Journal of Classification
https://doi.org/10.1007/s00357-021-09392-7

A Gibbs Sampling Algorithm with Monotonicity Constraints for Diagnostic Classification Models

Kazuhiro Yamaguchi[1] · Jonathan Templin[2]

Accepted: 27 May 2021/Published online: 31 July 2021

Abstract
Diagnostic classification models (DCMs) are restricted latent class models with a set of cross-class equality constraints and additional monotonicity constraints on their item parameters, both of which are needed to ensure the meaning of classes and model parameters. In this paper, we develop an efficient, Gibbs sampling-based Bayesian Markov chain Monte Carlo estimation method for general DCMs with monotonicity constraints. A simulation study was conducted to evaluate parameter recovery of the algorithm which showed accurate estimation of model parameters. Moreover, the proposed algorithm was compared to a previously developed Gibbs sampling algorithm which imposed constraints on only the main effect item parameters of the log-linear cognitive diagnosis model. The newly proposed algorithm showed less bias and faster convergence. An analysis of the 2000 Programme for International Student Assessment reading assessment data using this algorithm was also conducted.

Keywords Diagnostic classification models · Markov chain Monte Carlo methods · Gibbs sampling · Bayesian inference

1 Introduction

Diagnostic classification models (DCMs; e.g., Rupp et al., 2010), also referred to as cognitive diagnosis models (CDMs; e.g., Leighton & Gierl, 2007), are latent variable models used to estimate the multidimensional knowledge state of examinees, which can be used to provide effective remedial instruction (e.g., Tatsuoka & Tatsuoka, 1997). DCMs differ from item

Data analysis syntax is available in Open Science Framework, page: https://osf.io/h9fm4/.

✉ Kazuhiro Yamaguchi
yamaguchi.kazuhir.ft@u.tsukuca.ac.jp

[1] Faculty of Human Science, University of Tsukuba, Institutes of Human Sciences, A314, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-0006, Japan

[2] Department of Psychological and Quantitative Foundations, University of Iowa, Iowa City, IA, USA

response theory models (e.g., Embretson & Reise, 2000) in that the latent traits in DCMs, often called attributes, are ordinal categorical variables which are often binary-valued. Therefore, DCMs are versions of restricted latent class models (Rupp & Templin, 2008). The focus of this study is to develop an efficient Bayesian estimation method satisfying a set of monotonicity constraints on a wide class of DCMs.

Although numerous DCMs have been developed, for example the deterministic input noisy "and"- gate model (DINA; e.g., Haertel, 1989; Junker & Sijtsma, 2001; Macready & Dayton, 1977), the deterministic noisy input "or"-gate model (DINO; Templin & Henson, 2006), or the reduced reparameterized unified model (RRUM; Hartz & Roussos, 2008), general DCMs allow for the most flexible modeling options as they subsume the most frequently used latent class-based sub-models. General DCMs include the log-linear cognitive diagnosis model (LCDM; Henson et al., 2009), general diagnostic model (GDM; von Davier, 2008, 2014), and generalized deterministic input noisy "and" gate model (G-DINA; de la Torre, 2011). The equivalency of the three models is provided by de la Torre (2011) and von Davier (2014). If all possible parameters are included in these models, they are also called saturated DCMs (Li, Hunter, & Lei, 2016; Yamaguchi & Okada, 2018), although this term has also been used to describe the DCM structural model (e.g., the model specifying the distributional parameters of the attributes, see Templin & Bradshaw, 2014, or Hu & Templin, 2019).

In the LCDM framework, for a binary item response X_{ij} of an individual i ($i = 1, \ldots, I$) for item j, the item response function of i-th individual for item j is defined as:

$$\mathrm{P}\left(X_{ij} = 1 | \lambda_{j,0}, \boldsymbol{\lambda}_j, \boldsymbol{\alpha}_i = \boldsymbol{\alpha}_c, \boldsymbol{q}_j\right) = \frac{1}{1 + \exp\left(-\left(\lambda_{j0} + \boldsymbol{\lambda}_j^{\top} h\left(\boldsymbol{\alpha}_c, \boldsymbol{q}_j\right)\right)\right)}, \tag{1}$$

where the kernel function $\boldsymbol{\lambda}_j^{\top} h(\boldsymbol{\alpha}_c \ , \boldsymbol{q}_j)$ is represented as

$$\boldsymbol{\lambda}_j^{\top} h\left(\boldsymbol{\alpha}_c, \boldsymbol{q}_j\right) = \sum_{a=1}^{A} \lambda_{j,1,(a)} \alpha_{ca} q_{ja} + \sum_{a=1}^{A-1} \sum_{a' > a}^{A} \lambda_{j,2,\left(a,a'\right)} \alpha_{ca} \alpha_{ca'} q_{ja} q_{ja'} + \cdots. \tag{2}$$

The $\boldsymbol{q}_j = (q_{j1}, \ldots, q_{jA})^{\top}$ is the j-th row vector of the Q-matrix (e.g., Tatsuoka, 1983), a matrix with binary indicators of whether an item j measures an attribute ($a = 1, \ldots, A$). From a latent class model perspective, each class represents a specific attribute mastery pattern. The vector $\boldsymbol{\alpha}_c = (\alpha_{c1}, \ldots, \alpha_{cA})^{\top}$ contains the attribute mastery pattern corresponding to the c-th latent class. Each attribute mastery status indicator α_{ca} is binary valued with zero representing non-masters and one representing masters. Under a general DCM, the total number of latent classes is $C = 2^A$ with each being a specific attribute profile—a permutation of the A binary attributes. A saturated general DCM, thus, has $2^{\sum_a q_{ja}}$ parameters for each item.

In a general latent class model, each item has one parameter which is the probability of a correct response for a person from that class. The Q-matrix, through the helper function $h(\boldsymbol{\alpha}_c, \boldsymbol{q}_j)$ in (1), specifies a set of equality constraints for the item response probability for each class. These constraints enable classes with the same Q-matrix-indicated attributes to have the same item response probability. However, there is another difference between general DCMs and general latent class models: DCM item parameter *monotonicity* constraints. Henson et al. (2009) provided constraints for two reasons: to identify LCDM parameters and to ensure the meaning of the latent classes. *Monotonicity* constraints are "defined as the property such that for any examinee that masters additional skills his or her probability of a correct response must be equal to or greater than the probability of a correct response prior to learning the additional skills" (Henson et al., 2009, p. 198).

Moreover, monotonicity constraints are strongly related to issues of identification in DCMs. Liu et al. (2013) studied identifiability conditions of the DINA model under known parameters. Xu and Zhang (2016) established conditions for the identifiability of the DINA model and Gu and Xu (2019) extended the result and established a set of sufficient and necessary conditions. Fang et al. (2019) established identifiability conditions for a more general class of DCMs. Xu (2017) and Xu and Shang (2018) treated DCMs as restricted latent class models and provided identifiability conditions with monotonicity constraints. Gu and Xu (2020) loosened identifiability conditions for DCMs previously proposed by Xu and Shang (2018). In addition, Chen et al. (2020) proposed generic identifiability conditions for DCMs.

Selecting an appropriate identification rule is important not only to identify the model parameters but also to prevent the latent class model label switching problem (e.g., Stephens, 2000). In DCMs, if class labels are permuted, not only does the meaning of the class change, but also as models are predicated on classes being specific attribute profiles, the values of the model parameters are meaningless. Monotonicity constraints are an essential part of identification and can avoid the incidence of label switching if the identification conditions in Xu and Shang (2018) are satisfied.

To estimate parameters with monotonicity constraints, maximum likelihood (ML)-based algorithms used to estimate DCMs have frequently been employed. Rupp et al. (2010) and Templin and Hoffman (2013) showed parameter estimation procedures under marginal maximum likelihood using Mplus (Muthén & Muthén, 1998-2017). In addition, Hong et al. (2016) proposed ordered restricted maximum likelihood estimation methods for the G-DINA model. However, ML-based algorithms have difficulties in estimating DCMs with monotonicity constraints. First, ML-based algorithms can converge at local maxima. Therefore, even if we employ a multiple starting value strategy to assess the optimality of parameter estimates, it is difficult to judge whether the parameter estimates are from a global maximum. Second, estimation of variability (i.e., standard errors) of parameter estimates relies on asymptotic theory in ML and the asymptotic distribution with parameter constraints may not be accurate in cases where small sample sizes are present.

Bayesian estimation methods are frequently employed to avoid the difficulties of ML-based algorithms. In Bayesian estimation, sampling methods are used to approximate the posterior distributions of parameters. Historically, Bayesian estimation procedures, especially, Markov chain Monte Carlo (MCMC; e.g., Brooks et al., 2011), have played a central role in the estimation of many DCMs. For example, Henson et al. (2009) used a Metropolis-Hastings (MH) MCMC algorithm to estimate LCDM parameters. Culpepper (2015) developed a Gibbs sampling algorithm for the DINA model. de la Torre and Douglas (2004) developed a MH algorithm for the higher order DINA model. Li, Cohen, et al. (2016) used the JAGS program (Plummer, 2003) to estimate a longitudinal DCM. DeCarlo (2012) used the BUGS language (Lunn et al., 2000) for estimating reparametrized DINA model parameters. Hartz and Roussos (2008) demonstrated the hierarchical Bayesian formulation of the Fusion Model using MH. Zhan et al. (2019) provided JAGS code for various types of DCMs. Jiang and Carter (2018) used Stan and Hamiltonian Monte Carlo (Carpenter et al., 2017) for the LCDM. Finally, Chen et al. (2018), Culpepper (2019), Culpepper and Hudson (2018), and Chung (2019) developed a Gibbs sampling algorithm for Q-matrix estimation.

Many previous studies using MCMC for DCM parameter estimation, however, did not consider general monotonicity constraints. For example, the Gibbs sampling algorithm in Culpepper (2015) used a constraint where the sum of slipping and guessing parameters was limited to the interval from 0 to 1, but the algorithm was limited to the DINA model. Other MCMC procedures were also limited to the DINA or DINO models (e.g., DeCarlo, 2012;

Zhan et al., 2019) or constraints for general DCMs such as the LCDM were only posed on the intercept and main effects (Zhan et al., 2019).

In general DCMs, if monotonicity constraints are incomplete or missing, there is a risk of label switching of attribute profiles within MCMC chains. If label switching occurs, posterior distributions become skewed and item parameter estimates are meaningless. Previous label switching studies (e.g., Papastamoulis, 2016) have provided ad-hoc procedures permuting MCMC samples. However, these methods are computationally intensive. This computational burden is crucial in DCMs because the number of attribute mastery profiles tends to be large. Therefore, ad-hoc label switching remedies may not be useful with DCMs. To gain reasonable parameter estimates with MCMC, monotonicity constraints are essential in practice.

The MCMC procedure in Henson et al. (2009) built the constraint for their example, but it was not general to all Q-matrices and used an inefficient MH rejection sampling algorithm. Further, the Henson et al. (2009) study used a limited-information structural model rather than a general structural model. These features meant the MCMC method employed in Henson et al. (2009) was limited and was not an effective method for general applications.

Summarizing our previous discussion, DCM monotonicity constraints are important to avoid the label switching problem in Bayesian estimation methods with MCMC, but efficient Gibbs sampling algorithms incorporating these constraints for general models have not been well-developed. More precisely, no MCMC methods with efficient algorithms for estimating general DCMs with saturated item and structural model parameters using full monotonicity constraints have been developed. The contribution of this paper is the development of a new algorithm that is an efficient Gibbs sampling algorithm for the estimation of saturated item and structural model parameters of general DCMs under item parameter monotonicity constraints. To achieve this objective, we use a reparameterization of general DCMs as latent class models, employed by Yamaguchi and Okada (2021), when estimated with saturated item parameters. We note that Liu and Johnson (2019) used a similar formulation and constraint approach, but they did not assess parameter recovery, convergence of MCMC samples, nor effectiveness of their method thoroughly using a simulation study. In this study, we assess not only parameter recovery but also effectiveness of convergence in a simulation study and compare our method to previous no- or partially constrained versions of MCMC DCM algorithms.

