A Mixed Stochastic Approximation EM (MSAEM) Algorithm for the Estimation of the Four-Parameter Normal-Ogive Model

Xiangbin Meng[1] and Gongjun Xu[2]

[1] School of Mathematics and Statistics, Northeast Normal University, Changchun, China

[2] Department of Statistics, University of Michigan

Abstract

In recent years, the four-parameter model (4PM) has received increasing attention in item response theory. The purpose of this article is to provide more efficient and more reliable computational tools for fitting the 4PM. In particular, this article focuses on the four-parameter Normal Ogive (4PNO) model and develops efficient stochastic approximation Expectation Maximization (SAEM) algorithms to compute the marginalized maximum a posteriori (MMAP) estimator. First, a data augmentation scheme is used for the 4PNO model, which makes the complete data model be an exponential family, and then a basic SAEM algorithm is developed for the 4PNO model. Second, to overcome the drawback of the SAEM algorithm, we develop an improved SAEM algorithm for the 4PNO model, which is called the mixed SAEM (MSAEM). Results from simulation studies demonstrate that: (1) the MSAEM provides more accurate or comparable estimates as compared with the other estimation methods, while computationally more efficient; (2) the MSAEM is more robust to the choices of initial values and the priors for item parameters, which is a valuable property for practice use. Finally, a real data set is analyzed to show the good performance of the proposed methods.

Key words: four-parameter Normal Ogive model, stochastic approximation EM algorithm, marginalized maximum a posteriori estimation.

1 Introduction

The four-parameter model (4PM) in item response theory (IRT) was first provided by Barton and Lord (1981), in which an upper asymptote (slipping) parameter is introduced to model the uncertainty of a high-ability examinee missing an easy item. However, the 4PM had not been widely discussed for a long time, since the difficulties in parameter estimation and a lack of evidence supporting the need for it (Feuerstahler and Waller, 2014; Loken and Rulison, 2010). In recent years, researchers have shown renewed interest in the 4PM (Culpepper, 2016, 2017; Kern and Culpepper, 2020; Loken and Rulison, 2010; Meng et al., 2020; Waller and Feuerstahler, 2017). Several studies have verified that the presence of an upper asymptote is tenable in the situations of psychological assessment (Reise and Waller, 2003), the computerized adaptive testing (Liao et al., 2012; Rulison and Loken, 2009), and the large-scale low-staks assessment (Culpepper, 2017). Furthermore, various estimation methods have been developed for the 4PM. For instance, Loken and Rulison (2010) proposed a Bayesian estimation with the Markov chain Monte Carlo (MCMC) implemented using WinBUGS for the four-parameter Logistic (4PL) model; Feuerstahler and Waller (2014) employed the marginal maximum likelihood (MML) method with an Expectation Maximization (EM) algorithm as implemented in the R package "mirt" to estimate the 4PL model; Culpepper (2016) developed a Gibbs sampling algorithm for the Bayesian estimation of the four-parameter Normal Ogive (4PNO) model; Waller and Feuerstahler (2017) employed the EM algorithm to compute the marginalized maximum a posteriori (MMAP) estimation of the 4PL model by implementing the R package "mirt"; Meng et al. (2020) proposed an EM algorithm for the MMAP estimation of the 4PL model under the mixture modeling framework; Zhang et al. (2020a) proposed a Gibbs-slice sampling algorithm for estimating the 4PL model; Battauz (2020) proposed a regularization approach for estimating the 4PL model based on the inclusion of a penalty term in the log-likelihood function.

In these existing studies, the MCMC sampler and the EM algorithm are two main computational tools in the estimation of the 4PM model. For a Markov chain sampling-based method, the main drawback of the MCMC sampling is that performing exact inference generally requires all of the data to be processed at each iteration of the algorithm. For large datasets, the computational cost of the MCMC sampler can be prohibitive. In contrast, the EM algorithm is computationally more efficient, but it still has several major drawbacks. For instance, in the literature (Meng et al., 2020; Waller and Feuerstahler, 2017), the EM algorithm is mainly developed for the 4PL model, while there is no data augmentation method that can make the complete data model of the 4PL be a member of the exponential family, thus the convergence of the EM may not be guaranteed (Baker and Kim, 2004; Meng and Schilling, 1996). In addition, the E-step is often implemented by a numerical integration via fixed-point quadrature or other approximation methods, and the numerical approximation error can not be avoided (Meng and Schilling, 1996). Moreover, the M-step often does not yield a closed form solution and still needs numerical methods to solve the corresponding optimization problem, such as Newton-Raphson (NR) iteration; however, an issue with using the NR-type iteration or other numerical methods is that the starting values must be within a neighborhood of the true value, otherwise divergence or convergence to a suboptimal solution. Furthermore, the convergence of the whole EM algorithm also highly depends on the initial values and is often easy to fall into saddle points or does not converge.

To address these computational challenges in the estimation of the 4PM, this article focuses on the 4PNO model and develops computationally efficient stochastic approximation EM (SAEM) algorithms to compute the MMAP estimator. In the SAEM algorithm, the E-step of the EM algorithm is replaced by a simulation step and a stochastic approximation step, thus the numerical integration is avoided. In particular, Delyon et al. (1999) proved that the SAEM is able to converge to the maximum or local maximum point when the complete-data likelihood belongs to an exponential family, and the ill-convergence of the EM algorithm that the sequence of parameter estimate converges to saddle points is avoided by the stochastic approximation noise. The SAEM algorithm has been commonly used for the estimation of the non-linear mixed effects model (NLMEM), and it has been proved to be powerful for the Probit normal models (Allassonnière et al., 2010; Delyon et al., 1999; Kuhn and Lavielle, 2004; Lavielle and Mbogning, 2014). Furthermore, the SAEM has been used for computing the estimators of some item response models (Camilli and Fox, 2015; Camilli and Geis, 2019). Inspired by these studies, the first contribution of this article is to develop a SAEM algorithm to compute the MMAP estimator of the 4PNO model. Here an important step in our derivation is that the 4PNO model is reformulated to be a mixture model, and a data augmentation scheme is used for it. We show that the corresponding complete-data likelihood belongs to an exponential family, and then derive the sufficient statistics to compute the MMAP estimator, making the implementation of the SAEM algorithm simplified.

However, the SAEM algorithm is likely to be unstable and may produce poor estimators for the mixture of NLMEM under some situations: small sample size, overlap between mixture components, heteroscedastic models, etc. Aiming to address these issues, focusing on NLMEM, Lavielle and Mbogning (2014) developed an improved SAEM algorithm, in which the simulation of the latent categorical or group variable is avoided and is replaced by a conditional expectation. And they verified that, for estimating the mixture of NLMEM, this new algorithm can provide a more accurate estimation and achieve strong robustness to the initial values.

As mentioned above, the 4PNO model can be viewed as a mixture model, therefore, motivated by the study of Lavielle and Mbogning (2014), taking the SAEM as the stepping stone, we propose an efficient mixed SAEM (MSAEM) algorithm for the 4PNO model. In the MSAEM algorithm, the simulation of the latent group variables, that is required in the SAEM, is avoided and replaced by their conditional expectations given the subjects' abilities. Thus only the latent abilities need to be randomly generated in the simulation step, and the stochastic approximation is implemented on the conditional expectation function. The obtained results from the simulation studies demonstrate that, for the 4PNO model, the MSAEM algorithm performs substantially better than or comparable to the SAEM algorithm, the Monte Carlo EM (MCEM) algorithm and the MCMC sampler of Culpepper (2016), while computationally more efficient. Moreover, the estimates from the MSAEM show great robustness to the initial values and the choices of the priors, which is a valuable property for practical use.

The rest of the article is organized as follows. Section 2 presents the 4PNO model under a hierarchical modeling framework and gives the exponential form of the complete data likelihood, which is very important for implementing the SAEM and the MSAEM algorithms. Section 3 is the major part of this article, in which we firstly present the SAEM procedure, and then develop the MSAEM algorithm for the MMAP estimation of the 4PNO model. Section 4 reports simulation studies that were constructed to evaluate the performance of the SAEM and MSAEM algorithms for estimating the 4PNO model. Section 5 presents an application of the 4PNO to an empirical dataset. Finally, we provide further discussions on some future research directions in Section 6, and additional simulation results are reported in the Appendix.

2 MMAP Estimation of the 4PNO Model with a Data Augmentation Scheme

In this section, we first present the 4PNO model as a hierarchical model using a data augmentation method, and then introduce the MMAP estimation of the 4PNO.

2.1 A Data Augmentation Scheme for the 4PNO Model

Let $i = 1, ..., N$ and $j = 1, ..., M$ index test takers and items, and U_{ij} (with realization u_{ij}) denote the dichotomous response variable of examinee i to item j, where $U_{ij} = 1$ denotes the correct response and $U_{ij} = 0$ otherwise.

Following the definition of 4PM, the item response function (IRF) of the 4PNO model is given by,

$$P_i(\theta_j) = P(U_{ij} = 1|\theta_i, \boldsymbol{\xi}_j) = c_j + (d_j - c_j)\Phi(a_j\theta_i + b_j), \tag{1}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, $\theta_i \in (-\infty, +\infty)$ is the latent trait or ability of examinee i, and $\boldsymbol{\xi}_j = \{a_j, b_j, c_j, d_j\}$ is the item parameter set of item j, with $a_j \in (0, +\infty)$, $b_j \in (-\infty, +\infty)$, $c_j \in [0, 1)$, and $d_j \in (c_j, 1]$ being the slope, intercept, lower asymptote, and upper asymptote parameters, respectively. Specifically, c_j and d_j represent the minimum and maximum probabilities for a correct response, respectively.

In the following, a data augmentation scheme is applied to the 4PNO model, which makes the complete-data distribution belong to an exponential family.

2.1.1 A Hierarchical Modeling Framework of the 4PNO Model

First, from Equation 1, an equivalent form of the 4PNO IRF is,

$$P(U_{ij}=1|\theta_i,\boldsymbol{\xi}_j)=c_j\times[1-\Phi(a_j\theta_i+b_j)]+d_j\times\Phi(a_j\theta_i+b_j), \tag{2}$$

which implies the 4PNO is a mixture of two Bernoulli distributions with the latent categorical probability $\Phi(a_j\theta_i+b_j)$. Then, we introduce a latent binary indicator variable W_{ij} and define,

$$\begin{aligned} P(U_{ij}=1|W_{ij}=w_{ij},\boldsymbol{\xi}_j,\theta_i) &= c_j^{1-w_{ij}}d_j^{w_{ij}}, & (3)\\ P(W_{ij}=1|\boldsymbol{\xi}_j,\theta_i) &= \Phi(a_j\theta_i+b_j), & (4)\end{aligned}$$

where $w_{ij}\in\{0,1\}$ denotes the observation of W_{ij}. Taking the law of total probability, the 4PNO IRF in Equation 2 can be written in the sum of $P(U_{ij}=1|W_{ij}=w_{ij},\boldsymbol{\xi}_j,\theta_i)P(W_{ij}=w_{ij}|\boldsymbol{\xi}_j,\theta_i)$ over $w_{ij}\in\{0,1\}$. The addition of the auxiliary variable W_{ij} was first proposed by Béguin and Glas (2001) in Bayesian estimation of the 3PNO, and it has been extended for handing the other three- and four-parameter IRT models (Culpepper, 2016; Guo and Zheng, 2019; Meng et al., 2020; von Davier, 2009). Following Béguin and Glas (2001), W_{ij} is defined as,

$$\begin{cases} W_{ij}=1, & \text{if examinee } i \text{ is able to correctly answer item } j;\\ W_{ij}=0, & \text{otherwise.}\end{cases} \tag{5}$$

Then, based on the conditional probability given in Equation 3, c_j and d_j can be interpreted as the guessing and slipping parameters.

Note that Equation 4 is a probit model for W_{ij} on $a_j\theta_i+b_j$, which is equivalent to the following model on a normal augmented variable Z_{ij},

$$\begin{aligned} W_{ij} &= I_{(Z_{ij}>0)}, & (6)\\ Z_{ij}|\boldsymbol{\xi}_j,\theta_i &\sim N(a_j\theta_i+b_j,1), & (7)\end{aligned}$$

where I_A denotes the indicator function of a set A.

In addition, the examinees are assumed to be a sample from a population where the latent trait or ability follows a normal distribution, and the examinees' latent traits or abilities can be viewed as missing data. In the estimation of the IRT model, it generally assumes that

$$\theta_i \overset{\text{i.i.d.}}{\sim} N(0,1) \text{ for } i = 1, \ldots, N, \tag{8}$$

which helps establish a scale for the latent trait.

Finally, taking the data augmentation approaches given in Equations 3-8, we have the following hierarchical modeling formulation of the 4PNO,

$$\begin{aligned}
U_{ij}|W_{ij}, \boldsymbol{\xi}_j, \theta_i &\overset{\text{independent}}{\sim} \begin{cases} \text{Bernoulli}(d_j), & W_{ij} = 1 \\ \text{Bernoulli}(c_j), & W_{ij} = 0 \end{cases} \\
W_{ij} &= I_{(Z_{ij}>0)} \\
Z_{ij}|\boldsymbol{\xi}_j, \theta_i &\overset{\text{independent}}{\sim} N(a_j\theta_i + b_j, 1), \\
\theta_i &\overset{\text{i.i.d}}{\sim} N(0,1),
\end{aligned}$$

for $i = 1, \ldots, N, j = 1, \ldots, M$. The distribution of the complete-data $(u_{ij}, w_{ij}, z_{ij}, \theta_i)$ can be written as,

$$\begin{aligned}
f(u_{ij}, w_{ij}, z_{ij}, \theta_i \,|\, \boldsymbol{\xi}_j) \;\propto\; & d_j^{w_{ij}u_{ij}}(1-d_j)^{w_{ij}(1-u_{ij})}c_j^{(1-w_{ij})u_{ij}}(1-c_j)^{(1-w_{ij})(1-u_{ij})} \\
& \times\phi(z_{ij} - a_j\theta_i - b_j)\left[I_{(z_{ij}>0)}I_{(w_{ij}=1)} + I_{(z_{ij}\leq 0)}I_{(w_{ij}=0)}\right] \\
& \times\phi(\theta_i),
\end{aligned} \tag{9}$$

where $\phi(\cdot)$ is the standard normal density function. Note that, this data augmentation scheme has been used by Culpepper (2016) to develop a Gibbs sampler for the 4PNO. It can be seen that $f(u_{ij}, w_{ij}, z_{ij}, \theta_i \,|\, \boldsymbol{\xi}_j)$ is the product of Bernoulli distribution and normal density distribution, thus it belongs to the exponential family, which is ideal for developing the SAEM and MSAEM algorithms to estimate the 4PNO model in this work. In what follows, the exponential family form of the complete-data likelihood and the corresponding sufficient statistics are given.

2.1.2 Complete-Data Likelihood and Sufficient Statistics

We first introduce some notation. Let $\mathbf{u}_{i\cdot} = (u_{i1}, ..., u_{iM})$ denote the observed response vector of examinee i, $\mathbf{u}_{\cdot j} = (u_{1j}, ..., u_{Nj})'$ denote the observed response vector of item j, and $\mathbf{u} = (\mathbf{u}_{\cdot 1}, ..., \mathbf{u}_{\cdot N})$ denotes the observed response data from a test. Let $\mathbf{W}_{\cdot j} = (W_{1j}, ..., W_{Nj})'$ and $\mathbf{Z}_{\cdot j} = (Z_{1j}, ..., Z_{Nj})'$ denote the vector of the latent variables for item j, $\mathbf{w}_{\cdot j} = (w_{1j}, ..., w_{Nj})'$ and $\mathbf{z}_{\cdot j} = (z_{1j}, ..., z_{Nj})'$ denote the observations of $\mathbf{W}_{\cdot j}$ and $\mathbf{Z}_{\cdot j}$; $\mathbf{W} = (\mathbf{W}_{\cdot 1}, ..., \mathbf{W}_{\cdot M})$ and $\mathbf{Z} = (\mathbf{Z}_{\cdot 1}, ..., \mathbf{Z}_{\cdot M})$ denote the matrix of latent response variables for a test, $\mathbf{w} = (\mathbf{w}_{\cdot 1}, ..., \mathbf{w}_{\cdot M})$ and $\mathbf{z} = (\mathbf{z}_{\cdot 1}, ..., \mathbf{z}_{\cdot M})$ denote the observation of $\mathbf{W}$ and $\mathbf{Z}$. Let $\boldsymbol{\theta} = (\theta_1, ..., \theta_N)'$ be the ability parameter vector of N examinees. Finally, let $\mathbf{x} = (\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\theta})$ denote the complete-data, and $(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta})$ denote the missing data.

