algorithm summarized above is that in CEM algorithm we want to estimate latent class membership $z_i^{(t+1)}$, hence we need to find the index that maximizes posterior probability $\{\varphi_{i,l}^{(t+1)}, l \in [L]\}$ for each i. Formally, given $(\boldsymbol{\theta}^{(t)}, \boldsymbol{p}^{(t)})$, in the E-step we compute $\varphi_{i,l}^{(t+1)}$ in (10). Then we let the estimated latent class membership for subject i at step $t+1$ be

$$z_i^{(t+1)} = \underset{l \in \{1,\ldots,L\}}{\arg\max}\, \varphi_{i,l}^{(t+1)}$$

and correspondingly set $\mathbf{Z}^{(t+1)}$. In the M-step the parameters $(\boldsymbol{\theta}, \boldsymbol{p})$ are updated using $\mathbf{Z}^{(t+1)}$ instead of $\varphi_{i,l}^{(t+1)}$ as follows

$$\theta_{j,l}^{(t+1)} = \frac{\sum_i Z_{i,l}^{(t+1)} R_{i,j}}{\sum_i Z_{i,l}^{(t+1)}}, \quad p_l^{(t+1)} = \frac{\sum_{i=1} Z_{i,l}^{(t+1)}}{N}.$$

When the algorithm converges, we output $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{Z}})$ (recall the parameters in fixed-effect LCM are $\boldsymbol{\theta}$ and $\mathbf{Z}$). Since the CEM algorithm only requires $(\boldsymbol{\theta}, \boldsymbol{p})$ to obtain the next-step updates, we can set the initial values to be $(\widehat{\boldsymbol{\theta}}_T, \widehat{\boldsymbol{p}}_T)$. When there are some components of $\boldsymbol{\theta}$ outside the range $[0, 1]$, we can set the negative components to be a small positive number (e.g. 0.001) and those over one to be a number close to 1 (e.g. 0.999).

Empirically, we find that the accuracy of tensor-EM method is comparable to estimates obtained with EM algorithm starting from true parameters, indicating that the tensor-EM method can give the MLE of latent class model when the model is correctly specified. Moreover, it is more computationally efficient than EM algorithm with several random initial values, especially in large-scale data. See our simulation study for more details.

3.5 Selecting the Number of Classes

In the discussion above we assume the number of classes L is known. Next we discuss the selection of L. There exists a rich literature in selecting number of classes. Nylund et al. (2007) performed a Monte Carlo simulation study on several commonly used methods and found BIC proposed in Schwarz (1978) and likelihood ratio test based on bootstrap in McLachlan and Peel (2004) have a better performance. They recommend BIC and likelihood ratio test based on bootstrap to select the number of classes. Since we focus on large-scale datasets containing many items and samples, we follow the discussion of Chen et al. (2017) and apply generalized information criterion proposed in Nishii (1984) to selecting the number of classes.

Specifically, for a candidate set $\mathcal{L}$ and any $L \in \mathcal{L}$, we apply the tensor-EM algorithm to learning the parameters and compute the generalized information criterion for random-effect and fixed-effect LCM as follows:

$$\mathrm{GIC}_R(L) = -2\,\ell(\mathbf{R};\, \widehat{\mathbf{p}}^L,\, \widehat{\boldsymbol{\theta}}^L) + a_N \dim(\boldsymbol{p}, \boldsymbol{\theta}),$$

$$\mathrm{GIC}_F(L) = -2\,\ell(\mathbf{R};\, \widehat{\mathbf{Z}}^L,\, \widehat{\boldsymbol{\theta}}^L) + a_N \dim(\mathbf{Z}, \boldsymbol{\theta})$$

where $\dim(\boldsymbol{p}, \boldsymbol{\theta})$ is the dimension of parameters to estimate and measures the model complexity. We have $\dim(\boldsymbol{p}, \boldsymbol{\theta}) = JL + L - 1$ in random-effect LCM and $\dim(\mathbf{Z}, \boldsymbol{\theta}) = JL + N$ in fixed-effect LCM. Sample size dependent quantity a_N measures the level of penalty on model complexity. Here we consider two choice of a_N.

- GIC_1: $a_N = \log(N)$. This case corresponds to BIC and it enjoys some consistent results shown in Nishii (1984) when the model has low complexity (i.e. the dimension of parameters is fixed).

- GIC_2: $a_N = \log[\log(N)]\log N$. This choice is considered in Fan and Tang (2013) in generalized linear model to address the case where the dimension of parameter space d increases at a polynomial order of N, that is, $d = O(N^c)$ for some $c > 0$. The large-scale latent class model we consider tends to have many items. The dimension of parameters $\dim(\boldsymbol{p}, \boldsymbol{\theta}) = J \times L + L - 1$ can be large and should not be treated as fixed. For instance, in the simulation study a random-effect LCM we consider has ten classes and one hundred items. This model has dimension $d = 1009$ while the sample size is $N = 1000$. So it is more appropriate to adopt this choice in this setting. See Fan and Tang (2013) for discussion about theoretical results of this choice of a_N.

After calculating the GIC for different models, we select the number of classes to minimize $\text{GIC}(L)$:

$$L^* = \arg\min_{L \in \mathcal{L}} \text{GIC}(L).$$

Although our simulation results show that the tensor-EM method gives the MLE of the model when the number of classes L is correctly selected, it may converge to some local optima when L is incorrect. It is also likely that the tensor method yields inaccurate results when L is misspecified as $\widetilde{L}$. For instance, some item parameters may be negative or over one, which happens when the sample size is small. In this case, we propose to revise the tensor estimates as follows: we set the negative components to be a small positive number (e.g. 0.001) and those over one to be a number close to 1 (e.g. 0.999). Such a procedure will help us select the number of classes because we know as long as the true number of classes is specified, the tensor-EM method can yield an ideal estimate (MLE) with a small GIC value. On the other hand, a poor estimate based on a misspecified model will give a larger GIC value, indicating misfit of the model. Hence, GIC values computed by tensor-EM method can provide useful information to select the number of classes. According to our simulation study, the proposed method can select the right model most of the time.

After proposing the computational methods to select number of classes and to find the MLE, we next examine the theoretical properties of MLE in the large-N and large-J scenario. For random-effect LCM with fixed number of items J, the MLE is known to be consistent. However, the joint MLE for fixed-effect LCM may not be consistent (Neyman and Scott, 1948) when J is

fixed. Intuitively, one cannot hope to recover each subject's latent class membership accurately with only a finite number of items observed for each subject. So in the next section, we will consider the consistency of joint MLE when J also goes to infinity in fixed-effect LCMs.

4 Clustering Consistency of the Joint MLE

In this section we consider large-scale fixed-effect LCMs and characterize the behavior of latent class assignment estimator $\widehat{\mathbf{Z}}$ under suitable conditions, where $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{Z}})$ is the joint maximum likelihood estimator (MLE). We use a similar proof technique as in Gu and Xu (2021) to establish the clustering consistency of the joint MLE for fixed-effect LCMs.

First we need to define some notations. Denote the true parameters by $(\boldsymbol{\theta}^0, \mathbf{Z}^0)$. Define

$$P_{i,j} = \mathbb{P}(R_{i,j} = 1) = \theta^0_{j,z^0_i},$$
$$M = \frac{1}{NJ} \sum_{i=1}^{N} \sum_{j=1}^{J} \mathbb{P}(R_{i,j} = 1).$$

The $M \in [0, 1]$ above measures the average positive response rate over all subjects and items. Denote the expectation of log-likelihood $\ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta})$ in (1) by

$$\bar{\ell}(\mathbf{Z}, \boldsymbol{\theta}) = \mathbb{E}[\ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta})] = \sum_{i=1}^{N} \sum_{j=1}^{N} \Big\{ P_{i,j} \log(\theta_{j,z_i}) + (1 - P_{i,j}) \log(1 - \theta_{j,z_i}) \Big\},$$

where the expectation is taken with respect to the distribution of $\mathbf{R}$.

Given arbitrary $\mathbf{Z}$, denote

$$\ell(\mathbf{R}; \mathbf{Z}) = \sup_{\boldsymbol{\theta}} \ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta}) = \ell(\mathbf{R}; \mathbf{Z}, \widehat{\boldsymbol{\theta}}^{(\mathbf{Z})}),$$
$$\bar{\ell}(\mathbf{Z}) = \sup_{\boldsymbol{\theta}} \bar{\ell}(\mathbf{Z}, \boldsymbol{\theta}) = \bar{\ell}(\mathbf{Z}, \bar{\boldsymbol{\theta}}^{(\mathbf{Z})}),$$

where $\widehat{\boldsymbol{\theta}}^{(\mathbf{Z})} = \arg\max_{\boldsymbol{\theta}} \ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta})$ and $\bar{\boldsymbol{\theta}}^{(\mathbf{Z})} = \arg\max_{\boldsymbol{\theta}} \bar{\ell}(\mathbf{Z}, \boldsymbol{\theta})$. Then under any realization of $\mathbf{Z}$, the following holds for any latent class $a \in [L]$,

$$\widehat{\theta}^{(z)}_{j,a} = \frac{\sum_i Z_{i,a} R_{i,j}}{\sum_i Z_{i,a}}, \quad \bar{\theta}^{(z)}_{j,a} = \frac{\sum_i Z_{i,a} P_{i,j}}{\sum_i Z_{i,a}}. \tag{11}$$

We consider the joint maximum likelihood estimator $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{Z}})$ subject to fitting a L-class fixed-effect LCM with true parameters $(\boldsymbol{\theta}^0, \mathbf{Z}^0)$,

$$(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{Z}}) = \arg\max_{(\boldsymbol{\theta}, \mathbf{Z})} \ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta}).$$

Note that $\widehat{\mathbf{Z}} = \arg\max_{\mathbf{Z}} \ell(\mathbf{R}; \mathbf{Z}, \widehat{\boldsymbol{\theta}}^{(\mathbf{Z})}) = \arg\max_{\mathbf{Z}} \ell(\mathbf{R}; \mathbf{Z})$, where $\widehat{\boldsymbol{\theta}}^{(\mathbf{Z})}$ maximizes the profile likelihood $\ell(\mathbf{R}; \mathbf{Z}, \boldsymbol{\theta})$ given a particular realization $\mathbf{Z}$. One can apply the procedures in Section 3 to compute the joint MLE efficiently.

We impose the following assumptions on the true parameters.

Assumption 1. *There exists a constant $\gamma > 0$ such that*

$$\frac{1}{J^\gamma} \leq \min_{\substack{1\leq j\leq J,\\ 1\leq a\leq L}} \theta^0_{j,a} \leq \max_{\substack{1\leq j\leq J,\\ 1\leq a\leq L}} \theta^0_{j,a} \leq 1 - \frac{1}{J^\gamma}. \tag{12}$$

Assumption 2. *There exists a positive sequence $\{\beta_J\}$ such that*

$$\frac{1}{J} \min_{1\leq a\neq b\leq L} \|\boldsymbol{\theta}^0_{\cdot,a} - \boldsymbol{\theta}^0_{\cdot,b}\|^2 \geq \beta_J, \tag{13}$$

where $\|\cdot\|$ denotes the ℓ_2 norm.

Assumption 1 guarantees that the components of $\boldsymbol{\theta}$ are bounded away from 0 and 1 but allowed to become very close to 0 or 1 as J becomes larger. It is a quite mild technical assumption. Assumption 2 is an identification condition for latent classes and guarantees that the item parameters of different classes are different enough. Note that we allow different classes to have same probability to answer a single item correctly (i.e. $\theta_{j,a} = \theta_{j,b}$ for some $a \neq b$). But their average performance on the J items should be different.

In fitting the latent class model, we are interested in controlling the number of incorrect latent class assignments. Formally, after obtaining some estimator $\widehat{\mathbf{z}}$, let $\widehat{C}_1, \ldots, \widehat{C}_L$ be clusters from our estimator $\widehat{\mathbf{z}}$ such that subjects sharing same estimated membership are in one cluster. For instance, suppose there are eight subjects whose latent class memberships are $\mathbf{z}^0 = (2, 2, 2, 2, 1, 2, 1, 1)$. Here "1" and "2" represents the true class index for each subject. The estimates are $\widehat{\mathbf{z}} = (1, 1, 1, 1, 1, 2, 2, 2)$ so we have $\widehat{C}_1 = \{\text{subject } 1, 2, 3, 4, 5\}$ and $\widehat{C}_2 = \{\text{subject } 6, 7, 8\}$. Let $m_l = \arg\max_{a\in[L]} \sum_{i\in\widehat{C}_l} Z^0_{i,a}$ be the majority of true class membership among subjects in $\widehat{C}_l$. In our example, $\widehat{C}_1$ has four subjects whose true class indices under $\mathbf{z}^0$ are "2" and hence $m_1 = 2$ and similarly $m_2 = 1$. Since the latent classes are identified up to permutations of class index, m_l should be viewed as the class index of $\widehat{C}_l$ corresponding to true class assignments $\mathbf{z}^0$. In our example, although the estimates indicate that subjects in $\widehat{C}_1$ are in the latent class "1", we should obtain the latent class index of $\widehat{C}_1$ under $\mathbf{z}^0$, which

is $m_1 = 2$. The number of correct latent class assignments under $\widehat{\mathbf{z}}$ in $\widehat{C}_l$ is then $\left|\{i \in \widehat{C}_l : z_i^0 = m_l\}\right|$. In the above example we have $m_1 = 2, \left|\{i \in \widehat{C}_1 : z_i^0 = 2\}\right| = 4$, indicating that in $\widehat{C}_1$ four subjects are correctly assigned to their true class membership. Similarly $m_2 = 1, \left|\{i \in \widehat{C}_1 : z_i^0 = 2\}\right| = 2$. The total number of correctly assigned subjects is then $\sum_l \left|\{i \in \widehat{C}_l : z_i^0 = m_l\}\right|$. In our example this number is 6. The number of incorrect class assignments under estimated latent class assignments $N_e(\widehat{\mathbf{z}})$ is defined as:

$$N_e(\widehat{\mathbf{z}}) = N - \sum_{l=1}^{L} \left|\{i \in \widehat{C}_l : z_i^0 = m_l\}\right| \tag{14}$$

So every subject $i \in \{1, \ldots, N\}$ whose true class under $\mathbf{z}^0$ is not in the majority within its estimated class under $\widehat{\mathbf{z}}$ is counted. In the example above $N_e(\widehat{\mathbf{z}}) = 2$.

We have the following main theorem on the clustering consistency of joint MLE for fixed-effect LCM, which characterizes the asymptotic behavior of error rate $N(\widehat{\mathbf{z}})/N$.

Theorem 3. *Under assumptions 1 and 2, assume the following when $N, J \to \infty$,*

$$\frac{MJ}{\log L} \to \infty \,, \frac{N}{L} \to \infty, \tag{15}$$

$$\sqrt{\frac{M}{J}} \left(\frac{N}{L}\right)^{1-\xi} \to \infty \text{ for some small } \xi > 0,$$

for the joint maximum likelihood estimator $\widehat{\mathbf{z}}$, we have

$$\frac{N_e(\widehat{\mathbf{z}})}{N} = o_P\left(\frac{(\log J)^{1+\eta} \cdot \sqrt{M \log L}}{\sqrt{J}\beta_J}\right) \tag{16}$$

for any $\eta > 0$.

Assigning each subject to latent class resembles the process of clustering and hence we name our results as "clustering consistency". In Theorem 3, we allow $L \to \infty$ or $L = O(1)$ and as long as the scaling conditions hold, our results hold. In particular, if J remains bounded as $N, J \to \infty$, we have the following corollary.

Corollary 1. *Under assumptions 1 and 2, assume $N, J \to \infty$ and $L = O(1)$,*

$$MJ \to \infty \,, \tag{17}$$

$$\sqrt{\frac{M}{J}} N^{1-\xi} \to \infty \text{ for some small } \xi > 0,$$

for the joint maximum likelihood estimator $\widehat{\mathbf{z}}$*, we have*

$$\frac{N_e(\widehat{\mathbf{z}})}{N} = o_P\left(\frac{(\log J)^{1+\eta}\cdot\sqrt{M}}{\sqrt{J}\beta_J}\right) \tag{18}$$

for any $\eta > 0$.

In particular if $M = \frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J}\mathbb{P}(R_{i,j}=1)$ is of the constant order (denoted by $M=\Theta(1)$), the only scaling condition would become $J = o(N^{2(1-\xi)})$ for some small $\xi > 0$ and this is very mild. The rate depends on β_J specified in (13). If the item parameters between different classes differ by a constant then we have $\beta_J = \Theta(1)$ and the final error rate $N_e(\widehat{\mathbf{z}})/N = o_P\left((\log J)^{1+\eta}/\sqrt{J}\right)$ decays towards zero as N, J increases.

The following corollary shows the item parameters can be consistently estimated via joint MLE under some conditions as $N, J \to \infty$.

Corollary 2. *Under assumptions 1 and 2 and the scaling conditions in Theorem 3, if we further assume clustering consistency holds:*

$$\frac{N_e(\widehat{\mathbf{z}})}{N} \xrightarrow{P} 0, \ \text{as } N, J \to \infty$$

and there exists some positive constant τ *such that* $n_l^0/N \geq \tau$ *for all* $l \in [L]$ *where* $n_l^0 = \sum_i Z_{i,l}^0$ *is the number of samples in latent class* l*. Then as* $N, J \to \infty$*, with probability approaching 1, for any* $l \in [L]$ *there exists a unique* $a \in [L]$ *such that* $m_a = l$*, i.e. the* a*-th cluster represents* l*-th class. Furthermore we have for any* $l \in [L]$

$$\max_j |\widehat{\theta}_{j,a} - \theta_{j,l}^0| \xrightarrow{P} 0.$$

The proof of Corollary 2 can be found in the appendix. The parameter estimation consistency relies on clustering consistency established in Theorem 3. The condition $n_l^0/N \geq \tau$ for all l guarantees that there are enough samples to estimate the item parameters for each class. According to the theory presented above, both latent class membership and item parameters can be consistently estimated under mild conditions, which provides theoretical guarantees for real-world applications of large-scale latent class analysis.

It is interesting to mention some pioneer work in high dimensional item factor analysis model (Chen et al., 2019a,b). The fixed-effect LCM can be viewed as a special case of multidimensional IRT

model in Chen et al. (2019a) where the person parameters correspond to latent class membership in this work and factor loadings correspond to item parameters. Then one may use constrained joint MLE approach in Chen et al. (2019a) to obtain estimates for person parameters and factor loadings with consistency guarantees. The main difference in our work and Chen et al. (2019a) is that to obtain the joint MLE, we maximize $\mathbf{Z}$ over matrices with one-hot rows (exactly one component in each row of $\mathbf{Z}$ can be 1) while the person parameters are optimized over any real number (satisfying certain constraints on the norm) in Chen et al. (2019a). Since the joint MLEs are obtained in a different way, it is unclear how the joint MLE in LCM and constrained joint MLE in Chen et al. (2019a) are correlated. The continuous estimates for person parameters cannot translate to discrete latent class membership directly, making it hard to examine the relations between two estimators. Hence different techniques to establish the consistency of joint MLE are applied in our work. We will leave the connections between IRT models and LCM models for future explorations. In applications, both IRT models and latent class models can be fit to have different interpretations on the data.

We further discuss the connection and difference between our results and those in Gu and Xu (2021). Gu and Xu (2021) considered large-scale structured latent attribute models and established consistency of the joint MLE in their models. They also treated the latent part in their model (latent attribute profile) as fixed and derived the consistency of estimating the latent attribute profiles. Our work differs from Gu and Xu (2021) in the following respects: First, the assumption 2 in Gu and Xu (2021) requires each component of item parameters to be quite different for respondents with different latent attribute profiles. However the assumption 2 in our work only requires the L_2 distance between item parameters for different latent classes to be quite different. The model structures considered in Gu and Xu (2021) are more delicate and may require stronger assumptions on the item parameters. The other main difference lies in the technical proof. After proving a bound on $\bar{\ell}\left(\mathbf{Z}^0\right) - \bar{\ell}(\widehat{\mathbf{Z}})$, we obtain the clustering consistency by a refined partition argument while Gu and Xu (2021) considered the structures implied by Q-matrix and identification assumptions to prove the consistency of estimating the Q-matrix vectors and latent attribute profiles. See the proof in the appendix for details.

5 Simulation Study

In this section, we perform simulation studies to assess the performance of the tensor-EM method. Specifically, in Section 5.1, we examine the estimation accuracy and speed of tensor-EM method under LCMs. In Section 5.2, we consider the setting where local independence is violated and evaluate the robustness of the tensor-EM method. The clustering consistency is verified empirically together with comparisons to several other clustering methods in Section 5.3. We also empirically evaluate the performance of GIC in selecting the number of classes in Section 5.4.