In the next section, we describe the mixture parameterization of general DCMs along with the fully Bayesian formulation used in our algorithm. In Section 3, we develop a Gibbs sampling algorithm with monotonicity constraints. Then, in Section 4, we show the results of a simulation study that was conducted to evaluate the parameter recovery efficacy of our new Gibbs sampling algorithm with monotonicity constraints. In Section 5, we compare our method to another MCMC methods employed in a previous study (Zhan et al., 2019) where non-constrained and partially constrained (i.e., positive main effects of the LCDM) were used. In this study, we name our constraints as “complete monotonicity constraints.” The results of an empirical data analysis are shown in Section 6. We conclude our paper with a discussion of our algorithm performance.

2 A Fully Bayesian Parameterization of General DCMs as Mixture Models

2.1 Item Response Function and Complete Data Likelihood

To treat DCMs as latent class (mixture) models where each class has a restricted item response probability, we introduce a latent indicator function. In a general latent class model with $C = 2^A$

latent classes, each item j has 2^A item response parameters. In a general DCM with saturated item parameters, however, this number is reduced to $2^{\sum_a q_{ja}}$, where $\sum_a q_{ja} \le A$. To demonstrate, consider a test with three attributes ($A = 3$). Further, consider an item j that measures attributes one and three, yielding a Q-matrix vector of $\boldsymbol{q}_j = [1, 0, 1]^\top$. As $\sum_a q_{ja} = 2$, the item has $2^2 = 4$ different correct item response parameters: one for each permutation of the measured attributes (attributes one and three). To notate these parameters, we use the index h $\left(h = 1, \ldots, 2^{\sum_a q_{ja}}\right)$.

$\boldsymbol{a}_{jh}$	θ_{jh}	Latent Class (Attribute Profile) $\boldsymbol{\alpha}_c$							
		[000]	[001]	[010]	[011]	[100]	[101]	[110]	[111]
[0*0]	θ_{j1}	1	0	1	0	0	0	0	0
[0*1]	θ_{j2}	0	1	0	1	0	0	0	0
[1*0]	θ_{j3}	0	0	0	0	1	0	1	0
[1*1]	θ_{j4}	0	0	0	0	0	1	0	1

, (3)

Equation (3) lists the four permutations of the measured attributes of item j and denotes these as $\boldsymbol{a}_j$ (with a script letter "$\boldsymbol{a}$" and vector-valued elements $\boldsymbol{a}_{jh}$) where, in the table, an asterisk is used to indicate an attribute that is *not* measured by the item and is not a part of this permutation. We use the vector $\theta_j = \left[\theta_{j1}, \ldots, \theta_{j2^{\sum_a q_{ja}}}\right]^\top$ with elements θ_{jh} for corresponding LCDM-based item response probabilities. With three attributes, there are a total of $2^3 = 8$ classes, which are listed across the columns of (3). The elements of the table in (3) are formed as indicators of $\boldsymbol{a}_{jh} = \boldsymbol{\alpha}_c$. We use Iverson notation $I\left[\boldsymbol{a}_{jh} = \boldsymbol{\alpha}_c\right]$ for shorthand to notate these elemental indicators. The values of the item response probabilities are given by the LCDM item response function given in (1) and (2), which, for this example, gives:

$$\theta_{j1} = \frac{\exp\left(\lambda_{j0}\right)}{1 + \exp\left(\lambda_{j0}\right)}, \tag{4}$$

$$\theta_{j2} = \frac{\exp\left(\lambda_{j0} + \lambda_{j,1,(3)}\right)}{1 + \exp\left(\lambda_{j0} + \lambda_{j,1,(3)}\right)}, \tag{5}$$

$$\theta_{j3} = \frac{\exp\left(\lambda_{j0} + \lambda_{j,1,(1)}\right)}{1 + \exp\left(\lambda_{j0} + \lambda_{j,1,(1)}\right)}, \tag{6}$$

$$\theta_{j4} = \frac{\exp\left(\lambda_{j0} + \lambda_{j,1,(1)} + \lambda_{j,1,(3)} + \lambda_{j,2,(1,3)}\right)}{1 + \exp\left(\lambda_{j0} + \lambda_{j,1,(1)} + \lambda_{j,1,(3)} + \lambda_{j,2,(1,3)}\right)}. \tag{7}$$

As there is a one-to-one mapping between the LCDM parameterization and the latent class parameterization, the parameters can be transformed into the other parameterization without loss of information.

We can re-express the item response probability of individual i for item j using the latent class formulation of the saturated DCM by:

$$\mathrm{P}\left(X_{ij} = x_{ij} | \theta_j, \boldsymbol{q}_j, \boldsymbol{\alpha}_i\right) = \prod_{h=1}^{2^{\sum_a q_{ja}}} \left[\theta_{jh}^{x_{ij}}\left(1 - \theta_{jh}\right)^{1 - x_{ij}}\right]^{I\left[\boldsymbol{a}_{jh} = \boldsymbol{\alpha}_i\right]}, \tag{8}$$

where x_{ij} is a realization of X_{ij}, $\boldsymbol{\theta}_j$ is a vector with an element θ_{jh} that gives the item response probability of the item-specific restricted attribute mastery pattern h provided by the LCDM item response function of (1) and (2). The product in (8) is taken over all possible item parameters for a given item j where the item response function is raised to the power of an indicator variable—as only one value of this indicator is equal to a value of one per individual i, this allows the notation to select the correct constrained item response probability.

Assuming local independence given the latent variable $\boldsymbol{\alpha}_i$ and exchangeability of individuals, the complete data likelihood function is

$$\mathrm{P}(\mathbf{X}|\boldsymbol{\Theta},\mathbf{Q},\boldsymbol{\alpha}) = \prod_{i=1}^{I}\prod_{j=1}^{J}\mathrm{P}\left(X_{ij}=x_{ij}|\boldsymbol{\theta}_j,\boldsymbol{q}_j,\boldsymbol{\alpha}_i\right) = \prod_{i=1}^{I}\prod_{j=1}^{J}\prod_{h=1}^{2^{\sum_a q_{ja}}}\left[\theta_{jh}^{x_{ij}}\left(1-\theta_{jh}\right)^{1-x_{ij}}\right]^{I\left[\boldsymbol{a}_{jh}=\boldsymbol{\alpha}_i\right]}, \tag{9}$$

where the $I \times J$ matrix $\mathbf{X}$ has elements x_{ij}, the $\boldsymbol{\Theta}$ is a set of all item parameters $\boldsymbol{\theta}_j$, $\mathbf{Q}$ is a Q-matrix, and $\boldsymbol{\alpha}$ is the matrix of attributes for all individuals.

2.2 Priors for the Saturated DCM

The priors for the latent variables and latent variable distribution parameters are chosen so that a Gibbs sampling algorithm—one where we can sample from the complete conditional distribution—can be derived. In general DCMs, the latent variables $\boldsymbol{\alpha}_i$ follow a categorical distribution (often called Multivariate Bernoulli). In latent class models, the parameters of this prior distribution are called the mixing proportion parameters reflecting the proportion of individuals with an attribute mastery pattern. We use $\boldsymbol{\pi} = [\pi_1, \dots, \pi_c, \dots, \pi_{2^A}]^\top$ to represent these parameters such that each $\pi_c > 0$ and $\sum_{c=1}^{2^A}\pi_c = 1$. The probability mass function of this prior comes from the multinomial/categorical distribution:

$$\mathrm{P}(\boldsymbol{\alpha}_i|\boldsymbol{\pi}) = \prod_{c=1}^{2^A}\pi_c^{I[\boldsymbol{\alpha}_i=\boldsymbol{\alpha}_c]}, \tag{10}$$

where $I[\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_c]$ is one when $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_c$ and zero otherwise. Exchangeability of individuals is also assumed, so the joint distribution of $\boldsymbol{\alpha}$ is

$$\mathrm{P}(\boldsymbol{\alpha}|\boldsymbol{\pi}) = \prod_{i=1}^{I}\prod_{c=1}^{2^A}\pi_c^{I[\boldsymbol{\alpha}_i=\boldsymbol{\alpha}_c]}. \tag{11}$$

A conjugate prior for $\boldsymbol{\pi}$ is the Dirichlet distribution with parameters $\boldsymbol{\delta}^0 = \left[\delta_1^0, \dots, \delta_\mathrm{c}^0, \dots, \delta_{2^A}^0\right]^\top$,

$$\mathrm{P}\left(\boldsymbol{\pi}|\boldsymbol{\delta}^0\right) \propto \prod_{c=1}^{2^A}\pi_c^{\delta_c^0-1}. \tag{12}$$

A prior for the correct response probability parameter θ_{jh} is the Beta distribution with hyperparameters a_{jh}^0 and b_{jh}^0 as the sample space of the Beta distribution is (0,1),

$$\mathrm{P}\left(\theta_{jh}|a_{jh}^0, b_{jh}^0\right) \propto \theta_{jh}^{a_{jh}^0-1}\left(1-\theta_{jh}\right)^{b_{jh}^0-1}. \tag{13}$$

The joint probability of θ_{jh} with independence of parameters is

$$\mathrm{P}\left(\boldsymbol{\Theta}|\mathbf{A}^0, \mathbf{B}^0\right) \propto \prod_{j=1}^{J}\prod_{h=1}^{2^{\sum_a q_{ja}}}\theta_{jh}^{a_{jh}^0-1}\left(1-\theta_{jh}\right)^{b_{jh}^0-1}, \tag{14}$$

where the $\mathbf{A}^0$and $\mathbf{B}^0$are the sets of hyperparameters a_{jh}^0 and b_{jh}^0.

The joint posterior distribution of $\boldsymbol{\alpha}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\pi}$ that is calculated from (11) through (14) and using Bayes theorem is:

$$\begin{aligned}
&\mathrm{P}\left(\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{\pi}|\mathbf{X},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\mathbf{Q}\right)\propto\mathrm{P}(\mathbf{X}|\boldsymbol{\alpha},\boldsymbol{\Theta},\mathbf{Q})\mathrm{P}(\boldsymbol{\alpha}|\boldsymbol{\pi})\mathrm{P}\left(\boldsymbol{\pi}|\delta^0\right)\mathrm{P}\left(\boldsymbol{\Theta}|\mathbf{A}^0,\mathbf{B}^0\right) \qquad (15)\\
&=\left(\prod_{i=1}^{I}\prod_{j=1}^{J}\prod_{h=1}^{2^{\Sigma_a q_{ja}}}\left[\theta_{jh}^{x_{ij}}\left(1-\theta_{jh}\right)^{1-x_{ij}}\right]^{I\left[\boldsymbol{a}_{jh}=\boldsymbol{\alpha}_i\right]}\right)\left(\prod_{i=1}^{I}\prod_{c=1}^{2^A}\pi_c^{I\left[\boldsymbol{\alpha}_i=\boldsymbol{\alpha}_c\right]}\right)\left(\prod_{c=1}^{2^A}\pi_c^{\delta_c^0-1}\right)\\
&\qquad\left(\prod_{j=1}^{J}\prod_{h=1}^{2^{\Sigma_a q_{ja}}}\theta_{jh}^{a_{jh}^0-1}\left(1-\theta_{jh}\right)^{b_{jh}^0-1}\right).
\end{aligned}$$

3 Gibbs Sampling Algorithms for the Saturated DCM

3.1 Unconstrained Gibbs Sampling Algorithm

The unconstrained Gibbs sampling algorithm is as follows. We use superscript (t) as the iteration number. The parameter vectors $\boldsymbol{\alpha}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\pi}$ are sampled from conditional distributions and each step is

$$\text{Step1}:\boldsymbol{\alpha}^{(t)}\sim\mathrm{P}\left(\boldsymbol{\alpha}|\mathbf{X},\boldsymbol{\Theta}^{(t-1)},\boldsymbol{\pi}^{(t-1)},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\mathbf{Q}\right), \qquad (16)$$

$$\text{Step2}:\boldsymbol{\Theta}^{(t)}\sim\mathrm{P}\left(\boldsymbol{\Theta}|\mathbf{X},\boldsymbol{\alpha}^{(t)},\boldsymbol{\pi}^{(t-1)},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\mathbf{Q}\right), \qquad (17)$$

$$\text{Step3}:\boldsymbol{\pi}^{(t)}\sim\mathrm{P}\left(\boldsymbol{\pi}|\mathbf{X},\boldsymbol{\alpha}^{(t)},\boldsymbol{\Theta}^{(t)},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\mathbf{Q}\right). \qquad (18)$$