From Equation 9, the complete-data likelihood of $\boldsymbol{\xi}_j$ can be written as

$$\begin{aligned} L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j) &= \prod_{i=1}^{N} d_j^{w_{ij}u_{ij}} (1-d_j)^{w_{ij}(1-u_{ij})} c_j^{(1-w_{ij})u_{ij}} (1-c_j)^{(1-w_{ij})(1-u_{ij})} \phi(z_{ij} - a_j\theta_i - b_j) \\ &\quad \times [I_{(z_{ij}>0)} I_{(w_{ij}=1)} + I_{(z_{ij}\le 0)} I_{(w_{ij}=0)}] \phi(\theta_i), \end{aligned} \tag{10}$$

where $\mathbf{x}_{\cdot j} = (\mathbf{u}_{\cdot j}, \mathbf{w}_{\cdot j}, \mathbf{z}_{\cdot j}, \boldsymbol{\theta})$ is the complete data of item j. Further, we have

$$\begin{aligned} L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j) &= \exp\{\ln L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j)\} \\ &\propto \exp\Bigg\{ -\frac{1}{2}(\mathbf{z}'_{\cdot j}\mathbf{z}_{\cdot j} + \boldsymbol{\theta}\boldsymbol{\theta}') + N \ln(1-c_j) + \ln\frac{d_j}{1-d_j}\sum_{i=1}^{N} w_{ij}u_{ij} \\ &\qquad + \ln\frac{c_j}{1-c_j}\sum_{i=1}^{N}(1-w_{ij})u_{ij} + \ln\frac{1-d_j}{1-c_j}\sum_{i=1}^{N} w_{ij} \\ &\qquad + (a_j, b_j)\boldsymbol{\Lambda}'\mathbf{z}_{\cdot j} - \frac{1}{2}(a_j, b_j)\boldsymbol{\Lambda}'\boldsymbol{\Lambda}(a_j, b_j)' \Bigg\} \end{aligned} \tag{11}$$

where $\boldsymbol{\Lambda} = (\boldsymbol{\theta}, \mathbf{1}_N)$, and $\mathbf{1}_N$ denotes the $N \times 1$ vector of 1s. Equation 11 is the exponential family form of $L(\mathbf{x}_{.j} \mid \boldsymbol{\xi}_j)$, and the sufficient statistics of $\boldsymbol{\xi}_j$ are

$$S(\mathbf{x}_{.j}) = (S_1(\mathbf{x}_{.j}), S_2(\mathbf{x}_{.j}), S_3(\mathbf{x}_{.j}), S_4(\mathbf{x}_{.j}), S_5(\mathbf{x}_{.j})) \tag{12}$$

where

$$\begin{aligned}
S_1(\mathbf{x}_{.j}) &= \sum_{i=1}^{N} w_{ij}u_{ij}, & (13)\\
S_2(\mathbf{x}_{.j}) &= \sum_{i=1}^{N} w_{ij}, & (14)\\
S_3(\mathbf{x}_{.j}) &= \sum_{i=1}^{N} (1-w_{ij})u_{ij}, & (15)\\
S_4(\mathbf{x}_{.j}) &= \boldsymbol{\Lambda}'\boldsymbol{\Lambda}, & (16)\\
S_5(\mathbf{x}_{.j}) &= \boldsymbol{\Lambda}'\mathbf{z}_{.j}. & (17)
\end{aligned}$$

2.2 MMAP Estimation for the 4PNO Model

2.2.1 Priors

Following Culpepper (2016), for $j = 1, ..., M$, the prior for (a_j, b_j) is a truncated bivariate normal distribution with the constraint of $a_j \geq 0$,

$$f(a_j, b_j|\mu_0, \Sigma_0) \propto N_2(\mu_0, \Sigma_0)I_{(a_j \geq 0)}, \tag{18}$$

where μ_0 is a 2×1 mean vector and Σ_0 is a 2×2 covariance matrix. And the prior for (c_j, d_j) is a bivariate Beta distribution with the constraint of $d_j \geq c_j$,

$$f(c_j, d_j|\alpha_c, \beta_c, \alpha_d, \beta_d) \propto c_j^{\alpha_c - 1}(1-c_j)^{\beta_c - 1} d_j^{\alpha_d - 1}(1-d_j)^{\beta_d - 1} I_{(1 \geq d_j \geq c_j \geq 0)}. \tag{19}$$

The priors of (a_j, b_j) and (c_j, d_j) are assumed to be independent, then the joint prior of $\boldsymbol{\xi}_j$ can be written as

$$f(\boldsymbol{\xi}_j|\Omega) = f(a_j, b_j|\mu_0, \Sigma_0)f(c_j, d_j|\alpha_c, \beta_c, \alpha_d, \beta_d), \tag{20}$$

where $\Omega := \{\mu_0, \Sigma_0, \alpha_c, \beta_c, \alpha_d, \beta_d\}$ is the set of hyper-parameters. The above two priors are conjugate to the complete-data likelihood $L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j)$, which is mathematically convenient for posterior inference.

Finally, it should be emphasized that, the noninformative uniform prior density for (a_j, b_j), $f(a_j, b_j|\mu_0, \Sigma_0) \propto I_{(a_j \geq 0)}$, can be obtained in the limit as $|\Sigma_0^{-1}| \to 0$ ($|\Sigma_0^{-1}|$ is called the prior precision) (Gelman et al., 2013). Moreover, in the limit of $|\Sigma_0^{-1}| = 0$ (infinite prior variance), the prior mean μ_0 is irrelevant, that is, it can be specified as any value. And the prior for (c_j, d_j), with the setting of $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$, is the noninformative uniform prior.

2.2.2 MMAP Estimation

For $j = 1, ..., M$, the marginalized posterior distribution of $\boldsymbol{\xi}_j$ can be calculated by

$$\begin{aligned} f(\boldsymbol{\xi}_j|\mathbf{u}_{\cdot j}, \Omega) &= \int\int \frac{L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j) f\left(\boldsymbol{\xi}_j \left|\Omega\right.\right)}{L(\mathbf{u}_{\cdot j}|\Omega)} d\mathbf{z}_{\cdot j} d\boldsymbol{\theta} \\ &\propto \int\int L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j) f\left(\boldsymbol{\xi}_j \left|\Omega\right.\right) d\mathbf{z}_{\cdot j} d\boldsymbol{\theta}, \end{aligned} \tag{21}$$

where $L(\mathbf{x}_{\cdot j}|\boldsymbol{\xi}_j)$ and $f\left(\boldsymbol{\xi}_j \left|\Omega\right.\right)$ given in Equations 10 and 20. And the mode of $f(\boldsymbol{\xi}_j|\mathbf{u}_{\cdot j}, \Omega)$ is defined as the MMAP estimation of $\boldsymbol{\xi}_j$, which is

$$\hat{\boldsymbol{\xi}}_j = \underset{\boldsymbol{\xi}_j \in \boldsymbol{\Theta}_{\boldsymbol{\xi}}}{\arg\max} f(\boldsymbol{\xi}_j|\mathbf{u}_{\cdot j}, \Omega) = \underset{\boldsymbol{\xi}_j \in \boldsymbol{\Theta}_{\boldsymbol{\xi}}}{\arg\max} \int\int L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j) f\left(\boldsymbol{\xi}_j \left|\Omega\right.\right) d\mathbf{z}_{\cdot j} d\boldsymbol{\theta}. \tag{22}$$

Based on the MMAP estimation of the 4PNO defined in Equation 22, the computation of

$$\underset{\boldsymbol{\xi}_j \in \Theta_{\boldsymbol{\xi}}}{\arg\max} L(\mathbf{x}_{\cdot j} \mid \boldsymbol{\xi}_j) f(\boldsymbol{\xi}_j|\Omega), \tag{23}$$

for $j = 1, ..., M$, which is the posterior mode of $\boldsymbol{\xi}_j$ under complete-data likelihood, is the Maximization Step (M-Step) of the EM-type algorithms (including the basic EM, the MCEM, the SAEM and the MSAEM). As given below, benefiting from the conjugate prior and the complete-data likelihood being in the exponential family, the solutions to Equation 23 are available in closed forms and more importantly, they can be formulated as functions of the sufficient statistics $S(\mathbf{x}_{.j})$.

2.2.3 Posterior Mode of $\boldsymbol{\xi}_j$ under Complete-Data Model

Let $f(\boldsymbol{\xi}_j|\mathbf{x}_{.j}, \Omega)$ denote the posterior distribution of $\boldsymbol{\xi}_j$ given $\mathbf{x}_{.j}$, and we have

$$f(\boldsymbol{\xi}_j|\mathbf{x}_{.j}, \Omega) \propto L(\mathbf{x}_{.j} \mid \boldsymbol{\xi}_j) f(\boldsymbol{\xi}_j|\Omega),$$

in the case of the complete data $\mathbf{x}_{.j}$ is observed. Then,

$$\underset{\boldsymbol{\xi}_j \in \Theta_{\boldsymbol{\xi}}}{\arg\max} L(\mathbf{x}_{.j} \mid \boldsymbol{\xi}_j) f(\boldsymbol{\xi}_j|\Omega) = \underset{\boldsymbol{\xi}_j \in \Theta_{\boldsymbol{\xi}}}{\arg\max} f(\boldsymbol{\xi}_j|\mathbf{x}_{.j}, \Omega).$$

As the priors given in Equations 18 and 19 are conjugate for $L(\mathbf{x}_{.j} \mid \boldsymbol{\xi}_j)$, the following results can be obtained. First, the posterior distribution for (a_j, b_j) is a truncated bivariate normal distribution with the constraint of $a_j \geq 0$,

$$f(a_j, b_j|\mathbf{x}_{.j}, \mu_0, \Sigma_0) \propto N_2(a_j, b_j|\mu_{(a_j,b_j)}, \Sigma_{(a_j,b_j)}) I_{(a_j \geq 0)},$$

where

$$\begin{aligned} \mu_{(a_j,b_j)} &= (S_4(\mathbf{x}_{.j}) + \Sigma_0^{-1})^{-1}(S_5(\mathbf{x}_{.j})) + \Sigma_0^{-1}\mu_0), \\ \Sigma_{(a_j,b_j)} &= (S_4(\mathbf{x}_{.j}) + \Sigma_0^{-1})^{-1}, \end{aligned}$$

which are the posterior mean vector and covariance matrix. Then, the maximum of the posterior of a_j and b_j is

$$\hat{a}_j(s_j) = \mu_{a_j} \times I_{(\mu_{a_j} \geq 0)}, \tag{24}$$

$$\hat{b}_j(s_j) = \mu_{b_j}, \tag{25}$$

where μ_{a_j} and μ_{b_j} are the first and the second element of $\mu_{(a_j,b_j)}$, and s_j denotes the values of $S(\mathbf{x}_{.j})$. It can be seen that, when $\Sigma_0^{-1} = \mathbf{0}_{2\times2}$, where $\mathbf{0}_{2\times2}$ denotes the 2×2 zero matrix, which is a noninformative prior, the obtained MAP estimation of a_j and b_j in Equations 24 and 25 are reduced to the ML estimation.

Second, the posterior distribution for (c_j, d_j) is

$$f(c_j, d_j|\mathbf{x}_{\cdot j}, \alpha_c, \beta_c, \alpha_d, \beta_d) \propto c_j^{\alpha_{c_j}-1}(1-c_j)^{\beta_{c_j}-1}d_j^{\alpha_{d_j}-1}(1-d_j)^{\beta_{d_j}-1}I_{(1\geq d_j\geq c_j\geq 0)},$$

where

$$\alpha_{c_j} = \alpha_c + S_3(\mathbf{x}_{\cdot j}), \qquad \beta_{c_j} = \beta_c + N - S_2(\mathbf{x}_{\cdot j}) - S_3(\mathbf{x}_{\cdot j}),$$
$$\alpha_{d_j} = \alpha_d + S_1(\mathbf{x}_{\cdot j}), \qquad \beta_{d_j} = \beta_d + S_2(\mathbf{x}_j) - S_1(\mathbf{x}_{\cdot j}).$$

Noting that the maximum of $p(c_j, d_j|\mathbf{x}_{\cdot j}, \alpha_c, \beta_c, \alpha_d, \beta_d)$ is a convex optimization problem with constraints, and using the Lagrange multiplier method, it can be obtained that when

$$\frac{\alpha_{c_j}-1}{\alpha_{c_j}+\beta_{c_j}-2} \geq \frac{\alpha_{d_j}-1}{\alpha_{d_j}+\beta_{d_j}-2},$$

the maximum of the posterior of c_j and d_j is

$$\hat{c}_j(s_j) = \hat{d}_j(s_j) = \frac{\alpha_{c_j}+\alpha_{d_j}-2}{\alpha_{c_j}+\beta_{c_j}+\alpha_{d_j}+\beta_{d_j}-4}; \tag{26}$$

otherwise,

$$\hat{c}_j(s_j) \;=\; \frac{\alpha_{c_j}-1}{\alpha_{c_j}+\beta_{c_j}-2}, \tag{27}$$
$$\hat{d}_j(s_j) \;=\; \frac{\alpha_{d_j}-1}{\alpha_{d_j}+\beta_{d_j}-2}. \tag{28}$$

It should be noted that, when $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$, which is the noninformative prior, the obtained MAP estimators of c_j and d_j become their ML estimators.

3 Estimation Methods

This section presents two versions of the SAEM algorithm for the MMAP estimation of the 4PNO model. A full SAEM algorithm is proposed in subsection 3.1. In subsection

3.2, an improved and more efficient SAEM algorithm, the MSAEM, is developed for the 4PNO model.

3.1 SAEM Algorithm for the MMAP Estimation of the 4PNO Model

First, we introduce the basic EM algorithm for the MMAP estimation of the 4PNO model. The EM algorithm consists of an expectation step (E-Step) and a maximization step (M-Step), and in the situation where the complete data likelihood belongs to the exponential family, the E-Step and the M-Step can be implemented using the sufficient statistics. The E-step of each iteration is taking the conditional expectation over the sufficient statistics given the current estimates of parameters, and the M-step computes the MAP or ML estimate of parameters using the updated expectation of the sufficient statistics in the E-Step.

Let $\boldsymbol{\xi}^0 = (\boldsymbol{\xi}_1^0, ..., \boldsymbol{\xi}_M^0)$ be the initial values, and $\boldsymbol{\xi}^{k-1} = (\boldsymbol{\xi}_1^{k-1}, ..., \boldsymbol{\xi}_M^{k-1})$ denote the parameter estimate at the end of the $(k-1)$th iteration. The kth iteration of the EM algorithm consists of the two steps:

- E-Step: Compute

$$s_j^k = E(S(\mathbf{x}_{\cdot j})|\mathbf{u}_{\cdot j}, \boldsymbol{\xi}^{k-1}), \tag{29}$$

  where the expectation is with respect to $f(\mathbf{w}_{\cdot j}, \mathbf{z}_{\cdot j}, \boldsymbol{\theta}|\mathbf{u}_{\cdot j}, \boldsymbol{\xi}^{k-1})$, and $j = 1, .., M$.

- M-Step: Update

$$\boldsymbol{\xi}_j^k = \hat{\boldsymbol{\xi}}_j(s_j^k),$$

  which is obtained by substituting s_j^k for s_j in Equations 24-28.

In the above EM iteration, the conditional expectation in the E-Step can not be done in closed form, which leads to the problem of integral computation. For this problem, the MCEM algorithm proposed by Wei and Tanner (1990) is a powerful computing tool, in which the conditional expectation is computed by means of Monte Carlo samples. Meng and Schilling (1996) proposed a MCEM algorithm for the estimation of the full-information item factor model, and they demonstrated that the MCEM algorithm have substantial improvement over the EM algorithm with a numerical integral.

But the drawback of the MCEM algorithm is that it has a high computational cost in many situations. To address this problem, Delyon et al. (1999) proposed the SAEM algorithm, in which a stochastic approximation is used to estimate the conditional expectation in the E-Step. In contrast, the SAEM algorithm is more efficient. Furthermore, Delyon et al. (1999) proved that, the convergence of the SAEM algorithm can be ensured under many practical situations, when the complete-data likelihood belongs to an exponential family. In the following, we develop a SAEM algorithm for the MMAP estimation of the 4PNO model.

3.1.1 General Description of the SAEM Algorithm

Let $\boldsymbol{\xi}^0 = (\boldsymbol{\xi}_1^0, ..., \boldsymbol{\xi}_M^0)$ be the initial values, and $\boldsymbol{\xi}^{k-1} = (\boldsymbol{\xi}_1^{k-1}, ..., \boldsymbol{\xi}_M^{k-1})$ denote the parameter estimate at the end of the $(k-1)$th iteration. The kth iteration of the SAEM algorithm for the MMAP estimatino of the 4PNO model consists of:

- Simulation-Step (S-Step): Sample m_k sets of $(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta})$ from $p(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta} | \mathbf{u}, \boldsymbol{\xi}^{k-1})$ to form m_k sets of complete data set $\{\mathbf{x}^l = (\mathbf{u}, \mathbf{w}^l, \mathbf{z}^l, \boldsymbol{\theta}^l); l = 1, ..., m_k\}$.

- Stochastic Approximation-Step (SA-Step): Compute the sufficient statistics $S(\mathbf{x}^l_{.j})$ $(l = 1, ..., m_k)$ in Equations 13-17, and update s_j^k according to

$$s_j^k = s_j^{k-1} + \gamma_k \left(\frac{\sum_{l=1}^{m_k} S(\mathbf{x}_j^l)}{m_k} - s_j^{k-1} \right),$$

where $\{\gamma_k\}_{k>0}$ is the Robbins-Monro (RM) gain coefficient such that

$$\gamma_k > 0, \quad \sum_{k=1}^{+\infty} \gamma_k = +\infty, \quad \sum_{k=1}^{+\infty} \gamma_k^2 < +\infty,$$

for item$j$, where $j = 1, .., M$.

- Maximization-Step (M-Step): Compute

$$\boldsymbol{\xi}_j^k = \hat{\boldsymbol{\xi}}_j(s_j^k),$$

by substituting s_j^k for s_j in Equations 24-28, where $j = 1, ..., M$.

In the SAEM iteration, the E-Step in Equation 29 is replaced by a stochastic approximation iteration of Robbins and Monro (1951), thus the obstacle integral computation is avoided. For the stochastic approximation procedure, the choice of γ_k and the specification of m_k are very important, and they both determine the performance of the SAEM algorithm.

Remark 1: The choice of step sizes $\{\gamma_k\}_{k>0}$ plays an important role in determining the convergence of the SAEM algorithm. A sequence of large step sizes puts the parameter estimates into the neighborhood of the solution rather quickly, but it also introduces a large amount of simulation noise. On the other hand, a sequence of smaller step sizes reduces the noise faster, but it results in a rather slowly moving algorithm (Jank, 2006). Gu and Zhu (2001) proposed to employ stochastic approximation in two stages where the first stage uses a rather large step size and then in the second stage, after the algorithm has reached the proximity of the solution, the method switches to a smaller step size selection. Some recent research (Camilli and Fox, 2015; Camilli and Geis, 2019; Galarza et al., 2017; Kuhn and Lavielle, 2004; Lavielle and Mbogning, 2014) suggested setting $\gamma_k = 1$ in the first K iterations (the first stage) and setting $\gamma_k = \frac{1}{(k-K)^\alpha}$ when $k > K$ (the second stage), where $0.5 < \alpha \leq 1$. In practice, typically specify $\alpha = 1.0$ or $2/3$. For presentation convenience, in this article, the step size sequence is written as

$$\left\{\gamma_k = I_{(k\leq K)} + I_{(k>K)} \times \frac{1}{(k-K)^{\alpha}}\right\}_{k>0}. \tag{30}$$

Remark 2: If specifying $\gamma_k = 1$ for all the iterations, the SA-Step becomes equivalent to that the conditional expectation of the sufficient statistics, $E(S(\mathbf{x}_{\cdot j})|\mathbf{u}_{\cdot j}, \boldsymbol{\xi}^{k-1})$, is computed by a Monte Carlo integral, then the SAEM iteration is reduced to a MCEM algorithm. This MCEM can be seen as an extension of the MCEM proposed by Meng and Schilling (1996) to the 4PNO model. To guarantee the accuracy of the Monte Carlo integral, a big number of the simulation m_k is often required, which leads to high computational cost of the MCEM algorithm in many situations.