5.1 Performance of tensor-EM method under local independence

We consider 24 different settings: $\{N = 1000, 10000, 20000\} \otimes \{J = 100, 200\} \otimes \{L = 5, 10\} \otimes$ {item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$ or $\{0.2, 0.4, 0.6, 0.8\}$}. By item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$ we mean we generate the true $\boldsymbol{\theta}$'s elements $\theta_{j,a}$ independently and uniformly from $\{0.1, 0.2, 0.8, 0.9\}$. Note that under the considered large-scale LCM with many items, the generic identifiability conditions stated in Corollary 5 in Allman et al. (2009) is guaranteed.

We compare the performance of the proposed tensor-EM method in Section 3.4 with three other methods:

(1) EM-true, which is the EM algorithm starting from the true parameters as initial values;

(2) EM-random, where we randomly generate the initial values for the EM algorithm. In random-effect LCM, we keep trying different initial values until we find the EM algorithm converges in 1000 iterations on five initial values and then we select the estimators corresponding to maximum log-likelihood. In fixed-effect LCM, we generate five initial values and run CEM algorithm on them until it converges or the number of iterations exceeds 1000, then we select the estimators corresponding to maximum log-likelihood. The first mechanism can hopefully find better solutions but the second one can save more time. We use EM-random algorithm with these two mechanisms and compare the results with tensor-EM to show the good performance of tensor-EM;

(3) the tensor method alone. In this third competitor, we permute the items randomly and

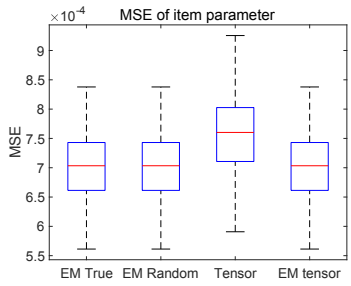
obtain a tensor estimate for each permutation. Let π be a permutation of $[J]$ and $\mathbf{R}_i^\pi$ be subject response vector corresponding to permutation π (i.e. $[\mathbf{R}_i^\pi]_k = R_{i,\pi(k)}$). We obtain tensor estimates $\widehat{\boldsymbol{\theta}}^\pi$ from cross moments of $\mathbf{R}_i^\pi$ and set the tensor estimates of original item parameters as $\widehat{\theta}_{j,a} = \widehat{\theta}^\pi_{\pi^{-1}(j),a}$. We then repeat this procedure five times and finally take average of them. This repetition can reduce the MSE of item parameters a little but will remain the same magnitude;

(4) EM-tensor. This is the tensor-EM algorithm we detail in Section 3.4. We first apply the tensor method and obtain the tensor estimates. We then use tensor estimates as initializations for EM and CEM algorithm in random-effect LCM and fixed-effect LCM, respectively.

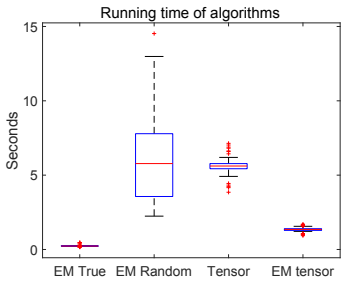
We emphasize that in the proposed tensor-EM method, we do not repeat the tensor power method or take any average. Empirically, just one implementation of the tensor power method gives good initial values for the EM algorithm.

The running time and MSE are reported, where MSE $= \sum_{j=1}^{J}\sum_{l=1}^{L}(\theta_{j,l} - \widehat{\theta}_{j,l})^2/(JL)$. In some settings, the EM-random estimates have too large MSEs, so we also present the plots excluding the EM-random estimates to better visualize the MSEs of the other three methods. The results are based on 100 replications in each simulation setting. For the random-effect LCM, the population proportion vector $\boldsymbol{p}$ and item parameters $\boldsymbol{\theta}$ are first generated and the process of generating samples and estimation is repeated. The proportion vector $\boldsymbol{p}$ is generated randomly to guarantee each class has enough samples (in settings with five classes we have $p_l \geq 0.1$ and in settings with ten classes we have $p_l \geq 0.08$ for all l). For the fixed-effect LCM, the latent class membership $\mathbf{Z}$ and $\boldsymbol{\theta}$ are generated and the process of sampling and estimation is repeated (so unlike in random-effect LCM, $\mathbf{Z}$ is the same for each replication). The convergence criterion for EM(CEM) algorithm is set as when the improvement in likelihood is less than 0.1 (which has a relative tolerance smaller than 10^{-5} in the considered simulation settings).

Due to the space limitation, we present here two representative figures for each type of LCM (i.e., random-effect LCM and fixed-effect LCM) in Figures 1–4, and provide the rest simulation results in Supplementary Material.

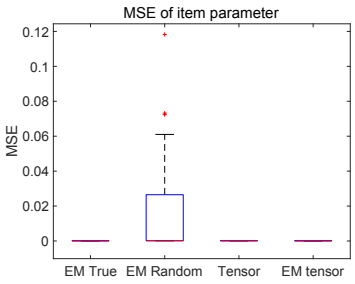


(a) MSE of item parameters

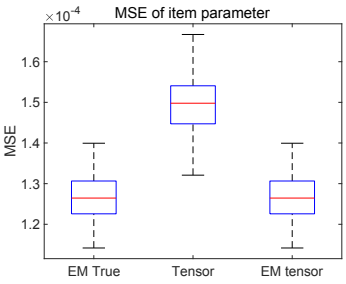


(b) Running time of the algorithms

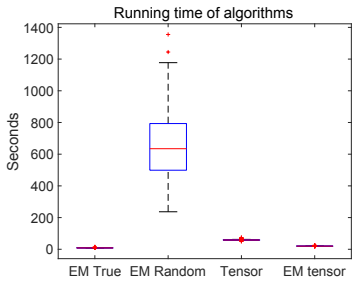
Figure 1: Random-effect LCM, $N = 1000, J = 100, L = 5, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

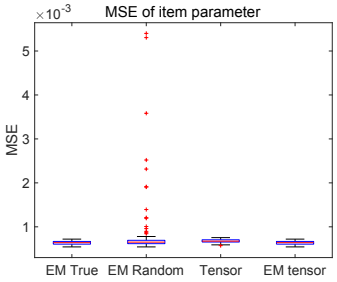


(b) MSE without EM-random

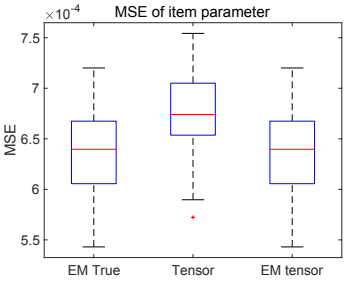


(c) Running time of the algorithms

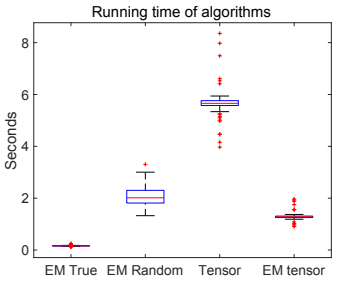
Figure 2: Random-effect LCM, $N = 10000, J = 100, L = 10, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

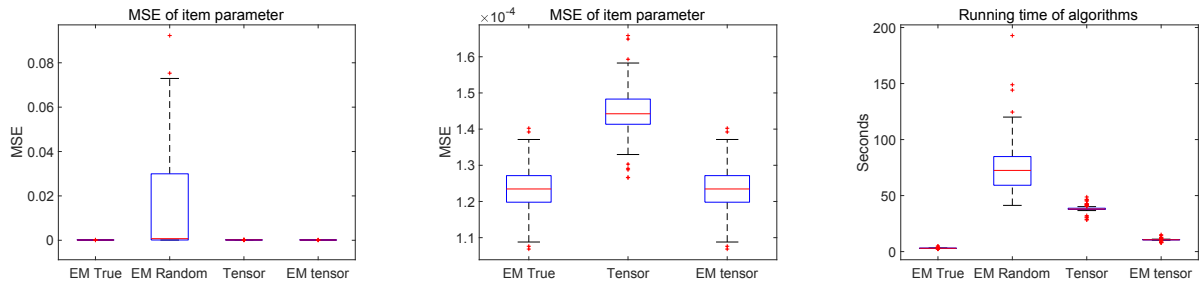


(b) MSE without EM-random



(c) Running time of the algorithms

Figure 3: Fixed-effect LCM, $N = 1000, J = 100, L = 5, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters (b) MSE without EM-random (c) Running time of the algorithms

Figure 4: Fixed-effect LCM, $N = 10000, J = 100, L = 10, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$

From the boxplots in Figures 1–4 and those in Supplementary Material, we can see that for each setting, the MSE of the tensor-EM method is almost the same as that of the EM-true method. The EM-random method sometimes yields local maximizer of log-likelihood function and thus its estimates have a large MSE. The tensor estimates alone have a larger MSE compared with the tensor-EM estimates but are more stable than the EM-random estimates. Comparing the running time of different algorithms, we can find that the tensor-EM method is computationally efficient, only second to the performance of EM-true with true parameters as initials values. On the other hand, the EM-random method can be computationally intensive because it needs more steps to converge.

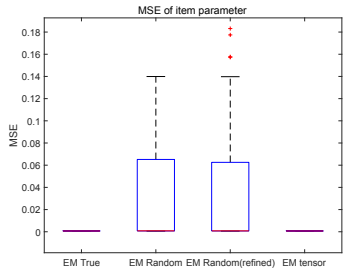
The sample size N, number of classes L and number of items J all affect the accuracy and running time of the methods examined. As the sample size N increases, the accuracy of the methods is improved while they all need more time. As the number of classes L increases, the MSE of EM-tensor estimates also becomes larger because we have more parameters to estimate. As the number of items J increases, the accuracy of EM-tensor remains comparable to EM-true while the accuracy of tensor method alone is improved. The running time increases as J, L becomes large. When the item parameters are generated in $\{0.1, 0.2, 0.8, 0.9\}$, the signal strength is strong and the estimates have smaller MSE compared with cases where item parameters are generated from $\{0.2, 0.4, 0.6, 0.8\}$ where the signal strength is weaker. We also note that the random-effect and fixed-effect LCMs with the same N, J, L and item parameters share similar orders of MSEs.

To further show the advantage of tensor-EM method over EM-random, we perform more

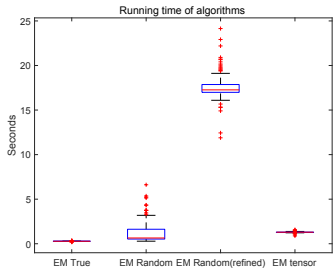
simulations. First, we let them start from the same initial points (under some transformations) and evaluate their estimation accuracy and running time. Recall we need to use second-order moments $\mathbf{M}_2$ to whiten $\mathbf{M}_3$ and get orthogonal decomposable tensor $\widetilde{\mathbf{M}}_3$. To ensure they start from the same initial points, we mimic the whitening process and take the following strategy.

We first divide the item parameters $\boldsymbol{\theta}$ into three parts as described in Section 3. Then we randomly generate initial values $\boldsymbol{\theta}_1^0$ for $\boldsymbol{\theta}_1$ from $U(0,1)$. For EM-random we also need initial values for $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$. From the relations between $\boldsymbol{\theta}_i$'s in Section 3.2, we set $\boldsymbol{\theta}_2^0 = \widehat{\mathbb{E}}[\mathbf{R}_i^2 \otimes \mathbf{R}_i^3]\widehat{\mathbb{E}}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^3]^+\boldsymbol{\theta}_1^0$ and $\boldsymbol{\theta}_3^0 = \widehat{\mathbb{E}}[\mathbf{R}_i^3 \otimes \mathbf{R}_i^2]\widehat{\mathbb{E}}[\mathbf{R}_i^1 \otimes \mathbf{R}_i^2]^+\boldsymbol{\theta}_1^0$ and concatenate them to form $\boldsymbol{\theta}^0$ and feed to EM-random algorithm. For Tensor-EM method, we need to transform $\boldsymbol{\theta}_1^0$. Recall the columns of $\boldsymbol{\theta}_1$ play the role of $\boldsymbol{\mu}_i$'s in (8) and after we define $\widetilde{\boldsymbol{\mu}}_i = \sqrt{w_i}\mathbf{W}^\top\boldsymbol{\mu}_i$, then $\widetilde{\boldsymbol{\mu}}_i$'s are sets of eigenvectors for an orthogonal decomposable tensor $\widetilde{\mathbf{M}}_3$, which we perform tensor power method on. Now let $\widetilde{\boldsymbol{\mu}}_i^0 = \sqrt{p_i}\mathbf{W}^\top\boldsymbol{\theta}_{1,i}^0$ for $i \in [L]$ and we use $\widetilde{\boldsymbol{\mu}}_i^0$ as the initial value in Algorithm 1 to obtain the i-th eigenvalue/eigenvector pair of the estimated $\mathbf{M}_3$ from the samples (i.e. we do not perform K random initializations as shown in Algorithm 1). After we obtain estimates of $\widetilde{\boldsymbol{\mu}}_i$, we use the relation shown in the last paragraph of Section 3.3.2 to get the tensor estimates of $(\boldsymbol{\mu}_i, \boldsymbol{\theta}_{1,i})$ and hence obtain the tensor estimates for $\boldsymbol{\theta}$. We then implement EM algorithms starting from it and evaluate its performance.

We also implement a "smarter" version of EM-random. The idea is to use a large tolerance to run EM with multiple random starting points and then run EM with a refined tolerance with the solution that gives largest likelihood in the first stage. We first run EM with 10 random initial values for 20 iterations and then find the solution that yields the largest likelihood. Then we run EM starting from that solution until convergence. The above three algorithms together with EM-true are run with 100 replications under settings: $\{N = 1000, 10000\} \otimes \{J = 100, 200\} \otimes \{L = 5, 10\} \otimes \{\text{item parameters} \in \{0.1, 0.2, 0.8, 0.9\} \text{ or } \{0.2, 0.4, 0.6, 0.8\}\}$. Some results are shown in Figures 5 and 6, where EM-random and EM-tensor use the same starting values as detailed above and EM-random(refined) uses the "smarter" refined-tolerance version of EM-random. More simulation results can be found in the appendix.

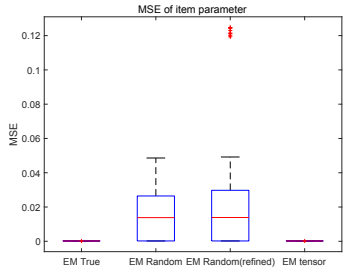


(a) MSE of item parameters

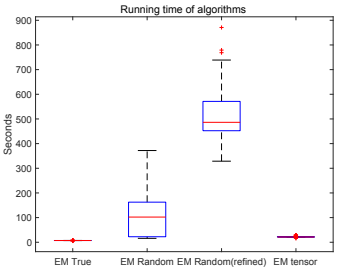


(b) Running time of the algorithms

Figure 5: Random-effect LCM, $N = 1000, J = 100, L = 5, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters



(b) Running time of the algorithms

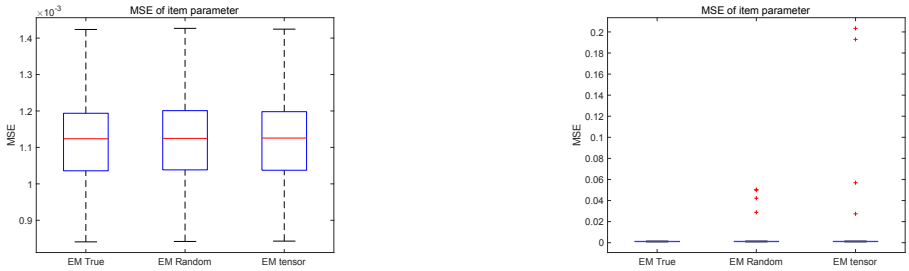
Figure 6: Random-effect LCM, $N = 10000, J = 200, L = 10$, $\theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$

As shown in Figures 5 and 6, the tensor-EM method has similar MSE with EM-true and outperforms the two EM-random algorithms. Comparing EM-random with tensor-EM, we see the tensor-EM has smaller MSE when they use the same initialization, indicating the better performance of tensor-EM. Comparing EM-random using refined tolerance with tensor-EM, the tensor-EM can yield more accurate results efficiently. One possible explanation is that the EM-random method using refined tolerance is a greedy method since it chooses the best solution only based on the first few iterations, which may not be very reliable.

5.2 Performance of tensor-EM method under local dependence

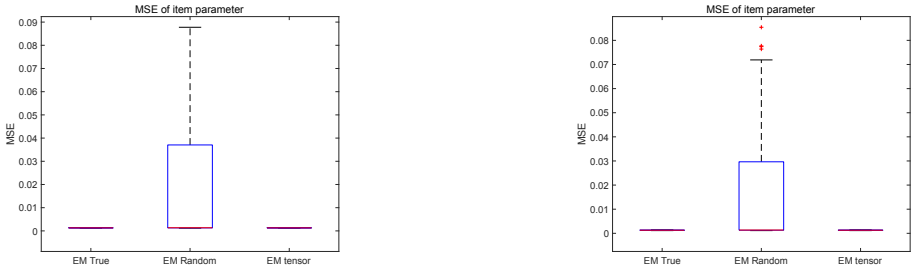
We further investigate the performance of EM algorithms under local dependence. The data generating process is as follows: After we generate z_i from the proportion vector $\boldsymbol{p}$, given $z_i = l$, we first obtain one sample $\mathbf{X}_i$ from J-dimensional multivariate normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\rho)$,

where $\mathbf{\Sigma}_\rho$ is the covariance matrix of an auto-regression model, i.e. $[\mathbf{\Sigma}_\rho]_{i,j} = \rho^{|i-j|}$ for some $0 < \rho < 1$. Then we let $R_{i,j} = I(X_{i,j} < q(\theta_{j,l}))$ where $q(\alpha)$ is the α-quantile of standard normal distribution. This guarantees marginally we have $\mathbb{P}(R_{i,j} = 1|z_i = l) = \theta_{j,l}$ but conditioning on z_i, the components of $\mathbf{R}_i$ are now correlated. The value of ρ controls the extent to which they are correlated. $\rho = 0$ corresponds to the conditional independent case. We run EM-random, tensor-EM and EM-true under the following settings: $\{N = 1000\} \otimes \{\rho = 0.3, 0.7\} \otimes \{J = 100, 200\} \otimes \{L = 5, 10\} \otimes \{\text{item parameters} \in \{0.1, 0.2, 0.8, 0.9\} \text{ or } \{0.2, 0.4, 0.6, 0.8\}\}$. These three algorithms are run in the same way as what we did in Figures 1–4. Some results are presented in Figures 7 and 8 and more results can be found in appendix.



(a) MSE of item parameters $\rho = 0.3$ (b) MSE of item parameters $\rho = 0.7$

Figure 7: Random-effect LCM, $N = 1000, J = 100, L = 5, \theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters $\rho = 0.3$ (b) MSE of item parameters $\rho = 0.7$

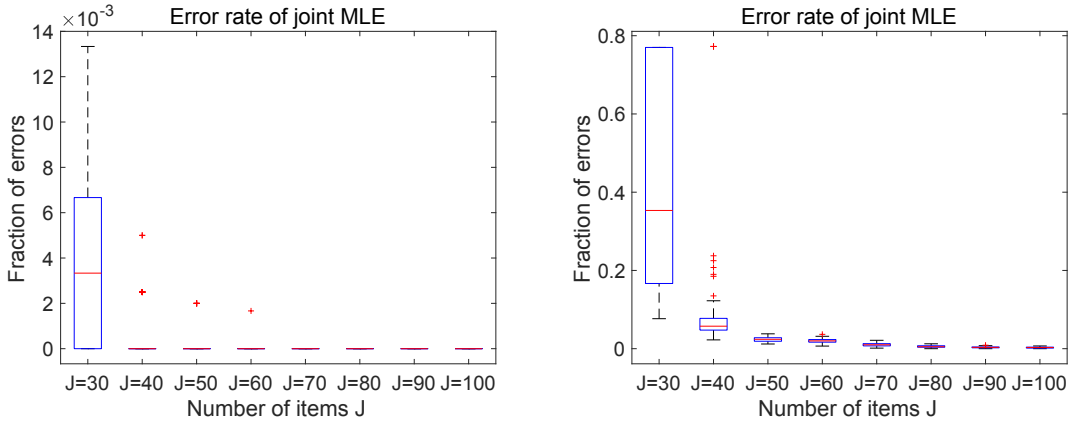
Figure 8: Random-effect LCM, $N = 1000, J = 100, L = 10, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$

We note that in the local dependent setting, the log-likelihood is no longer valid and hence there is no guarantee on the EM-true method, which is also based on the log-likelihood. However, EM-true can still yield accurate estimates. The proposed tensor-EM method has similar performance with

EM-true and is also robust against violations of local independence. In contrast, the EM-random method may not work well under local dependence settings.