From the joint posterior (16), the full conditional distribution of $\boldsymbol{\alpha}_i$, $\mathrm{P}(\boldsymbol{\alpha}_i|\,\boldsymbol{x}_i,\boldsymbol{\Theta},\boldsymbol{\pi},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\mathbf{Q})$, is independent for all individuals i and it becomes a categorical distribution whose parameters are is calculated as:

$$\mathrm{P}\left(\boldsymbol{\alpha}_i=\boldsymbol{\alpha}_c|\boldsymbol{x}_i,\boldsymbol{\Theta},\boldsymbol{\pi},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\mathbf{Q}\right)=\frac{\left[\prod_{j=1}^{J}\prod_{h=1}^{2^{\Sigma_a q_{ja}}}\left[\theta_{jh}^{x_{ij}}\left(1-\theta_{jh}\right)^{1-x_{ij}}\right]^{I\left[\boldsymbol{a}_{jh}=\boldsymbol{\alpha}_c\right]}\right]\pi_c}{\sum_{c=1}^{2^A}\left[\prod_{j=1}^{J}\prod_{h=1}^{2^{\Sigma_a q_{ja}}}\left[\theta_{jh}^{x_{ij}}\left(1-\theta_{jh}\right)^{1-x_{ij}}\right]^{I\left[\boldsymbol{a}_{jh}=\boldsymbol{\alpha}_c\right]}\right]\pi_c}. \qquad (19)$$

The conditional posterior $\mathrm{P}(\theta_{jh}|\,\mathbf{X},\boldsymbol{\pi},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\boldsymbol{\alpha},\mathbf{Q})$ is also independent for all j and h. The conditional distribution follows a Beta distribution with parameters:

$$\begin{cases}
a_{jh}^{*}=\sum_{i=1}^{I}\sum_{c=1}^{2^A}I\left[\boldsymbol{a}_{jh}=\boldsymbol{\alpha}_c\right]x_{ij}+a_{jh}^0,\\
b_{jh}^{*}=\sum_{i=1}^{I}\sum_{c=1}^{2^A}I\left[\boldsymbol{a}_{jh}=\boldsymbol{\alpha}_c\right]\left(1-x_{ij}\right)+b_{jh}^0.
\end{cases} \qquad (20)$$

Finally, the conditional posterior $\mathrm{P}(\boldsymbol{\pi}|\,\mathbf{X},\boldsymbol{\Theta},\mathbf{A}^0,\mathbf{B}^0,\delta^0,\boldsymbol{\alpha},\mathbf{Q})$ follows a Dirichlet distribution with parameters

$$\delta_c^{*}=\sum_{i=1}^{I}I\left[\boldsymbol{\alpha}_i=\boldsymbol{\alpha}_c\right]+\delta_c^0,\quad\forall c. \qquad (21)$$

3.2 Gibbs Sampling Algorithm with Monotonicity Constraints

The Gibbs sampling algorithm outlined above does not impose the DCM monotonicity constraints which are needed for item parameter identification and latent class meaning. The monotonicity constraints stipulate a partially ordered relationship among the correct item response probabilities as additional attributes are mastered. The monotonicity constraints for an example item measuring three attributes are shown in Table 1. In this item, θ_{j1} corresponds to a person who is not a master of all attributes, θ_{j2} to θ_{j4} are correct item response probabilities for attribute mastery patterns with only one mastered attribute, corresponding to attribute mastery patterns [1, 0, 0], [0, 1, 0], and [0, 0, 1]. Similarly, θ_{j5} to θ_{j7} are correct response probabilities for attribute mastery patterns with two mastered attributes with patterns [1, 1, 0], [1, 0, 1], and [0, 1, 1].

Finally, θ_{j8} is the correct response probability for the attribute mastery pattern where all attributes are mastered. Under our monotonicity constraints, the parameters have a partially ordered relationship:

$$\begin{gathered}
0\le\theta_{j1}\le\min\left(\theta_{j2},\theta_{j3},\theta_{j4}\right),\\
\theta_{j1}\le\theta_{j2}\le\min\left(\theta_{j5},\theta_{j6}\right),\\
\theta_{j1}\le\theta_{j3}\le\min\left(\theta_{j5},\theta_{j7}\right),\\
\theta_{j1}\le\theta_{j4}\le\min\left(\theta_{j6},\theta_{j7}\right),\\
\max\left(\theta_{j3},\theta_{j4}\right)\le\theta_{j7}\le\theta_{j8},\\
\max\left(\theta_{j2},\theta_{j3}\right)\le\theta_{j5}\le\theta_{j8},\\
\max\left(\theta_{j2},\theta_{j4}\right)\le\theta_{j6}\le\theta_{j8},\\
\max\left(\theta_{j5},\theta_{j6},\theta_{j7}\right)\le\theta_{j8}\le1.
\end{gathered}\tag{22}$$

The probability of a correct response for the class of individuals mastering none of the attributes, θ_{j1}, should be the smallest of the eight correct item response probabilities as no attributes are mastered. Similarly, the probability of a correct response for the class of individuals mastering all attributes, θ_{j8} should be highest. The probability of individuals mastering the first attribute, θ_{j2} is greater than individuals mastering no attributes, θ_{j1}, but is less than the probability of individuals mastering the first attribute and one additional attribute (either attribute two or attribute three), min(θ_{j5}, θ_{j6}). There is no ordinal relationship between [1, 0, 0] and [0, 1, 0], [0, 0, 1], or [0, 1, 1]. Likewise, the probability of individuals mastering both attributes one and two, θ_{j5}, should be greater than or equal to that of individuals mastering

Table 1 An example of monotonicity constraints for an item measuring three attributes

Restricted attribute mastery pattern number h	Attribute			Correct item response probability	Minimum value	Maximum value
	α_1	α_2	α_3			
1	0	0	0	θ_{j1}	0	min(θ_{j2}, θ_{j3}, θ_{j4})
2	1	0	0	θ_{j2}	θ_{j1}	min(θ_{j5}, θ_{j6})
3	0	1	0	θ_{j3}		min(θ_{j5}, θ_{j7})
4	0	0	1	θ_{j4}		min(θ_{j6}, θ_{j7})
5	1	1	0	θ_{j5}	max(θ_{j2}, θ_{j3})	θ_{j8}
6	1	0	1	θ_{j6}	max(θ_{j2}, θ_{j4})	
7	0	1	1	θ_{j7}	max(θ_{j3}, θ_{j4})	
8	1	1	1	θ_{j8}	max(θ_{j5}, θ_{j6}, θ_{j7})	1

min() is a function to return minimum value of the arguments and max() is a function to return the maximum value of the arguments just one of those attributes, $\max(\theta_{j2}, \theta_{j3})$. Similar logic applies to all other parameters for this item and produces partially ordered relationships among their respective parameters.

To satisfy monotonicity constraints, we modify Step 2 in the unconstrained Gibbs sampling algorithm to use the sampling schema employed in order-constrained latent class models (e.g., Hoijtink, 1998; Laudy et al., 2004). The strategy is to use a truncated Beta distribution with an inverse probability sampling method. A truncated Beta distribution has a lower bound we denote as *Low* and an upper bound we denote as *Upp*. To sample from this distribution, first, sample U from a uniform distribution on [0, 1]. Then, calculate the $\{\text{Low} + U \times (1 - \text{Upp} - \text{Low})\}$-th quantile of the Beta distribution with parameters a and b. This quantile value is the random sample from the truncated Beta distribution.

For our algorithm, let $\theta_j^{(t)}$ be the t-th iteration in an MCMC algorithm with of the probability of a correct response to an item j. Using the example in Table 1, Step 2 is now:

- Step 2.1: Calculate (20) for all parameters in item j,
- Step 2.2: Sample $\theta_{j1}^{(t)}$ from the truncated Beta distribution with parameters $\boldsymbol{\alpha}_{jh}^*$ and b_{jh}^* whose lower bound is 0 and upper bound is $\min\left(\theta_{j2}^{(t-1)}, \theta_{j3}^{(t-1)}, \theta_{j4}^{(t-1)}\right)$,
- Step 2.3: Sample $\theta_{j2}^{(t)}$ to $\theta_{j4}^{(t)}$ to from the truncated Beta distribution with parameters $\boldsymbol{\alpha}_{jh}^*$ and b_{jh}^* whose lower bound is $\theta_{j1}^{(t)}$ and each upper bound is $\min\left(\theta_{j5}^{(t-1)}, \theta_{j6}^{(t-1)}\right)$, $\min\left(\theta_{j5}^{(t-1)}, \theta_{j7}^{(t-1)}\right)$, or $\min\left(\theta_{j6}^{(t-1)}, \theta_{j7}^{(t-1)}\right)$,
- Step 2.4: Sample $\theta_{j5}^{(t)}$ to $\theta_{j7}^{(t)}$ from the truncated Beta distribution with parameters a_{jh}^* and b_{jh}^* whose corresponding lower bounds are $\max\left(\theta_{j2}^{(t)}, \theta_{j3}^{(t)}\right)$, $\max\left(\theta_{j2}^{(t)}, \theta_{j4}^{(t)}\right)$, and $\max\left(\theta_{j3}^{(t)}, \theta_{j4}^{(t)}\right)$, and upper bound is $\theta_{j8}^{(t-1)}$,
- Step 2.5: Sample $\theta_{j8}^{(t)}$ from the truncated Beta distribution with parameters a_{jh}^* and b_{jh}^* whose lower bounds is $\max\left(\theta_{j5}^{(t)}, \theta_{j6}^{(t)}, \theta_{j7}^{(t)}\right)$ and upper bound is 1.

This procedure can be easily modified for any number of attributes measured by an item. Steps 1 and 3 are same as the unconstrained Gibbs sampling algorithm.

More generally, Step 2 can be written for parameters in the item measuring more than two attributes. For k in 0 to $\sum_a q_{ja}$, let a parameter set $\theta_{h,(k)}=\{\theta_h | \sum_a \boldsymbol{a}_{jha} = k$, where $\boldsymbol{a}_{jha}$ is an element of a_{jh} except $*\}$. Note $\theta_{h,(0)}$ and $\theta_{h,\left(\sum_a q_{ja}\right)}$ are always unitary. We introduce a symbol "$\prec$" to represent partial order relationship between two attribute mastery patterns: $\boldsymbol{a}_{jh} \prec \boldsymbol{a}_{jh'}$ if $\boldsymbol{a}_{jha} \leq \boldsymbol{a}_{jh'a}, \forall a$. Finally, let subscript sets for fixed h be $\Lambda_h^+ = \left\{h' | \boldsymbol{a}_{jh} \prec \boldsymbol{a}_{jh'}\right\}$ and $\Lambda_h^- = \left\{h'' | \boldsymbol{a}_{jh''} \prec \boldsymbol{a}_{jh}\right\}$. Then, the general description of the Step 2 is the following:

- Step 2.1: Calculate (20) for all parameters in item j,
- Step 2.2: Sample $\theta_{h,(0)}^{(t)}$ from the truncated Beta distribution with parameters a_{jh}^* and b_{jh}^* whose lower bound is 0 and upper bound is $\min\left(\left\{\theta_{h',(1)}^{(t-1)} | h' \in \Lambda_h^+\right\}\right)$,

- Step 2.3: For $k = 1, \ldots, \sum_a q_{ja} - 1$, sample each $\theta^{(t)}_{jh,(k)} \in \theta^{(t)}_{j,(k)}$ from the truncated Beta distribution with parameters a^*_{jh} and b^*_{jh} whose lower bound is $\max\left(\left\{\theta^{(t)}_{h'',(k-1)} | h'' \in \Lambda^-_h\right\}\right)$ and upper bound is $\min\left(\left\{\theta^{(t-1)}_{h',(k+1)} | h' \in \Lambda^+_h\right\}\right)$,
- Step 2.4: Sample $\theta^{(t)}_{h,\left(\sum_a q_{ja}\right)}$ from the truncated Beta distribution with parameters a^*_{jh} and b^*_{jh} whose corresponding lower bounds is $\max\left(\left\{\theta^{(t)}_{h'',\left(\sum_a q_{ja}-1\right)} | h'' \in \Lambda^-_h\right\}\right)$, and upper bound is 1.