Remark 3: When the complete-data model belongs to the exponential family, the convergence of the SAEM algorithm can be guaranteed with the setting of the number of the simulation $m_k = 1$. In comparison with the MCEM algorithm, the simulation is cheaper, thus the SAEM algorithm is more efficient. Furthermore, we have investigated in simulation studies that, as m_k increases, the computational time of the SAEM algorithm increases significantly, but the accuracy and the stability of the obtained SAEM estimates show no significant changes.

It can be seen that, the complete-data model is a member of the exponential family, the SA-Step and the M-Step can be directly computed using the sufficient statistics, thus the intractable numerical computations are avoided. To simulate $(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta})$ in S-Step, which is not available in closed form, we propose a two-step sampling procedure, where a discrete-grid method is employed to generate the realizations of the missing data $(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta})$, and the detailed sampling procedure is given below.

3.1.2 Generating Missing Data in the S-Step

In the S-Step, drawing $(\mathbf{w}^l, \mathbf{z}^l, \boldsymbol{\theta}^l)$ from $f(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{u}, \boldsymbol{\xi}^{k-1})$ consists of the following two steps: first sampling θ_i^l from its marginal posterior distribution $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$, and then drawing (w_{ij}^l, z_{ij}^l) from $f(w_{ij}, z_{ij}|\theta_i^l, u_{ij}, \boldsymbol{\xi}_j^{(k-1)})$, where $i = 1, ..., N, j = 1, ..., M$. As the closed form of $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$ is not available, it can not be used directly for simulation. To deal with this problem, a discrete-grid method is used. The S-Step proceeds as follows:

1. For $i = 1, ..., N$, θ_i^l $(l = 1, ..., m_k)$ is directly simulated from $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$ using a discrete-grid approximation method.

   (a) A grid including a set of evenly spaced values, $\theta_1^*, ..., \theta_T^*$, is selected on a broad range of the parameter space for θ. The posterior density function of θ_i, $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$, is computed on the grid values,

   $$p(\theta_t^*|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1}) = \frac{p(\mathbf{u}_{i.}|\theta_t^*, \boldsymbol{\xi}^{k-1})\phi(\theta_t^*)}{\sum_{t=1}^{T} p(\mathbf{u}_{i.}|\theta_t^*, \boldsymbol{\xi}^{k-1})\phi(\theta_t^*)}, t = 1, ..., T, \tag{31}$$

   which is a discrete-grid approximation of $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$.

   (b) Randomly draw m_k values of θ_i^l, $\{\theta_i^l; l = 1, ..., m_k\}$, from the discrete-grid approximation of $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$ given in Equation 31.

2. For $i = 1, ..., N$ and $j = 1, ...M$, sample (w_{ij}^l, z_{ij}^l) $(l = 1, ..., m_k)$ from $p(w_{ij}, z_{ij}|u_{ij}, \boldsymbol{\xi}_j^{(k-1)}, \theta_i^l)$ according to the following procedure,

   (a) Sample w_{ij}^l from,

   $$W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l \sim Bernoulli(p_{W=1})$$

   where

   $$p_{W=1} = \begin{cases} \frac{d_j^{k-1}\Phi(a_j^{k-1}\theta_i^l+b_j^{k-1})}{d_j^{k-1}\Phi(a_j^{k-1}\theta_i^l+b_j^{k-1})+c_j^l(1-\Phi(a_j^{k-1}\theta_i^l+b_j^{k-1}))}, & u_{ij} = 1 \ ; \\ \frac{(1-d_j^{k-1})\Phi(a_j^{k-1}\theta_i^l+b_j^{k-1})}{(1-d_j^{k-1})\Phi(a_j^{k-1}\theta_i^l+b_j^{k-1})+(1-c_j^{k-1})(1-\Phi(a_j^{k-1}\theta_i^l+b_j^{k-1}))}, & u_{ij} = 0. \end{cases}$$

(b) Sample z_{ij}^{l} from,

$$Z_{ij}|w_{ij}^{l}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l \sim N(a_j^{k-1}\theta_i^l + b_j^{k-1}, 1)[I_{(z_{ij}>0)}w_{ij}^l + I_{(z_{ij}\leq 0)}(1 - w_{ij}^l)].$$

which is a truncated normal distribution.

Remark 4: For the discrete-grid approximation approach, the selection of the grid points plays a key role. Understandably, a grid defined on too small an area may miss important features of the distribution that fall outside the grid, but on a large area with wide intervals between points may miss important features that fall between the grid points. In this study, we use an equal-spacing grid of θ_i in the interval of $[-3.0, +3.0]$. As given in Equation 8, θ_i is assumed to follow $N(0,1)$, and $[-3.0, +3.0]$ is the range of ± 3 standard deviation, thus it can cover a broad range of θ-space. The equal-spacing grid is a regular grid design and is commonly used in the grid-sampling method. Furthermore, we have verified in simulation studies that 30 grid points is enough to ensure the accuracy of the SAEM algorithm.

In Bayesian statistics, the discrete-grid approximation provides a powerful computational and sampling approach for the situations where the posterior distribution is low-dimensional and has no closed-form expression (Gelman et al., 2013). Using the discrete-grid approximation sampling method makes it possible to generate the missing data of θ_i from its marginal posterior distribution, and then the MCMC sampler is avoided. Another advantage with using the discrete-grid method is that it allows some important indicators or statistics to be computed directly. For instance, the expected a posteriori (EAP) estimate of θ_i can be calculated by

$$\text{EAP}(\theta_i) = \frac{\sum_{t=1}^{T} \theta_t^* \times p(\mathbf{u}_{i.}|\theta_t^*, \boldsymbol{\xi}^k)\phi(\theta_t^*)}{\sum_{t=1}^{T} p(\mathbf{u}_{i.}|\theta_t^*, \boldsymbol{\xi}^k)\phi(\theta_t^*)}, i = 1, ..., N. \tag{32}$$

Furthermore, the logarithm of marginal posterior odds for $\boldsymbol{\xi}^k$ to $\boldsymbol{\xi}^{k-1}$,

$$\Delta L^k = |\log f(\boldsymbol{\xi}^k|\mathbf{u}, \Omega) - \log f(\boldsymbol{\xi}^{k-1}|\mathbf{u}, \Omega)|, \tag{33}$$

can be easily calculated, and it is used to check the the convergences of the SAEM iteration. As discussed in Meng and Schilling (1996), ΔL^k is a powerful tool for monitoring the convergence of the EM-type algorithms, and it is recommended for the problems when it is easily computed (Gelman et al., 2013).

3.2 MSAEM Algorithm for the MMAP Estimation of the 4PNO Model

Celeux et al. (2000) pointed out that, because of the well known label switching phenomenon in the estimation of the mixture models, the simulation of latent categorical variables is likely to impact the convergence of the MCMC and the SAEM algorithm for the mixture of NLMEM. To cope with this problem, Lavielle and Mbogning (2014) developed an modified SAEM algorithm for computing the maximum likelihood estimation of the mixture of NLMEM.

As given in Equation 2, the 4PNO model belongs to a two-classify mixture Bernoulli model, where W_{ij} is the latent categorical variable. Furthermore, according to the relationship between W_{ij} and Z_{ij} in Equation 6, Z_{ij} indirectly play the role of the latent categorical variable. It is indicated that, the simulation of $(\mathbf{w}, \mathbf{z})$ is likely to impact the performance of the SAEM algorithm for the 4PNO model. Inspired by the the studies of Celeux et al. (2000) and Lavielle and Mbogning (2014), we develop an improved and more efficient SAEM algorithm, which is called the MSAEM, for computing the MMAP estimate of the 4PNO model.

3.2.1 General Description of the MSAEM Algorithm

By taking the law of total expectation, the E-Step in Equation 29 can be calculated by,

$$E(S(\mathbf{x}_{\cdot j})|\mathbf{u}_{\cdot j}, \boldsymbol{\xi}) = E_{\boldsymbol{\theta}}(E_{(\mathbf{W}_{\cdot j}, \mathbf{Z}_{\cdot j})}(S(\mathbf{x}_{\cdot j})|\boldsymbol{\theta}, \mathbf{u}_{\cdot j}, \boldsymbol{\xi})|\mathbf{u}_{\cdot j}, \boldsymbol{\xi}), \tag{34}$$

where $E_{(\mathbf{w}_{.j},\mathbf{z}_{.j})}(\cdot|\boldsymbol{\theta},\mathbf{u}_{.j},\boldsymbol{\xi})$ is the expectation with respect to $f(\mathbf{w}_{.j},\mathbf{z}_{.j}|\mathbf{u}_{.j},\boldsymbol{\theta},\boldsymbol{\xi})$, and $E_{\boldsymbol{\theta}}(\cdot|\mathbf{u}_{.j},\boldsymbol{\xi})$ is the expectation with respect to $f(\boldsymbol{\theta}|\mathbf{u}_{.j},\boldsymbol{\xi})$. Then $E(S(\mathbf{x}_{.j})|\mathbf{u}_{.j},\boldsymbol{\xi})$ can be estimated using a stochastic approximation procedure on $E_{(\mathbf{w}_{.j},\mathbf{z}_{.j})}(S(\mathbf{x}_{.j})|\boldsymbol{\theta},\mathbf{u}_{.j},\boldsymbol{\xi})$, when $E_{(\mathbf{w}_{.j},\mathbf{z}_{.j})}(S(\mathbf{x}_{.j})|\boldsymbol{\theta},\mathbf{u}_{.j},\boldsymbol{\xi})$ can be done in closed form. From this point, an improved SAEM algorithm is developed for the 4PNO model, which is given in the following.

Let $\boldsymbol{\xi}^0 = (\boldsymbol{\xi}_1^0, ..., \boldsymbol{\xi}_M^0)$ denote the initial values, and $\boldsymbol{\xi}^{k-1} = (\boldsymbol{\xi}_1^{k-1}, ..., \boldsymbol{\xi}_M^{k-1})$ denote the parameter estimate at the end of the $(k-1)$th iteration. The kth iteration of the MSAEM algorithm consists of the following steps:

- S-Step: Sample θ_i^l $(l = 1, ..., m_k)$ from $f(\theta_i|\mathbf{u}_{i.},\boldsymbol{\xi}^{k-1})$, where $i = 1, ..., N$, using the discrete-grid method that is given in subsection 3.1.3. Here $\{\boldsymbol{\theta}^l = (\theta_1^l, ..., \theta_N^l)'; l = 1, ..., m_k\}$ denotes m_k vectors of the abilities.

- E-Step: Compute

  $$S(\mathbf{u}_{.j},\boldsymbol{\theta}^l,\boldsymbol{\xi}_j^{k-1}) = E_{(\mathbf{w}_{.j},\mathbf{z}_{.j})}(S(\mathbf{x}_{.j})|\mathbf{u}_{.j},\boldsymbol{\theta}^l,\boldsymbol{\xi}_j^{k-1}),$$

  according to the Equations 35-39, where $j = 1, .., M$, $l = 1, ..., m_k$.

- SA-Step: Update s_j^k according to

  $$s_j^k = s_j^{k-1} + \gamma_k(\frac{\sum_{l=1}^{m_k} S(\mathbf{u}_{.j},\boldsymbol{\theta}^l,\boldsymbol{\xi}_j^{k-1})}{m_k} - s_j^{k-1}),$$

  where the specifications of γ_k and m_k are the same as that in the SAEM algorithm; please see Remarks 1 and 3.

- M-Step: Compute

  $$\boldsymbol{\xi}_j^k = \hat{\boldsymbol{\xi}}_j(s_j^k),$$

  by substituting s_j^k for s_j in Equations 24-28, where $j = 1, ..., M$.

In the above procedures, the missing data of $(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta})$ is dealt with using a mixture of the two methods: $\boldsymbol{\theta}$ are simulated from their marginal posterior distributions, and $(\mathbf{w}, \mathbf{z})$ are replaced by their conditional expectations. Therefore, the improved SAEM algorithm is called the mixed SAEM (MSAEM) algorithm. It can be seen that, the use of the discrete-grid sampling method is very important for the implementation of the MSAEM iteration, since in which randomly drawing from $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$ is the first step, and the closed form of $f(\theta_i|\mathbf{u}_{i.}, \boldsymbol{\xi}^{k-1})$ can not be obtained.

In comparison with the SAEM algorithm, the simulation of missing data of the MSAEM algorithm is computationally cheaper, then the MSAEM algorithm becomes more efficient. More importantly, as verified by the simulation studies in Section 4, the accuracy of the MSAEM are higher than those of the SAEM algorithm.

For implementing the MSAEM algorithm, the closed form of the conditional expectation of the sufficient statistics, $E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1})$, need to be obtained and they are derived in the following.

3.2.2 Implementation Details of the E-Step

First, according to the definition of $S(\mathbf{x}_{.j})$ $(j = 1, ..., M)$ in Equations 13-17, all the elements of $E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1})$ can be formulated as,

$$E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S_1(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}) = \sum_{i=1}^{N} E(W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l) u_{ij}, \quad (35)$$

$$E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S_2(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}) = \sum_{i=1}^{N} E(W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l), \quad (36)$$

$$E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S_3(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}) = \sum_{i=1}^{N} (1 - E(W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l)) u_{ij}, \quad (37)$$

$$E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S_4(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}) = \boldsymbol{\Lambda}' \boldsymbol{\Lambda}, \quad (38)$$

$$E_{(\mathbf{w}_{.j}, \mathbf{z}_{.j})}(S_5(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}) = \boldsymbol{\Lambda}' E(\mathbf{Z}_{.j}|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}), \quad (39)$$

where $\mathbf{\Lambda} = (\boldsymbol{\theta}^l, \mathbf{1}_N)$ and $E(\mathbf{Z}_{.j}|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1}) = (E(Z_{1j}|u_{1j}, \theta_1^l, \boldsymbol{\xi}_j^{k-1}), ..., E(Z_{Nj}|u_{Nj}, \theta_N^l, \boldsymbol{\xi}_j^{k-1}))'$. It can be seen that, if $E(W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l)$ and $E(Z_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l)$ can be done in closed forms, the closed form of $E_{(\mathbf{W}_{.j}, \mathbf{Z}_{.j})}(S(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1})$ can be obtained consequently.

From Equations 2 and 9, the conditional densities of Z_{ij} given $U_{ij} = u_{ij}$ can be written as

$$\begin{aligned} & f(z_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^l) \\ = & \begin{cases} \frac{d_j^{k-1}\phi(z_{ij}-a_j^{k-1}\theta_i^l-b_j^{k-1})I_{(z_{ij}>0)}+c_j^{k-1}\phi(z_{ij}-a_j^{k-1}\theta_i^l-b_j^{k-1})I_{(z_{ij}\leq 0)}}{P(U_{ij}=1|\xi_j^{k-1},\theta_i^l)}, & u_{ij} = 1 \ ; \\ \frac{(1-d_j^{k-1})\phi(z_{ij}-a_j^{k-1}\theta_i^l-b_j^{k-1})I_{(z_{ij}>0)}+(1-c_j^{k-1})\phi(z_{ij}-a_j^{k-1}\theta_i^l-b_j^{k-1})I_{(z_{ij}\leq 0)}}{1-P(U_{ij}=1|\xi_j^{k-1},\theta_i^l)}, & u_{ij} = 0. \end{cases} \end{aligned}$$

where $P(U_{ij} = 1|\boldsymbol{\xi}_j^{k-1}, \theta_i^l)$ is given in Equation 2.

Further, we have

$$E(Z_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^k) = \begin{cases} \frac{d_j^{k-1}E_1+c_j^{k-1}E_2}{P(U_{ij}=1|\xi_j^{k-1},\theta_i^k)}, & u_{ij} = 1 \ ; \\ \frac{(1-d_j^{k-1})E_1+(1-c_j^{k-1})E_2}{1-P(U_{ij}=1|\xi_j^{k-1},\theta_i^k)}, & u_{ij} = 0, \end{cases} \tag{40}$$

where

$$\begin{aligned} E_1 &= \phi(a_j^{k-1}\theta_i^l + b_j^{k-1}) + (a_j^{k-1}\theta_i^l + b_j^{k-1})\Phi(a_j^{k-1}\theta_i^l + b_j^{k-1}), \\ E_2 &= -\phi(a_j^{k-1}\theta_i^l + b_j^{k-1}) + (a_j^{k-1}\theta_i^l + b_j^{k-1})\Phi(-a_j^{k-1}\theta_i^l - b_j^{k-1}), \end{aligned}$$

and

$$E(W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^k) = \begin{cases} \frac{d_j^{k-1}\Phi(a_j^{k-1}\theta_i^k+b_j^{k-1})}{P(U_{ij}=1|\xi_j^{k-1},\theta_i^k)}, & u_{ij} = 1 \ ; \\ \frac{(1-d_j^{k-1})\Phi(a_j^{k-1}\theta_i^k+b_j^{k-1})}{1-P(U_{ij}=1|\xi_j^{k-1},\theta_i^k)}, & u_{ij} = 0. \end{cases} \tag{41}$$

Finally, by plugging $E(W_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^k)$ and $E(Z_{ij}|u_{ij}, \boldsymbol{\xi}_j^{k-1}, \theta_i^k)$ given in Equations 40 and 41 into Equations 35-39, the closed form of $E_{(\mathbf{W}_{.j}, \mathbf{Z}_{.j})}(S(\mathbf{x}_{.j})|\mathbf{u}_{.j}, \boldsymbol{\theta}^l, \boldsymbol{\xi}_j^{k-1})$ is obtained.

4 Monte Carlo Simulation

In this section, two Monte Carlo simulation studies are reported. The first simulation study was constructed to investigate the impact of K (the number of steps with $\gamma_k = 1$; see Remark 1) on the performance of the SAEM and the MSAEM, and assess the sensitivity (or robustness) of the two algorithms on the initial values. The second simulation was constructed to evaluate the properties (recovery accuracy, insensitivity on the priors, computational efficiency) of the SAEM and MSAEM algorithms by comparing with the MCEM algorithm and the Gibbs sampler algorithm of Culpepper (2016). A simulation study was also constructed to compare the recovery accuracy of MSAEM for 4PNO with two commonly used EM algorithms for the 4PL model.