5.3 Verification of Clustering Consistency

In this subsection, we empirically verify Theorem 3 in fixed-effect LCMs with diverging N and J. Specifically, we consider fixed-effect LCM with $L = 5$ classes. We let J increase from 30 to 100 by 10 and set $N = 10J$ in all the simulations. The only purpose of setting $N = 10J$ is that we can visualize the error rate $N_e(\widehat{\mathbf{z}})/N$ as a function of J in a plot to see the trend. Item parameters are generated uniformly from either $\{0.1, 0.2, 0.8, 0.9\}$ or $\{0.2, 0.4, 0.6, 0.8\}$ and latent class assignments $\mathbf{z}$ are sampled uniformly over $[L]$. After item parameters and latent class assignments are generated, we generate response $\mathbf{R}$ accordingly. Since we have shown that tensor-EM method has good performance in Section 5.1, we then apply tensor-EM method to obtain the joint MLE $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{Z}})$. The error of the estimated latent class assignments $\widehat{\mathbf{Z}}$ is evaluated and the number of incorrect assignments $N_e(\widehat{\mathbf{z}})$ defined in (14) is computed for each replication. This process of generating $\mathbf{R}$, estimating $\mathbf{Z}$ and evaluating error is replicated 100 times and the boxplots of error rate $N_e(\widehat{\mathbf{z}})/N$ are shown in Figure 9. According to these plots, the error rate of the estimated latent class membership $\mathbf{z}$ decays to zero as N, J increases. Again in the strong signal setting where $\theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$ the error rate converges to zero faster.



(a) Item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$ (b) Item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

Figure 9: Error rate of joint MLE in latent class assignments versus number of items J

We also compare the clustering performance of the proposed tensor-EM method with the performance of several other commonly used clustering algorithms. To be concrete, we consider the following clustering algorithms.

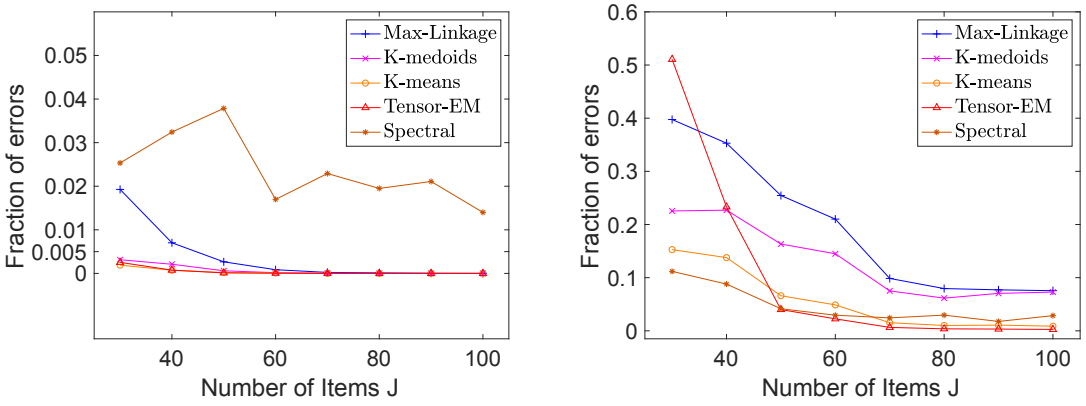
1. Max linkage clustering. In our simulation, max linkage is found to have a better performance compared with single linkage and average linkage, and hence here we only present the results of the max linkage clustering.

2. K-medoids. Considering the binary response in our setting, Hamming distance is applied as a metric. For each replication, the algorithm is repeated ten times with different initial cluster centroid positions and the best is chosen as the final result.

3. K-means. Similarly to K-medoids, Hamming distance is used and for each replication, the algorithm is performed with ten different initial values.

4. Spectral clustering with normalization. Hamming distance is used to compute the similarity between data points.

We use functions from *Statistics and Machine Learning Toolbox* in Matlab to implement the first three methods. Spectral clustering is implemented using the normalized random-walk Laplacian matrix (see Shi and Malik (2000) for details). Under the same settings as in Figure 9, we generate samples from LCM and apply the above algorithms and tensor-EM to cluster the data. The average error rates of all the algorithms over 100 replications are computed for each J. The trend of error rate versus the number of items J is shown in Figure 10. We can see all the algorithms have small error rates as J becomes large and the tensor-EM method has the best performance when J is moderately large. One possible explanation is that the tensor-EM method is more tailored for LCMs and may have advantage over other methods in the LCM setting.



(a) Item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$ (b) Item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

Figure 10: Error rate of different clustering algorithms

5.4 Performance of GIC in Selecting Number of Classes

We consider the accuracy of GIC in selecting L, which needs to be estimated in practice. The settings we consider are the same as in Section 5.1. The performance of GIC to select the number of classes in the random-effect LCM and the fixed-effect LCM are reported in Table 1. For settings with the true number of classes L being five, we let the candidate set of L be $\{2, 3, 4, 5, 6, 7\}$; while for settings with the true L being ten, we let the candidate set of L be $\{7, 8, 9, 10, 11, 12\}$. We see that for all these settings GIC_1 can always select the correct number of classes. And for most of the settings GIC_2 can choose the right model. The only setting where GIC_2 performs not so well is random-effect LCM with $N = 1000, J = 100, L = 10, \theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$. In general, both GIC_1 and GIC_2 enjoy desirable performance in selecting the correct number of classes.

6 Real Data Analysis

In this section we apply the proposed method to real data from Trends in International Mathematics and Science Study (TIMSS). We use a subset of TIMSS 2011 Austrian data (George and Robitzsch, 2015; Sedat and Arican, 2015) in R package **CDM**. 47 items are available to measure students' abilities in 9 mathematical sub-competences, including (DA) Data and Applying, (DK) Data and Knowing, (DR) Data and Reasoning, (GA) Geometry and Applying, (GK) Geometry and Knowing,

Signal strength	N	J	L	Random-effect		Fixed-effect	
				GIC_1	GIC_2	GIC_1	GIC_1
$\theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$	1000	100	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
		200	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
	10000	100	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
		200	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
	20000	100	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
		200	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
$\theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$	1000	100	5	1.00	1.00	1.00	1.00
			10	1.00	0.77	1.00	1.00
		200	5	1.00	1.00	1.00	1.00
			10	1.00	0.99	1.00	1.00
	10000	100	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
		200	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
	20000	100	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00
		200	5	1.00	1.00	1.00	1.00
			10	1.00	1.00	1.00	1.00

Table 1: The fraction of correctly selecting the number of classes

(GR) Geometry and Reasoning, (NA) Numbers and Applying, (NK) Numbers and Knowing, and (NR) Numbers and Reasoning. The first Q-matrix in the R package **CDM** indicates the relations on the items and sub-competences measured, which are summarized in Table 2.

One feature of large scale education assessment data is that we only have response on a subset of items for each student. Here 48% of the components in the response matrix $\mathbf{R}$ are missing. Under the missing at random (MAR) assumption, we use the multiple imputation (MI) to obtain five complete datasets. Same analysis is performed on each of the dataset and the final results (GIC and

Sub-competences	Item index that measures the sub-competences
DA	46,47
DK	20,34
DR	21,35
GA	17,18,30,31,32,42,44
GK	7,8,16,19,28,29,43
GR	33,45
NA	1,6,10,15,23,24,37,38,40
NK	11,14,22,25,26,27,36
NR	2,3,4,5,9,12,13,39,41

Table 2: Relations between sub-competences and items

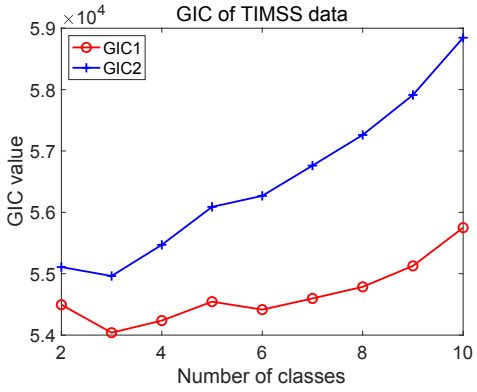
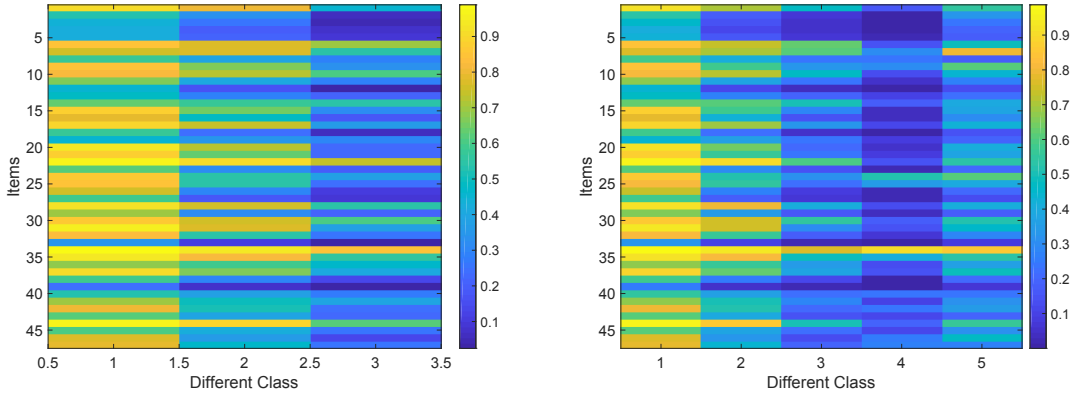


Figure 11: GIC of models with different number of classes

item parameters) presented are the average of results obtained from the five datasets. This average pooling strategy is often used in analyzing missing data with MI to account for the randomness of MI. We refer to Gelman and Hill (2006) and Van Buuren and Groothuis-Oudshoorn (2011) for more details on missing data and MI.

After completing the data with MI, we fit a random-effect LCM on each of the dataset. We apply the GIC method to selecting the number of latent classes. According to Figure 11, $L = 3$ and $L = 6$ are plausible options and we fit two LCMs with $L = 3$ and $L = 6$. In the case $L = 6$, one component in the estimated proportion vector $\widehat{\boldsymbol{p}}$ is fairly small (smaller than 0.005). The corresponding latent class can be neglected for better interpretability and parsimony. So we instead fit two models with $L = 3$ and $L = 5$. The estimated proportion vectors are $\widehat{\boldsymbol{p}}_3 = (0.25, 0.46, 0.29)$ and $\widehat{\boldsymbol{p}}_5 = (0.29, 0.37, 0.2, 0.02, 0.12)$ and the item parameters $\widehat{\boldsymbol{\theta}}$ are visualized in Figure 12.



(a) $\widehat{\boldsymbol{\theta}}_3$ (b) $\widehat{\boldsymbol{\theta}}_5$

Figure 12: Estimated item parameters for $L = 3$ and $L = 5$. A brighter color indicates a larger item parameter and hence larger probability to answer the corresponding item correctly.

In the $L = 3$ case, we see students in the three classes behave differently on the 47 items. Students in the first class top three classes and display best competence in all the sub-competences while students in the third class do not perform well on the test and need to improve their sub-competences. The competence of students in the second class is in the middle and may indicate students' average competence. Comparing two figures for $L = 3$ and $L = 5$, we see the first two classes roughly have the same performance on the items. Thus the $L = 5$ case can be viewed as a refined analysis on $L = 3$, where the third class is further divided into three parts. For $L = 5$, according to the relations between items and sub-competences summarized in Table 2, we can compare the sub-competences of different classes. For most of the sub-competences (DK, DR, GA, GK, GR, NA, NK), the first class performs best, followed by the second class, which further outperforms third and fifth classes. The fourth class has the worst performance. Note that the estimated proportion for the fourth class is 0.02 and hence only a few students have such unsatisfactory performance. For sub-competences DA and NR, the first class remains the top. The second and fifth classes follow the first class and have similar performance. The third and fourth classes have an unsatisfactory performance compared with the others.

We can also assess the difficulty of items. One goal of latent class analysis is to find items that best distinguish different classes. From the plots we see different classes have different item

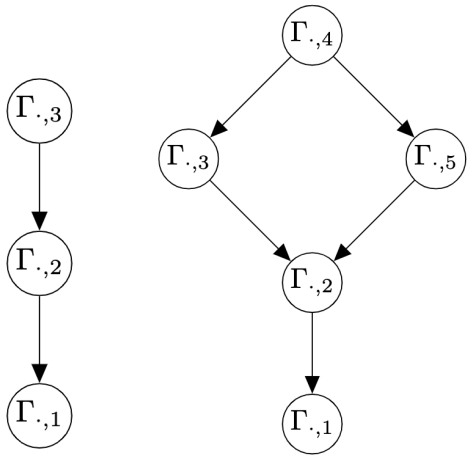
parameters on most of the items, indicating these items can distinguish between classes. However, there exist some items that are not ideal in this respect. For instance, all classes have relatively good performance on item 33 (M041335 in original index of the tests). This question presents four bar plots of three colors (red, green and blue) and asks students which bar plot has the smallest value for blue. Since the frequency is clearly shown in the plots, this question may be easy for fourth graders and may not be an ideal item to distinguish between different classes. Furthermore, all classes have relatively poor performance on item 39 (M051006 Cost of ice cream), indicating item 39 is of high difficulty level.

We further explore the latent hierarchical structures of the latent classes and their interpretations. Following the idea in Section 4.2 of Ma et al. (2022), we first define

$$\mathbf{\Gamma} = \left(\mathbb{I}\left\{ \left| \widehat{\theta}_{j,l} - \max_{m\in[L]} \widehat{\theta}_{j,m} \right| \le \tau \right\} : j \in [J], l \in [L] \right) \in \{0,1\}^{J\times L}$$

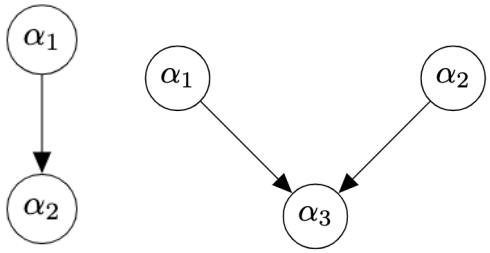
for a small $\tau > 0$. Note that $\Gamma_{j,l} = 1$ indicates that $\widehat{\theta}_{j,l}$ is close to $\max_{m\in[L]} \widehat{\theta}_{j,m}$ and hence latent class l possesses relatively high level of item parameter for item j. We say a latent class l_1 is more capable than a latent class l_2 and represent it as $\mathbf{\Gamma}_{\cdot,l_2} \to \mathbf{\Gamma}_{\cdot,l_1}$ if $\mathbf{\Gamma}_{\cdot,l_1} \succeq \mathbf{\Gamma}_{\cdot,l_2}$, where for two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, we write $\boldsymbol{v}_1 \succeq \boldsymbol{v}_2$ if $v_{1,i} \ge v_{2,i}$ for each i. With this definition we can get partial orders among latent classes. In our real data analysis, we let $\tau = 0.25$ (roughly the standard error of all item parameters) and relax the definition of $\mathbf{\Gamma}_{\cdot,l_1} \succeq \mathbf{\Gamma}_{\cdot,l_2}$ to $\Gamma_{j,l_1} \ge \Gamma_{j,l_2}$ for 90% of items when j varies in $\{1,\ldots,J\}$. The resulting partial orders based on item parameters can be represented as directed acyclic graphs (DAGs) in Figure 13.

With the obtained partial orders, we can apply the latent hierarchy recovering algorithm in Ma et al. (2022) to obtain the latent attribute representations of the latent classes under the cognitive diagnostic modeling framework. In particular, for $L = 3$, the three latent classes can be represented as $\{(1,1),(1,0),(0,0)\}$, and for $L = 5$, the five latent classes can be represented as $\{(1,1,1),(1,1,0),(1,0,0),(0,0,0),(0,1,0)\}$. The hierarchies among the learned latent attributes can be represented as in Figure 14. Specifically, in the case $L = 3$, α_1 is a more basic prerequisite for α_2; similarly in the case $L = 5$, α_1 and α_2 may be more basic prerequisites for attribute α_3.



(a) $L = 3$ (b) $L = 5$

Figure 13: Partial orders among latent classes



(a) $L = 3$ (b) $L = 5$

Figure 14: Latent hierarchical structures of attributes

The estimated item parameters may help us identify which sub-competences each learned attribute α_i corresponds to. In the case $L = 3$, we find $\widehat{\theta}_{j,1} - \widehat{\theta}_{j,2}$ is fairly large (greater than 0.2) for most items j corresponding to DA, GR, NA, NK, which may be the sub-competences represented by attribute α_2. Similarly students in class 2 greatly improve their competences in DK, DR and GA compared with students in class 3. Hence the more basic attribute α_1 may represent DK, DR and GA. Similarly in the case $L = 5$, we find α_1 corresponds to sub-competences NA, GK, NK and GA, α_2 corresponds to NA, GA, DK and DR, α_3 corresponds to DA, NA, NK and NR. According to the hierarchical structure in Figure 14, DA and number-related skills NA, NK and NR may be more advanced sub-competences than others.

We conclude this section by discussing the connection and difference between latent class analysis and more delicate cognitive diagnostic models in real data analysis. CDMs can be viewed as special cases of LCMs with more delicate structures on the item parameters; indeed, CDMs belong to a family of *restricted* latent class models (Xu, 2017; Xu and Shang, 2018). With the structure specified by Q-matrix, CDMs can provide more fine-grained analysis than LCMs (e.g. we can estimate the latent profile of each individual). On the other hand, LCMs do not require the prior knowledge on the Q-matrix and may serve the purpose of exploratory analysis before modeling the data with more delicate CDMs, as shown in Ma et al. (2022) and explored in our data analysis. Hence LCMs and CDMs are closely related and both can be useful in various applications.

7 Discussion

This paper investigates the computation and theory for large-scale latent class models. In terms of computation, commonly used likelihood-based methods (e.g. EM algorithm) suffer from slow convergence rate and potential convergence to local optima under poor initializations. Recent developments in tensor decomposition and its applications provide a computationally efficient moment-based method to estimate the parameters. However, such tensor method is based on low-order moments of the observed variables rather than the entire likelihood function. Hence it is not statistically efficient and generally requires a large number of samples to ensure the accuracy of estimates. In this work, we propose a two-stage tensor-EM estimation pipeline which combines these two methods. Simulation studies empirically show the proposed procedure is both computationally efficient and statistically accurate. Moreover, based on our simulations in tensor methods the moments in (7) need to be estimated accurately. When the sample size is not very large (e.g. $N = 100$), the tensor method may not yield good estimates since the estimation of moments is not accurate. In applications ideally we need at least $N = 500$ samples to ensure the good performance. Note that Condition 1 implicitly constrains the number of classes L. If we want to fit a LCM with L classes, then condition 1 requires $L \leq \min\{J_1, J_2, J_3\} \leq J/3$. In applications we should always fit LCM with $L \leq J/3$ classes so that the Condition 1 is satisfied and tensor method can be applied. Furthermore, we theoretically establish the clustering consistency (consistency of latent

class membership) of large-scale fixed-effect latent class analysis where sample size N and number of items J both go to infinity. Consistency of item parameters is proved as a corollary of clustering consistency. In terms of consistency under random-effect LCM, we empirically verify it in simulations. However, in the likelihood function of random-effect LCM, the latent variable is marginalized out, making the log-likelihood more challenging to analyze the consistency of parameters. We will leave it as future work.

We also note that the response variables do not have to be binary for applying the tensor method; the tensor power method can also apply to models with polytomous or continuous responses. This is because the low-order moments constructed in and exploited by the tensor power method here (also see Anandkumar et al., 2014) essentially result from the local independence in LCMs; that is, the observed variables are conditionally independent given the latent variable. Regarding the theoretical results for polytomous response, we believe similar proof techniques can be applied to establish the clustering consistency. The details are left as future work. It is also possible to find similar tensor structures in more complicated latent structure models, for example, the diagnostic classification models or cognitive diagnostic models (CDMs) (Rupp and Templin, 2008; von Davier and Lee, 2019). Since CDMs can be viewed as extensions of LCMs where the local independence assumption also holds, the same tensor power method used here may also be used to find rough estimates of parameters in a CDM. However, CDMs also have a unique feature, the Q-matrix (Tatsuoka, 1983), which induces complicated parameter constraints on top of a latent class model. Thus a plain tensor power method for unrestricted LCMs would not give estimates that satisfy the equality constraints under a Q-matrix. How to develop an efficient estimation procedure for CDMs by integrating the tensor method is an interesting question left for future investigation.