4 Simulation Study 1

4.1 Simulation Settings

A simulation study was used to evaluate the ability of our complete monotonicity-constrained Gibbs sampling algorithm to recover known parameter values. In the simulation study, the sample sizes were set to 200 and 1000, which we refer to as the small and large sample size conditions. The Q-matrix used is shown in Table 2 with 19 items and four attributes. Items 1–8 measure only one attribute, items 9–14 measure two attributes, items 15 to 18 measure three attributes, and item 19 measures four attributes. Three attribute correlation conditions were used where the tetrachoric correlation between attributes was set to 0, 0.5, and 0.8, representing are zero, moderate, and high correlations. The attribute mastery patterns were generated using a continuous, underlying multivariate normal distribution, as assumed by the tetrachoric correlation. First, for each simulated examinee, a continuous four-dimensional attribute vector $\widetilde{\boldsymbol{\alpha}}$ was generated from a multivariate normal distribution with a zero mean vector and a covariance matrix with ones for diagonal elements and 0, 0.5, and 0.8 for the off-diagonal elements. The cut-point for each underlying variable, which sets the marginal proportion of masters for an attribute, was set to be increasing across the attributes. That is, if the a-th element of $\widetilde{\boldsymbol{\alpha}}$, denote as $\widetilde{\alpha}_a$, was greater than $\Phi^{-1}(a/(1+A))$, where $\Phi^{-1}(\cdot)$ is the inverse function of the cumulative standard normal distribution, then α_a becomes 1 otherwise 0. From this procedure, true attribute mastery patterns and mixing parameters $\boldsymbol{\pi}$ were generated. This procedure was based on Chiu and Douglas (2013).

Item response probability parameters $\boldsymbol{\theta}_j$ were determined based on guessing and slipping parameters, denoted g_j and s_j. Here, guessing was the correct response probability for the attribute pattern where no attributes from item j were mastered and slipping was the probability of an incorrect item response for the attribute pattern where all attributes measured by item j were mastered. We first determined the two true parameters and, if an item measured more than two attributes, the correct answer probability of θ_{jh} was determined as $g_j + a_h\left(1-s_j-g_j\right) / \sum_a q_{ja}$, where a_h is the number of attributes mastered in the h-th item-specific attribute mastery pattern.

For example, assume $g_j = s_j = 0.2$, then $\theta_{j1} = g_j = 0.2$ and $\theta_{j4} = 1 - s_j = 0.8$. In addition, θ_{j2} and θ_{j3} had only one mastered attribute and, thus, the parameters were $\theta_{j2} = \theta_{j3} = 0.2 + (1 - 0.2$

Table 2 Q-matrix for simulation study one

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

$-0.2)/2=0.5$. Two guessing and slipping parameter conditions were assumed based on the simulation in Culpepper (2015): $g_j=s_j=0.1$ and $g_j=s_j=0.2$. The former was considered a high-quality items condition and the latter was a low-quality items condition. In the simulation study, there were two sample size conditions, two item quality conditions, and three correlation conditions, yielding 12 simulation conditions in total. The simulation was repeated 50 times for each condition.

In the Bayesian estimation procedure, we estimated four separate Markov chains with the number of iterations set to 10,000 for each chain. The thinning interval was one and the burn-in period was the first 5000 iterations. We selected random starting values for $\mathbf{\Theta}$ that satisfied our monotonicity constraints. To evaluate chain convergence, we employed the Gelman-Rubin $\widehat{R}$ index (Brooks & Gelman, 1998) for all parameters. We judged a chain to be converged when the maximum $\widehat{R} < 1.05$. In addition, we used the autocorrelation for MCMC samples to assess the efficiency of the Gibbs sampling algorithm as algorithms resulting in lower autocorrelations cover more of the posterior distribution for each iteration and thus need fewer iterations. In our parameter evaluation, we used EAP estimates from the posterior distributions formed by the MCMC samples. The quality of the parameter estimates was evaluated based on bias and root mean square error (RMSE). The simulation and real data analysis code were written in 64-bit R for Windows version 3.6.1 (R core team, 2019) and used the CODA package (Plummer et al., 2006) for convergence diagnostics.

4.2 Results

Figure 1 shows the multivariate $\widehat{R}$ for each simulation condition with chain lengths from 100 to 10,000 in iteration increments of 100. The panels in the first and third rows of Fig. 1 indicate the high-quality items conditions under small and large sample size settings. Across all replications, the multivariate version of $\widehat{R}$ was less than 1.05 within 2000 iterations. In addition, the panels in the second and fourth rows of Fig. 1 were the $\widehat{R}$ in the low-quality items condition under small and large sample sizes. Across all replications, the multivariate version of $\widehat{R}$ reached less than 1.05 at between 2000 and 5000 iterations. Therefore, as to be expected, the high-quality items conditions resulted in faster convergence of the MCMC algorithm than the low-quality conditions. Moreover, sample size was related to convergence speed: the small sample size conditions were slower to converge than the large sample size conditions. The attribute correlation conditions did not affect the reduction of the $\widehat{R}$.

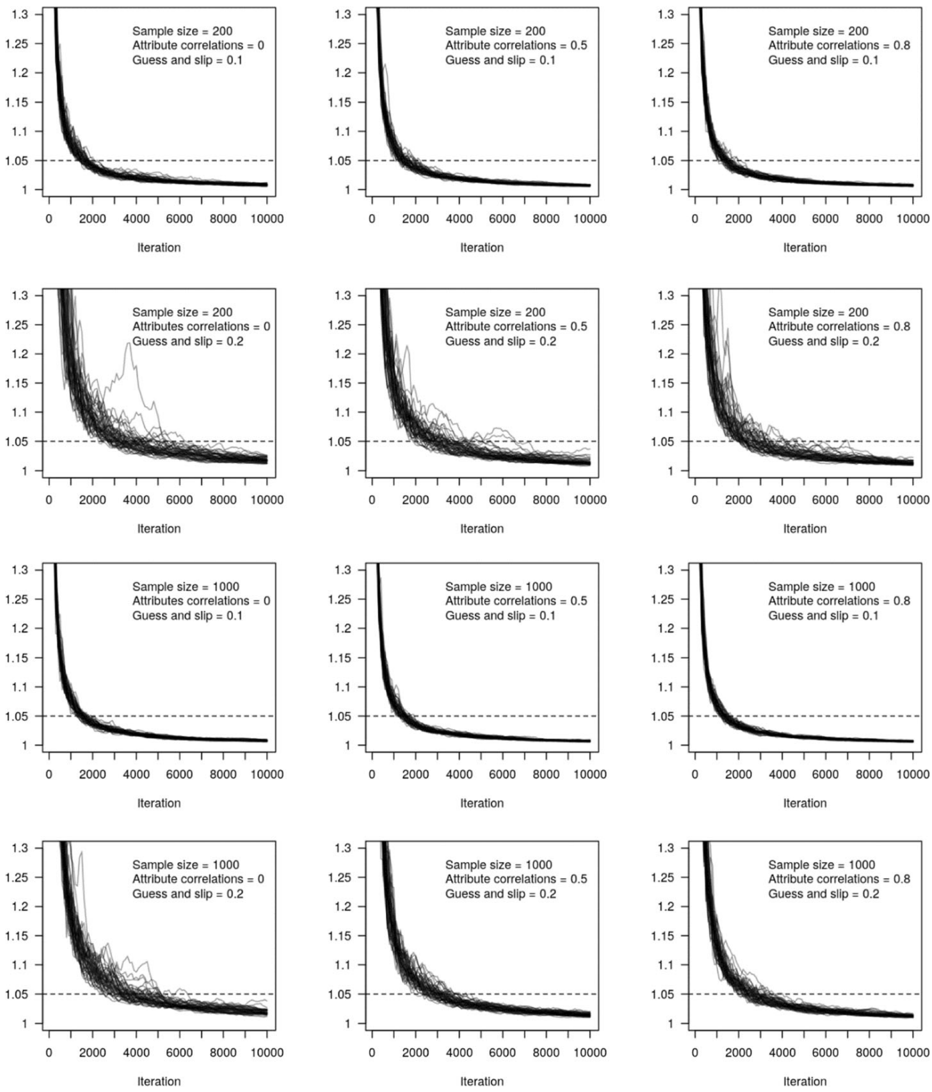

Fig. 1 Gelman-Rubin $\widehat{R}$-hat for all parameters. Each plot contains 50 replications

Figure 2 shows the average autocorrelation across all item and mixing parameters whose lags were from 1 to 50 for each simulation condition and each plot contains 50 replications. The first and third rows of Fig. 2, which were high-quality item conditions, indicated that the average autocorrelations in ten lags were less than 0.05 and the values quickly decreased to 0 with increases of lagged iterations. In other words, the MCMC samples indicated low dependencies between concurrent iterations. However, when the quality of items was low, more iterations were needed to decrease the auto correlation, which is shown in the second (small sample size condition) and fourth (large sample size condition) rows of Fig. 2. The correlation among attributes also affected MCMC sample dependency especially in low-

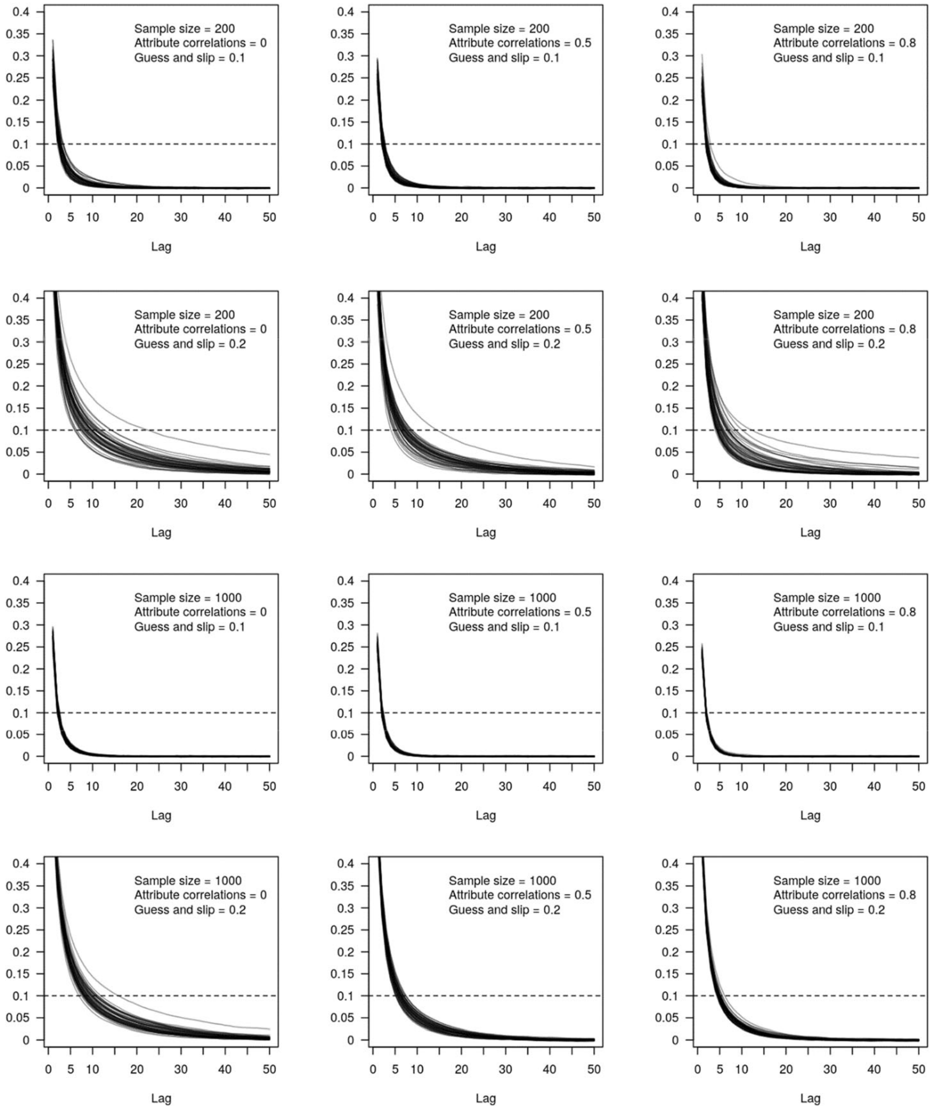

Fig. 2 Average autocorrelation for all parameters. Each plot contains 50 replications

quality items conditions: the high correlation conditions helped to decrease the autocorrelation more than the middle or zero correlation conditions. However, even in the low-quality items condition, most of the 15-lag autocorrelations were less than 0.1.