4.1 Study 1

4.1.1 Design

In this simulation, the test length of the artificial test was $M = 30$, and the true values of $\boldsymbol{\xi}_j$ $(j = 1, ..., M)$ were randomly generated from the following distributions: $a_j \sim U(0.5, 3.0)$, $b_j \sim N(0.0, 1.0)$, $c_j \sim U(0.0, 0.35)$ and $d_j \sim U(0.65, 1.0)$, where $U(\cdot, \cdot)$ denotes the uniform distribution. The sample size of test takers is an important factor affecting the estimation of item parameters in IRT models, and three different levels of sample size were considered in this simulation, they were $N = 500$, 1000 and 5000. Under each of the three sample conditions, the true values of $\boldsymbol{\theta}$ were randomly generated from the standard normal distribution, $\theta_i \sim N(0, 1)$ $(i = 1, .., N)$.

In this simulation study, to reduce sampling error, 200 replications were generated under each of the three sample sizes. For each replication, the MMAP estimation of the 4PNO model was computed by using the SAEM and the MSAEM algorithms. The priors for (a_j, b_j) in Equations 18 were specified to be: $(\mu_0, \Sigma_0^{-1}) = (\mathbf{0}_2, (2\mathbf{I}_2)^{-1})$, where $\mathbf{0}_2$ is a two-dimensional vector of zeros and $\mathbf{I}_2$ is a two-dimensional identity matrix, which is the same as that in Culpepper(2016); the priors for c_j and d_j in Equation 19 were specified to be, $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$, which were suggested by Loken and Rulison (2010). These priors are commonly used in the estimation of the IRT models. Other priors can also be employed; for instance, the non-informative priors are given in the next simulation study, and the obtained conclusions are almost identical.

To study the impact of K on the convergence of the SAEM and MSAEM algorithms, the step size sequence $\{\gamma_k\}$ given in Equation 30 was specified with 4 levels of K, they were $K = 100$, 500, 1000 and 1500, when $k > K$, the gain speed control parameter was set to be $\alpha = 2/3$. Furthermore, in this study, to assess the sensitivity to the initial values, the SAEM and MSAEM algorithms were ran starting from three different initial values: (I1) randomly generated from the distributions, $a_j^0 \sim U(0.5, 3), b_j^0 \sim U(-2, 2), c_j^0 \sim U(0, 0.3)$ and $d_j^0 \sim U(0.7, 1)$; (I2) a group of representative values, $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2, d_j^0 = 0.8$; (I3) the true values of $\boldsymbol{\xi}_j (j = 1, ..., M)$ were taken as the initial values. Therefore, for each of the three sample sizes, the SAEM and the MSAEM were implemented under the 12 conditions (4 levels of $K \times 3$ groups of initial values) separately, and 200 replications were performed for each simulation condition.

In the S-Step, the grid sampling procedure used the range of $\theta \in [-3.0, 3.0]$, and the number of the grid points was $T = 30$, as discussed in Remark 4. The convergences of the SAEM and MSAEM iterations were checked using ΔL^k that is given in Equation 33, and if $\Delta L^k \leq 10^{-4}$, the iteration was terminated.

4.1.2 Results

The root mean squire error (RMSE) of each parameter was calculated across the 200 replications to evaluate the recovery accuracy. Let δ_j denote one of the four characteristic parameters (a_j, b_j, c_j, d_j) of item j, and $\hat{\delta}_{jg}$ denote the estimate obtained with the g-th simulated data, then the RMSE of δ_j is computed as

$$RMSE_{\delta_j} = \sqrt{200^{-1}\sum\nolimits_{g=1}^{200}\left(\hat{\delta}_{jg} - \delta_j\right)^2}, \tag{42}$$

where $j = 1, ..., M$. Further, let $RMSE_\delta = \{RMSE_{\delta_1}, \ldots, RMSE_{\delta_M}\}$, which denotes the RMSEs of δs for all the items in test. In this simulation, under each of the 3 sample sizes ($N = 500, 2000$ and 5000), the Box-plots of $RMSE_\delta$ obtained from the two algorithms (SAEM and MSAEM) implemented with the 12 settings (4 levels of K and 3 groups of initial values) are displayed in Figures 1-6.

Observing these figures, it can be found the following trends: First, across the three sample sizes (N = 500, 1000 and 5000), the Box-plots of RMSEs obtained from the SAEM and the MSAEM with setting $K = 100$ were significantly different from those obtained with the settings of $K = 500, 1000$ and 1500. Moreover, in the case of $K = 100$, there were substantial differences in the Box-plots of RMSEs corresponding to three different initial values (I1, I2, and I3). These phenomena indicate that the convergence of the SAEM and the MSAEM algorithms with $K = 100$ may not be guaranteed.

Second, under the conditions of $N = 500$ and 1000, the Box-plots of RMSE obtained from the SAEM and the MSAEM with the same initial values were almost unchanged with the increase of K ($K = 500, 1000, 1500$). These results demonstrate that the convergence of the SAEM and the MSAEM can be guaranteed under the setting of $K = 500$. Furthermore, the Box-plots of SAEM and MSAEM under different initial values were almost identical; even when the initial values are selected randomly (denoted as I1), the obtained results were very close to those obtained with the true values (denoted as I3). This trend indicates that both the SAEM and the MSAEM are highly robust on the choice of the initial values. But it can be seen that the Boxplots of the MSAEM under different initial values were more stable, indicating the better robustness of the MSAEM on initial values.

Third, the Boxplots obtained under the sample size of $N = 5000$ were significantly lower than those under $N = 500$ and 1000, which is expected since the accuracy of estimators should be improved with the increase of sample size. It can be seen that, in the case of $K = 500$, the Box-plots of RMSEs of $\boldsymbol{a} = \{a_1, ..., a_M\}$ from the initial values of I1 (randomly selected) and I2 were a little higher than those from the initial value I3, but this phenomenon disappeared when $K = 1000$ and 1500. Then to ensure the estimation accuracy, we suggest to set $K = 1000$. It should be emphasized that, even under $K = 1000$, the computing times of the SAEM and the MSAEM were not much. In the following section of Study 2, the computing times of the two algorithms were investigated under different conditions, and the obtained results are reported in Table 5.

Finally, it can be seen that, in comparison with the SAEM algorithm, the Box-plots of the MSAEM algorithm are located lower; moreover, they are stable under different levels of K and different initial values. These phenomena are also obvious under the relatively larger sample size $N = 5000$. It thus indicates that the MSAEM algorithm, as an improvement of the SAEM algorithm, has more desired properties for estimating the 4PNO model.

4.2 Study 2

4.2.1 Design

In this simulation, two test lengths ($M = 20$ or 40) and two sample sizes ($N = 1000$ or 5000) were considered, thus there were in total $2 \times 2 = 4$ testing conditions. The same as the design of Study 1, for each of the four testing conditions, the true values of $\boldsymbol{\xi}_j$

$(j = 1, ..., M)$ were randomly generated from the following distributions, $a_j \sim U(0.5, 3)$, $b_j \sim N(0, 1)$, $c_j \sim U(0, 0.35)$ and $d_j \sim U(0.65, 1)$, and the true values of $\boldsymbol{\theta}$ were randomly generated from the standard normal distribution, $\theta_i \sim N(0, 1)$ $(i = 1, .., N)$.

In practice, the choice of the prior is subjective, thus it is desirable that the Bayesian inference is robust to the specification of the prior distribution in many situations (Berger, 1990; Gelman et al., 2013). From this point of view, it makes sense to perform a sensitivity analysis of the estimation of the 4PNO on the prior of $\boldsymbol{\xi}_j$ $(j = 1, ..., M)$. To do it, in this simulation, two priors were specified for (a_j, b_j), an informative prior, $(\mu_0, \Sigma_0^{-1}) = (\mathbf{0}_2, (2\mathbf{I}_2)^{-1})$, which is the same as that used in Study 1, and a noninformative prior, $(\mu_0, \Sigma_0^{-1}) = (\mathbf{0}_2, \mathbf{0}_{2\times 2})$, where $\mathbf{0}_{2\times 2}$ denotes the 2×2 zero matrix. Note that, this noninformative prior is an improper prior, which is given in subsection 2.1.1, and for more detailed theoretical concerns, see Gelman et al. (2013). Furthermore, two priors were specified for (c_j, d_j), an informative prior, $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$, which is the same as that used in Study 1, and a noninformative prior, $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$. The two priors of (a_j, b_j) and the two priors of (c_j, d_j) were crossed, leading to four priors for $\boldsymbol{\xi}_j$ $(j = 1, ..., M)$, which are shown in Table 1. The 4PNO model was estimated with the four priors separately to check the differences in the estimation accuracy under different priors.

The four testing conditions and the four priors for $\boldsymbol{\xi}_j$ were crossed, leading to in total $4 \times 4 = 16$ simulation conditions. In each of the 16 conditions, three algorithms were implemented to compute the MMAP estimates of the 4PNO model, including MSAEM, SAEM, and MCEM. According to the results in Study 1, in this simulation, the SAEM and the MSAEM were implemented with the settings: $K = 1000$, $\alpha = 2/3$ and $m_k = 1$. Furthermore, as discussed in Remark 4, the MCEM algorithm is the SAEM iteration with $\gamma_k = 1$ across all iterations. For MCEM, the accuracy of the Monte Carlo integral improves with the increase of m_k, but with a high computational cost. In this simulaiton, the

MCEM algorithm was implemented with $m_k = 30$, which was also verified to be enough for guaranteeing the performance of the MCEM algorithm by a separate simulation study (not reported here).

Furthermore, for comparison, the Gibbs sampler of Culpepper (2016) was implemented with the R package "fourPNO" to compute Bayesian estimates of the 4PNO. Here 30000 Gibbs iterations were generated, and the first 10000 iterations were discarded as burn-in. Note that, different from the three stochastic versions of EM algorithm proposed by this study, the Gibbs sampler is used to compute the expectation a posterior (EAP) estimation of the 4PNO model. Finally, the initial values of the four algorithms (SAEM, MSAEM, MCEM and Gibbs sampler) were identical, $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2$, and $d_j^0 = 0.8$.

The above procedures (data generation and parameter estimation) were repeated 200 times. Evaluation criteria of parameter recovery include: the average RMSE of each parameter over all items (which is denoted as ARMSE), the average of the correlation between the estimates and the true parameter across 200 replications (which is denoted as ACor), and the average of the item response function (IRF) recovery across all the items and the 200 replications (which is denoted as AIRF). They are calculated by

$$\text{ARMSE}_{\hat{\delta}} = \frac{1}{M}\sum_{j=1}^{M}\sqrt{200^{-1}\sum_{g=1}^{200}\left(\hat{\delta}_{jg} - \delta_j\right)^2},$$

and

$$\text{ACor}_{\hat{\delta}} = \frac{1}{200}\sum_{g=1}^{200}\frac{\sum_{j=1}^{M}\hat{\delta}_{jg}\delta_j - \frac{1}{M}\sum_{j}^{M}\hat{\delta}_{jg}\sum_{j}^{M}\delta_j}{\sqrt{(\sum_{j=1}^{M}\hat{\delta}_{jg}^2 - \frac{1}{M}(\sum_{j=1}^{M}\hat{\delta}_{jg})^2)(\sum_{j=1}^{M}\delta_j^2 - \frac{1}{M}(\sum_{j=1}^{M}\delta_j)^2)}},$$

where δ_j and $\hat{\delta}_{jg}$ are defined in Equation 42. Different from the ARMSE and the ACor, the AIRF is used for assessing the recovery accuracy at item-level. First, the IRF recovery is calculated by

$$\mathrm{IRF}_{jg} = \frac{1}{100}\sum_{t=1}^{100} |P(U_j = 1|\theta_t^*, \hat{\boldsymbol{\xi}}_{jg}) - P(U_j = 1|\theta_t^*, \boldsymbol{\xi}_j)|, \tag{43}$$

where $P(U_j = 1|\theta_t^*, \boldsymbol{\xi}_j)$ denotes the real item response probability of item j at θ_t^*, and $P(U_j = 1|\theta_t^*, \hat{\boldsymbol{\xi}}_{jg})$ denotes the estimated item response probability with the g-th simulated data. Here, θ_t^* belongs to the 100 evenly spaced θ points, $(\theta_1^*, ..., \theta_{100}^*)$, ranging from -3.0 to 3.0, which is suggested by Wollack et al. (2002). And then, the AIRF across the 200 replications and the M items is calculated by,

$$\mathrm{AIRF} = \frac{\sum_{g=1}^{200}\sum_{j=1}^{M} \mathrm{IRF}_{jg}}{200 \times M}. \tag{44}$$

4.2.2 Results

The obtained values of ARMSE, ACor and AIRF under the conditions of $N = 1000$ and 5000 are given in Tables 2-3. Observing these results, the following trends can be found.

Recovery Accuracy and Sensitivity Analysis on Priors. First, under most conditions, the ARMSEs and the AIRFs of the MSAEM algorithm were the smallest, and the ACor of the MSAEM were the biggest. These results provide evidence that the MMAP estimates computed by the MSAEM algorithm have the highest accuracy. Additionally, it can be seen that, there were very small differences in each of the three recovery evaluation indices of the MSAEM algorithm across the four priors. It is indicated that, the MMAP estimates computed by the MSAEM algorithm are highly robust to the prior of $\boldsymbol{\xi}_j$.

Second, under most conditions, the three recovery accuracy evaluation indices of the MCEM algorithm were very close to those of the MSAEM algorithm, and were obviously smaller than those of the SAEM algorithm. The MCEM algorithm yielded good performance for computing the MMAP estimation of the 4PNO. But the drawback of the MCEM algorithm is that it was quiet time-consuming in many situations. Please see

Table 5, compared with the MSAEM and the SAEM, the MCEM required much more computing time. Finally, similar to the MSAEM, the MMAP estimates computed by the MCEM were robust with respect to the prior of $\boldsymbol{\xi}_j$.

Third, compared with the MSAEM and the MCEM, the performance of the SAEM was poorer. Under most conditions, the estimation accuracy was lower, and the robustness with regard to the prior was weaker. As discussed by Lavielle and Mbogning (2014), when the sample size is small, the SAEM algorithm tends to be unstable and produce poor accuracy estimation for the mixture models, which is due to the simulation of the latent categorical variable in S-Step. It can be seen that, as the increase of the sample size, the estimation accuracy of the SAEM algorithm was greatly improved, and the inferiority of the SAEM algorithm was weaken. These observations are consistent with the views of Lavielle and Mbogning (2014). Therefore, it can be concluded that, for the 4PNO model, the simulation of $\mathbf{w}$ and $\mathbf{z}$ in S-Step leads to the relative poorer behavior of the SAEM algorithm.

Fourth, the ARMSEs and the AIRFs from the Gibbs sampler under the informative-prior of (a, b) (Prior1 and Prior 2) were very close to or even smaller than these of the MSAEM algorithm. In particular, under Prior 2, in which the informative prior for (a_j, b_j) and the noninformative prior for (c_j, d_j), the Gibbs sampler algorithm performed the highest recovery accuracy, which is consistent with the conclusion in Culpepper (2016). However, under the noninformative prior for (a, b) (Prior 3 and Prior 4), the recovery accuracy of the Gibbs sampler was the lowest, and this trend was more significant in the case of $N = 1000$ and $M = 20$. As mentioned above, the Bayesian estimation computed by the Gibbs sampler was the EAP estimate of the 4PNO model. Therefore, it can be concluded that, the EAP estimation of the 4PNO model is sensitive to the prior of (a_j, b_j); in contrast, the MMAP estimation of the 4PNO is more robust to the prior of $\boldsymbol{\xi}_j$.

Finally, it can be found that, the recovery accuracy of the estimates computed by the four algorithms were improved as the increase of N and M. When $N = 5000$ and $M = 40$, the differences in the recovery metrics between the four algorithms were very small. It is indicated that, the sample size is very important for parameter estimation; when the sample size is large, the estimation accuracy can be guaranteed, and the lack of the computational accuracy can be greatly diminished. But, for a large sample size, the estimation of parameter tends to require a high computational effort, and the computing efficiency becomes an important consideration for an estimation method.

Computing Efficiency Assessment. The average computing time (in Second) of the four algorithms (SAEM, MSAEM, MCEM and Gibbs sampler) across the 200 replications were calculated to assess their computing efficiency; please see Table 5. Note that the simulation study was run on a PC with an Intel Core i9-10900K (3.7 GHz) processor with 64GB of RAM. It can be seen that the MSAEM algorithm was the most efficient, followed by the SAEM algorithm, and both were much more efficient than the MCEM and the Gibbs sampler algorithm. This superiority of the computing efficiency of the MSAEM algorithm was much larger in the case of $N = 5000$ and $M = 40$.

The complete-data model of the 4PNO belongs to an exponential family, thus the computations of SAEM and the MSAEM are simplified, such as the numerical calculations are avoided in each iteration, which results in the computation cost of each iteration is small. In addition, compared to the SAEM, the simulation of missing data in MSAEM was much cheaper, which makes the computational cost of the MSAEM algorithm further lower than that of the SAEM algorithm.

These obtained results provide evidence that the MSAEM algorithm has overall advantages over the other three algorithms. The estimator obtained by the MSAEM algorithm not only have the highest recovery accuracy, but also is highly robust to the prior for the item parameters; more importantly, the MSAEM algorithm is computationally more efficient than the other algorithms. Finally, it should be noted that, the MSAEM can be directly used for the 2PNO and 3PNO models, and thus the MSAEM provides a powerful computing tool for the estimation of the NO models.

4.2.3 Recovery accuracy of the 4PNO under small sample size

According to the suggestion of one reviewer, the estimation accuracy of the 4PNO model is assessed under the sample size of $N = 500$, and the obtained results are reported in Table 4. One message from these results is under the sample size of $N = 500$, the estimation of the 4PNO model is of rather poor accuracy. By comparison, the three evaluation criteria of the MSAEM and the MCEM were a little smaller than those of the other two algorithms; moreover, in the cases of the noninformative prior for (a_j, b_j) and the test length is $M = 20$, the estimates computed by the Gibbs sampler displayed great large errors.