References

Allman, E. S., Matias, C., and Rhodes, J. A. (2009). Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132.

Anandkumar, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and Liu, Y.-K. (2012a). A spectral

algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832.

Anandkumar, A., Hsu, D., and Kakade, S. M. (2012b). A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pages 33–1.

Balakrishnan, S., Wainwright, M. J., and Yu, B. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120.

Bandeen-Roche, K., Miglioretti, D. L., Zeger, S. L., and Rathouz, P. J. (1997). Latent variable regression for multiple discrete outcomes. *Journal of the American Statistical Association*, 92(440):1375–1386.

Bucholz, K., Hesselbrock, V., Heath, A., Kramer, J., and Schuckit, M. (2000). A latent class analysis of antisocial personality disorder symptom data from a multi-centre family study of alcoholism. *Addiction*, 95(4):553–567.

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

Chaganty, A. T. and Liang, P. (2013). Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning*, pages 1040–1048. PMLR.

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., Li, X., and Zhang, S. (2019a). Joint maximum likelihood estimation for high-dimensional exploratory item factor analysis. *Psychometrika*, 84(1):124–146.

Chen, Y., Li, X., and Zhang, S. (2019b). Structured latent factor analysis for large-scale data: Identifiability, estimability, and their implications. *Journal of the American Statistical Association*, pages 1–15.

Collins, L. M. and Lanza, S. T. (2009). *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences*, volume 718. John Wiley & Sons.

De, L., De Moor, B., and Vandewalle, J. (2000). On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342.

De Lathauwer, L. and De Moor, B. (1998). From matrix to tensor: Multilinear algebra and signal processing. In *Institute of mathematics and its applications conference series*, volume 67, pages 1–16. Citeseer.

Dean, N. and Raftery, A. E. (2010). Latent class analysis variable selection. *Annals of the Institute of Statistical Mathematics*, 62(1):11.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Dunn, K. M., Jordan, K., and Croft, P. R. (2006). Characterizing the course of low back pain: a latent class analysis. *American journal of epidemiology*, 163(8):754–761.

Fan, Y. and Tang, C. Y. (2013). Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):531–552.

Gelman, A. and Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

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. and Xu, G. (2021). A joint MLE approach to large-scale structured latent attribute analysis. *Journal of the American Statistical Association*, DOI:10.1080/01621459.2021.1955689.

Hagenaars, J. A. and McCutcheon, A. L. (2002). *Applied latent class analysis*. Cambridge University Press.

Harshman, R. A. (1970). *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis*. University of California at Los Angeles Los Angeles, CA.

Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189.

Hsu, D. and Kakade, S. M. (2013). Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM.

Keel, P. K., Fichter, M., Quadflieg, N., Bulik, C. M., Baxter, M. G., Thornton, L., Halmi, K. A., Kaplan, A. S., Strober, M., Woodside, D. B., et al. (2004). Application of a latent class analysis to empirically define eatingdisorder phenotypes. *Archives of General Psychiatry*, 61(2):192–200.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Kongsted, A. and Nielsen, A. M. (2017). Latent class analysis in health research. *Journal of physiotherapy*, 63(1):55–58.

Kruskal, J. B. (1976). More factors than subjects, tests and treatments: an indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika*, 41(3):281–293.

Lanza, S. T. and Rhoades, B. L. (2013). Latent class analysis: an alternative perspective on subgroup analysis in prevention and treatment. *Prevention Science*, 14(2):157–168.

Lazarsfeld, P. F. and Henry, N. W. (1968). *Latent structure analysis*. Houghton Mifflin Co.

Lubke, G. H. and Muthén, B. (2005). Investigating population heterogeneity with factor mixture models. *Psychological methods*, 10(1):21.

Ma, C., Ouyang, J., and Xu, G. (2022). Learning latent and hierarchical structures in cognitive diagnosis models. *Psychometrika, to appear, doi.org/10.1007/s11336-022-09867-5*.

McCullagh, P. (2018). *Tensor Methods in Statistics: Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC.

McLachlan, G. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

Muthén, B. and Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the EM algorithm. *Biometrics*, 55(2):463–469.

Neyman, J. and Scott, E. L. (1948). Consistent estimates based on partially consistent observations. *Econometrica: Journal of the Econometric Society*, pages 1–32.

Nishii, R. (1984). Asymptotic properties of criteria for selection of variables in multiple regression. *The Annals of Statistics*, pages 758–765.

Nylund, K. L., Asparouhov, T., and Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural equation modeling: A multidisciplinary Journal*, 14(4):535–569.

Ouyang, J. and Xu, G. (2022). Identifiability of latent class models with covariates. *Psychometrika, to appear, doi.org/10.1007/s11336-022-09852-y*.

Rupp, A. A. and Templin, J. L. (2008). Unique characteristics of diagnostic classification models: A comprehensive review of the current state-of-the-art. *Measurement*, 6(4):219–262.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Sedat, Ş. and Arican, M. (2015). A diagnostic comparison of turkish and korean students' mathematics performances on the timss 2011 assessment. *Journal of Measurement and Evaluation in Education and Psychology*, 6(2).

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

Smilde, A., Bro, R., and Geladi, P. (2005). *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons.

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

Tucker, L. R. (1964). The extension of factor analysis to three-dimensional matrices. *Contributions to Mathematical Psychology*, 110119.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

Van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67.

Vermunt, J. K. (2003). Applications of latent class analysis in social science research. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 22–36. Springer.

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. *Political analysis*, pages 450–469.

von Davier, M. and Lee, Y.-S. (2019). Handbook of diagnostic classification models. *Cham: Springer International Publishing*.

Wang, M. and Hanges, P. J. (2011). Latent class procedures: Applications to organizational research. *Organizational Research Methods*, 14(1):24–31.

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.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. (2014). Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Advances in neural information processing systems*, 27.

Appendices

In this supplementary material, the proof of Theorem 3 (clustering consistency) is presented in Appendix 1. Then Corollary 2 (consistency of item parameters) is proved in Appendix 2. In Appendix 3, more simulation results are provided to show the good performance of the proposed EM-tensor method.

Appendix 1: Proof of Theorem 3

Outline of proof idea. The proof follows the following 8 steps.

Step 1: Express $\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})$ in terms of $\sum_a n_a \sum_j D(\widehat{\theta}_{j,a} \| \bar{\theta}_{j,a}) + X - \mathbb{E}(X)$, where X is a random variable depending on $\mathbf{R}$ and $\bar{\boldsymbol{\theta}}^{(\mathbf{Z})}$ under $\mathbf{Z}$, and $n_a = \sum_{i=1}^N Z_{i,a}$.

Step 2: Bound the first term $\sum_j \sum_a n_{j,a} D(\widehat{\theta}_{j,a} \| \bar{\theta}_{j,a})$ in the above display uniformly over all possible $\mathbf{Z}$.

Step 3: Bound the second term $X - \mathbb{E}(X)$ using Bernstein type inequality. Combine this and Step 2 to obtain a bound for $\sup_{\mathbf{Z}} |\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})|$.

Step 4: (Denote the true latent class memberships by $\mathbf{Z}^0$ and joint MLE by $\widehat{\mathbf{Z}}$.) Establish $\bar{\ell}(\mathbf{Z}^0) \geq \bar{\ell}(\mathbf{Z})$ for all $\mathbf{Z}$. Use triangle inequality to upper-bound the non-negative quantity $\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\widehat{\mathbf{Z}})$.

$$0 \leq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\widehat{\mathbf{Z}}) \leq [\bar{\ell}(\mathbf{Z}^0) - \ell(\mathbf{R};\,\mathbf{Z}^0)] + [\ell(\mathbf{R};\,\mathbf{Z}^0) - \ell(\mathbf{R};\,\widehat{\mathbf{Z}})] + [\ell(\mathbf{R};\,\widehat{\mathbf{Z}}) - \bar{\ell}(\widehat{\mathbf{Z}})]$$

Since in the above display the middle group of terms $[\ell(R;\,\mathbf{Z}^0) - \ell(R;\,\widehat{\mathbf{Z}})] \leq 0$, we have $0 \leq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\widehat{\mathbf{Z}}) \leq 2\sup_{\mathbf{Z}} |\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})|$.

Step 5: Introduce the notion of partitions and generalize $\bar{\ell}(\mathbf{Z})$ to $\bar{\ell}(\Pi)$.

Step 6: Show that a refined partition increases $\bar{\ell}(\cdot)$. To be concrete, let Π^* be a refined partition of Π, then we have $\bar{\ell}(\Pi^*) \geq \bar{\ell}(\Pi)$.

Step 7: Show that for any latent class assignment $\mathbf{Z}$, we can find a partition Π^* that refines $\Pi^{\mathbf{Z}}$ and $\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) \geq \frac{1}{2} J \beta_J N_e(\mathbf{z})$.

Step 8: Apply results in step 6 and step 7 to MLE $\widehat{\mathbf{z}}$, we have

$$\text{bound in step 4} \geq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^{\widehat{\mathbf{Z}}}) \geq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) \geq \frac{1}{2} J \beta_J N_e(\widehat{\mathbf{z}}).$$

Now we formally begin the proof of Theorem 3 in the above several steps. In derivations below we abbreviate $\bar{\boldsymbol{\theta}}^{(\mathbf{Z})}$ as $\bar{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\theta}}^{(\mathbf{Z})}$ as $\widehat{\boldsymbol{\theta}}$ to simplify notations.

Step 1. Define $D(p\|q) = p\log(p/q)+(1-p)\log((1-p)/(1-q))$, the Kullback-Leibler divergence of a Bernoulli distribution with parameter p from that with parameter q. In this step we prove a lemma as follows.

Lemma 1. *Let $(R_{i,j};\, 1 \le N, 1 \le J)$ denote independent Bernoulli trials with parameters $(P_{i,j};\, 1 \le N, 1 \le J)$. Under a general latent class model, given an arbitrary $\mathbf{Z}$, there is*

$$\begin{aligned} &\sup_{\boldsymbol{\theta}} \ell(\mathbf{R};\, \mathbf{Z},\, \boldsymbol{\theta}) - \sup_{\boldsymbol{\theta}} \mathbb{E}[\ell(\mathbf{R};\, \mathbf{Z},\, \boldsymbol{\theta})] \\ =& \sum_{a=1}^{L} n_a \sum_j D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a}) + \sum_i \sum_j (R_{i,j} - P_{i,j}) \log\Big(\frac{\bar{\theta}_{j,z_i}}{1-\bar{\theta}_{j,z_i}}\Big) \\ =& \sum_{a=1}^{L} n_a \sum_j D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a}) + X - \mathbb{E}X, \end{aligned} \tag{19}$$

where $X = \sum_i \sum_j R_{i,j} \log\left(\frac{\bar{\theta}_{j,z_i}}{1-\bar{\theta}_{j,z_i}}\right)$ is random variable depending on $\mathbf{Z}$ and

$$\widehat{\theta}_{j,a} = \frac{\sum_i Z_{i,a} R_{i,j}}{\sum_i Z_{i,a}}, \quad \bar{\theta}_{j,a} = \frac{\sum_i Z_{i,a} P_{i,j}}{\sum_i Z_{i,a}} \tag{20}$$

Given a fixed $\mathbf{Z}$, denote $n_a^{(\mathbf{Z})} = \sum_{i=1}^{N} Z_{i,a}$.The maximizing properties of $\widehat{\theta}_{j,a}$ and $\bar{\theta}_{j,a}$ in 20 imply that

$$n_a \widehat{\theta}_{j,a} = \sum_i Z_{i,a} R_{i,j}, \quad n_a \bar{\theta}_{j,a} = \sum_i Z_{i,a} P_{i,j}. \tag{21}$$

Using (21), we have the following,

$$\begin{aligned} &\ell(\mathbf{R};\, \mathbf{Z}) - \bar{\ell}(\mathbf{Z}) \\ =& \sum_i \sum_j \sum_{a=1}^{L} Z_{i,a}[R_{i,j} \log \widehat{\theta}_{j,a} + (1 - R_{i,j}) \log(1 - \widehat{\theta}_{j,a})] \\ &\qquad - \sum_i \sum_j \sum_{a=1}^{L} Z_{i,a}[P_{i,j} \log \bar{\theta}_{j,a} + (1 - P_{i,j}) \log(1 - \bar{\theta}_{j,a})] \\ =& \sum_j \sum_{a=1}^{L} n_a[\widehat{\theta}_{j,a} \log \widehat{\theta}_{j,a} + (1 - \widehat{\theta}_{j,a}) \log(1 - \widehat{\theta}_{j,a})] - \sum_j \sum_{a=1}^{L} n_a[\bar{\theta}_{j,a} \log \bar{\theta}_{j,a} + (1 - \bar{\theta}_{j,a}) \log(1 - \bar{\theta}_{j,a})] \\ =& \sum_j \sum_{a=1}^{L} \Big\{ n_a[\widehat{\theta}_{j,a} \log \widehat{\theta}_{j,a} + (1 - \widehat{\theta}_{j,a}) \log(1 - \widehat{\theta}_{j,a})] - n_a[\widehat{\theta}_{j,a} \log \bar{\theta}_{j,a} + (1 - \widehat{\theta}_{j,a}) \log(1 - \bar{\theta}_{j,a})] \Big\} \end{aligned}$$

$$+\sum_j\sum_{a=1}^{L}\Big\{n_a[\widehat{\theta}_{j,a}\log\bar{\theta}_{j,a}+(1-\widehat{\theta}_{j,a})\log(1-\bar{\theta}_{j,a})]-n_a[\bar{\theta}_{j,a}\log\bar{\theta}_{j,a}+(1-\bar{\theta}_{j,a})\log(1-\bar{\theta}_{j,a})]\Big\}$$

$$=\sum_{a=1}^{L}n_a\sum_j D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a})+\sum_i\sum_j\Big\{[R_{i,j}\log\bar{\theta}_{j,z_i}+(1-R_{i,j})\log(1-\bar{\theta}_{j,z_i})]$$

$$-[P_{i,j}\log\bar{\theta}_{j,z_i}+(1-P_{i,j})\log(1-\bar{\theta}_{j,z_i})]\Big\}$$

$$=\sum_{a=1}^{L}n_a\sum_j D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a})+\sum_i\sum_j R_{i,j}\log\Big(\frac{\bar{\theta}_{j,z_i}}{1-\bar{\theta}_{j,z_i}}\Big)-\sum_i\sum_j P_{i,j}\log\Big(\frac{\bar{\theta}_{j,z_i}}{1-\bar{\theta}_{j,z_i}}\Big).$$

Define the random variable

$$X=\sum_i\sum_j R_{i,j}\log(\bar{\theta}_{j,z_i}/(1-\bar{\theta}_{j,z_i})), \tag{22}$$

then X depends on $\mathbf{Z}$ and the above display becomes the summation of $\sum_{a=1}^{L}n_a\sum_j D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a})$ and $X-\mathbb{E}[X]$. This establishes (19) in Lemma 1. In the following, we bound the first term $\sum_{a=1}^{L}n_a\sum_j D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a})$ and the second term $X-\mathbb{E}[X]$ in the above display uniformly over all possible $\mathbf{Z}$, respectively in Step 2 and Step 3.

Step 2. In this step we prove the following theorem.

Theorem 4. *The following event happens with probability at least* $1-\delta$,

$$\max_{\mathbf{Z}}\left\{\sum_j\sum_a n_a D(\widehat{\theta}^{\mathbf{Z}}_{j,a}\|\bar{\theta}^{\mathbf{Z}}_{j,a})\right\}<N\log L+JL\log\Big(\frac{N}{L}+1\Big)-\log\delta.$$

Given any fixed latent class memberships $\mathbf{Z}$, every $\widehat{\theta}_{j,a}$ is an average of n_a independent Bernoulli random variables $R_{1,j},\ldots,R_{N,j}$ with mean $\bar{\theta}_{j,a}$. We apply the Chernoff-Hoeffding theorem to obtain

$$\mathbb{P}(\widehat{\theta}_{j,a}\geq\bar{\theta}_{j,a}+t)\leq e^{-n_aD(\bar{\theta}_{j,a}+t\|\bar{\theta}_{j,a})},\quad \mathbb{P}(\widehat{\theta}_{j,a}\leq\bar{\theta}_{j,a}-t)\leq e^{-n_aD(\bar{\theta}_{j,a}-t\|\bar{\theta}_{j,a})}. \tag{23}$$

Note that given a fixed $\mathbf{Z}$, each $\widehat{\theta}_{j,a}$ can take values only in the finite set $\{0,1/n_a,2/n_a,\ldots,1\}$ of cardinality n_a+1. We denote this range of $\widehat{\theta}_{j,a}$ by $\widehat{\Theta}^{j,a}$. Then $\mathbb{P}(\widehat{\theta}_{j,a}=\vartheta)\leq\exp\{-n_aD(\vartheta\|\bar{\theta}_{j,a})\}$ for any $\vartheta\in\widehat{\Theta}^{j,a}$. Now consider the cardinality of the set $\widehat{\Theta}$ given $\mathbf{Z}$. Since for each of the $J\times L$ entries in $\widehat{\boldsymbol{\theta}}$, $\widehat{\theta}_{j,a}$ can independently take on n_a+1 different values, there is $|\widehat{\Theta}|=[\prod_a(n_a+1)]^J$. Considering the natural constraint $\sum_{a=1}^{L}n_a=N$, we have

$$|\widehat{\Theta}|=\Big[\prod_{a=1}^{L}(n_a+1)\Big]^J\leq\Big[\Big(\frac{N}{L}+1\Big)^L\Big]^J. \tag{24}$$

Define $\widehat{\Theta}_\epsilon = \{\widetilde{\boldsymbol{\theta}} \in \widehat{\Theta} : \sum_j \sum_a n_a D(\widetilde{\theta}_{j,a} \| \bar{\theta}_{j,a}) \geq \epsilon\}$, then $\widehat{\Theta}_\epsilon \subseteq \widehat{\Theta}$. Note that the components of $\widehat{\boldsymbol{\theta}}$ depend on different components of $\{R_{i,j}, i \in [N], j \in [J]\}$ and thus are independent. We have

$$
\begin{aligned}
&\mathbb{P}\Big(\sum_{j=1}^{J}\sum_{a=1}^{L} n_a D(\widehat{\theta}_{j,a} \| \bar{\theta}_{j,a}) \geq \epsilon\Big) \\
&= \sum_{\widetilde{\boldsymbol{\theta}} \in \widehat{\Theta}_\epsilon} \mathbb{P}\Big(\widehat{\boldsymbol{\theta}} = \widetilde{\boldsymbol{\theta}}\Big) \\
&\leq \sum_{\widetilde{\boldsymbol{\theta}} \in \widehat{\Theta}_\epsilon} \prod_j \prod_a \exp\{-n_a D(\widetilde{\theta}_{j,a} \| \bar{\theta}_{j,a})\} \\
&\leq \sum_{\widetilde{\boldsymbol{\theta}} \in \widehat{\Theta}_\epsilon} \exp\{-n_a \sum_j \sum_a D(\widetilde{\theta}_{j,a} \| \bar{\theta}_{j,a})\} \\
&\leq \sum_{\widetilde{\boldsymbol{\theta}} \in \widehat{\Theta}_\epsilon} \exp\{-\epsilon\} \\
&\leq |\widehat{\Theta}_\epsilon| e^{-\epsilon} \leq |\widehat{\Theta}| e^{-\epsilon} \leq \Big(\frac{N}{L} + 1\Big)^{JL} e^{-\epsilon}.
\end{aligned}
$$

The above result holds for fixed $\mathbf{Z}$, so applying a union bound over all the L^N possible assignment $\mathbf{Z}$, there is

$$
\mathbb{P}\Big(\max_{\mathbf{Z}} \Big\{\sum_j \sum_a n_a D(\widehat{\theta}_{j,a} \| \bar{\theta}_{j,a})\Big\} \geq \epsilon\Big) \leq L^N \Big(\frac{N}{L} + 1\Big)^{JL} e^{-\epsilon}.
$$

Now take $\delta = L^N \Big(\frac{N}{L} + 1\Big)^{JL} e^{-\epsilon}$, then $\epsilon = N \log L + JL \log(\frac{N}{L} + 1) - \log \delta$. Therefore the following event happens with probability at least $1 - \delta$,

$$
\max_{\mathbf{Z}} \Big\{\sum_j \sum_a n_a D(\widehat{\theta}_{j,a} \| \bar{\theta}_{j,a})\Big\} < \epsilon = N \log L + JL \log\Big(\frac{N}{L} + 1\Big) - \log \delta.
$$

This concludes the proof of Theorem 4.