Figures 3 and 4 show average bias and RMSE of item parameters for each item under the small and large sample size conditions. In Figs. 3 and 4, the upper panels were bias and lower panels were RMSE. In addition, the left column shows the high-quality items conditions and the right column shows the low-quality items conditions. Under the small sample size conditions (Fig. 3), there were small bias results (− 0.06 to 0.06) for the first to eighth items. This indicates that items measuring easily mastered attributes tend to have a slight positive bias and items measuring relatively difficult attributes were negatively biased. The other items showed almost no bias. In addition, the RMSEs of item parameters in the lower quality items were worse than the ones in the high-quality items. However, the largest RMSE was less than 0.10 in the lower quality items.

In contrast, Fig. 4 shows the average bias and RMSE of item parameters for each item under the large sample size condition. From Fig. 3, the bias of parameter estimates was small for all conditions. Figure 4 also shows that the RMSE of the first 8 items (which measured only one attribute) was less than 0.03 in the high-quality items condition and that the low-quality items condition indicated greater RMSEs than the high-quality items condition. Most of the RMSEs showed similar values which were between 0.01 and 0.05 among all simulation conditions. These RMSE results were similar to the results of Culpepper (2015).

Figures 5 and 6 show bias and RMSE values for mixing parameters under the small and large sample size conditions, indicating small bias in all simulation conditions. In Fig. 5, the

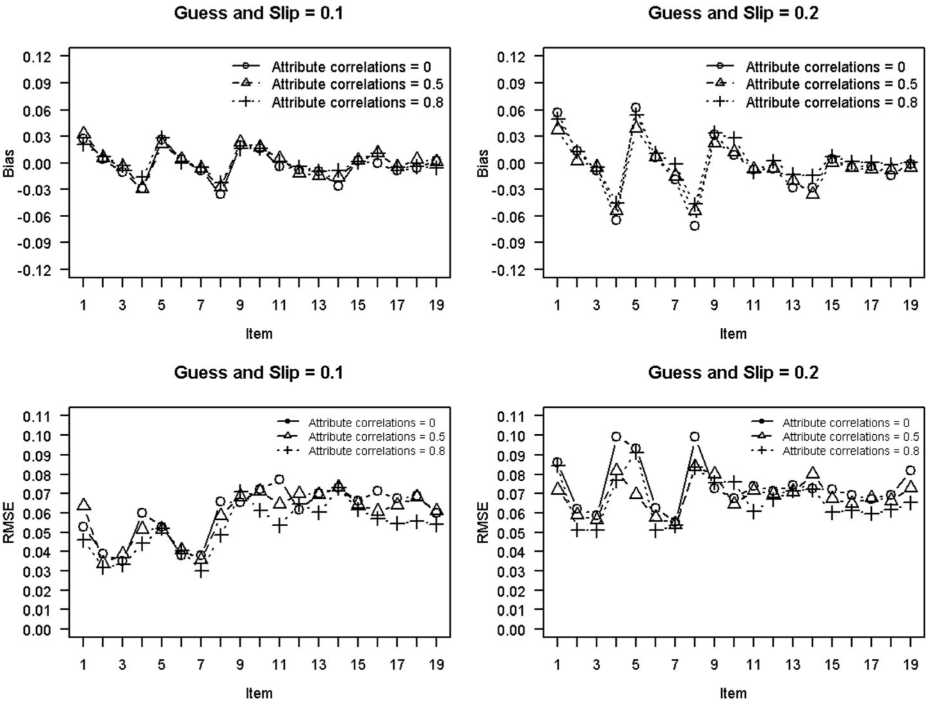

Fig. 3 Average bias (upper panels) and RMSE (lower panels) of the correct item response probability for each item for the sample size 200 condition

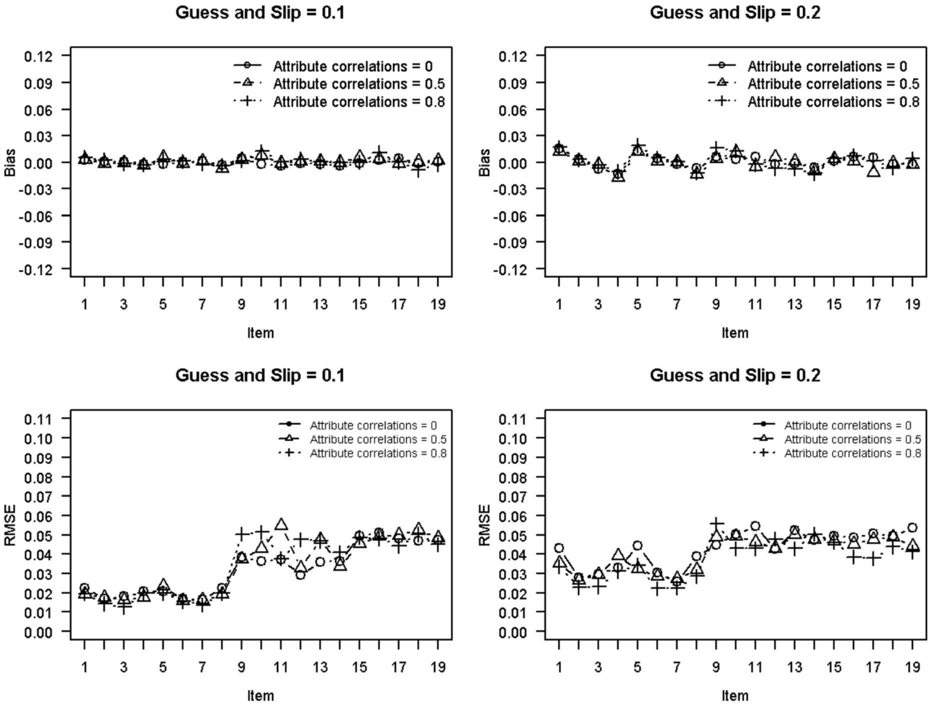

Fig. 4 Average bias (upper panels) and RMSE (lower panels) of the correct item response probability for each item for the sample size 1000 condition

RMSE in the low-quality items condition was at most just above 0.07 and most of the values were less than 0.06. Figure 6 also showed that the RMSEs in the high-quality items were less than 0.02 and almost all RMSEs in the low-quality items conditions were less than 0.03. The small RMSEs were gained with the monotonicity-constrained Gibbs sampling method.

Finally, Table 3 indicates recovery rates of each attribute mastery and whole attribute mastery patterns in each simulation condition. In the high-quality items condition, a difference between sample sizes could not be detected, and recovery rates of each attribute were greater than 0.95. The whole attribute mastery pattern recovery rates were larger than 0.85. In the low-quality items condition, the large sample size conditions were better than the small sample size conditions. In addition, recovery rates were not as good as in the high-quality items conditions. However, recovery rates for each attribute were at least 0.85 even in small sample size settings. The recovery rates of all mastery patterns were 0.587 to 0.699 in small sample sizes.The values in the large sample size conditions were slightly better than the small sample size conditions. In general, the recovery rates of true attribute mastery statuses were adequate for practical use.

In summary, with respect to the complete monotonicity-constrained algorithm, the MCMC chains showed relatively fast convergence and the parameters were recovered well, even in the small sample size conditions. Moreover, parameter biases were reduced in the large sample size conditions. In addition, the RMSE values were small across simulation conditions. Furthermore, the attribute mastery patterns were also recovered well, even under small sample size conditions.

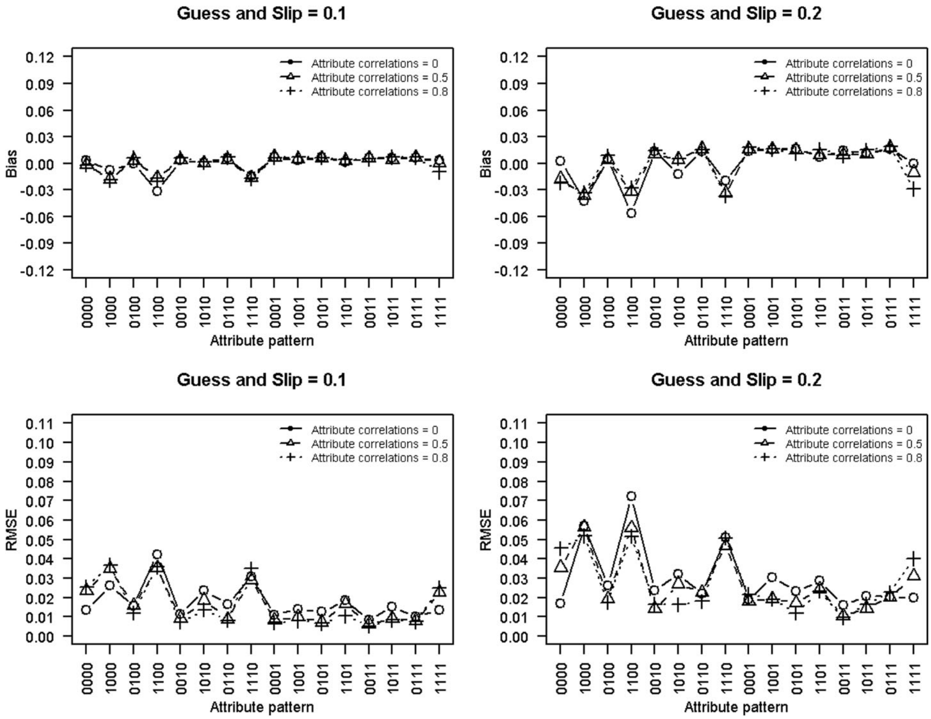

Fig. 5 Average bias (upper panels) and RMSE (lower panels) of the mixing parameter for each attribute mastery pattern for the sample size 200 condition

5 Simulation Study 2

5.1 Simulation Settings

The purpose of simulation study two was to compare our proposed Gibbs sampling algorithm and previous non- and partially constrained (main effects only) MCMC algorithms for the LCDM using JAGS. We evaluated estimation time, convergence ratio of parameters, estimation bias, and RMSE. Note that the JAGS automatically selects the appropriate sampler from model description and sometimes employed MH sapling method. However, all samplers employed in this simulation were Gibbs sampling-based ones. Detailed description of selection of the sampler was shown the user manual of the JAGS program (Plummer, 2017).

The basic simulation settings, data generating process, and convergence criteria were left unchanged from simulation study one, but we changed some specifics. The number of attributes in simulation study two was six with a Q-matrix shown in Table 4. We made this change to assess comparative algorithm performance across a more difficult estimation task. The Q-matrix was set to satisfy identifiability conditions and had 30 items: items one through twelve measured only one attribute, items 13 to 22 measured two attributes, items 23 to 26 measured three attributes, and items 29 and 30 items measured four attributes.