The poor recovery accuracy is mainly due to the small sample size for the estimation of the complex 4PNO model. In the IRT literature, some studies (Patsula, 1995; Tang et al., 1993; Yen, 1987; Yoes, 1995) supported that 1000 was taken as the minimum sample size required for accurate item-parameter estimation in IRT. Thissen (1982) suggested that 500 was the minimum feasible sample size for the dichotomous unidimensional model. Yen (1987) supported that at least a sample of 1000 with 20 items for the three parameter model estimation. According to these studies and the obtained results in this simulation, it can be concluded that the sample size of $N = 500$ may not be adequate for the 4PNO model estimation. This is consistent with the previous studies (Culpepper, 2016; Waller and Feuerstahler, 2017) that a relatively larger sample size is required to accurately recover 4PNO parameter values.

4.2.4 Estimation comparison with EM algorithms for the 4PL model

Following one reviewer's suggestion, to further investigate the performance of the proposed MSAEM algorithm, a simulation study is constructed to compare the recovery accuracy of MSAEM for 4PNO with two commonly used EM algorithms for the 4PL model. The first is the EM algorithm implemented in R package "mirt", and the second is the EM algorithm proposed by Meng et al. (2020). To save space, this simulation design and the obtained results are reported in Appendix. Note that due to the difference of the 4PNO and 4PL models, a scale constant of $D = 1.702$ is used to ensure the approximately equivalence of the item response functions under the two models. Comparing the results in Tables A2 and A3 to these in Tables 2-4, the three evaluation criteria consistently support that the estimates from the MSAEM for 4PNO are more accurate than those from the two EM algorithms for 4PL. Furthermore, the estimation results of the two EM algorithms more depend on the priors for (a_j, b_j, c_j, d_j), especially when the sample size is small. In particular, the estimation accuracy of the 4PL model under the noninformative priors is difficult to guarantee. Overall, compared with the EM algorithms for the 4PL model, the MSAEM algorithm provides a better performance.

5 Empirical study

In this section, we demonstrate the application of the 4PNO model via an empirical example that is from a state math assessment test (Tao et al., 2012). This data set includes 65 dichotomous items and 2000 test takers. For this example, the 4PNO model was estimated using the two methods: the MSAEM algorithm proposed by this article and the Gibbs sampler of Culpepper (2016) that is implemented in the R package of "fourPNO" separately. When the estimates of one model obtained from different methods display some large differences, the accuracy of at least one method should be doubted; otherwise, the obtained estimates are likely to be credible. Therefore, in this empirical study, the accuracy of the MSAEM algorithm was evaluated by comparing with the results from the Gibbs sampler. The two algorithms were carried out under the same priors that are shown in the row of "Prior 2" of Table 1, because with the priors in "Prior 2", the Gibbs sampler performed the highest recovery accuracy. The other settings (such as the specifications of K, the initial values of iteration, the number of the Gibbs iterations and the period of burn in) were the same as that in simulation study 2.

The estimates of the 4PNO model from the two algorithms were displayed using the scatter plots in Figure 7. From the four plots, it can be seen that for each item parameter $(a, b, c$ or $d)$, almost all the points fall on the diagonal, and the corresponding four correlation coefficients ≥ 0.98. It indicates that the estimates obtained from the MSAEM and the Gibbs sampler are consistent with each other. On the other hand, the MSAEM has substantially computational advantage over the Gibbs sampler; for fitting this data, the computing time of the MSAEM algorithm is less than 8 seconds, while the Gibbs sampler requires at least 1600 seconds (the PC information is given in the simulation study 2). In the following, the fit of the 4PNO model to this data was assessed at the item and the test levels respectively.

To evaluate the performance of the 4PNO model, the 3PNO and the 2PNO were selected as comparison models. Noting that the MSAEM algorithm can be directly implemented for computing the MMAP estimates of the two more restricted models. According to the recommendation of one reviewer, the S-X^2 statistic of Orlando and Thissen (2000) was used to assess the model fit at the item level. For item j, the S-X_j^2 statistic is calculated by

$$S - X_j^2 = \sum_{m=1}^{M-1} N_m \frac{(O_{jm} - E_{jm})^2}{E_{jm}(1 - E_{jm})},$$

where O_{jm} and E_{jm} denote the observed and expected proportion correct of examinees with total score m who get item j correct, N_m is the observed number of persons with test score m, and M is the maximal possible test score. The $S\text{-}X_j^2$ statistic approximately follows the chi-square distribution with the degrees of freedom $M - 1 - n_{pj}$, where n_{pj} denotes the number of the item parameters of item j. Thus an item fit chi-square significance test can be constructed based on the $S\text{-}X^2$ statistics. As verified by Orlando and Thissen (2000, 2003), the $S\text{-}X^2$ performed better than the traditional item fit statistics such as the Q_1 statistic (Yen, 1981) and the G_2 statistic (McKinley and Mills, 1985) for dichotomous IRT models.

The obtained values of $S\text{-}X^2$ of the three models (4PNO, 3PNO and 2PNO) are shown in Table 6, where a value labeled " $*$ " indicates the model shows misfit at the significance level of 0.05. It can be seen that, the values of $S\text{-}X^2$ of the 4PNO model for most items are the smallest, the values of $S\text{-}X^2$ of the 3PNO model are a little bigger, and the values of $S\text{-}X^2$ of the 2PNO are great larger than that of the other two models. Moreover, it can be seen that 1 item is significantly misfitted by the 4PNO, 3 items are significantly misfitted by the 3PNO, and 21 items are significantly misfitted by the 2PNO. These findings provide evidence that the fit of the 4PNO to the item response data is a little better than that of the 3PNO, and the fits of them are substantially better than that of the 2PNO model.

At the test level, the assessment of model fit was carried out using the Akaike information criterion (AIC) (Akaike, 1998),

$$\text{AIC} = -2 \ln L(\mathbf{u} \mid \boldsymbol{\xi}) + 2n_p, \tag{45}$$

where

$$L(\mathbf{u} \mid \boldsymbol{\xi}) = \prod_{j=1}^{M} L(\mathbf{u}_j | \boldsymbol{\xi}_j)$$

is the marginal likelihood that can be approximately calculated using the discrete grid method, and n_p is the number of the item parameters of all items. Note that in AIC, $-2 \ln L(\mathbf{u} \mid \boldsymbol{\xi})$ is used for evaluating the model fit, and n_p is the penalty for model complexity. As shown in Table 6, for different items, the best fitting item-level models could be different. Therefore, to fully evaluate the models, the data set was also fitted by a item-level hybrid model, where each item was modeled by the best among the three models in terms of the smallest S-X^2 statistic.

The obtained results are displayed in Table 7. It shows that the AIC of the hybrid model was the smallest, while close to that of the 4PNO. This finding indicates that the hybrid model slightly improved the fit of the 4PNO model for the whole test. In addition, the AIC of the 3PNO was bigger than that of the 4PNO, suggesting the 4PNO has some superiority over the 3PNO. The values of AIC and $-2\ln L(\mathbf{u} \mid \boldsymbol{\xi})$ of the 2PNO were substantially larger than those of the 4PNO and 3PNO, suggesting that the 2PNO model provided the worst fit to this data set. Overall, both the S-X^2 statistics and AIC support that the 4PNO was the best, followed by the 3PNO, and the 2PNO model was the worst.

6 Discussion

In IRT, the 4PM has received increasing attention in recent years. Noting the computational challenges in the 4PM estimation, this work aims to offer powerful computational tools by using recent advances in statistical computation. We focus on the 4PNO model and develop two versions of the SAEM algorithm to compute the MMAP estimators of the item parameters. Specifically, the 4PNO model is reformulated to be a hierarchical model by using a data augmentation method, and an important property is that the corresponding complete-data model belongs to an exponential family, which is convenient for developing the EM-type algorithms. We first develop a SAEM algorithm to compute the MMAP estimator of the 4PNO model, which includes the MCEM algorithm as a special case, but this algorithm is likely to be unstable due to the mixture modeling nature of the

4PNO model. To overcome the drawback of the SAEM algorithm, we further develop an improved SAEM algorithm for the 4PNO model, which is called the MSAEM algorithm.

The results from the simulation studies demonstrate that the obtained estimators from the MSAEM algorithm are more accurate than those from the Gibbs sampler of Culpepper (2016) and the SAEM algorithm; moreover, the MSAEM algorithm is more robust to the choices of starting values and priors. The recovery accuracy of the MCEM algorithm is close to that of the MSAEM algorithm, while the MCEM algorithm is more time-consuming. Overall the MSAEM algorithm is computationally more efficient than the other methods. Furthermore, as suggested by a reviewer, to fully investigate the performance of the proposed methods, the recovery accuracy of MSAEM for the 4PNO model was also compared with that of two existing EM algorithms for the 4PL model. The results demonstrate that the MSAEM algorithm is more accurate and robust than the EM algorithms for the 4PL model.

There are several future research directions. First, the step size sequence $\{\gamma_k\}$ plays an important role in the performance of the SAEM and the MASEM algorithms. In this study, we investigated the influence of the number of steps with $\gamma_k = 1$ (which is denoted as K) on the estimation accuracy, and the choice of K was suggested based on the simulation results. A more systematic strategy for setting $\{\gamma_k\}$ under different settings needs to be further explored. Second, as discussed in Kern and Culpepper (2020) and Meng et al. (2020), the 4PM can be viewed as a J-attribute higher-order DINA model, and it is valuable to extend the MSAEM algorithm to such higher-order models. Furthermore, Kern and Culpepper (2020) discussed the identifiability of the 4PM, and proposed a restricted version of 4PNO model, the dyad 4PNO model, based on the identifiability results for cognitive diagnosis models. The dyad 4PNO model was estimated using the MCMC algorithm in Kern and Culpepper (2020), and it is interesting to extend the SAEM and MSAEM algorithms to the dyad 4PNO model. Third, in this study, the MMAP estimation is defined under the assumption that the latent trait follows the normal distribution, however, some recent studies showed that the normal assumption is likely to fail in practice and the estimates based on the false assumption may lead to large estimation error (DeMars, 2012; Svetina et al., 2017; Wang et al., 2018; Zhang et al., 2020c). Therefore, it is valuable to develop an estimation method under a general distributional assumption on the latent trait. Finally, other stochastic versions of the EM algorithm have also been used for the estimation of some important IRT models (Fox, 2003; Zhang et al., 2020b) with good performance, and it is interesting to develop other stochastic EM algorithms for the 4PNO model using the data augmentation scheme given in this work.

Table 1: Four prior distributions for the item parameters, $\boldsymbol{\xi}_j = (a_j, b_j, c_j, d_j)$, in the 4PNO.

	$(a_j, b_j) \sim N_2(\mu_0, \Sigma_0) I_{(a_j \geq 0)}$	$(c_j, d_j) \sim Beta_2(\alpha_c, \beta_c, \alpha_d, \beta_d) I_{(1 \geq d_j \geq c_j \geq 0)}$
Prior 1 (Informative+Informative)	$(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2,(2\mathbf{I}_2)^{-1})$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 2 (Informative+Noninformative)	$(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2,(2\mathbf{I}_2)^{-1})$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$
Prior 3 (Noninformative+Informative)	$(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2,\mathbf{0}_{2\times2})$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 4 (Noninformative+Noninformative)	$(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2,\mathbf{0}_{2\times2})$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Note: $\mathbf{0}_2$: two-dimensional vector of zeros; $\mathbf{0}_{2\times2}$: 2×2 matrix of zeros; $\mathbf{I}_2$: two-dimensional identity matrix.

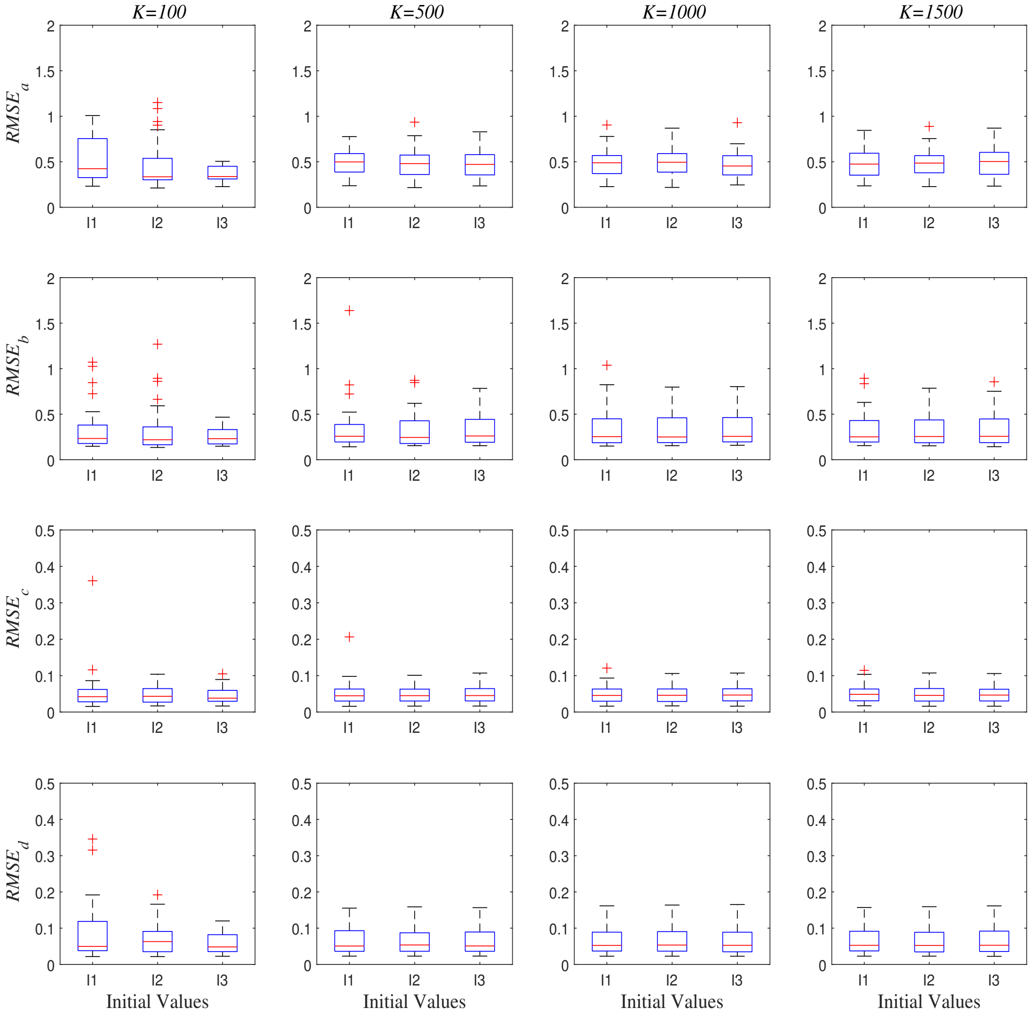

Figure 1: Box-plots of RMSEs of the MMAP estimates of a_j, b_j, c_j and d_j over all items in the test, and the MMAP estimates are computed by the SAEM algorithms under 4 levels of the number (K) of $\gamma_k = 1$ $(K = 100, 500, 1000, 1500)$ and 3 groups of initial values (I1: randomly selection, where $a^0 \sim U(0, 3), b^0 \sim U(-2, 2), c^0 \sim U(0, 0.3)$, $d^0 \sim U(0.7, 1)$; I2: a group of common values, where $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2, d_j^0 = 0.8$; I3: the true values of (a_j, b_j, c_j, d_j) are used as $(a_j^0, b_j^0, c_j^0, d_j^0)$). Sample size is $N = 500$.

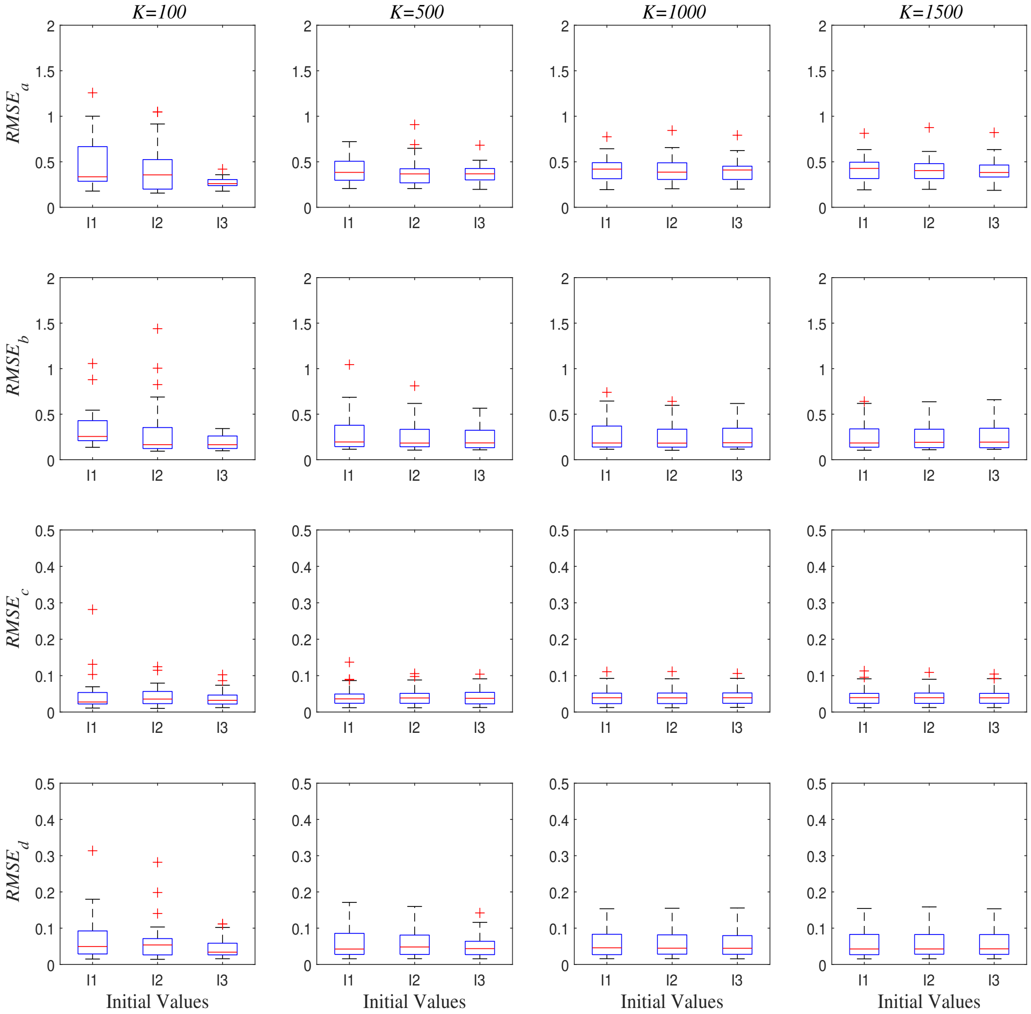

Figure 2: Box-plots of RMSEs of the MMAP estimates of a_j, b_j, c_j and d_j over all items in the test, and the MMAP estimates are computed by the SAEM algorithms under 4 levels of the number (K) of $\gamma_k = 1$ $(K = 100, 500, 1000$ and $1500)$ and 3 groups of initial values (I1: randomly selection, where $a^0 \sim U(0,3), b^0 \sim U(-2,2), c^0 \sim U(0,0.3)$, $d^0 \sim U(0.7,1)$; I2: a group of common values, where $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2, d_j^0 = 0.8$; I3: the true values of (a_j, b_j, c_j, d_j) are used as $(a_j^0, b_j^0, c_j^0, d_j^0)$). Sample size is $N = 1000$.