Step 3. In this step we bound $|X - \mathbb{E}[X]|$, with X defined in (22). Denote $X_{i,j} = R_{i,j} \log(\bar{\theta}_{j,z_i}/(1 - \bar{\theta}_{j,z_i}))$, then $X = \sum_i \sum_j X_{i,j}$. Under Assumption 1, we have $|X_{i,j}| \leq \gamma \log J$. Then we have $\sum_i \sum_j \mathbb{E}[X_{i,j}^2] \leq \sum_i \sum_j \mathbb{P}(R_{i,j} = 1) \gamma^2 (\log J)^2 = \gamma^2 \sum_i \sum_j P_{i,j} (\log J)^2 = \gamma^2 MNJ (\log J)^2$. Applying the Bernstein's inequality to the sum of independent bounded random variables, we have the following holds for any fixed $\mathbf{Z}$,

$$
\mathbb{P}(|X - \mathbb{E}[X]| \geq \epsilon) \leq 2 \exp\left\{-\frac{(1/2)\epsilon^2}{\sum_i \sum_j \mathbb{E}[X_{i,j}^2] + (1/3)\gamma \log(J)\epsilon}\right\}
$$

$$\leq 2\exp\left\{-\frac{(1/2)\epsilon^2}{\gamma^2 MNJ(\log J)^2 + (1/3)\gamma\log(J)\epsilon}\right\}.$$

We next prove the following theorem.

Theorem 5. *Under the following scaling (as $N, J \to \infty$),*

$$\frac{MJ}{\log L} \to \infty\,, \frac{N}{L} \to \infty, \tag{25}$$

$$\sqrt{\frac{M}{J}}\left(\frac{N}{L}\right)^{1-\xi} \to \infty \text{ for some small } \xi > 0,$$

we have

$$\frac{1}{NJ}\max_{\mathbf{Z}}|\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})| = o_P\left(\frac{\sqrt{M\log L}}{\sqrt{J}}(\log J)^{1+\eta}\right)$$

for any $\eta > 0$.

We need to bound $|\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})|$ uniformly over all the $\mathbf{Z}$. Combining the results of Step 2 and Step 3, since there are L^N possible assignments of $\mathbf{Z}$, we apply the union bound to obtain

$$\begin{aligned}
&\mathbb{P}(\max_{\mathbf{Z}}|\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})| \geq 2\epsilon\delta_{NJ}) \qquad (26)\\
&\leq L^N\mathbb{P}\left[\left\{\sum_j\sum_a n_a D(\widehat{\theta}_{j,a}\|\bar{\theta}_{j,a}) \geq \epsilon\delta_{NJ}\right\} \cup \{|X - \mathbb{E}[X]| \geq \epsilon\delta_{NJ}\}\right]\\
&\leq \exp\left\{N\log L + JL\log\left(\frac{N}{L}+1\right) - \epsilon\delta_{NJ}\right\}\\
&\quad + 2\exp\left\{N\log L - \frac{(1/2)\epsilon^2\delta_{NJ}^2}{\gamma^2 MNJ(\log J)^2 + (1/3)\gamma\log(J)\epsilon\delta_{NJ}}\right\}.
\end{aligned}$$

In order for the term on the right hand side of the above display to go to zero, the following of δ_{NJ} would suffice,

$$\delta_{NJ} = N\sqrt{MJ\log L}(\log J)^{1+\eta}. \tag{27}$$

for a small positive constant η. Then the right hand side of (26) goes to zero as N, J go large and hence the scaling of J described in the theorem yields $\mathbb{P}(\max_{\mathbf{Z}}|\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})| \geq 2\epsilon\delta_{NJ}) = o(1)$, which implies

$$\frac{1}{NJ}\max_{\mathbf{Z}}|\ell(\mathbf{R};\,\mathbf{Z}) - \bar{\ell}(\mathbf{Z})| = o_P\left(\frac{\sqrt{M\log L}}{\sqrt{J}}(\log J)^{1+\eta}\right). \tag{28}$$

This proves Theorem 5.

Step 4. Denote the true class assignments by $\mathbf{Z}^0$. We first establish

$$\bar{\ell}(\mathbf{Z}^0) \geq \bar{\ell}(\mathbf{Z}), \quad \text{for all } \mathbf{Z}. \tag{29}$$

First note that $\theta^0_{j,z_i^0} = P_{i,j}$, and

$$\bar{\theta}_{j,z_i^0} = \frac{\sum_{m=1}^N Z^0_{m,z_i^0} P_{m,j}}{\sum_{m=1}^N Z^0_{m,z_i^0}} = \frac{\sum_{m=1}^N Z^0_{m,z_i^0} P_{i,j}}{\sum_{m=1}^N Z^0_{m,z_i^0}} = P_{i,j}.$$

The difference $\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\mathbf{Z})$ can be written as

$$\begin{aligned}
\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\mathbf{Z}) &= \sum_j \sum_i [P_{i,j} \log\Big(\frac{\bar{\theta}^0_{j,z_i^0}}{\bar{\theta}^{\mathbf{Z}}_{j,z_i}}\Big) + (1 - P_{i,j}) \log\Big(\frac{1 - \bar{\theta}^0_{j,z_i^0}}{1 - \bar{\theta}^{\mathbf{Z}}_{j,z_i}}\Big)] \\
&= \sum_j \sum_i [P_{i,j} \log\Big(\frac{P_{i,j}}{\bar{\theta}^{\mathbf{Z}}_{j,z_i}}\Big) + (1 - P_{i,j}) \log\Big(\frac{1 - P_{i,j}}{1 - \bar{\theta}^{\mathbf{Z}}_{j,z_i}}\Big)] = \sum_i \sum_j D(P_{i,j} \| \bar{\theta}^{\mathbf{Z}}_{j,z_i}) \geq 0,
\end{aligned}$$

therefore establishing (29). Since the above holds for every $\mathbf{Z}$, it also holds for the maximum likelihood estimator $\widehat{\mathbf{Z}}$. We further upper bound $\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\mathbf{Z})$ from above as follows,

$$0 \leq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\widehat{\mathbf{Z}}) \leq [\bar{\ell}(\mathbf{Z}^0) - \ell(\mathbf{R}; \mathbf{Z}^0)] + \underbrace{[\ell(\mathbf{R}; \mathbf{Z}^0) - \ell(\mathbf{R}; \widehat{\mathbf{Z}})]}_{\leq 0} + [\ell(\mathbf{R}; \widehat{\mathbf{Z}}) - \bar{\ell}(\widehat{\mathbf{Z}})],$$

where $[\ell(\mathbf{R}; \mathbf{Z}^0) - \ell(\mathbf{R}; \widehat{\mathbf{Z}})] \leq 0$ results from the definition of $\widehat{\mathbf{Z}}$ as the MLE, that is $\mathbf{Z}$ maximizes the $\ell(\mathbf{R}; \mathbf{Z}, \widehat{\boldsymbol{\theta}}^{\mathbf{Z}})$. Therefore

$$\begin{aligned}
0 \leq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\widehat{\mathbf{Z}}) &\leq [\bar{\ell}(\mathbf{Z}^0) - \ell(\mathbf{R}; \mathbf{Z}^0)] + [\ell(\mathbf{R}; \widehat{\mathbf{Z}}) - \bar{\ell}(\widehat{\mathbf{Z}})] \\
&\leq 2 \sup_{\mathbf{Z}} |\bar{\ell}(\mathbf{Z}) - \ell(\mathbf{R}; \mathbf{Z})| \\
&= o_p(\delta_{NJ}).
\end{aligned}$$

Step 5. To establish the consistency of MLE in clustering subjects into latent classes, we need to introduce the notion of partitions. First we observe that any latent class assignment $\mathbf{Z}$ defines a partition on $[N]$ into T subsets $(S_1, \ldots, S_T)$ via mapping $\Pi^{\mathbf{Z}}$ from $[N]$ to $[T]$ such that for any subject we have $\theta^0_{j,z_i} = \theta^0_{j,\Pi_i^{\mathbf{Z}}}$ for all j. We now generalize this notion. For any partition on $[N]$, define

$$\bar{\theta}^{\Pi}_{j,a} = \frac{1}{|S_a|} \sum_{i=1}^N \theta^0_{j,z_i^0} I(i \in S_a) = \frac{1}{|S_a|} \sum_{i=1}^N P_{i,j} I(i \in S_a)$$

as the average over all i in the subset S_a indexed by $\Pi_i = a$. We then define generalization of $\bar{\ell}(\mathbf{Z})$ as

$$\bar{\ell}(\Pi) = \sum_i \sum_j [P_{i,j} \log(\bar{\theta}^{\Pi}_{j,\Pi_i}) + (1 - P_{i,j}) \log(1 - \bar{\theta}^{\Pi}_{j,\Pi_i})].$$

Note that $\bar{\theta}^{\Pi^{\mathbf{Z}}}_{j,a} = \bar{\theta}^{\mathbf{Z}}_{j,a}$ and hence $\bar{\ell}(\Pi^{\mathbf{Z}}) = \bar{\ell}(\mathbf{Z})$ when the partition $\Pi^{\mathbf{Z}}$ is induced by latent class assignment $\mathbf{Z}$.

We will proceed as follows: in step 6 we show a refined partition increases $\bar{\ell}(\cdot)$. We then construct a refined partition Π^* for every partition $\Pi^{\mathbf{Z}}$ induced by $\mathbf{Z}$ and prove $\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) \geq \frac{1}{2} N_e(\mathbf{z}) \beta_J$ in step 7. Finally we apply the results to MLE $\widehat{\mathbf{Z}}$ and obtain the desired results in step 8.

Step 6. We prove the following lemma:

Lemma 2. *Let Π^* be a refinement of any partition Π of $[N]$, then we have $\bar{\ell}(\Pi^*) \geq \bar{\ell}(\Pi)$.*

Given $a \in [T^*]$ indexing S^*_a in Π^*, since $S^*_a \subseteq S_b$ for some S_b in Π, let $F(a)$ denote its index under Π (i.e. b). We have

$$\begin{aligned}
\bar{\ell}\left(\Pi^*\right) &= \sum_{a=1}^{T^*} |S^*_a| \sum_{j=1}^{J} \left\{ \bar{\theta}^{\Pi^*}_{j,a} \log \bar{\theta}^{\Pi^*}_{j,a} + \left(1 - \bar{\theta}^{\Pi^*}_{j,a}\right) \log \left(1 - \bar{\theta}^{\Pi^*}_{j,a}\right) \right\} \\
&\geq \sum_{a=1}^{T^*} |S^*_a| \sum_{j=1}^{J} \left\{ \bar{\theta}^{\Pi^*}_{j,a} \log \bar{\theta}^{\Pi}_{j,F(a)} + \left(1 - \bar{\theta}^{\Pi^*}_{j,a}\right) \log \left(1 - \bar{\theta}^{\Pi}_{j,F(a)}\right) \right\} \\
&= \sum_{b=1}^{T} |S_b| \sum_{j=1}^{J} \left\{ \bar{\theta}^{\Pi}_{j,b} \log \bar{\theta}^{\Pi}_{j,b} + \left(1 - \bar{\theta}^{\Pi}_{j,b}\right) \log \left(1 - \bar{\theta}^{\Pi}_{j,b}\right) \right\} = \bar{\ell}(\Pi).
\end{aligned}$$

The first equality is obtained by rewriting $\bar{\ell}(\Pi)$ in terms of subsets. The inequality follows from non-negativity of K-L distance. Then we combine terms in same class under Π and obtain the second equality.

Step 7. Now we prove a result on refinement.

Lemma 3. *For any latent class assignment $\mathbf{Z}$, there exists a partition Π^* that refines $\Pi^{\mathbf{Z}}$ and*

$$\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) \geq \frac{1}{2} N_e(\mathbf{z}) J \beta_J$$

For a given $\mathbf{Z}$, partition each latent class assigned by $\mathbf{Z}$ into sub-classes according to true assignments $\mathbf{Z}^0$ of each sample. For each sample i_1 that is incorrectly assigned by $\mathbf{Z}$ (by definition

this means its true class under $\mathbf{Z}^0$ is not in the majority within its estimated class under $\mathbf{Z}$), we find another sample i_2 assigned to same class under $\mathbf{Z}$ but i_1 and i_2 belong to different class under $\mathbf{Z}^0$ and make these two samples (i_1, i_2) a pair. We allow two misclassified samples to form a pair. Note that since incorrectly assigned samples are not in the majority of that class, we can find a pair for each of them.

Here is a simple example. Suppose in one class of $\mathbf{Z}$, we have 7 samples and $\mathbf{Z}^0$ (true latent class assignments) assigns them as three sub-classes $\{1,2,3,4\},\{5,6\},\{7\}$. In this example samples indexed by 5,6 and 7 are misclassified. We can find pairs $(4,5),(6,7)$.

The refined partition Π^* contains all such pairs and remaining correctly assigned samples in all classes assigned by $\mathbf{Z}$. So for the above example, the refined subset for that class is $\{1,2,3\},\{4,5\},\{6,7\}$. Let $e(\mathbf{z})$ by the set of incorrectly assigned sample. Clearly Π^* is a refinement and we have

$$\begin{aligned}
\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) &= \sum_i \sum_j D(P_{i,j} \| \bar{\theta}^{\Pi^*}_{j,\Pi^*_i}) \\
&\geq \sum_{i \in e(\mathbf{z})} \sum_j D(P_{i,j} \| \bar{\theta}^{\Pi^*}_{j,\Pi^*_i}) \\
&= \sum_{i \in e(\mathbf{z})} \sum_j D(P_{i,j} \| \frac{P_{i,j} + P_{i',j}}{2})
\end{aligned}$$

where i and i' are in different classes under $\mathbf{Z}^0$ while in same subset under Π^* by definition. Apply Pinsker's inequality we have

$$\begin{aligned}
D(P_{i,j} \| \frac{P_{i,j} + P_{i',j}}{2}) &\geq \frac{1}{2} \left[|P_{i,j} - \frac{P_{i,j} + P_{i',j}}{2}| + |1 - P_{i,j} - (1 - \frac{P_{i,j} + P_{i',j}}{2})| \right]^2 \\
&= \frac{1}{2}(P_{i,j} - P_{i',j})^2 \\
&= \frac{1}{2}(\theta^0_{j,z^0_i} - \theta^0_{j,z^0_{i'}})^2
\end{aligned}$$

Hence we have

$$\begin{aligned}
\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) &\geq \sum_{i \in e(\mathbf{z})} \sum_j \frac{1}{2}(\theta^0_{j,z^0_i} - \theta^0_{j,z^0_{i'}})^2 \\
&\geq \sum_{i \in e(\mathbf{z})} \frac{1}{2} \| \boldsymbol{\theta}^0_{\cdot,z^0_i} - \boldsymbol{\theta}^0_{\cdot,z^0_{i'}} \|^2 \\
&\geq \frac{1}{2} N_e(\mathbf{z}) J \beta_J
\end{aligned}$$

Step 8. Apply Lemma 3 to MLE $\widehat{\mathbf{z}}$, there exists a refinement of $\Pi^{\widehat{\mathbf{Z}}}$ denoted as Π^* such that

$$\bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) \geq \frac{1}{2} N_e(\mathbf{z}) J \beta_J$$

By Lemma 2 we have $\bar{\ell}(\Pi^*) \geq \bar{\ell}(\Pi^{\widehat{\mathbf{Z}}})$. So we conclude that

$$\begin{aligned} o_P(\delta_{NJ}) &= \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\widehat{\mathbf{Z}}) \\ &\geq \bar{\ell}(\mathbf{Z}^0) - \bar{\ell}(\Pi^*) \\ &\geq \frac{1}{2} N_e(\mathbf{z}) J \beta_J \end{aligned}$$

which completes the proof.

Appendix 2: Proof of Corollary 2

Recall $m_a = \arg\max_{l \in [L]} \sum_{i \in \widehat{C}_a} Z^0_{i,l}$ is the class index under $\mathbf{Z}^0$ for cluster $\widehat{C}_a$. For any $0 < \epsilon < \tau$, define the following event

$$A^\epsilon_N = \{N_e(\widehat{\mathbf{z}})/N \leq \epsilon\}.$$

On the event A^ϵ_N, for any $l \in [L]$, since we assume $n^0_l/N \geq \tau > 0$, we claim that there is exactly one $a \in [L]$ such that $m_a = l$, i.e. the $a-$th cluster represents the $l-$th class. To see the existence of such a, assume by contradiction that for some l there is no a such that $m_a = l$, then all subjects in class l are misclassified and we have

$$N_e(\widehat{\mathbf{z}})/N \geq n^0_l/N \geq \tau > \epsilon,$$

a contradiction. Since for each $l-$th class we can find $a - th$ cluster to represent it and there are exactly L clusters $\widehat{C}_1, \ldots, \widehat{C}_L$, such a must be unique for all the L classes. Note that $\mathbb{P}(A^\epsilon_N) \to 1$, the first statement in the corollary is proved.

For any $\epsilon \in (0, \tau)$, from the argument above, on the event A^ϵ_N for each l we can find exactly one $a \in [L]$ such that $m_a = l$, then the joint MLE for $\theta^0_{j,l}$ is

$$\widehat{\theta}_{j,a} = \widehat{\theta}^{(\widehat{\mathbf{z}})}_{j,a} = \frac{\sum_{i=1}^N \widehat{Z}_{i,a} R_{i,j}}{\sum_{i=1}^N \widehat{Z}_{i,a}}.$$

Recall we can rewrite $\theta^0_{j,l}$ as

$$\theta^0_{j,l} = \frac{\sum_{i=1}^N Z^0_{i,l} \theta^0_{j,l}}{\sum_{i=1}^N Z^0_{i,l}} = \frac{\sum_{i=1}^N Z^0_{i,l} P_{i,j}}{\sum_{i=1}^N Z^0_{i,l}}.$$

By triangle inequality we have

$$
\begin{aligned}
&\max_j |\widehat{\theta}_{j,a} - \theta^0_{j,l}| \\
&= \max_j \left| \frac{\sum_{i=1}^N \widehat{Z}_{i,a} R_{i,j}}{\sum_{i=1}^N \widehat{Z}_{i,a}} - \frac{\sum_{i=1}^N Z^0_{i,l} P_{i,j}}{\sum_{i=1}^N Z^0_{i,l}} \right| \\
&\le \max_j \left| \frac{\sum_{i=1}^N \widehat{Z}_{i,a} R_{i,j}}{\sum_{i=1}^N \widehat{Z}_{i,a}} - \frac{\sum_{i=1}^N \widehat{Z}_{i,a} R_{i,j}}{\sum_{i=1}^N Z^0_{i,l}} \right| + \max_j \left| \frac{\sum_{i=1}^N \widehat{Z}_{i,a} R_{i,j}}{\sum_{i=1}^N Z^0_{i,l}} - \frac{\sum_{i=1}^N Z^0_{i,l} R_{i,j}}{\sum_{i=1}^N Z^0_{i,l}} \right| \\
&+ \max_j \left| \frac{\sum_{i=1}^N Z^0_{i,l} R_{i,j}}{\sum_{i=1}^N Z^0_{i,l}} - \frac{\sum_{i=1}^N Z^0_{i,l} P_{i,j}}{\sum_{i=1}^N Z^0_{i,l}} \right| \\
&\equiv I_1 + I_2 + I_3.
\end{aligned}
$$

We then analyze these three terms.

$$
I_1 \le \max_j \sum_i \widehat{Z}_{i,a} R_{i,j} \frac{\sum_i |\widehat{Z}_{i,a} - Z^0_{i,l}|}{n^0_l \sum_i \widehat{Z}_{i,a}} \le \frac{\sum_i |\widehat{Z}_{i,a} - Z^0_{i,l}|}{n^0_l}.
$$

There are two cases in which $|\widehat{Z}_{i,a} - Z^0_{i,l}| = 1$:

- $\widehat{Z}_{i,a} = 1, Z^0_{i,l} = 0$, i.e. subject i is in cluster a but not in class l. Since $m_a = l$, subject i is misclassified and counted in $N_e(\widehat{\mathbf{z}})$.
- $\widehat{Z}_{i,a} = 0, Z^0_{i,l} = 1$, i.e. subject i is in class l but not in cluster a. Since cluster a is the only cluster that represents class l, subject i must be misclassified and counted in $N_e(\widehat{\mathbf{z}})$.