Due to the size of the number of attributes for this Q-matrix, we expected a long estimation time duration. Therefore, the number of manipulated conditions was reduced. Two guessing and slipping parameter conditions were assumed again: $g_j = s_j = 0.1$ and $g_j = s_j = 0.2$. In

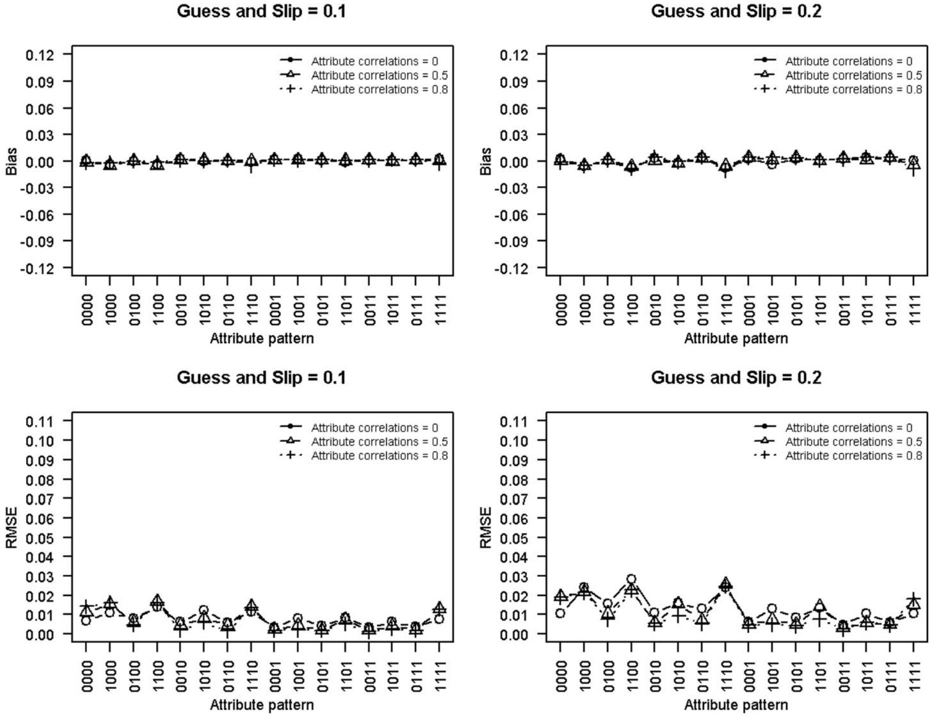

Fig. 6 Average bias (upper panels) and RMSE (lower panels) of the mixing parameter for each attribute mastery pattern for sample size 1000 condition

addition, attribute correlations were set to 0 or 0.8 as the three conditions from simulation study one had only a small difference in results. We examined 2 × 2 = 4 simulation conditions. Sample size was fixed to 500. The number of iterations were 10,000 of which the first 5000 iterations were discarded as a burn-in period. Three MCMC chains were estimated. Thirty replications were performed for each condition. The JAGS code for comparative methods was

Table 3 Recovery rate of each attribute mastery and whole mastery pattern in 12 simulation conditions in simulation one

Sample size	Item parameter conditions	Attribute correlations	Attribute				All
			1	2	3	4	
200	High-quality	0	.966	.955	.958	.962	.852
		.5	.966	.960	.962	.969	.864
		.8	.974	.971	.971	.976	.896
	Low-quality	0	.887	.867	.865	.879	.587
		.5	.906	.888	.891	.906	.651
		.8	.920	.907	.912	.914	.699
1000	High- quality	0	.969	.960	.961	.968	.867
		.5	.974	.966	.966	.974	.885
		.8	.981	.976	.976	.981	.916
	Low-quality	0	.906	.877	.877	.907	.635
		.5	.921	.898	.897	.924	.684
		.8	.936	.919	.922	.936	.743

Table 4 Q-matrix for simulation study two

Item	Attribute					
	1	2	3	4	5	6
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1
7	1	0	0	0	0	0
8	0	1	0	0	0	0
9	0	0	1	0	0	0
10	0	0	0	1	0	0
11	0	0	0	0	1	0
12	0	0	0	0	0	1
13	1	1	0	0	0	0
14	0	0	1	1	0	0
15	0	0	0	0	1	1
16	1	0	1	0	0	0
17	0	1	0	1	0	0
18	0	0	1	0	1	0
19	0	0	0	1	0	1
20	1	0	0	1	0	0
21	0	1	0	0	1	0
22	0	0	1	0	0	1
23	1	1	1	0	0	0
24	0	1	1	1	0	0
25	0	0	1	1	1	0
26	0	0	0	1	1	1
27	1	0	1	0	1	0
28	0	1	0	1	0	1
29	1	1	1	1	0	0
30	0	0	1	1	1	1

based on Zhan et al. (2019) leaving the settings for prior distributions unchanged from their study. Our proposed Gibbs sampling algorithm and the other two MCMC methods were applied for the same data to compare estimation time, convergence rates, and parameter recovery. The University of Iowa's Argon High Performance Computing system was used for this simulation.

5.2 Results

Table 5 shows the average estimation time of the three algorithms for each simulation condition. The estimation times for the complete monotonicity constraints ranged from 422.626 to 498.836 s—about five times faster than the other constraint types (we note these differences were likely due to the use of R for the complete monotonicity constraints algorithm as compared to JAGS for the other two). Figures 7 and 8 show a boxplot for the ratio of correct item response/mixing parameters satisfying our convergence criteria over 30 replications. The complete monotonicity constraint algorithm employed in the proposed Gibbs sampling algorithm (condition "C" in the figures) always converged for all conditions. For all conditions, the median ratio of the no-constraint Gibbs sampling (represented as "N" in the figures) was less than 20%. Therefore, we omit the results of the no-constraint method. The main effects

Table 5 Estimation time of three estimation algorithms in simulation study two

Item parameters conditions	Attribute correlations	Constraints type					
		No-constraints		Main effects		Complete	
		Mean	(SD)	Mean	(SD)	Mean	(SD)
High-quality	0	3195.508	(1634.273)	3062.796	(837.152)	422.626	(111.064)
	.8	3446.409	(986.397)	2860.615	(724.028)	454.784	(112.190)
Low-quality	0	2960.413	(1178.284)	3015.640	(760.136)	498.836	(167.962)
	.8	2704.048	(336.428)	2601.063	(513.711)	495.695	(168.905)

Constraints type represents constraints of estimation method: "No-constraints" did not have any constraints on LCDM parameters, "Main effects" meant positivity constraints on main effects in LCDM parameters, and "Complete" was the full monotonicity constraint employed in the proposed method

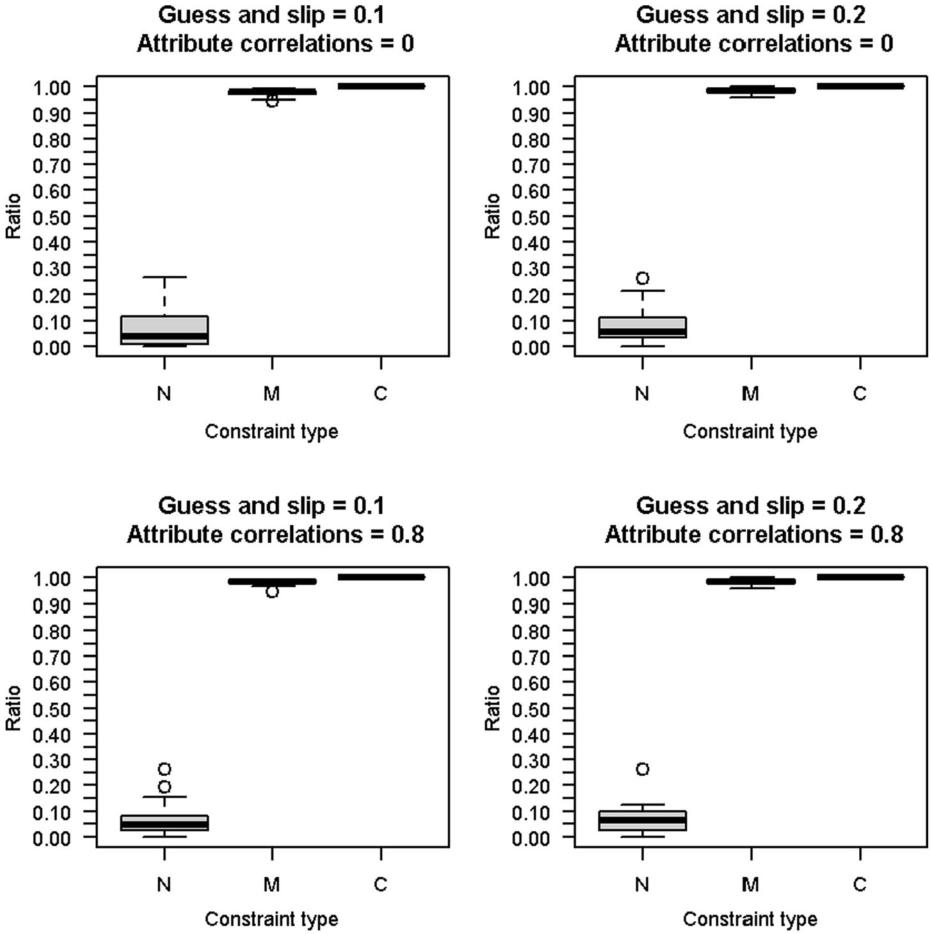

Fig. 7 Box plots for the ratio of satisfied convergence criteria ($\widehat{R} < 1.05$) correct item response probability parameters over 30 replications. The constraint types are no-constraints (marked "N"), main effects only monotonicity ("M"), and complete monotonicity ("C")

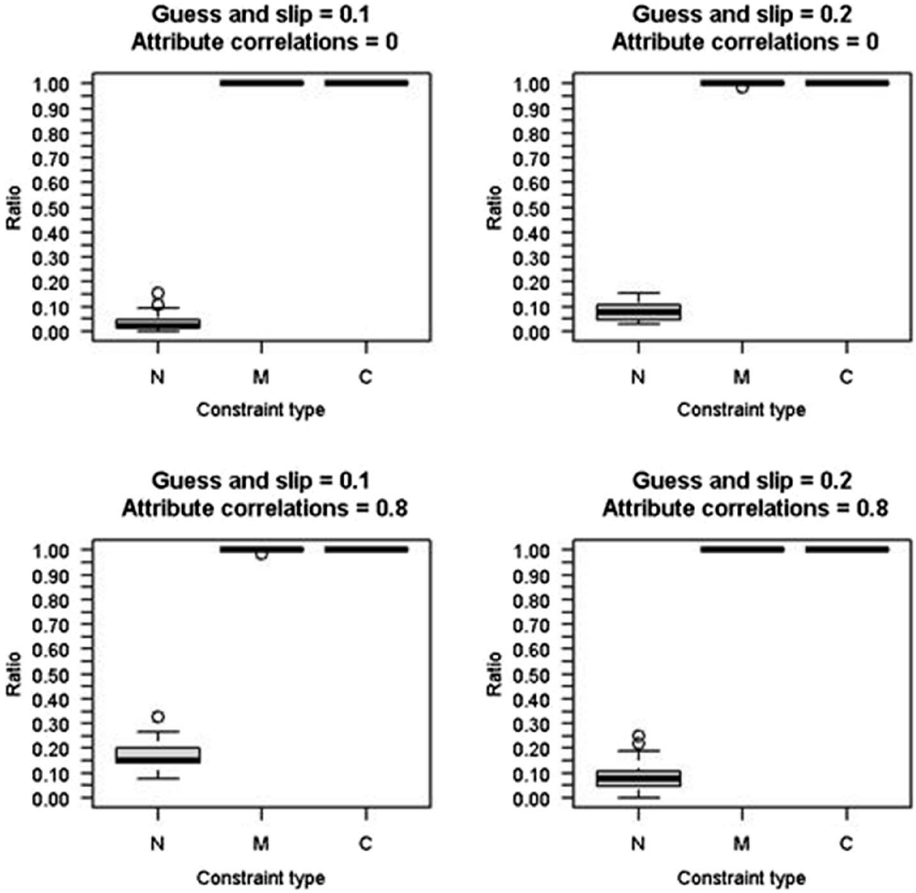

Fig. 8 Box plots for the ratio of satisfied convergence criteria ($\widehat{R} < 1.05$) mixing parameters over 30 replications. The constraint types are no-constraints (marked "N"), main effects only monotonicity ("M"), and complete monotonicity ("C")

constraints (indicated "M" in the figures) showed much better convergence than the no-constraints method with converged ratios that were very close to one. However, it did not always satisfy convergence criteria. From these results, we show the set of complete monotonicity constraints is important to improve convergence of MCMC algorithms, showing that our proposed algorithm is computationally effective.