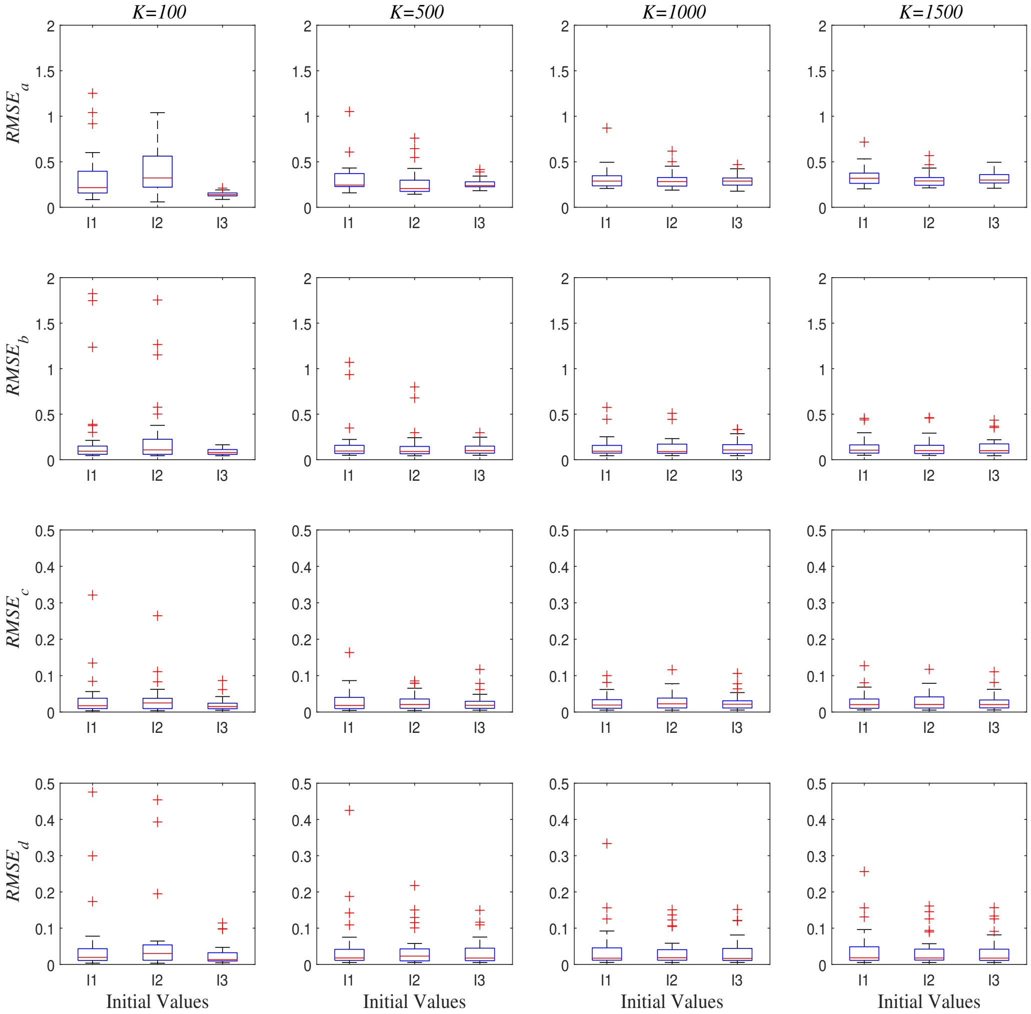

Figure 3: Box-plots of RMSEs of the MMAP estimates of a_j, b_j, c_j and d_j over all items in the test, and the MMAP estimates are computed by the SAEM algorithms under 4 levels of the number (K) of $\gamma_k = 1$ $(K = 100, 500, 1000$ and $1500)$ and 3 groups of initial values (I1: randomly selection, where $a^0 \sim U(0,3), b^0 \sim U(-2,2), c^0 \sim U(0,0.3)$, $d^0 \sim U(0.7,1)$; I2: a group of common values, where $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2, d_j^0 = 0.8$; I3: the true values of (a_j, b_j, c_j, d_j) are used as $(a_j^0, b_j^0, c_j^0, d_j^0)$). Sample size is $N = 5000$.

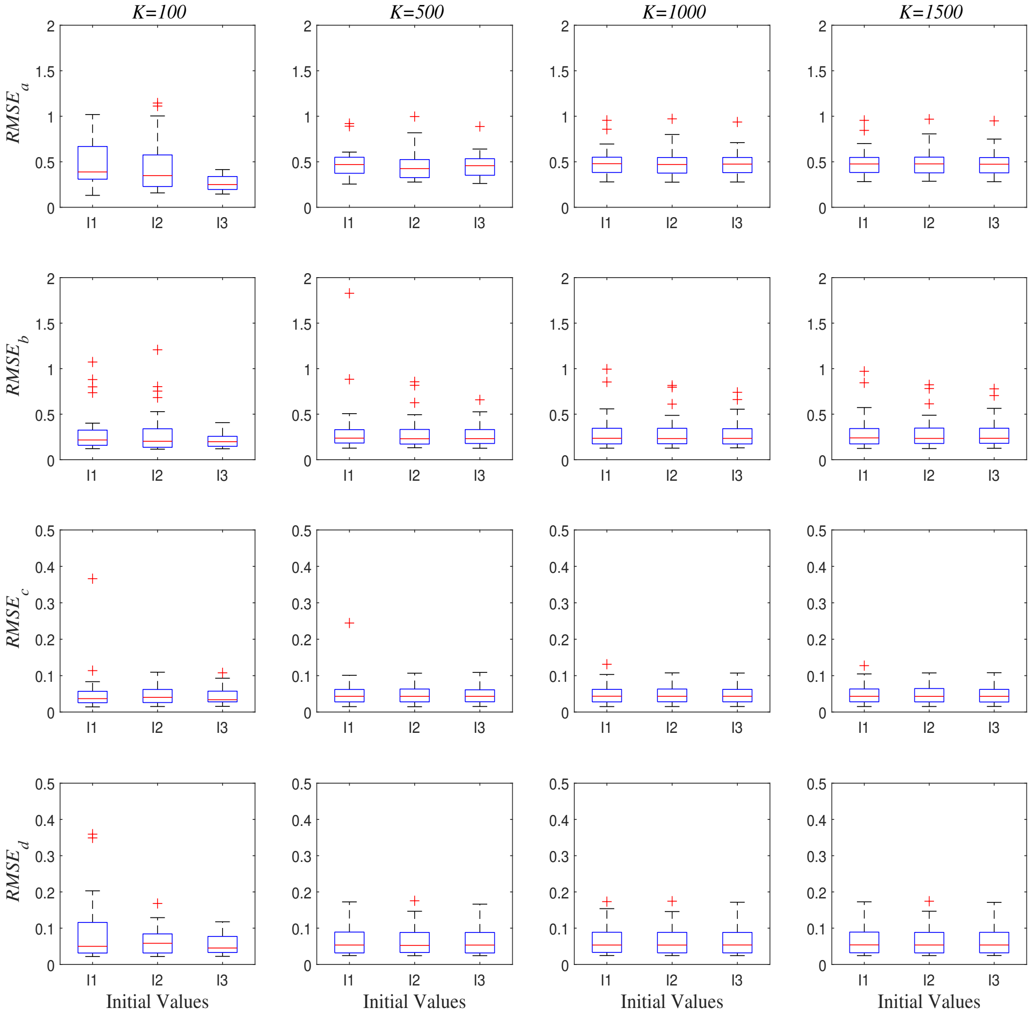

Figure 4: Box-plots of RMSEs of the MMAP estimates of a_j, b_j, c_j and d_j over all items in the test, and the MMAP estimates are computed by the MSAEM algorithms under 4 levels of the number (K) of $\gamma_k = 1$ $(K = 100, 500, 1000$ and $1500)$ and 3 groups of initial values (I1: randomly selection, where $a^0 \sim U(0, 3), b^0 \sim U(-2, 2), c^0 \sim U(0, 0.3)$, $d^0 \sim U(0.7, 1)$; I2: a group of common values, where $a^0_j = 1, b^0_j = 0, c^0_j = 0.2, d^0_j = 0.8$; I3: the true values of (a_j, b_j, c_j, d_j) are used as $(a^0_j, b^0_j, c^0_j, d^0_j)$). Sample size is $N = 500$.

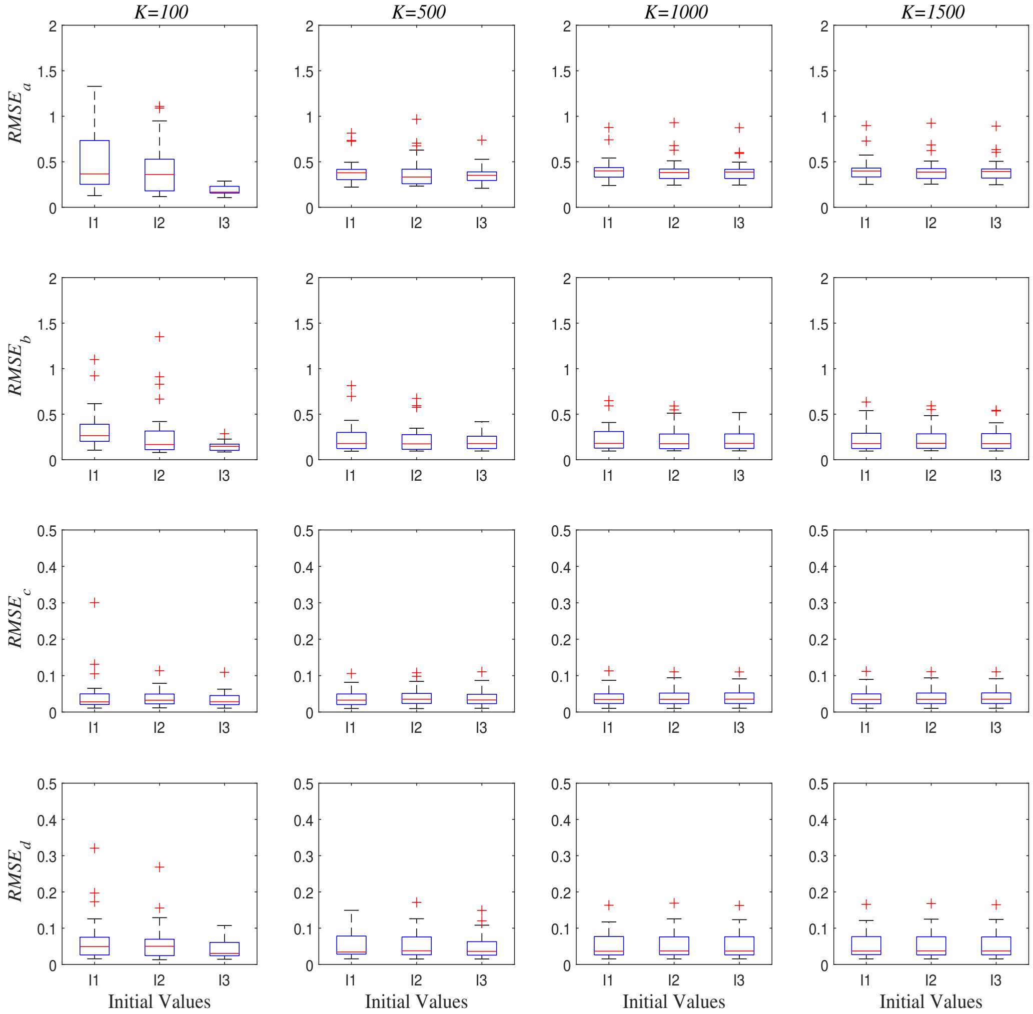

Figure 5: Box-plots of RMSEs of the MMAP estimates of a_j, b_j, c_j and d_j over all items in the test, and the MMAP estimates are computed by the MSAEM algorithms under 4 levels of the number (K) of $\gamma_k = 1$ ($K = 100, 500, 1000$ and 1500) and 3 groups of initial values (I1: randomly selection, where $a^0 \sim U(0,3), b^0 \sim U(-2,2), c^0 \sim U(0,0.3)$, $d^0 \sim U(0.7,1)$; I2: a group of common values, where $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2, d_j^0 = 0.8$; I3: the true values of (a_j, b_j, c_j, d_j) are used as $(a_j^0, b_j^0, c_j^0, d_j^0)$). Sample size is $N = 1000$.

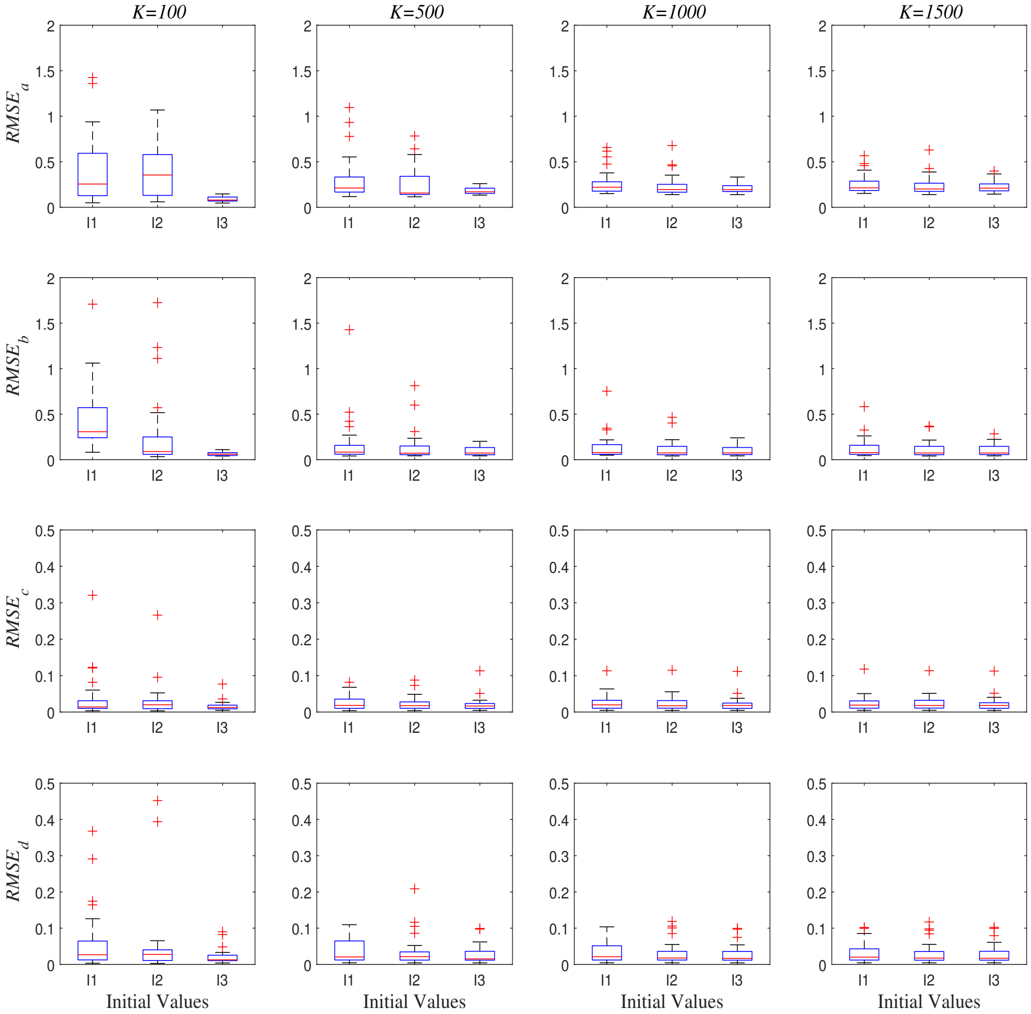

Figure 6: Box-plots of RMSEs of the MMAP estimates of a_j, b_j, c_j and d_j over all items in the test, and the MMAP estimates are computed by the MSAEM algorithms under 4 levels of the number (K) of $\gamma_k = 1$ ($K = 100, 500, 1000$ and 1500) and 3 groups of initial values (I1: randomly selection, where $a^0 \sim U(0,3), b^0 \sim U(-2,2), c^0 \sim U(0,0.3)$, $d^0 \sim U(0.7,1)$; I2: a group of common values, where $a_j^0 = 1, b_j^0 = 0, c_j^0 = 0.2, d_j^0 = 0.8$; I3: the true values of (a_j, b_j, c_j, d_j) are used as $(a_j^0, b_j^0, c_j^0, d_j^0)$). Sample size is $N = 5000$.

Table 2: ARMSE, ACor and AIRF of the estimates of a_j, b_j, c_j and d_j across the test for SAEM, MSAEM, MCEM and Gibbs Sampler, under the four priors of $\boldsymbol{\xi}_j = (a_j, b_j, c_j, d_j)$ and two sample sizes ($N = 1000$ and 5000). The test length is $M = 20$.