By clustering consistency we have

$$
I_1 \le \frac{N_e(\widehat{\mathbf{z}})}{n^0_l} \le \frac{N_e(\widehat{\mathbf{z}})}{\tau N} \xrightarrow{P} 0.
$$

For the second term we have

$$
I_2 = \frac{\max_j |\sum_i R_{i,j}(\widehat{Z}_{i,a} - Z^0_{i,l})|}{n^0_l} \le \frac{\sum_i |\widehat{Z}_{i,a} - Z^0_{i,l}|}{n^0_l} \le \frac{N_e(\widehat{\mathbf{z}})}{n^0_l} \xrightarrow{P} 0.
$$

For the third term, we apply Hoeffding's inequality and obtain

$$
\mathbb{P}(I_3 \ge \delta) = \mathbb{P}\left(\max_j \frac{|\sum_i Z^0_{i,l}(R_{i,j} - P_{i,j})|}{n^0_l} \ge \delta \right) \le J \exp(-2 n^0_l \delta^2) \le J \exp(-2\tau N \delta^2) \to 0
$$

where in the last step we use the scaling condition $\sqrt{\frac{M}{J}} \left(\frac{N}{L}\right)^{1-\xi} \to \infty$. This shows

$$
I_3 \xrightarrow{P} 0.
$$

Note that on $(A_N^\epsilon)^c$ we may not be able to define $\widehat{\theta}_{j,a}$ since the first statement in the corollary may not hold (we may not find the $a-th$ cluster for each $l-$th class). Mathematically we can arbitrarily define any $\widehat{\theta}_{j,a}$ as long as it is in $[0,1]$. Since $\mathbb{P}(A_N^\epsilon) \to 1$, we only need to focus on the situation on A_N^ϵ. We then have

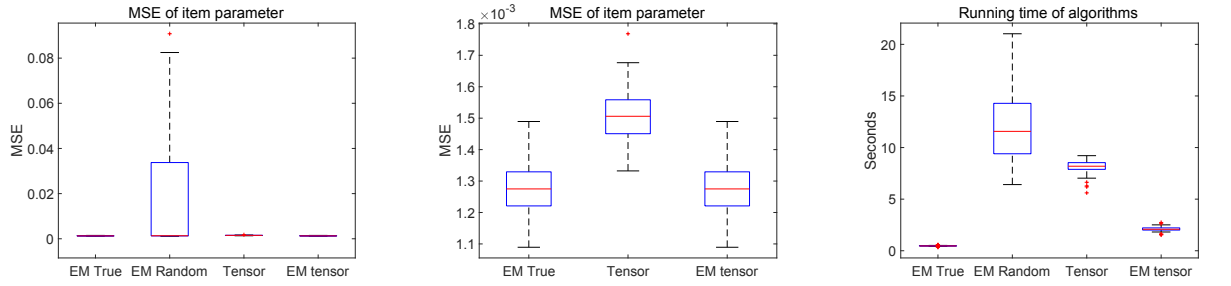
$$\begin{aligned}
&\mathbb{P}(\max_j |\widehat{\theta}_{j,a} - \theta^0_{j,l}| \geq \epsilon) \\
\leq &\mathbb{P}(\max_j |\widehat{\theta}_{j,a} - \theta^0_{j,l}| I_{(A_N^\epsilon)^c} \geq \epsilon/2) + \mathbb{P}(\max_j |\widehat{\theta}_{j,a} - \theta^0_{j,l}| I_{A_N^\epsilon} \geq \epsilon/2) \\
\leq &\mathbb{P}((A_N^\epsilon)^c) + \mathbb{P}(I_1 I_{A_N^\epsilon} \geq \epsilon/6) + \mathbb{P}(I_2 I_{A_N^\epsilon} \geq \epsilon/6) + \mathbb{P}(I_3 I_{A_N^\epsilon} \geq \epsilon/6) \to 0.
\end{aligned}$$

This completes the proof.

Appendix 3: More simulation results

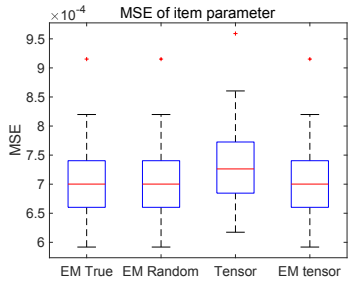
Random-effect LCM

We first present more simulation results for random-effect LCM.

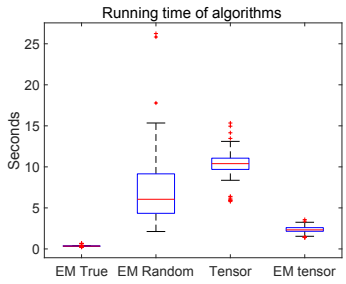


(a) MSE of item parameters (b) MSE without EM-random (c) Running time of the algorithms

Figure 15: $N = 1000, J = 100, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

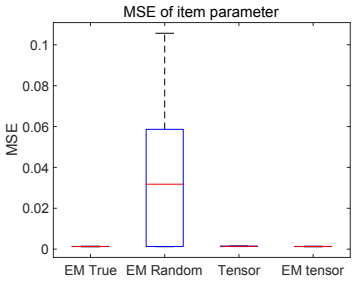


(a) MSE of item parameters

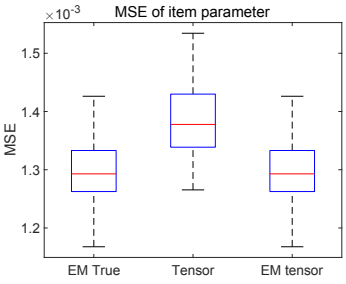


(b) Running time of the algorithms

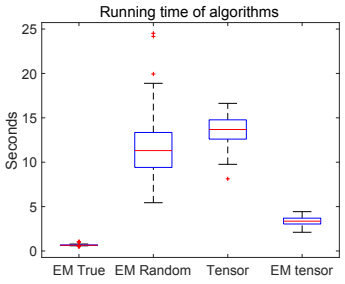
Figure 16: $N = 1000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

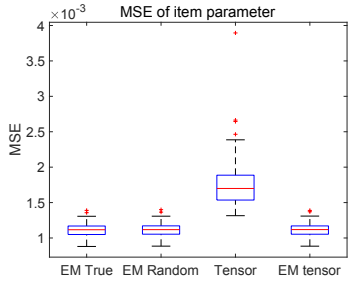


(b) MSE without EM-random

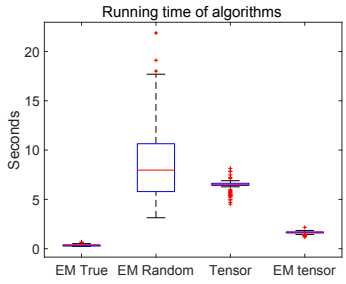


(c) Running time of the algorithms

Figure 17: $N = 1000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

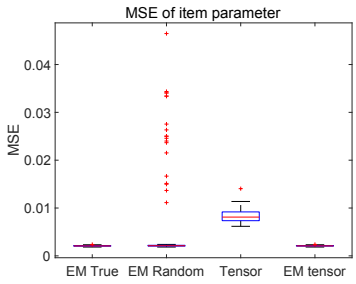


(a) MSE of item parameters

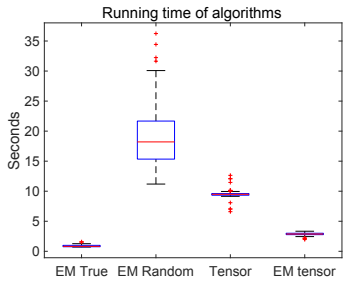


(b) Running time of the algorithms

Figure 18: $N = 1000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

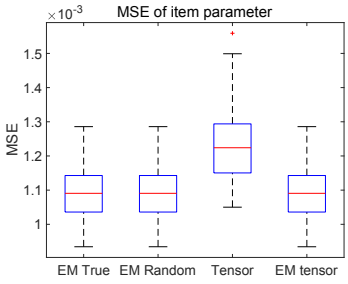


(a) MSE of item parameters

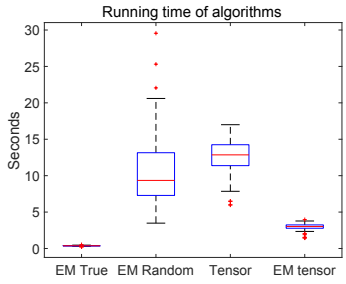


(b) Running time of the algorithms

Figure 19: $N = 1000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

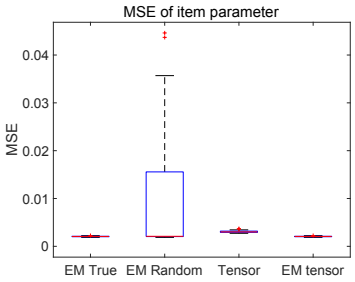


(a) MSE of item parameters

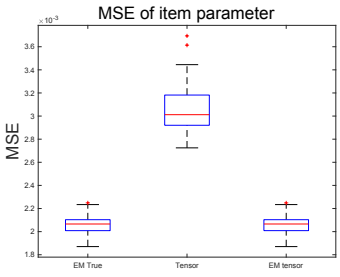


(b) Running time of the algorithms

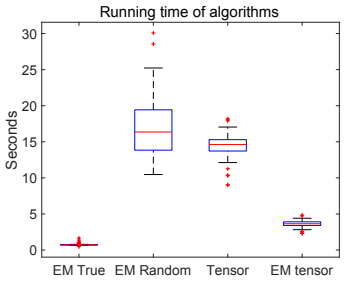
Figure 20: $N = 1000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

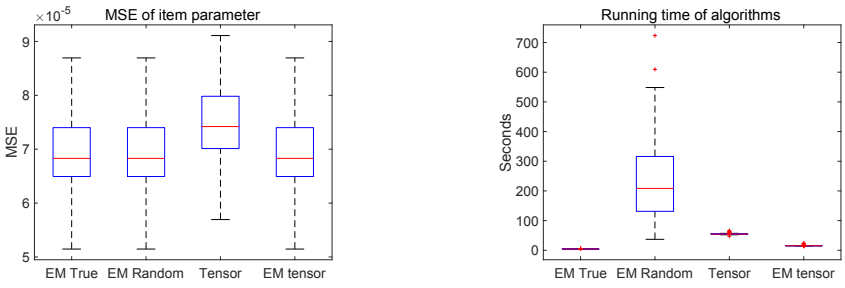


(b) MSE without EM-random



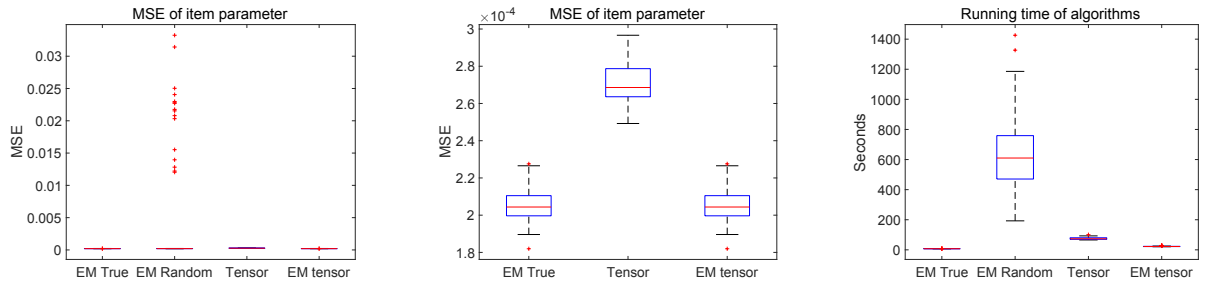
(c) Running time of the algorithms

Figure 21: $N = 1000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



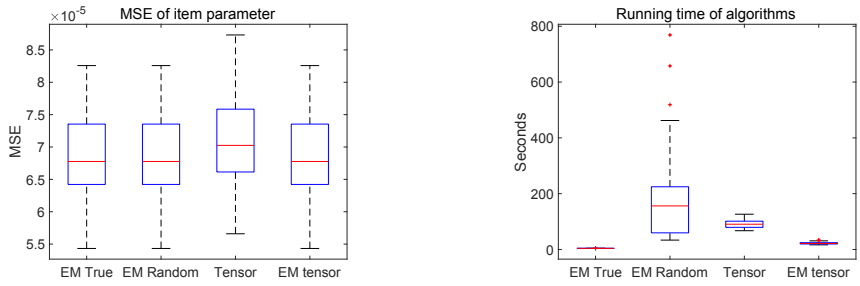
(a) MSE of item parameters (b) Running time of the algorithms

Figure 22: $N = 10000, J = 100, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



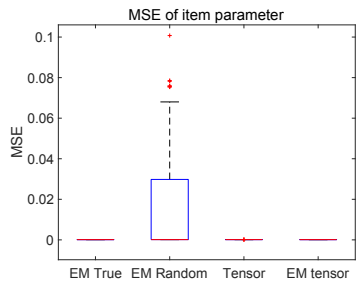
(a) MSE of item parameters (b) MSE without EM-random (c) Running time of the algorithms

Figure 23: $N = 10000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

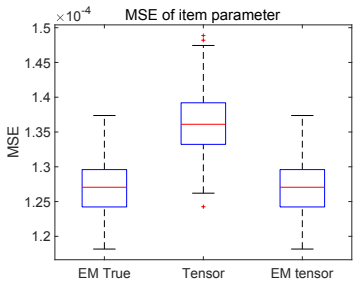


(a) MSE of item parameters (b) Running time of the algorithms

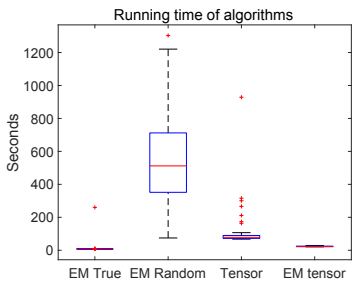
Figure 24: $N = 10000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

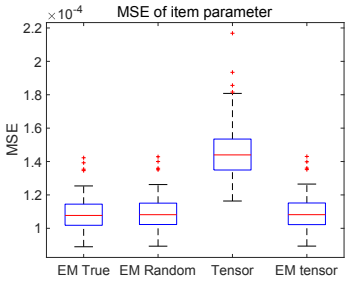


(b) MSE without EM-random

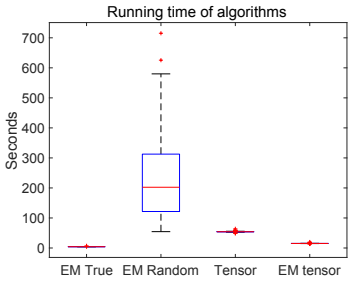


(c) Running time of the algorithms

Figure 25: $N = 10000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

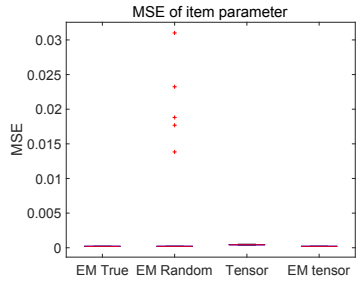


(a) MSE of item parameters

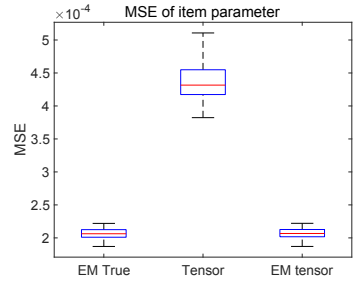


(b) Running time of the algorithms

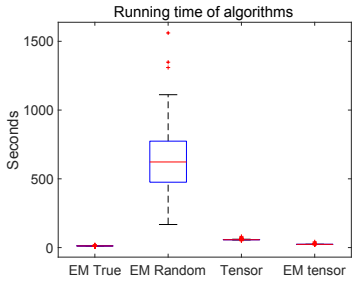
Figure 26: $N = 10000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

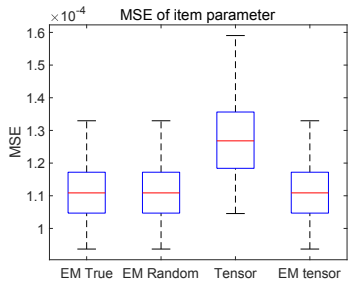


(b) MSE without EM-random

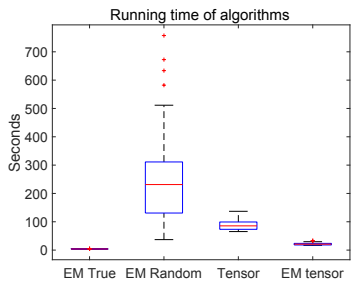


(c) Running time of the algorithms

Figure 27: $N = 10000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

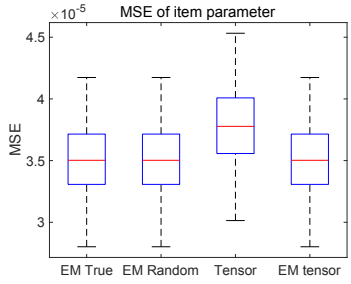


(a) MSE of item parameters

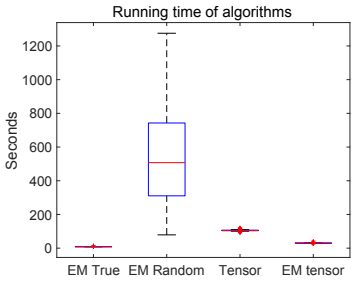


(b) Running time of the algorithms

Figure 28: $N = 10000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

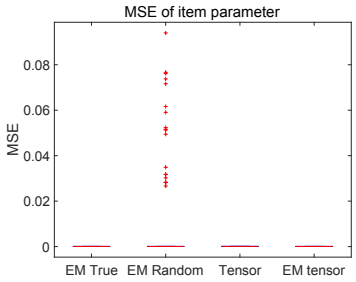


(a) MSE of item parameters

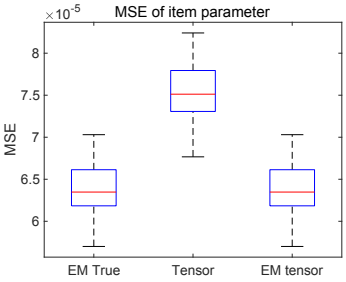


(b) Running time of the algorithms

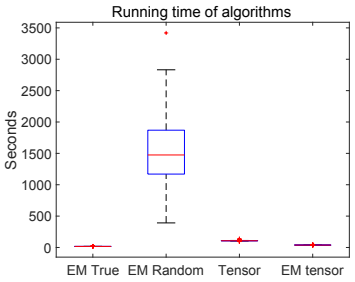
Figure 29: $N = 20000, J = 100, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

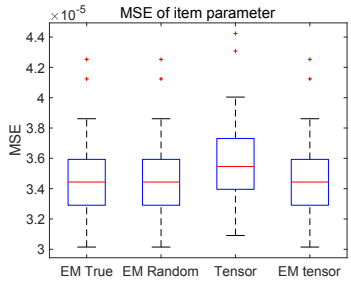


(b) MSE without EM-random

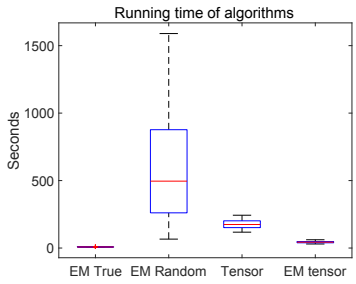


(c) Running time of the algorithms

Figure 30: $N = 20000, J = 100, L = 10$,item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