Table 6 showed average biases and RMSEs which are shown in units multiplied by 1000 for item parameters. The columns "# attributes measured per item" represent the mean of bias and RMSE for items that measure only one to four attributes in both constraint conditions. The high-quality items conditions showed smaller bias values than the low-quality items conditions. In addition, higher attribute correlations conditions tended to have smaller bias values than the no correlation conditions. The absolute bias of items that were measuring only one or four attributes tended to be small. Importantly, the absolute bias values of the complete monotonicity constraints were almost always smaller than the ones of partial (main effects) constraints, especially for items measuring two or three attributes. Even in conditions in which

Table 6 Average estimation bias and RMSE for each condition in simulation two

Constraints type	Item parameters conditions	Attribute correlations	Bias×1000				RMSE×1000			
			# attributes measured per item				# attributes measured per item			
			1	2	3	4	1	2	3	4
Main effects	High-quality	0	3.722	12.335	10.989	9.445	38.699	31.367	65.420	40.900
		.8	− 0.722	8.545	2.439	− 2.693	30.900	27.930	60.695	39.354
	Low-quality	0	10.962	46.233	34.562	6.416	95.737	81.810	152.373	41.439
		.8	6.525	24.119	30.974	− 3.177	70.913	60.370	125.201	45.188
Complete	High-quality	0	− 1.109	6.771	3.449	6.788	36.805	28.492	62.022	40.866
		.8	− 1.744	6.635	6.551	− 0.638	29.210	26.675	60.578	39.925
	Low-quality	0	− 0.434	32.324	17.065	− 2.702	88.131	73.113	135.808	42.612
		.8	0.673	16.464	16.925	− 4.415	64.400	54.750	108.905	44.482

Constraints type represents constraints of estimation method: “Main effects” meant positivity constraints on main effects in LCDM parameters and “Complete” was the fully monotonicity constraint employed in proposed method

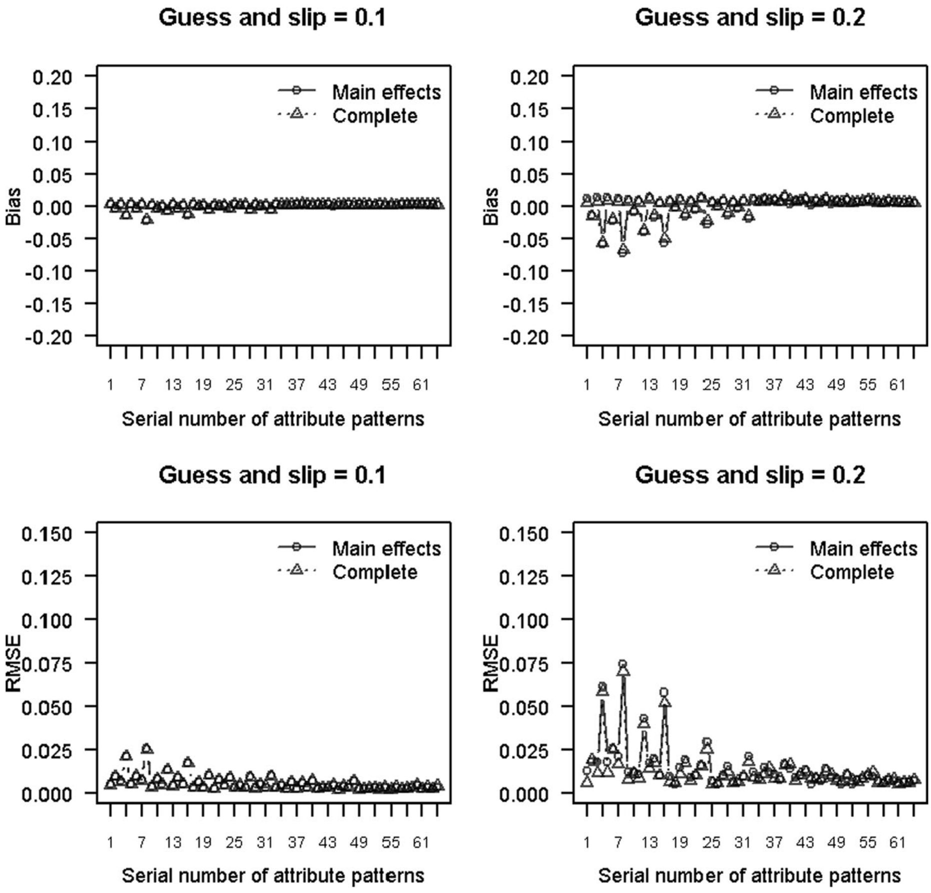

Fig. 9 Average bias (upper panels) and RMSE (lower panels) of the mixing parameter for 0 attribute correlations condition. "Main effects" represented the positivity constraint on the LCDM parameter and "Complete" represented complete monotonicity constraints

the bias values of partial (main effects) constraints were slightly smaller than the complete monotonicity constraints method, the values were reasonably small.

For RMSE, item quality and attribute correlations conditions showed similar results to that for bias. The RMSEs of the complete monotonicity constraint algorithm showed smaller values than the partial (main effects) constraints in the low-quality items conditions. However, the difference between the two constraints was small in the high-quality items conditions.

Moreover, the biases and RMSEs of mixing parameters shown in the Figs. 9 and 10 were not different between the two constraints. However, the bias and RMSE in the high-quality items conditions were reasonably small. Some bias and RMSE values in the low-quality items conditions were close to − 0.05 or 0.075 but others were very small. Finally, Table 7 shows the results of correct attribute recovery rate. The values were very similar between the two types of constraints in high-quality items conditions. In the low-quality items conditions, the correct recovery rates in the complete monotonicity constraints algorithms were slightly better than the partial (main effects) constraints algorithm.

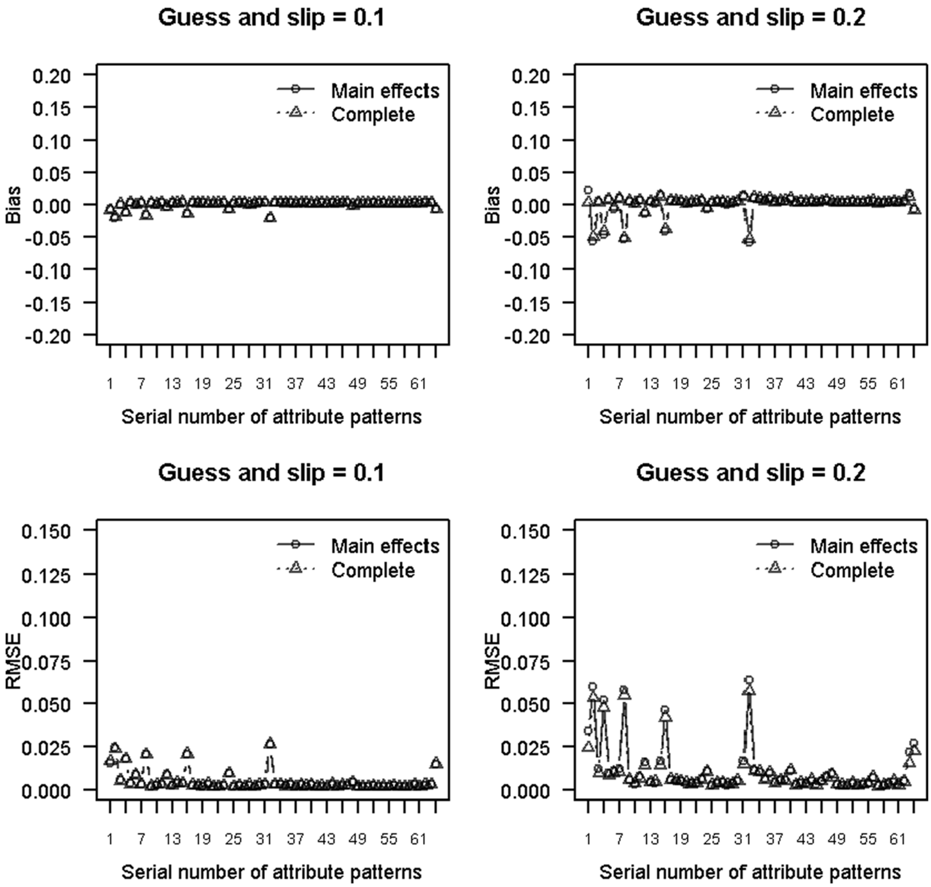

Fig. 10 Average bias (upper panels) and RMSE (lower panels) of the mixing parameter for 0.8 attribute correlations condition. "Main effects" represented the positivity constraint on the LCDM parameter and "Complete" represented complete monotonicity constraints

Table 7 Average attribute mastery recovery rate for each condition in simulation study two

Constraints type	Item parameters conditions	Attribute correlations	Attribute						All
			1	2	3	4	5	6	
Main effects	High-quality	0	.964	.960	.967	.965	.961	.965	.804
		.8	.979	.976	.979	.978	.978	.977	.875
	Low-quality	0	.729	.863	.876	.886	.874	.812	.350
		.8	.855	.915	.927	.928	.926	.896	.567
Complete	High-quality	0	.967	.961	.967	.965	.961	.964	.807
		.8	.980	.977	.980	.978	.978	.979	.879
	Low-quality	0	.805	.877	.884	.889	.874	.813	.399
		.8	.885	.923	.928	.929	.925	.902	.593

Constraints type represents constraints of estimation method: "Main effects" meant positivity constraints on main effects in LCDM parameters and "Complete" was the fully monotonicity constraint employed in proposed method

Table 8 Q-matrix employed in Chen and de la Torre (2014)

Item	Attribute					
	A1	A2	A3	A4	A5	A6
R040Q02	0	0	1	1	0	0
R04Q03A	1	0	0	1	0	0
R040Q3B	0	0	0	1	1	0
R040Q04	0	1	0	1	0	0
R04Q06	0	0	1	1	0	0
R077Q02	1	0	0	0	0	0
R077Q03	0	1	0	0	1	0
R077Q04	0	0	1	0	1	0
R077Q05	0	1	0	0	1	0
R077Q06	1	0	1	0	0	0
R088Q01	0	1	0	1	0	0
R088Q03	1	0	0	1	0	0
R088Q04T	0	0	1	1	0	0
R088Q05T	0	1	0	1	0	0
R088Q07	0	1	0	1	1	0
R11Q01	0	1	0	0	1	0
R11Q04	1	0	1	0	0	0
R110Q05	1	0	1	0	0	0
R110Q06	1	0	1	0	0	0
R216Q01	0	1	0	0	0	1
R216Q02	0	0	1	0	1	1
R216Q03T	1	0	1	0	0	1
R216Q04	0	0	1	0	0	1
R216Q06	1	0	1	0	0	1
R23Q01	1	0	1	0	0	1
R23Q02	0	0	1	0	0	1