			ARMSE				ACor				AIRF
Sample Size	Prior	Algorithm	a	b	c	d	a	b	c	d	
$N = 1000$	Prior 1	MSAEM	0.40	0.29	0.05	0.05	0.74	0.92	0.86	0.76	0.0306
		SAEM	0.48	0.34	0.06	0.06	0.66	0.90	0.82	0.70	0.0330
		MCEM	0.41	0.30	0.05	0.05	0.74	0.92	0.86	0.75	0.0308
		Gibbs Sampler	0.47	0.31	0.05	0.05	0.72	0.92	0.87	0.75	0.0319
	Prior 2	MSAEM	0.44	0.33	0.05	0.09	0.67	0.90	0.86	0.56	0.0329
		SAEM	0.55	0.39	0.06	0.10	0.56	0.85	0.81	0.46	0.0373
		MCEM	0.44	0.33	0.05	0.09	0.67	0.90	0.86	0.56	0.0328
		Gibbs Sampler	0.46	0.32	0.05	0.06	0.73	0.90	0.89	0.67	0.0332
	Prior 3	MSAEM	0.45	0.31	0.05	0.05	0.75	0.93	0.85	0.75	0.0308
		SAEM	0.57	0.35	0.06	0.06	0.67	0.90	0.81	0.70	0.0332
		MCEM	0.46	0.31	0.05	0.05	0.75	0.92	0.85	0.76	0.0309
		Gibbs Sampler	3.30	1.28	0.06	0.07	0.41	0.85	0.85	0.75	0.0470
	Prior 4	MSAEM	0.51	0.35	0.05	0.09	0.64	0.90	0.86	0.53	0.0347
		SAEM	0.64	0.42	0.06	0.10	0.53	0.86	0.83	0.47	0.0382
		MCEM	0.50	0.36	0.06	0.09	0.64	0.90	0.86	0.54	0.0346
		Gibbs Sampler	3.55	1.20	0.06	0.08	0.22	0.80	0.85	0.59	0.0477
$N = 5000$	Prior 1	MSAEM	0.29	0.17	0.03	0.04	0.86	0.97	0.93	0.83	0.0176
		SAEM	0.33	0.18	0.03	0.04	0.83	0.97	0.92	0.79	0.0190
		MCEM	0.29	0.17	0.03	0.04	0.86	0.97	0.98	0.82	0.0174
		Gibbs Sampler	0.33	0.17	0.03	0.04	0.88	0.94	0.95	0.84	0.0181
	Prior 2	MSAEM	0.32	0.18	0.03	0.06	0.83	0.97	0.93	0.71	0.0185
		SAEM	0.39	0.20	0.04	0.07	0.76	0.96	0.90	0.63	0.0213
		MCEM	0.33	0.19	0.03	0.06	0.83	0.97	0.93	0.68	0.0188
		Gibbs Sampler	0.35	0.16	0.03	0.05	0.82	0.97	0.95	0.75	0.0183
	Prior 3	MSAEM	0.28	0.15	0.03	0.04	0.87	0.98	0.93	0.84	0.0173
		SAEM	0.34	0.18	0.04	0.05	0.82	0.97	0.91	0.81	0.0198
		MCEM	0.29	0.16	0.03	0.04	0.86	0.98	0.94	0.82	0.0175
		Gibbs Sampler	0.69	0.26	0.04	0.03	0.79	0.97	0.92	0.96	0.0247
	Prior 4	MSAEM	0.31	0.17	0.03	0.06	0.85	0.97	0.94	0.71	0.0180
		SAEM	0.39	0.20	0.04	0.07	0.77	0.96	0.92	0.63	0.0212
		MCEM	0.33	0.18	0.03	0.06	0.83	0.97	0.93	0.70	0.0185
		Gibbs Sampler	0.72	0.27	0.03	0.05	0.72	0.96	0.94	0.75	0.0212

Note: Prior 1: (μ_0, Σ_0^{-1})= ($\mathbf{0}_2$,$(2\mathbf{I}_2)^{-1}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 2: (μ_0, Σ_0^{-1}) = ($\mathbf{0}_2$, $(2\mathbf{I}_2)^{-1}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$
Prior 3: (μ_0, Σ_0^{-1})= ($\mathbf{0}_2$,$\mathbf{0}_{2\times2}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 4: (μ_0, Σ_0^{-1})= ($\mathbf{0}_2$,$\mathbf{0}_{2\times2}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

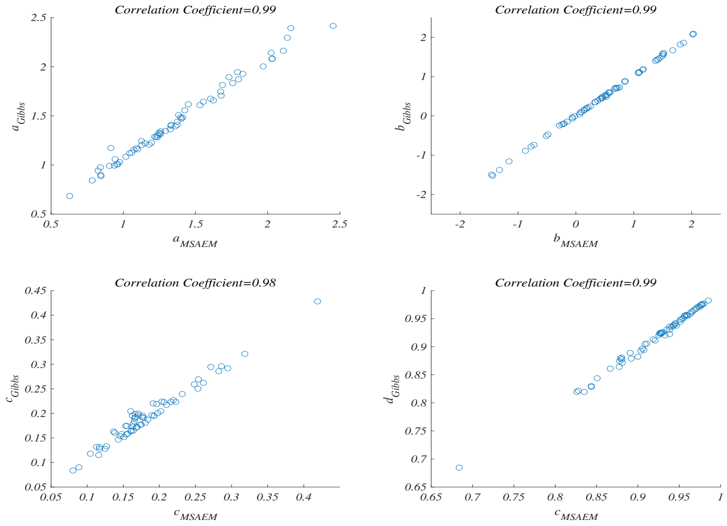

Figure 7: The scatter plots between the estimators of a_j (or b_j, c_j, d_j) obtained using the MSAEM and the Gibbs sampler, across the test.

Table 3: ARMSE, ACor and AIRF of the estimates of a_j, b_j, c_j and d_j across the test for SAEM, MSAEM, MCEM and Gibbs Sampler, under the four priors of $\boldsymbol{\xi}_j = (a_j, b_j, c_j, d_j)$ and two sample sizes ($N = 1000$ and 5000). The test length is $M = 40$.

			ARMSE				ACor				
Sample Size	Prior	Algorithm	a	b	c	d	a	b	c	d	AIRF
$N = 1000$	Prior 1	MSAEM	0.37	0.25	0.04	0.04	0.83	0.97	0.82	0.80	0.0272
		SAEM	0.44	0.28	0.05	0.05	0.78	0.96	0.78	0.77	0.0292
		MCEM	0.39	0.25	0.05	0.04	0.82	0.97	0.81	0.82	0.0277
		Gibbs Sampler	0.43	0.26	0.04	0.05	0.83	0.97	0.84	0.80	0.0273
	Prior 2	MSAEM	0.39	0.33	0.06	0.08	0.82	0.90	0.79	0.32	0.0286
		SAEM	0.46	0.32	0.07	0.08	0.76	0.90	0.60	0.63	0.0313
		MCEM	0.41	0.32	0.07	0.07	0.80	0.90	0.76	0.33	0.0290
		Gibbs Sampler	0.43	0.30	0.05	0.06	0.78	0.93	0.81	0.43	0.0270
	Prior 3	MSAEM	0.38	0.25	0.05	0.04	0.83	0.97	0.81	0.82	0.0272
		SAEM	0.52	0.29	0.05	0.05	0.76	0.95	0.77	0.81	0.0298
		MCEM	0.42	0.28	0.05	0.05	0.83	0.97	0.81	0.81	0.0280
		Gibbs Sampler	2.38	1.14	0.06	0.06	0.51	0.87	0.80	0.79	0.0377
	Prior 4	MSAEM	0.42	0.35	0.07	0.07	0.79	0.90	0.76	0.31	0.0305
		SAEM	0.55	0.35	0.07	0.07	0.75	0.92	0.73	0.71	0.0327
		MCEM	0.47	0.34	0.07	0.08	0.76	0.89	0.72	0.32	0.0302
		Gibbs Sampler	2.61	1.05	0.07	0.07	0.31	0.80	0.75	0.49	0.0389
$N = 5000$	Prior 1	MSAEM	0.23	0.14	0.03	0.03	0.94	0.99	0.92	0.84	0.0140
		SAEM	0.26	0.15	0.03	0.03	0.92	0.96	0.90	0.84	0.0150
		MCEM	0.24	0.15	0.05	0.03	0.94	0.99	0.92	0.82	0.0142
		Gibbs Sampler	0.25	0.14	0.03	0.03	0.93	0.99	0.92	0.86	0.0142
	Prior 2	MSAEM	0.25	0.18	0.03	0.05	0.94	0.94	0.90	0.34	0.0143
		SAEM	0.29	0.20	0.04	0.06	0.90	0.93	0.88	0.37	0.0157
		MCEM	0.25	0.18	0.03	0.05	0.93	0.94	0.90	0.38	0.0144
		Gibbs Sampler	0.27	0.17	0.03	0.05	0.92	0.96	0.91	0.50	0.0142
	Prior 3	MSAEM	0.22	0.13	0.03	0.03	0.94	0.99	0.92	0.88	0.0139
		SAEM	0.26	0.16	0.04	0.03	0.92	0.99	0.89	0.87	0.0153
		MCEM	0.23	0.14	0.03	0.03	0.94	0.99	0.92	0.85	0.0141
		Gibbs Sampler	0.44	0.23	0.03	0.03	0.87	0.97	0.92	0.88	0.0156
	Prior 4	MSAEM	0.25	0.18	0.03	0.05	0.94	0.94	0.90	0.37	0.0147
		SAEM	0.28	0.20	0.04	0.06	0.91	0.93	0.88	0.37	0.0161
		MCEM	0.25	0.18	0.03	0.05	0.94	0.94	0.90	0.40	0.0147
		Gibbs Sampler	0.46	0.23	0.04	0.04	0.83	0.96	0.91	0.56	0.0156

Note: Prior 1: $(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2,(2\mathbf{I}_2)^{-1})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 2: $(\mu_0,\ \Sigma_0^{-1}) = (\mathbf{0}_2, (2\mathbf{I}_2)^{-1})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$
Prior 3: $(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2,\mathbf{0}_{2\times2})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 4: $(\mu_0, \Sigma_0^{-1})= (\mathbf{0}_2,\mathbf{0}_{2\times2})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Table 4: ARMSE, ACor and AIRF of the estimates of a_j, b_j, c_j and d_j across the test for SAEM, MSAEM, MCEM and Gibbs Sampler, under the four priors of $\boldsymbol{\xi}_j = (a_j, b_j, c_j, d_j)$ and two test lengths ($M = 20$ and 40). The sample size is $N = 500$.

			ARMSE				ACor				AIRF
Test Length	Prior	Algorithm	a	b	c	d	a	b	c	d	
$M = 20$	Prior 1	MSAEM	0.46	0.36	0.06	0.05	0.67	0.87	0.80	0.71	0.0388
		SAEM	0.56	0.43	0.07	0.06	0.59	0.84	0.76	0.65	0.0411
		MCEM	0.46	0.36	0.06	0.06	0.67	0.87	0.80	0.71	0.0388
		Gibbs Sampler	1.07	2.39	0.08	0.57	0.00	0.12	0.53	-0.39	0.0766
	Prior 2	MSAEM	0.49	0.42	0.06	0.11	0.58	0.83	0.81	0.48	0.0430
		SAEM	0.63	0.50	0.07	0.12	0.45	0.76	0.78	0.42	0.0466
		MCEM	0.50	0.42	0.06	0.11	0.58	0.83	0.81	0.49	0.0430
		Gibbs Sampler	0.80	1.20	0.10	0.21	0.00	0.02	0.34	0.47	0.0966
	Prior 3	MSAEM	0.52	0.38	0.06	0.06	0.67	0.88	0.79	0.70	0.0397
		SAEM	0.65	0.46	0.07	0.07	0.57	0.84	0.74	0.62	0.0428
		MCEM	0.50	0.38	0.06	0.06	0.67	0.88	0.79	0.70	0.0396
		Gibbs Sampler	2.20	9.53	0.08	0.57	0.01	0.03	0.54	-0.17	0.0778
	Prior 4	MSAEM	0.57	0.45	0.08	0.11	0.53	0.83	0.77	0.48	0.0453
		SAEM	0.71	0.48	0.10	0.11	0.46	0.81	0.64	0.43	0.0495
		MCEM	0.61	0.48	0.07	0.11	0.55	0.83	0.81	0.45	0.0453
		Gibbs Sampler	4.80	6.66	0.09	0.24	−0.06	0.12	0.44	0.15	0.0825
$M = 40$	Prior 1	MSAEM	0.42	0.32	0.05	0.05	0.79	0.95	0.77	0.79	0.0339
		SAEM	0.50	0.37	0.06	0.06	0.72	0.93	0.69	0.71	0.0357
		MCEM	0.44	0.31	0.06	0.06	0.76	0.95	0.74	0.74	0.0353
		Gibbs Sampler	0.46	0.31	0.05	0.05	0.76	0.95	0.76	0.74	0.0353
	Prior 2	MSAEM	0.49	0.43	0.07	0.09	0.68	0.88	0.70	0.30	0.0380
		SAEM	0.58	0.43	0.07	0.09	0.66	0.90	0.69	0.61	0.0428
		MCEM	0.51	0.43	0.08	0.09	0.68	0.87	0.71	0.34	0.0393
		Gibbs Sampler	0.49	0.37	0.07	0.07	0.71	0.91	0.72	0.39	0.0357
	Prior 3	MSAEM	0.49	0.32	0.06	0.05	0.79	0.95	0.75	0.75	0.0352
		SAEM	0.55	0.40	0.06	0.06	0.72	0.92	0.69	0.73	0.0375
		MCEM	0.50	0.43	0.06	0.08	0.74	0.95	0.76	0.76	0.0358
		Gibbs Sampler	4.80	2.19	0.08	0.07	0.33	0.83	0.68	0.76	0.0530
	Prior 4	MSAEM	0.50	0.42	0.08	0.09	0.70	0.88	0.72	0.32	0.0405
		SAEM	0.61	0.45	0.09	0.09	0.60	0.91	0.69	0.62	0.0450
		MCEM	0.55	0.45	0.08	0.09	0.68	0.88	0.69	0.40	0.0412
		Gibbs Sampler	5.35	2.29	0.10	0.09	0.12	0.72	0.67	0.48	0.0556

Note: Prior 1: (μ_0, Σ_0^{-1})= ($\mathbf{0}_2$,$(2\mathbf{I}_2)^{-1}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 2: (μ_0, Σ_0^{-1}) = ($\mathbf{0}_2$, $(2\mathbf{I}_2)^{-1}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$
Prior 3: (μ_0, Σ_0^{-1})= ($\mathbf{0}_2$,$\mathbf{0}_{2\times 2}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 4: (μ_0, Σ_0^{-1})= ($\mathbf{0}_2$,$\mathbf{0}_{2\times 2}$) and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Table 5: The average computing times of SAEM, MSAEM, MCEM, and Gibbs Sampler, under two sample sizes ($N = 1000$ and 5000) and two test lengths ($M = 20$ and 40) across the 200 replications.

Testing Conditions		Average Computing Time (in Second)			
Sample Size	Test Length	SAEM	MSAEM	MCEM	Gibbs Sampler
$N = 1000$	$M = 20$	6	4	135	166
$N = 5000$	$M = 20$	20	15	550	771
$N = 1000$	$M = 40$	11	5	240	332
$N = 5000$	$M = 40$	65	46	1650	1692

Table 6: The S-X^2 statistics for 4PNO, 3PNO and 2PNO models at the item level.

	4PNO	3PNO	2PNO
Item 1	44.27	64.69	75.31
Item 2	53.62	69.60	92.13*
Item 3	45.14	51.32	65.59
Item 4	83.92*	86.67*	100.24*
Item 5	47.93	47.79	49.39
Item 6	39.93	41.12	40.15
Item 7	59.71	75.38	163.33*
Item 8	50.64	101.96*	226.04*
Item 9	50.73	65.51	104.41*
Item 10	39.97	39.08	84.94*
Item 11	45.19	68.23	92.74*
Item 12	51.70	55.39	134.12*
Item 13	42.77	45.50	49.36
Item 14	63.90	65.98	77.46
Item 15	42.43	43.04	46.06
Item 16	44.02	54.97	72.49
Item 17	62.68	74.10	86.16*
Item 18	50.12	51.44	53.66
Item 19	49.41	50.13	51.77
Item 20	54.52	66.09	109.74*
Item 21	48.12	50.94	50.93
Item 22	49.83	62.81	83.15*
Item 23	49.84	54.00	80.76
Item 24	48.74	49.66	54.54
Item 25	66.44	71.43	90.42*
Item 26	37.62	36.86	38.66
Item 27	58.77	71.29	81.83*
Item 28	38.89	41.40	55.65
Item 29	51.64	52.96	59.32
Item 30	59.00	58.58	65.30
Item 31	64.73	62.83	61.52
Item 32	60.49	55.39	61.24
Item 33	56.61	55.90	56.26
Item 34	57.37	56.54	63.23
Item 35	35.83	49.99	78.53
Item 36	48.32	49.13	47.61
Item 37	34.95	36.03	45.24
Item 38	44.49	79.93	98.39*
Item 39	47.37	53.06	52.03
Item 40	53.97	54.50	50.47
Item 41	42.46	42.24	65.10
Item 42	54.10	47.92	54.24
Item 43	47.20	59.59	78.31
Item 44	73.84	68.50	79.61
Item 45	58.04	64.57	82.04*
Item 46	53.62	54.65	69.72
Item 47	38.63	40.24	44.94
Item 48	54.22	66.99	89.88*
Item 49	43.43	43.82	43.15
Item 50	43.82	45.58	50.36
Item 51	53.80	58.56	69.49
Item 52	52.49	50.24	109.89*
Item 53	54.92	51.86	65.30
Item 54	50.17	48.57	53.23
Item 55	58.71	67.83	90.89*
Item 56	50.81	54.71	49.72
Item 57	71.09	71.21	73.46
Item 58	64.67	75.02	87.27*
Item 59	44.03	45.04	71.55
Item 60	40.27	40.72	47.71
Item 61	38.77	39.39	41.12
Item 62	48.99	54.52	61.87
Item 63	72.25	87.89*	94.91*
Item 64	61.31	74.14	174.88*
Item 65	50.82	52.31	50.86

Note:* denotes S-X^2 is greater than the critical values at the significance level of %5.

Table 7: The values of AIC for 4PNO, 3PNO, 2PNO and a hybrid model of the three models (2PNO, 3PNO and 4PNO) at the test level.