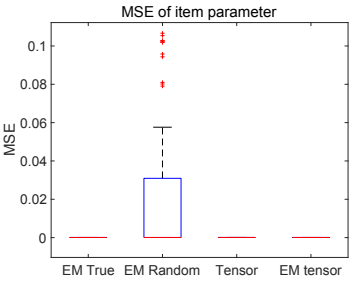


(a) MSE of item parameters

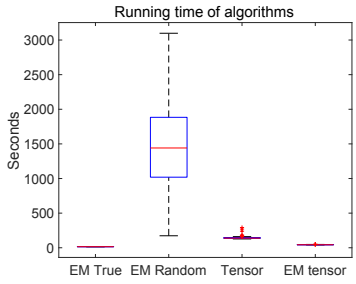


(b) Running time of the algorithms

Figure 31: $N = 20000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

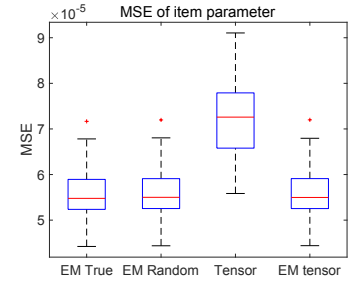


(a) MSE of item parameters

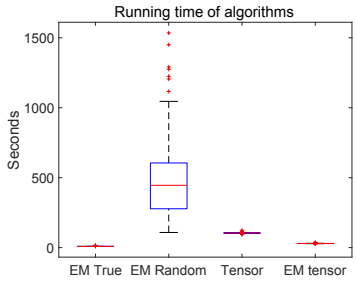


(b) Running time of the algorithms

Figure 32: $N = 20000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

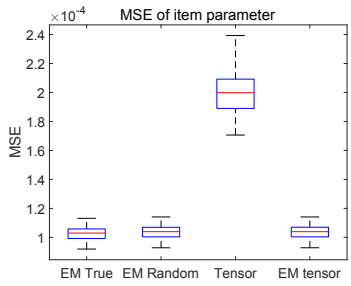


(a) MSE of item parameters

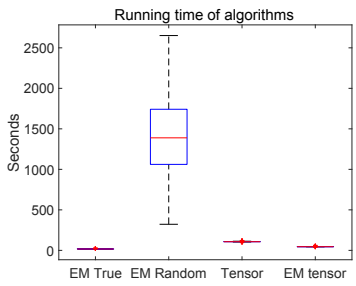


(b) Running time of the algorithms

Figure 33: $N = 20000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

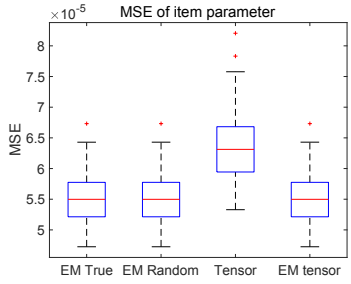


(a) MSE of item parameters

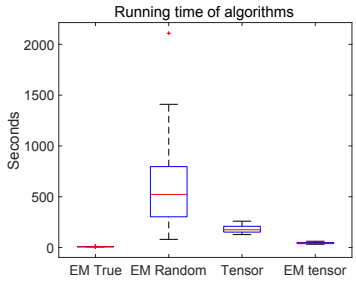


(b) Running time of the algorithms

Figure 34: $N = 20000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

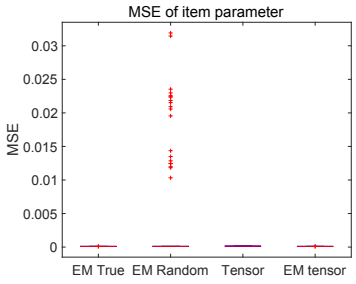


(a) MSE of item parameters

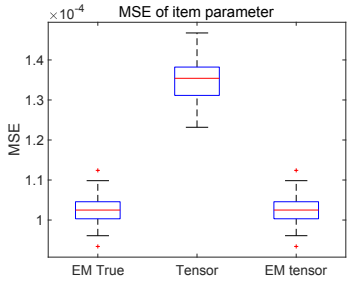


(b) Running time of the algorithms

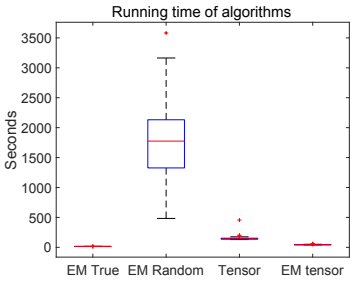
Figure 35: $N = 20000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters



(b) MSE without EM-random

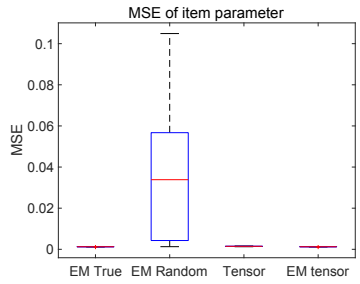


(c) Running time of the algorithms

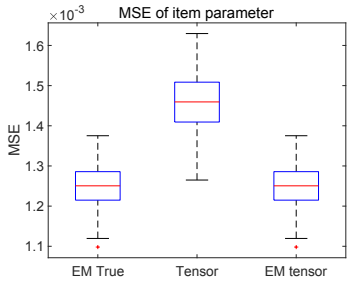
Figure 36: $N = 20000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

Fixed-effect LCM

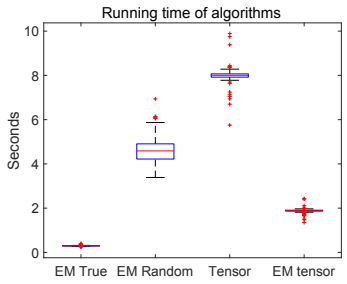
Then the simulation results of fixed-effect LCM are presented.



(a) MSE of item parameters

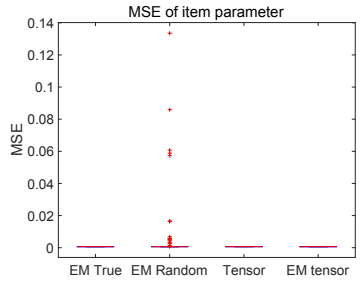


(b) MSE without EM-random

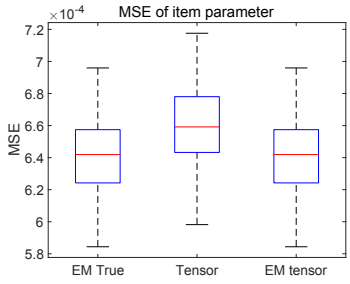


(c) Running time of the algorithms

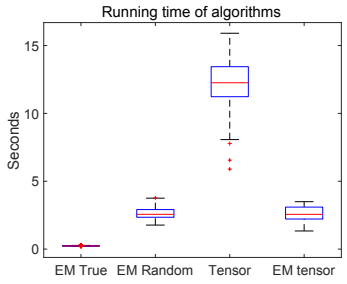
Figure 37: $N = 1000, J = 100, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

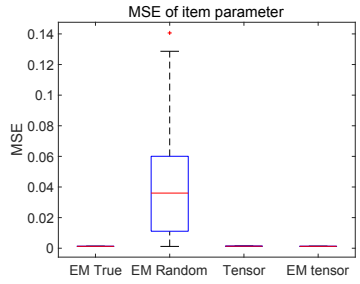


(b) MSE without EM-random

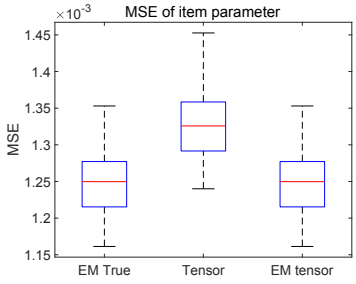


(c) Running time of the algorithms

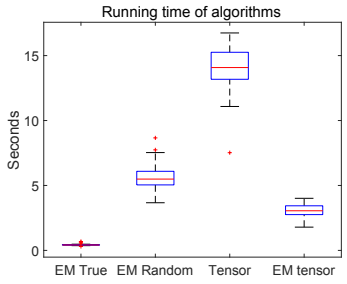
Figure 38: $N = 1000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

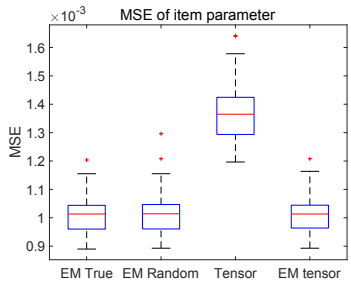


(b) MSE without EM-random

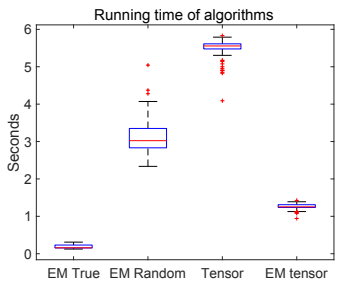


(c) Running time of the algorithms

Figure 39: $N = 1000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

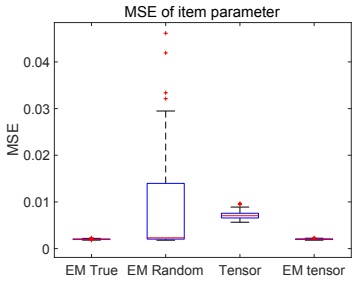


(a) MSE of item parameters

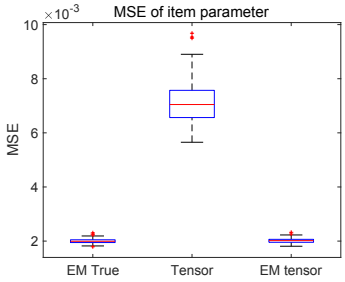


(b) Running time of the algorithms

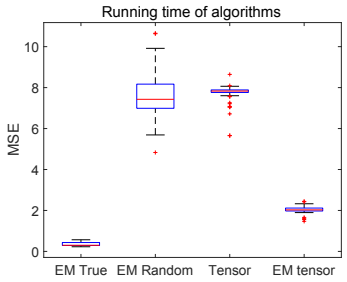
Figure 40: $N = 1000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

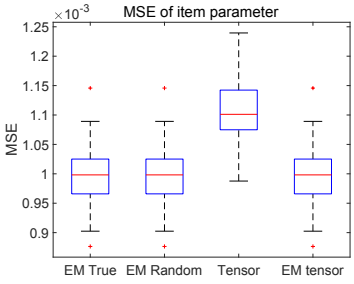


(b) MSE without EM-random

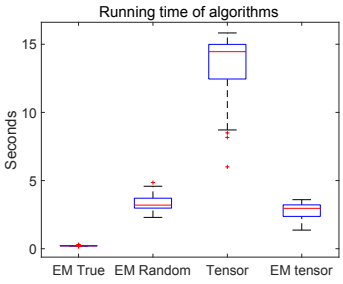


(c) Running time of the algorithms

Figure 41: $N = 1000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

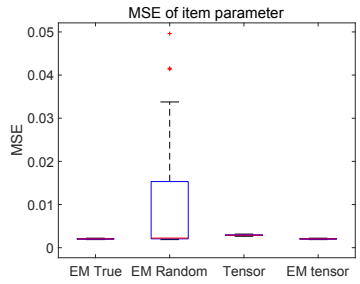


(a) MSE of item parameters

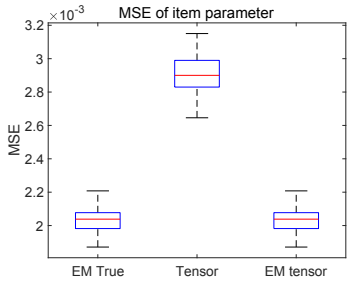


(b) Running time of the algorithms

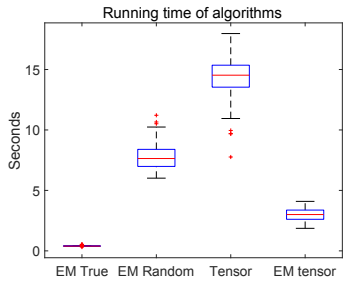
Figure 42: $N = 1000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

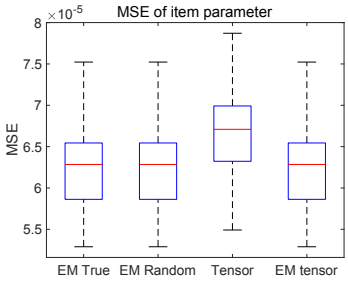


(b) MSE without EM-random

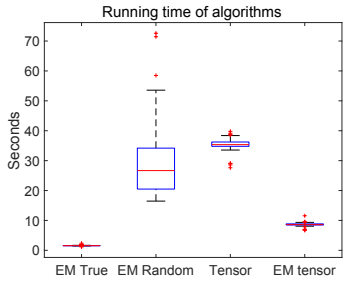


(c) Running time of the algorithms

Figure 43: $N = 1000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

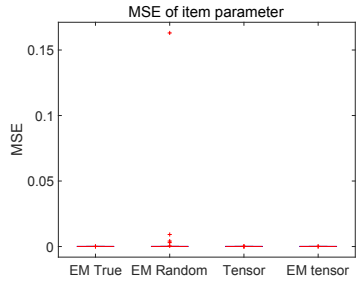


(a) MSE of item parameters

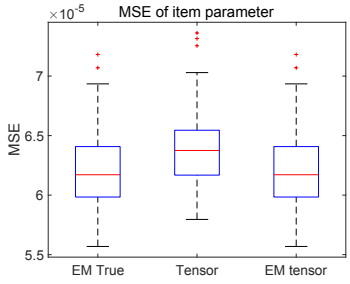


(b) Running time of the algorithms

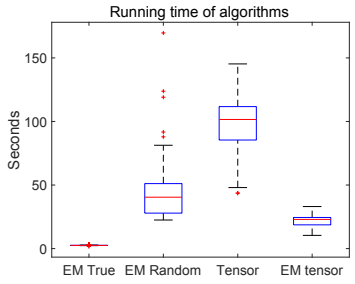
Figure 44: $N = 10000, J = 100, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

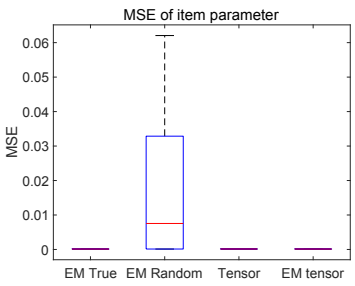


(b) MSE without EM-random

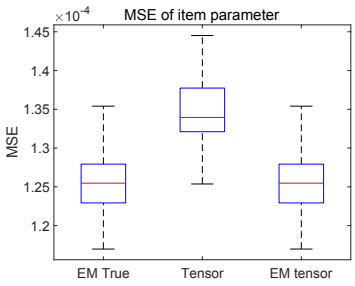


(c) Running time of the algorithms

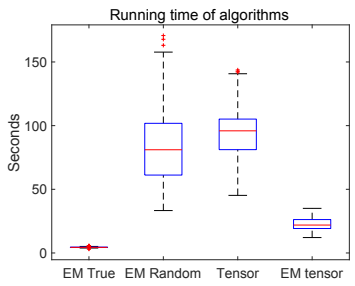
Figure 45: $N = 10000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

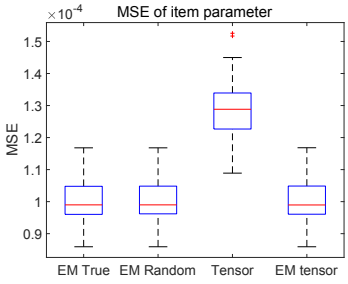


(b) MSE without EM-random

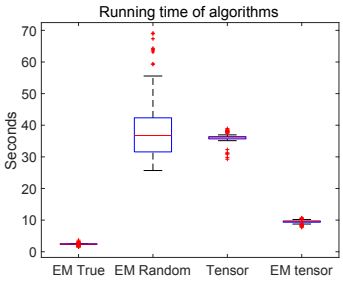


(c) Running time of the algorithms

Figure 46: $N = 10000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

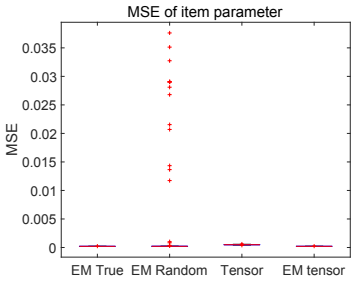


(a) MSE of item parameters

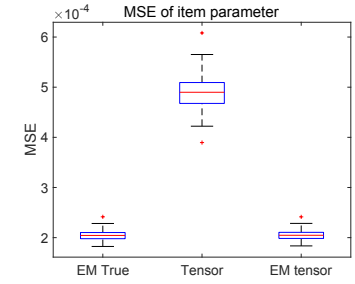


(b) Running time of the algorithms

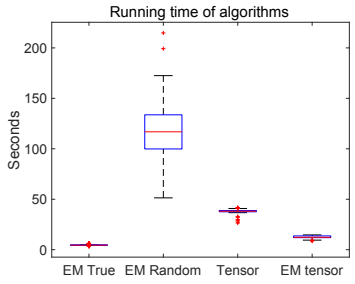
Figure 47: $N = 10000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

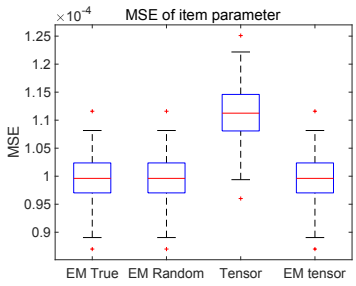


(b) MSE without EM-random

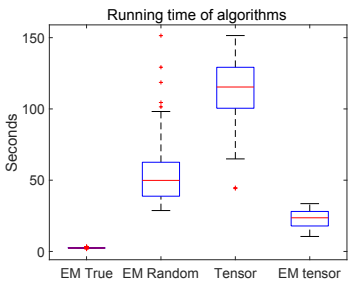


(c) Running time of the algorithms

Figure 48: $N = 10000, J = 100, L = 10$,item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

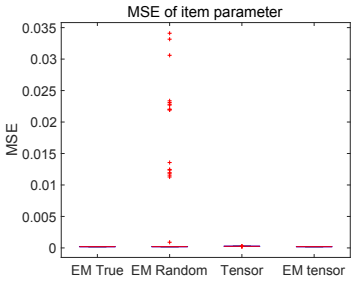


(a) MSE of item parameters

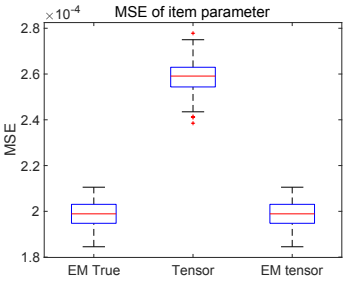


(b) Running time of the algorithms

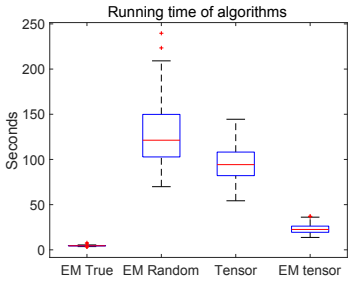
Figure 49: $N = 10000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

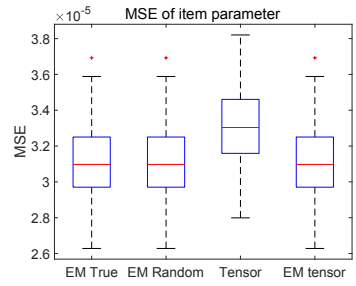


(b) MSE without EM-random

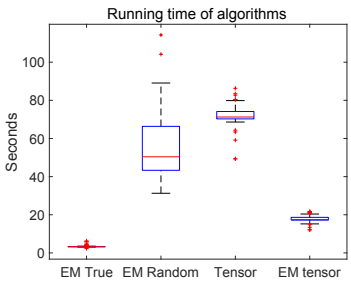


(c) Running time of the algorithms

Figure 50: $N = 10000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

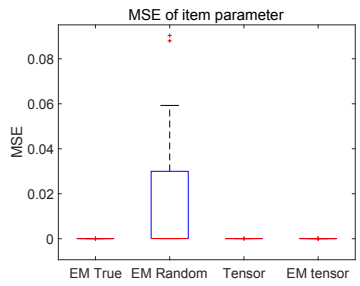


(a) MSE of item parameters

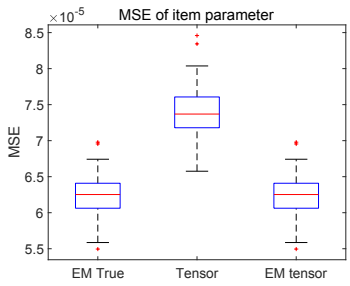


(b) Running time of the algorithms

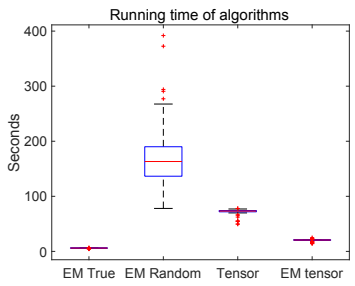
Figure 51: $N = 20000, J = 100, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

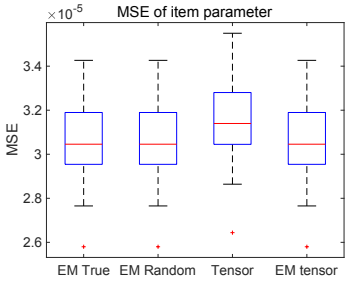


(b) MSE without EM-random

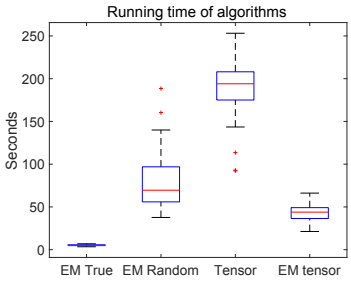


(c) MSE without EM-random

Figure 52: $N = 20000, J = 100, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

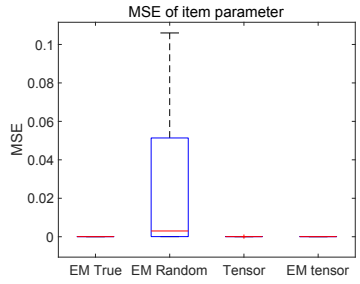


(a) MSE of item parameters

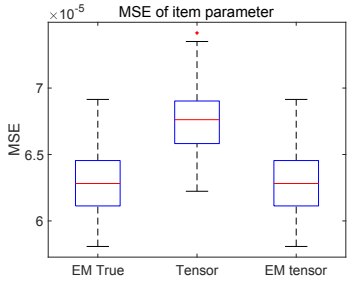


(b) Running time of the algorithms

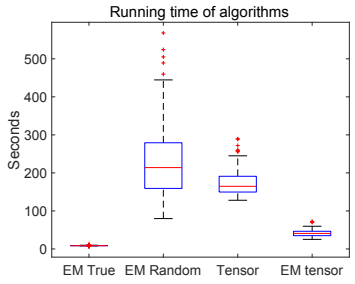
Figure 53: $N = 20000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$



(a) MSE of item parameters

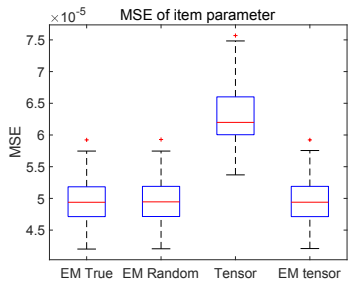


(b) MSE without EM-random

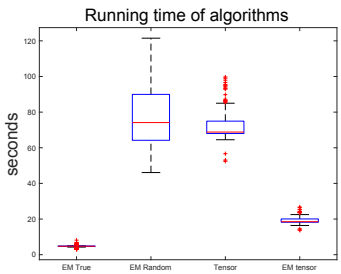


(c) Running time of the algorithms

Figure 54: $N = 20000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

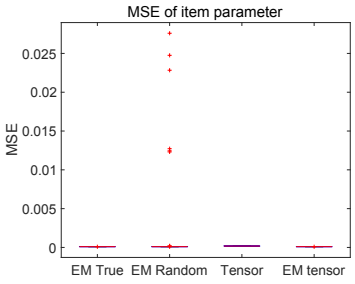


(a) MSE of item parameters

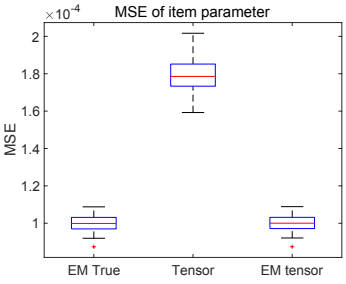


(b) Running time of the algorithms

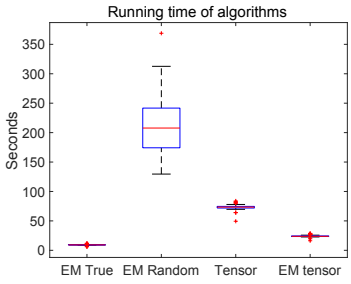
Figure 55: $N = 20000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters

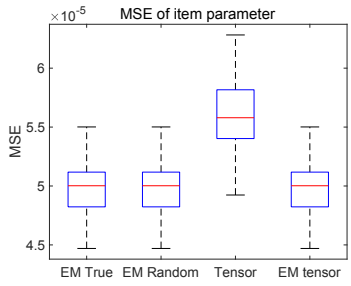


(b) MSE without EM-random

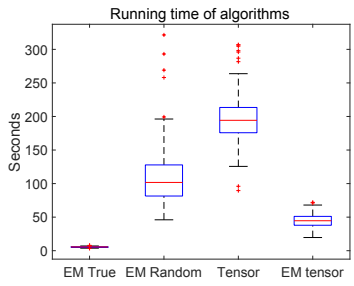


(c) Running time of the algorithms

Figure 56: $N = 20000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

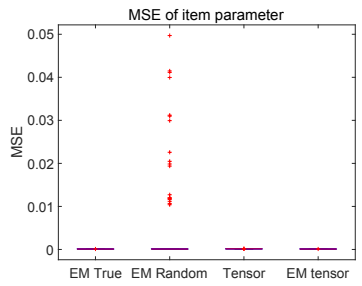


(a) MSE of item parameters

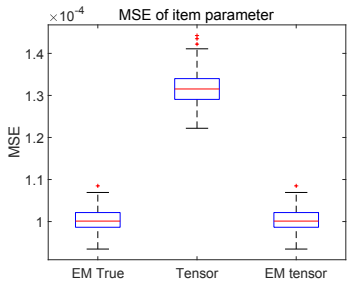


(b) Running time of the algorithms

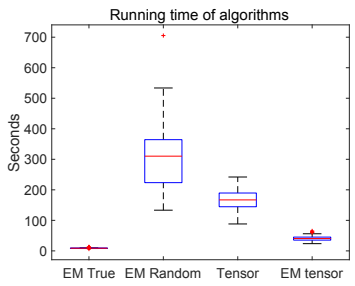
Figure 57: $N = 20000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters



(b) MSE without EM-random

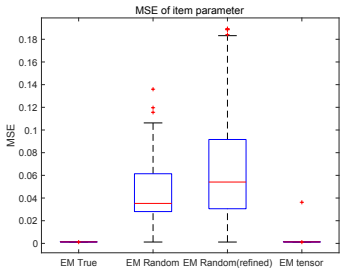


(c) Running time of the algorithms

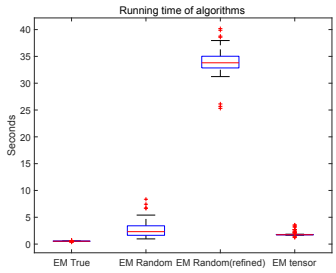
Figure 58: $N = 20000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

EM-random and Tensor-EM with same initializations

We then present the simulation results when EM-random and Tensor-EM are implemented with same initializations together with the "smarter" version of EM-random.

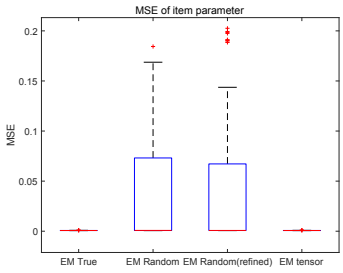


(a) MSE of item parameters

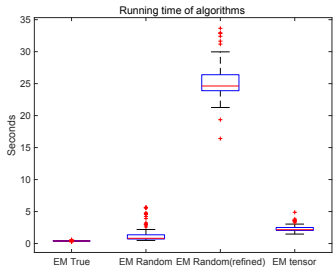


(b) Running time of the algorithms

Figure 59: $N = 1000, J = 100, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

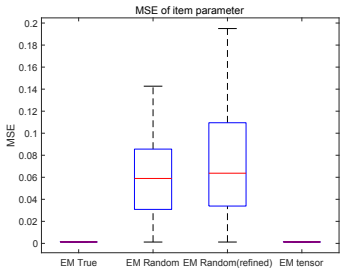


(a) MSE of item parameters

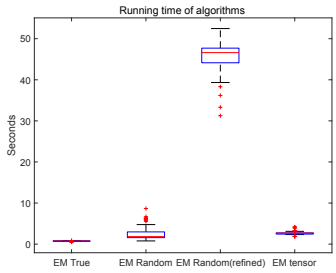


(b) Running time of the algorithms

Figure 60: $N = 1000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

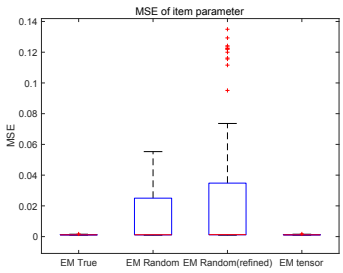


(a) MSE of item parameters

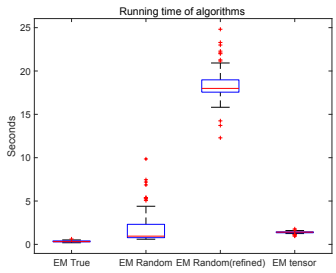


(b) Running time of the algorithms

Figure 61: $N = 1000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

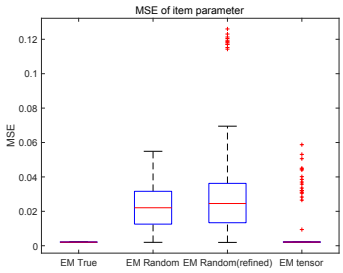


(a) MSE of item parameters

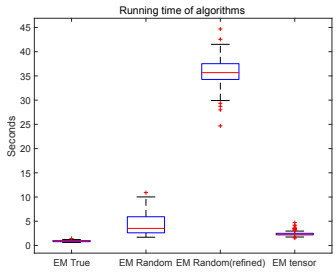


(b) Running time of the algorithms

Figure 62: $N = 1000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

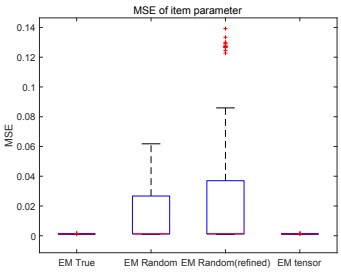


(a) MSE of item parameters

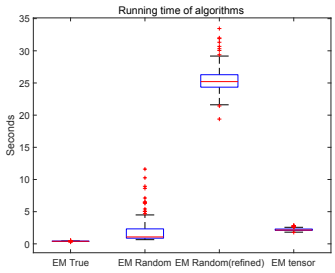


(b) Running time of the algorithms

Figure 63: $N = 1000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

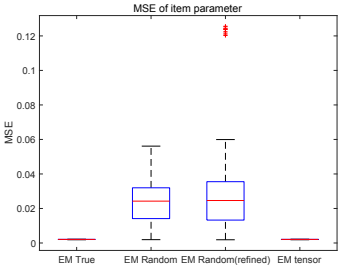


(a) MSE of item parameters

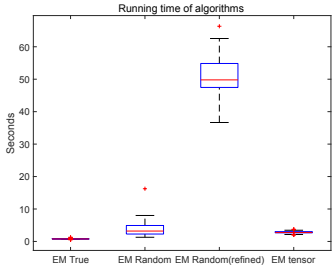


(b) Running time of the algorithms

Figure 64: $N = 1000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

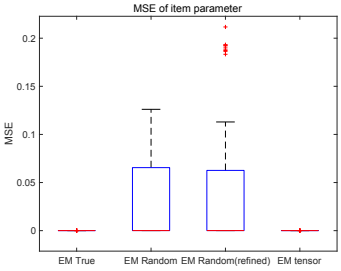


(a) MSE of item parameters

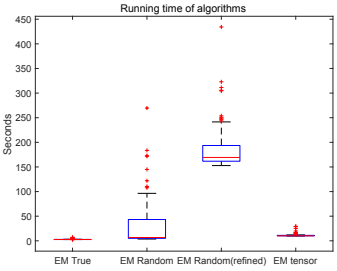


(b) Running time of the algorithms

Figure 65: $N = 1000, J = 200, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

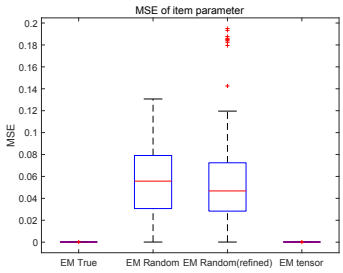


(a) MSE of item parameters

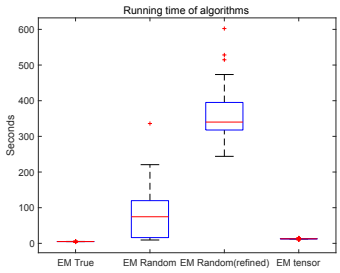


(b) Running time of the algorithms

Figure 66: $N = 10000, J = 100, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

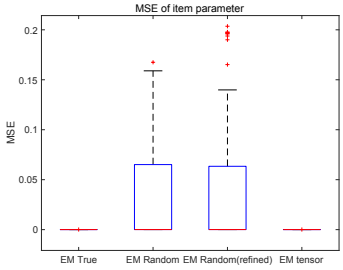


(a) MSE of item parameters

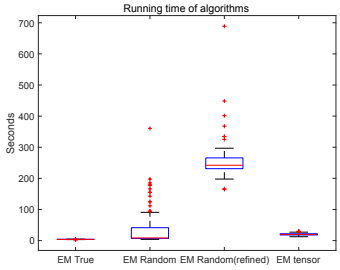


(b) Running time of the algorithms

Figure 67: $N = 10000, J = 100, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

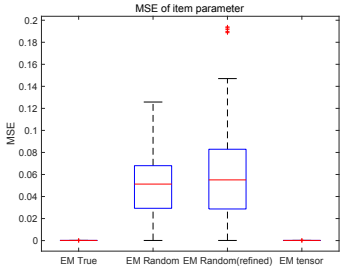


(a) MSE of item parameters

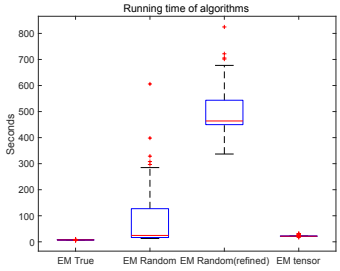


(b) Running time of the algorithms

Figure 68: $N = 10000, J = 200, L = 5$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

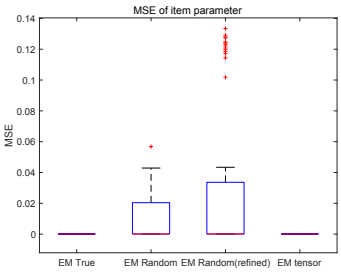


(a) MSE of item parameters

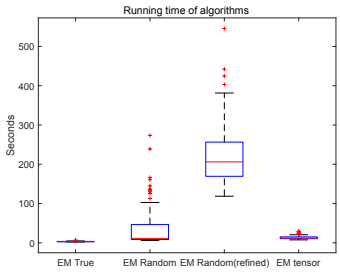


(b) Running time of the algorithms

Figure 69: $N = 10000, J = 200, L = 10$, item parameters $\in \{0.1, 0.2, 0.8, 0.9\}$

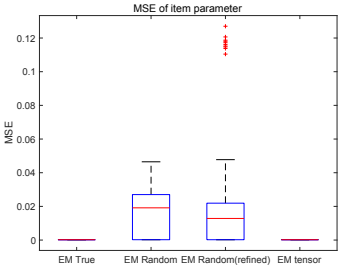


(a) MSE of item parameters

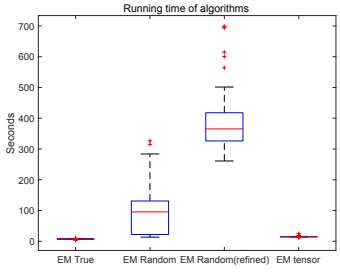


(b) Running time of the algorithms

Figure 70: $N = 10000, J = 100, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

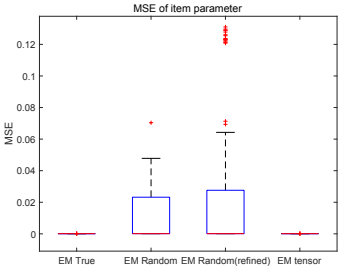


(a) MSE of item parameters

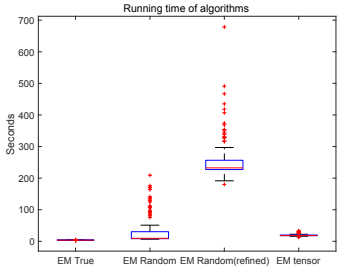


(b) Running time of the algorithms

Figure 71: $N = 10000, J = 100, L = 10$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$



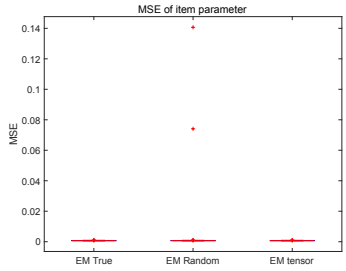
(a) MSE of item parameters



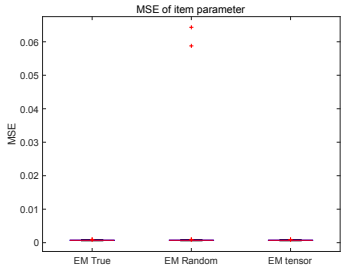
(b) Running time of the algorithms

Figure 72: $N = 10000, J = 200, L = 5$, item parameters $\in \{0.2, 0.4, 0.6, 0.8\}$

Simulations under local dependence

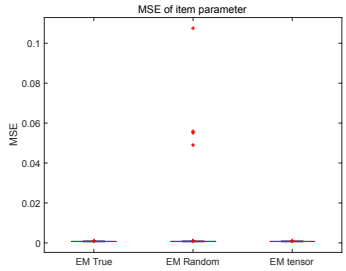


(a) MSE of item parameters $\rho = 0.3$

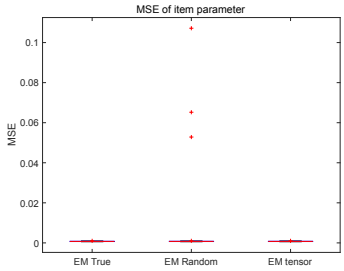


(b) MSE of item parameters $\rho = 0.7$

Figure 73: Random-effect LCM, $N = 1000, J = 100, L = 5, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$

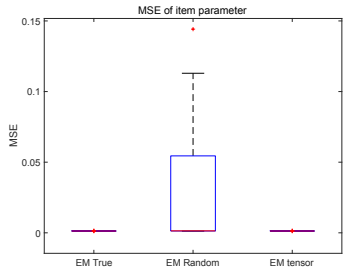


(a) MSE of item parameters $\rho = 0.3$

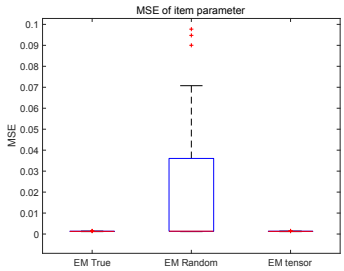


(b) MSE of item parameters $\rho = 0.7$

Figure 74: Random-effect LCM, $N = 1000, J = 200, L = 5, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$

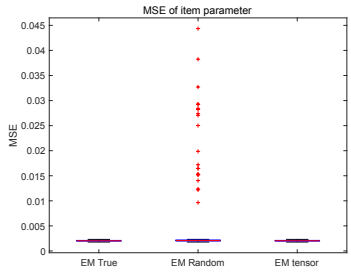


(a) MSE of item parameters $\rho = 0.3$

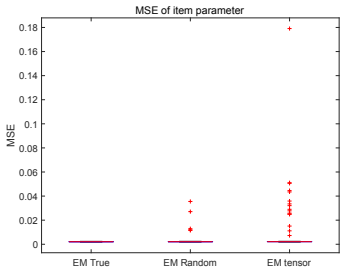


(b) MSE of item parameters $\rho = 0.7$

Figure 75: Random-effect LCM, $N = 1000, J = 200, L = 10, \theta_{j,a} \in \{0.1, 0.2, 0.8, 0.9\}$

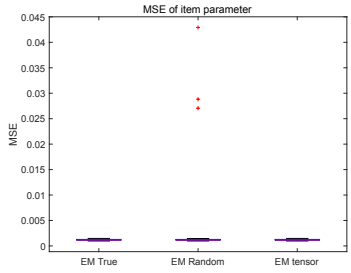


(a) MSE of item parameters $\rho = 0.3$

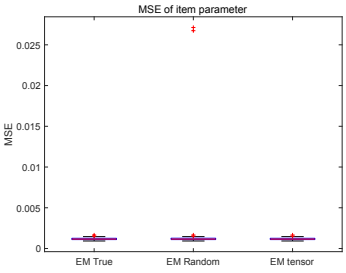


(b) MSE of item parameters $\rho = 0.7$

Figure 76: Random-effect LCM, $N = 1000, J = 100, L = 10, \theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$

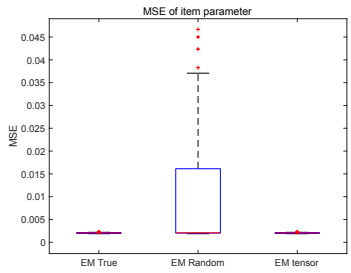


(a) MSE of item parameters $\rho = 0.3$

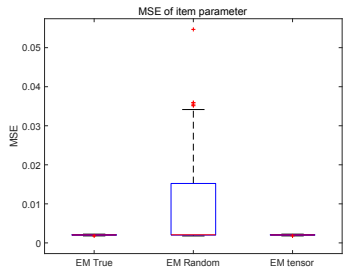


(b) MSE of item parameters $\rho = 0.7$

Figure 77: Random-effect LCM, $N = 1000, J = 200, L = 5, \theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$



(a) MSE of item parameters $\rho = 0.3$



(b) MSE of item parameters $\rho = 0.7$

Figure 78: Random-effect LCM, $N = 1000, J = 200, L = 10, \theta_{j,a} \in \{0.2, 0.4, 0.6, 0.8\}$