	$-2\ln L(\mathbf{u} \mid \boldsymbol{\xi})$	$2n_p$	AIC
4PNO	132891	520	133411
3PNO	133081	390	133471
2PNO	133590	260	133850
Hybrid model	132916	472	133388

Hybrid model: is the combination of 2PNO, 3PNO and 4PNO

Appendix: The Simulation Result on the Recovery Accuracy of the 4PL Model

In this simulation, the 4PL is the data-generating model,

$$P_i(\theta_j) = P(U_{ij} = 1|\theta_i, \boldsymbol{\xi}_j) = c_j + (d_j - c_j)\frac{\exp[D(a_j\theta_i + b_j)]}{1 + \exp[D(a_j\theta_i + b_j)]},$$

where $D = 1.702$ is the scale constant. To keep comparability with the results shown in Table 2-4, the testing conditions as well as the true values of a_j, b_j, c_j and d_j are the same as that of "Study 2". The MMAP estimate of the 4PL model is computed by the EM algorithm implemented in the R package of "mirt" and the EM algorithm proposed by Meng et al. (2020) separately.

The prior for (a_j, b_j) is $(\ln a_j, b_j)' \sim N_2(\mu_0, \Sigma_0)$, which is different from that for the 4PNO model, since the above two EM algorithms are implemented under a lognormal prior for a_j. The prior for (c_j, d_j) is the same as that of the 4PNO, which is a bivariate Beta distribution given in Equation 19.

Following the design of "Study 2", the MMAP estimate of the 4PL model is computed under the four different priors, please see Table A1. Note that the variance of $\ln a_j$ is 1, which is to make the prior information close to the truncated normal distribution in Equation 18 with the variance is 2. The same as that of "Study 2", the simulation study generated 200 replications, and the parameter recovery is assessed by computing the three criteria (ARMSE, ACor and AIRF) across the 200 replications. The obtained results are given in Tables A2 and A3.

Table A1: Four prior distributions for the item parameters (i.e., a_j, b_j, c_j and d_j) in the 4PL.

	$(\ln a_j, b_j) \sim N_2(\mu_0, \Sigma_0)$	$(c_j, d_j) \sim Beta_2(\alpha_c, \beta_c, \alpha_d, \beta_d) I_{(1 \geq d_j \geq c_j \geq 0)}$
Prior 1 (Informative+Informative)	$(\mu_0,\ \Sigma_0^{-1}) = \left(\mathbf{0}_2, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\right)$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 2 (Informative+Noninformative)	$(\mu_0,\ \Sigma_0^{-1}) = \left(\mathbf{0}_2, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\right)$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$
Prior 3 (Noninformative+Informative)	$(\mu_0,\ \Sigma_0^{-1}) = (\mathbf{0}_2, \mathbf{0}_{2\times 2})$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$
Prior 4 (Noninformative+Noninformative)	$(\mu_0,\ \Sigma_0^{-1}) = (\mathbf{0}_2, \mathbf{0}_{2\times 2})$	$(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Note: $\mathbf{0}_2$: two-dimensional vector of zeros; $\mathbf{0}_{2\times 2}$: 2 × 2 matrix of zeros.

Acknowledgements

The authors are greatly indebted to the editor, an associate editor, and three anonymous reviewers for their valuable comments and suggestions. Meng is partially supported by the National Natural Science Foundation of China (11501094, 11571069), and Xu is partially supported by National Science Foundation (SES-1846747) and Institute of Education Sciences (R305D200015).

Table A2: ARMSE, ACor and AIRF of the estimates of 4PL model obtained by the EM algorithm implemented by the R package of "mirt" under the 24 simulation conditions (two test lengths ($M = 20$ and 40)× three sample sizes($N = 500, 1000$ and 5000) × four priors of (a_j, b_j, c_j, d_j)).

			ARMSE				ACor				AIRF
Test Length	Sample Size	Prior	a	b	c	d	a	b	c	d	
$M = 20$	$N = 500$	Prior1	0.65	0.36	0.07	0.06	0.47	0.85	0.75	0.62	0.0462
		Prior2	0.80	0.41	0.15	0.14	0.49	0.84	0.40	0.27	0.0505
		Prior3	0.65	0.42	0.07	0.06	0.39	0.82	0.69	0.54	0.0454
		Prior4	5.16	1.86	0.12	0.13	0.13	0.66	0.50	0.35	0.0584
	$N = 1000$	Prior1	0.56	0.32	0.06	0.06	0.49	0.88	0.75	0.67	0.0380
		Prior2	0.68	0.34	0.12	0.12	0.57	0.89	0.52	0.36	0.0399
		Prior3	0.58	0.34	0.06	0.06	0.48	0.88	0.72	0.65	0.0371
		Prior4	2.20	0.79	0.10	0.11	0.31	0.81	0.59	0.45	0.0431
	$N = 5000$	Prior1	0.45	0.20	0.05	0.05	0.59	0.96	0.83	0.79	0.0249
		Prior2	0.43	0.19	0.06	0.07	0.77	0.97	0.79	0.66	0.0210
		Prior3	0.47	0.21	0.05	0.05	0.60	0.96	0.82	0.77	0.0252
		Prior4	0.55	0.22	0.05	0.06	0.73	0.97	0.71	0.73	0.0202
$M = 40$	$N = 500$	Prior1	0.59	0.44	0.06	0.07	0.68	0.87	0.73	0.71	0.0553
		Prior2	0.70	0.47	0.13	0.12	0.70	0.83	0.18	0.54	0.0517
		Prior3	0.52	0.47	0.06	0.07	0.71	0.88	0.75	0.71	0.0496
		Prior4	3.47	1.55	0.09	0.10	0.28	0.70	0.54	0.58	0.0458
	$N = 1000$	Prior1	0.45	0.39	0.05	0.06	0.75	0.91	0.79	0.79	0.0424
		Prior2	0.49	0.35	0.08	0.07	0.76	0.88	0.83	0.72	0.0290
		Prior3	0.41	0.44	0.05	0.06	0.77	0.90	0.79	0.79	0.0396
		Prior4	1.06	0.61	0.07	0.07	0.56	0.89	0.63	0.69	0.0317
	$N = 5000$	Prior1	0.32	0.21	0.04	0.04	0.88	0.97	0.87	0.89	0.0222
		Prior2	0.28	0.18	0.06	0.04	0.92	0.95	0.34	0.84	0.0148
		Prior3	0.36	0.33	0.04	0.04	0.79	0.88	0.88	0.90	0.0221
		Prior4	0.32	0.17	0.04	0.04	0.91	0.98	0.73	0.85	0.0152

Note: Prior 1: $(\mu_0,\ \Sigma_0^{-1})= \left(\mathbf{0}_2, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\right)$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$

Prior 2: $(\mu_0,\ \Sigma_0^{-1})= \left(\mathbf{0}_2, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\right)$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Prior 3: $(\mu_0,\ \Sigma_0^{-1})= (\mathbf{0}_2, \mathbf{0}_{2\times 2})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$

Prior 4: $(\mu_0, \Sigma_0^{-1})= (\mathbf{0}_2, \mathbf{0}_{2\times 2})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Table A3: ARMSE, ACor and AIRF of the estimates of 4PL model obtained by the EM algorithm proposed by Meng et al. (2020) under the 24 simulation conditions (two test lengths ($M = 20$ and 40) × three sample sizes($N = 500, 1000$ and 5000) × four priors of (a_j, b_j, c_j, d_j)).

Test Length	Sample Size	Prior	ARMSE				ACor				AIRF
			a	b	c	d	a	b	c	d	
$M = 20$	$N = 500$	Prior1	0.59	0.33	0.06	0.06	0.50	0.89	0.73	0.65	0.0380
		Prior2	0.69	0.44	0.11	0.12	0.41	0.80	0.64	0.55	0.0475
		Prior3	8.14	3.81	0.07	0.07	0.29	0.64	0.62	0.49	0.0445
		Prior4	13.12	5.02	0.11	0.12	0.09	0.54	0.58	0.46	0.0578
	$N = 1000$	Prior1	0.53	0.27	0.06	0.06	0.57	0.93	0.79	0.73	0.0305
		Prior2	0.63	0.34	0.08	0.09	0.47	0.89	0.72	0.61	0.0367
		Prior3	4.46	2.07	0.06	0.06	0.39	0.75	0.70	0.61	0.0356
		Prior4	6.76	2.61	0.09	0.09	0.21	0.67	0.67	0.55	0.0436
	$N = 5000$	Prior1	0.35	0.16	0.03	0.04	0.81	0.98	0.91	0.85	0.0168
		Prior2	0.41	0.18	0.04	0.06	0.75	0.97	0.89	0.78	0.0198
		Prior3	0.62	0.31	0.04	0.04	0.76	0.96	0.89	0.83	0.0181
		Prior4	0.75	0.31	0.04	0.06	0.66	0.95	0.88	0.77	0.0206
$M = 40$	$N = 500$	Prior1	0.51	0.35	0.05	0.05	0.72	0.93	0.76	0.78	0.0347
		Prior2	0.60	0.45	0.10	0.09	0.63	0.82	0.29	0.64	0.0396
		Prior3	6.20	4.94	0.06	0.06	0.33	0.68	0.76	0.76	0.0383
		Prior4	9.10	5.83	0.17	0.11	0.08	0.54	0.30	0.59	0.0447
	$N = 1000$	Prior1	0.43	0.28	0.04	0.05	0.81	0.96	0.81	0.85	0.0269
		Prior2	0.49	0.35	0.08	0.07	0.76	0.88	0.83	0.72	0.0290
		Prior3	2.15	1.51	0.05	0.06	0.51	0.80	0.81	0.84	0.0285
		Prior4	3.48	1.96	0.08	0.08	0.25	0.68	0.33	0.68	0.0321
	$N = 5000$	Prior1	0.24	0.14	0.03	0.03	0.94	0.99	0.87	0.93	0.0131
		Prior2	0.30	0.19	0.05	0.04	0.92	0.94	0.37	0.87	0.0155
		Prior3	0.34	0.33	0.04	0.04	0.80	0.89	0.88	0.90	0.0221
		Prior4	0.38	0.20	0.05	0.05	0.83	0.95	0.37	0.86	0.0151

Note: Prior 1: $(\mu_0,\, \Sigma_0^{-1})= \left(\mathbf{0}_2, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\right)$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$

Prior 2: $(\mu_0,\, \Sigma_0^{-1})= \left(\mathbf{0}_2, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\right)$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

Prior 3: $(\mu_0,\, \Sigma_0^{-1})= (\mathbf{0}_2, \mathbf{0}_{2\times 2})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (5, 17, 17, 5)$

Prior 4: $(\mu_0, \Sigma_0^{-1})= (\mathbf{0}_2, \mathbf{0}_{2\times 2})$ and $(\alpha_c, \beta_c, \alpha_d, \beta_d) = (1, 1, 1, 1)$

References

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer.

Allassonnière, S., Kuhn, E., Trouvé, A., et al. (2010). Construction of bayesian deformable models via a stochastic approximation algorithm: a convergence study. *Bernoulli*, 16(3):641–678.

Baker, F. B. and Kim, S.-H. (2004). *Item response theory: Parameter estimation techniques*. CRC Press.

Barton, M. A. and Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. *ETS Research Report Series*, 1981(1):i–8.

Battauz, M. (2020). Regularized estimation of the four-parameter logistic model. *Psych*, 2(4):269–278.

Béguin, A. A. and Glas, C. A. (2001). MCMC estimation and some model-fit analysis of multidimensional IRT models. *Psychometrika*, 66(4):541–561.

Berger, J. O. (1990). Robust bayesian analysis: sensitivity to the prior. *Journal of statistical planning and inference*, 25(3):303–328.

Camilli, G. and Fox, J.-P. (2015). An aggregate IRT procedure for exploratory factor analysis. *Journal of educational and behavioral statistics*, 40(4):377–401.

Camilli, G. and Geis, E. (2019). Stochastic approximation EM for large-scale exploratory IRT factor analysis. *Statistics in medicine*, 38(21):3997–4012.

Celeux, G., Hurn, M., and Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970.

Culpepper, S. A. (2016). Revisiting the 4-parameter item response model: Bayesian estimation and application. *Psychometrika*, 81(4):1142–1163.

Culpepper, S. A. (2017). The prevalence and implications of slipping on low-stakes, large-scale assessments. *Journal of Educational and Behavioral Statistics*, 42(6):706–725.

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

DeMars, C. E. (2012). A comparison of limited-information and full-information methods in M plus for estimating item response theory parameters for nonnormal populations. *Structural Equation Modeling: A Multidisciplinary Journal*, 19(4):610–632.

Feuerstahler, L. M. and Waller, N. G. (2014). Estimation of the 4-parameter model with marginal maximum likelihood. *Multivariate behavioral research*, 49(3):285.

Fox, J.-P. (2003). Stochastic EM for estimating the parameters of a multilevel IRT model. *British Journal of Mathematical and Statistical Psychology*, 56(1):65–81.

Galarza, C. E., Lachos, V. H., and Bandyopadhyay, D. (2017). Quantile regression in linear mixed models: a stochastic approximation em approach. *Statistics and its Interface*, 10(3):471.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Gu, M. G. and Zhu, H.-T. (2001). Maximum likelihood estimation for spatial models by markov chain monte carlo stochastic approximation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):339–355.

Guo, S. and Zheng, C. (2019). The bayesian expectation-maximization-maximization for the 3plm. *Frontiers in psychology*, 10:1175.

Jank, W. (2006). Implementing and diagnosing the stochastic approximation EM algorithm. *Journal of Computational and Graphical Statistics*, 15(4):803–829.

Kern, J. L. and Culpepper, S. A. (2020). A restricted four-parameter IRT model: The dyad four-parameter normal ogive (Dyad-4PNO) model. *Psychometrika*, 85(3):575–599.

Kuhn, E. and Lavielle, M. (2004). Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM: Probability and Statistics*, 8:115–131.

Lavielle, M. and Mbogning, C. (2014). An improved SAEM algorithm for maximum likelihood estimation in mixtures of non linear mixed effects models. *Statistics and Computing*, 24(5):693–707.

Liao, W.-W., Ho, R.-G., Yen, Y.-C., and Cheng, H.-C. (2012). The four-parameter logistic item response theory model as a robust method of estimating ability despite aberrant responses. *Social Behavior and Personality: an international journal*, 40(10):1679–1694.

Loken, E. and Rulison, K. L. (2010). Estimation of a four-parameter item response theory model. *British Journal of Mathematical and Statistical Psychology*, 63(3):509–525.

McKinley, R. L. and Mills, C. N. (1985). A comparison of several goodness-of-fit statistics. *Applied Psychological Measurement*, 9(1):49–57.

Meng, X., Xu, G., Zhang, J., and Tao, J. (2020). Marginalized maximum a posteriori estimation for the four-parameter logistic model under a mixture modelling framework. *British Journal of Mathematical and Statistical Psychology*, 73:51–82.

Meng, X.-L. and Schilling, S. (1996). Fitting full-information item factor models and an empirical investigation of bridge sampling. *Journal of the American Statistical Association*, 91(435):1254–1267.

Orlando, M. and Thissen, D. (2000). Likelihood-based item-fit indices for dichotomous item response theory models. *Applied psychological measurement*, 24(1):50–64.

Orlando, M. and Thissen, D. (2003). Further investigation of the performance of s-x2: An item fit index for use with dichotomous item response theory models. *Applied Psychological Measurement*, 27(4):289–298.

Patsula, L. (1995). *A comparison of item parameter estimates and ICCs produced with TESTGRAF and BILOG under different test lengths and sample sizes.* University of Ottawa (Canada).

Reise, S. P. and Waller, N. G. (2003). How many IRT parameters does it take to model psychopathology items? *Psychological Methods*, 8(2):164–184.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Rulison, K. L. and Loken, E. (2009). I've fallen and I can't get up: can high-ability students recover from early mistakes in CAT? *Applied Psychological Measurement*, 33(2):83–101.

Svetina, D., Valdivia, A., Underhill, S., Dai, S., and Wang, X. (2017). Parameter recovery in multidimensional item response theory models under complexity and nonnormality. *Applied psychological measurement*, 41(7):530–544.

Tang, K. L., Way, W. D., and Carey, P. A. (1993). The effect of small calibration sample sizes on toefl irt-based equating. *ETS Research Report Series*, 1993(2):i–38.

Tao, J., Shi, N.-Z., and Chang, H.-H. (2012). Item-weighted likelihood method for ability estimation in tests composed of both dichotomous and polytomous items. *Journal of Educational and Behavioral Statistics*, 37(2):298–315.

Thissen, D. (1982). Marginal maximum likelihood estimation for the one-parameter logistic model. *Psychometrika*, 47(2):175–186.

von Davier, M. (2009). Is there need for the 3pl model? guess what? *Measurement: Interdisciplinary Research and Perspectives*, 7(2):110–114.

Waller, N. G. and Feuerstahler, L. (2017). Bayesian modal estimation of the four-parameter item response model in real, realistic, and idealized data sets. *Multivariate behavioral research*, 52(3):350–370.

Wang, C., Su, S., and Weiss, D. J. (2018). Robustness of parameter estimation to assumptions of normality in the multidimensional graded response model. *Multivariate behavioral research*, 53(3):403–418.

Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

Wollack, J. A., Bolt, D. M., Cohen, A. S., and Lee, Y.-S. (2002). Recovery of item parameters in the nominal response model: A comparison of marginal maximum likelihood estimation and markov chain monte carlo estimation. *Applied psychological measurement*, 26(3):339–352.

Yen, W. M. (1981). Using simulation results to choose a latent trait model. *Applied Psychological Measurement*, 5(2):245–262.

Yen, W. M. (1987). A comparison of the efficiency and accuracy of bilog and logist. *Psychometrika*, 52(2):275–291.

Yoes, M. (1995). An updated comparison of micro-computer based item parameter estimation procedures used with the 3-parameter irt model. *St. Paul, MN: Assessment Systems Corporation.*

Zhang, J., Du, H., Zhang, Z., and Tao, J. (2020a). Gibbs-slice sampling algorithm for estimating the four-parameter logistic model. *Frontiers in Psychology*, 11:2121.

Zhang, S., Chen, Y., and Liu, Y. (2020b). An improved stochastic EM algorithm for large-scale full-information item factor analysis. *British Journal of Mathematical and Statistical Psychology*, 73(1):44–71.

Zhang, X., Wang, C., Weiss, D. J., and Tao, J. (2020c). Bayesian inference for IRT models with non-normal latent trait distributions. *Multivariate Behavioral Research*, pages 1–21.