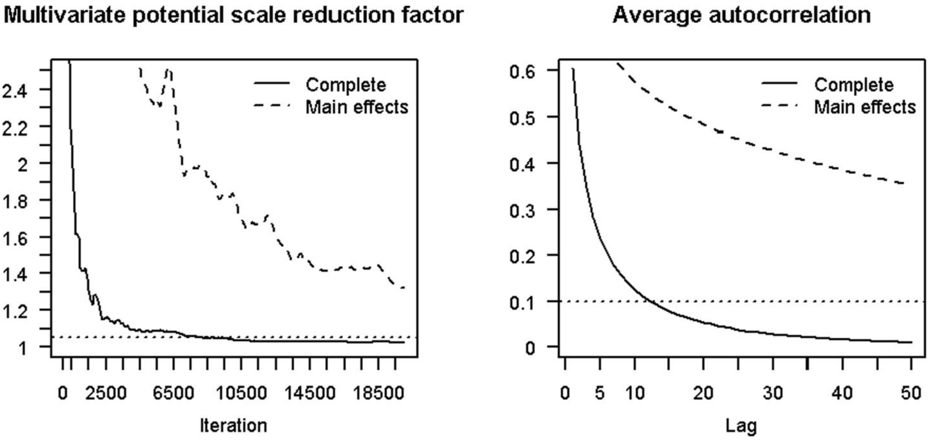

Fig. 11 Gelman-Rubin multivariate scale reduction factor, $\hat{R}$(left panel), and average autocorrelation (right panel) of all item and mixing parameters in the Programme for International Student Assessment (PISA) 2000 reading assessment data. The multivariate version of potential scale reduction factor was calculated based on four independent chains of length from 100 to 20,000 with iterations of 100 and lags of average autocorrelation from 1 to 50. The solid line represents the complete monotonicity constraints algorithm and dashed line represents the positivity constraint on the LCDM parameters

Table 9 Correct item response probability parameter estimates of the Programme for International Student Assessment 2000 data

	Estimates (SD)															
Item	0		1													
	00		10		01		11									
	000		100		010		110		001		101		011		111	
R040Q02	.222	(.043)	.436	(.080)	.725	(.036)	.807	(.021)								
R040Q03A	.072	(.027)	.190	(.050)	.530	(.062)	.786	(.025)								
R040Q03B	.018	(.010)	.188	(.048)	.044	(.026)	.581	(.028)								
R040Q04	.272	(.046)	.590	(.074)	.858	(.033)	.963	(.012)								
R040Q06	.165	(.036)	.333	(.079)	.556	(.042)	.848	(.023)								
R077Q02	.377	(.033)	.947	(.013)												
R077Q03	.077	(.023)	.332	(.098)	.548	(.069)	.757	(.025)								
R077Q04	.354	(.030)	.497	(.074)	.618	(.039)	.676	(.024)								
R077Q05	.055	(.017)	.176	(.062)	.329	(.045)	.406	(.025)								
R077Q06	.201	(.024)	.370	(.052)	.330	(.083)	.638	(.028)								
R088Q01	.221	(.037)	.795	(.053)	.415	(.047)	.880	(.022)								
R088Q03	.013	(.009)	.214	(.047)	.114	(.038)	.507	(.025)								
R088Q04T	.023	(.010)	.079	(.033)	.046	(.017)	.238	(.023)								
R088Q05T	.206	(.038)	.598	(.068)	.613	(.043)	.871	(.019)								
R088Q07	.094	(.026)	.328	(.107)	.349	(.053)	.786	(.097)	.144	(.050)	.620	(.117)	.476	(.071)	.912	(.020)
R110Q01	.548	(.033)	.844	(.055)	.855	(.044)	.944	(.013)								
R110Q04	.536	(.031)	.776	(.046)	.851	(.076)	.973	(.010)								
R110Q05	.340	(.033)	.750	(.043)	.708	(.085)	.839	(.020)								
R110Q06	.474	(.030)	.745	(.047)	.609	(.082)	.885	(.019)								
R216Q01	.297	(.035)	.576	(.100)	.872	(.048)	.976	(.012)								
R216Q02	.077	(.024)	.317	(.111)	.280	(.098)	.596	(.104)	.357	(.078)	.620	(.132)	.708	(.069)	.857	(.024)
R216Q03T	.014	(.009)	.052	(.036)	.137	(.065)	.246	(.072)	.126	(.054)	.393	(.073)	.583	(.163)	.816	(.032)
R216Q04	.035	(.016)	.312	(.067)	.258	(.045)	.675	(.032)								
R216Q06	.190	(.033)	.274	(.069)	.325	(.093)	.538	(.092)	.793	(.051)	.856	(.040)	.870	(.047)	.931	(.018)
R236Q01	.107	(.024)	.344	(.068)	.414	(.095)	.604	(.068)	.178	(.047)	.419	(.058)	.51	(.108)	.752	(.028)
R236Q02	.024	(.011)	.245	(.054)	.058	(.022)	.421	(.031)								

6 Application to Empirical Data

6.1 Analysis of Programme for International Student Assessment Data

To evaluate the performance of our algorithm on empirical data, we used the Programme for International Student Assessment (PISA) 2000 reading assessment data employed Chen and de la Torre (2014), as retrieved from the CDM package in R (George et al., 2016). The data consists of a sample of 1095 individuals from the UK with no missing responses to 26 items from PISA booklets 8 and 9. Items were multiple choice scored as binary indicators of correct responses.

The Q-matrix is shown in Table 8 and has six attributes. The attributes were α_1: Locating information, α_2: Forming a broad general understanding, α_3: Developing a logical interpretation, α_4: Evaluating number-rich text with number sense, α_5: Evaluating the quality or appropriateness of text, and α_6: test speediness. The definition of first 5 attributes were shown Table 7 in Chen and de la Torre (2014). The sixth attribute was that an individual can completely consider all items within the test time if he or she masters the attribute.

In addition to the proposed Gibbs sampling with complete monotonicity constraints, the no-constraint and partial (main effects only) constraint MCMC methods were estimated with JAGS. To compare the three algorithms, we employed posterior predictive *p*-value (e.g., Gelman et al., 1996; Meng, 1994) as a model-data fit index. The calculation procedure of posterior predictive *p*-value in DCMs was shown in Zhan et al. (2019). This value indicates better model-data fit as the *p*-value approaches 0.5. In this analysis, we used four chains with the number of iterations set to 20,000 of which the first 10,000 iterations were discarded as the burn-in period. The algorithm was considered converged when parameters had $\widehat{R} < 1.05$. Data analysis syntax is available in Open Science Framework page: https://osf.io/h9fm4/. The analysis was conducted on the University of Iowa's Argon High Performance Computing system.

Table 10 Mixing parameter estimates of the Programme for International Student Assessment 2000 data

Attribute mastery pattern	Estimates (SD)	Attribute mastery pattern	Estimates (SD)	Attribute mastery pattern	Estimates (SD)	Attribute mastery pattern	Estimates (SD)
000000	**.145** (.017)	000010	.006 (.005)	000001	.010 (.007)	000011	.004 (.004)
100000	.003 (.003)	100010	.014 (.007)	100001	.003 (.003)	100011	.008 (.006)
010000	.003 (.003)	010010	.002 (.002)	010001	.004 (.003)	010011	.002 (.002)
110000	.005 (.004)	110010	.002 (.002)	110001	.014 (.009)	110011	.017 (.011)
001000	.007 (.005)	001010	.007 (.005)	001001	.001 (.001)	001011	.002 (.002)
101000	.002 (.002)	101010	.003 (.003)	101001	.001 (.001)	101011	.003 (.002)
011000	.008 (.006)	011010	.005 (.004)	011001	.003 (.003)	011011	.004 (.003)
111000	**.024** (.011)	111010	.010 (.009)	111001	.005 (.005)	111011	.019 (.009)
000100	**.063** (.015)	000110	.003 (.003)	000101	**.048** (.015)	000111	.015 (.010)
100100	.003 (.003)	100110	**.022** (.008)	100101	.004 (.004)	100111	**.051** (.018)
010100	.003 (.003)	010110	.002 (.002)	010101	.004 (.004)	010111	.002 (.002)
110100	.004 (.004)	110110	.003 (.003)	110101	.017 (.010)	110111	**.026** (.015)
001100	.004 (.004)	001110	.006 (.004)	001101	.002 (.002)	001111	.006 (.005)
101100	.002 (.002)	101110	.006 (.005)	101101	.003 (.002)	101111	.006 (.005)
011100	.005 (.005)	011110	.003 (.003)	011101	.003 (.003)	011111	.004 (.003)
111100	.014 (.010)	111110	**.037** (.018)	111101	.005 (.005)	111111	**.280** (.024)

Large values are in bold

6.2 Parameter Estimation Result of PISA Data Analysis

The estimation times of the no-constraints, main effects constraints, and complete constraints were 11,954.670 s, 12,395.330 s, and 1510.836 s, respectively. The proposed algorithm was the fastest, which likely reflects the use of custom R-syntax rather than the general estimation of JAGS. The $\widehat{R}$s of the proposed Gibbs sampling algorithm were all less than 1.05, and most of the partial (main effect) constraint algorithm were less than 1.05 except for two parameters ($\lambda_{12,0} = 1.059$, $\lambda_{12,1,1} = 1.053$). Most parameters of the no-constraints algorithm did not satisfy this criterion. As such, all results for the no-constraints algorithm have been omitted.

We calculated the multivariate version of $\widehat{R}$ with chain lengths from 100 to 20,000 in iterations of 100 (left panel of Fig. 11) and the average autocorrelation of the chain over all parameters with lags from 1 to 50 (right panel of Fig. 11) for the complete and partial (main effects) monotonicity constraints algorithms. The multivariate $\widehat{R}$ for the complete monotonicity-constrained algorithm was less than 1.05 near iteration 9000 but the value of the partial (main effects) constraints algorithm decreased relatively slowly. This result indicated the relative efficiency of the complete constraint Gibbs sampling algorithm when compared to the algorithm with partial (main effects) constraints.

Moreover, the average autocorrelation of the complete monotonicity-constrained algorithm dropped to less than 0.1 by 15 lags. On the other hand, the average autocorrelation of the partial (main effects) constraints was larger than 0.3 until lag 50. This value indicated the MCMC samples of the complete monotonicity-constrained algorithm were not as strongly correlated, yielding a more efficient sample from posterior distribution parameter space than the partial (main effects) constraint algorithm. The posterior predictive p-values of the complete monotonicity and partial (main effects) constraints were 0.412 and 0.346, respectively with the complete monotonicity constraints algorithm result being closer to the ideal value of 0.5.

Tables 9 and 10 show parameter estimates of $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}$ for the complete monotonicity-constrained Gibbs sampling algorithm. The estimates satisfied monotonicity constraints (e.g., item R216Q06). For attribute patterns where all Q-matrix indicated attributes were mastered, many correct item response probabilities were greater than 0.7 (e.g., item R040Q02 or R040Q03A). However, some items, such as R236Q02, had correct item response probabilities less than 0.5. This means that some items might be difficult even for examinees mastering all attributes measured by the item which can indicate a lack of model fit or other Q-matrix misspecification. For attribute patterns where no Q-matrix indicated attributes were mastered, the correct item response probabilities were generally less than 0.4. The correct item response probabilities of R110Q01, R110Q04, and R110Q06 were greater than 0.45. Therefore, these items were easily answered without mastering any attributes.

The mixing parameters indicated that the attribute patterns where all attributes were mastered and where all attributes were not mastered had the most examinees, with more than 40% of examinees belonging to one of the two attribute mastery patterns. About 6% of examinees mastered the fourth attribute and 5.1% of examinees were estimated to have mastered the first, fourth, fifth, and sixth attributes. Many attribute mastery pattern probabilities were less than 0.01.

7 Conclusion and Discussion

In this study, we showed our Gibbs sampling algorithm with monotonicity constraints for general DCMs with saturated item parameters could recover true parameter values. Simulation studies one and two showed the proposed complete monotonicity-constrained Gibbs sampling algorithm could recover true parameters in fewer iterations and faster estimation times than previously developed no-constraint or partial (main effects) constraints algorithms. The empirical data example indicated reasonable estimates and it showed the efficiency of our proposed algorithm when compared to the other two constraints algorithms. Our algorithm is unique in that it (1) uses an effective Gibbs sampler, (2) is built for general DCMs with saturated item structural model parameters, and (3) satisfies the monotonicity constraints for all item parameters across all MCMC iterations. The use of Bayesian methods allows for the estimation of such DCMs in a direct manner. In comparison, general marginal maximum likelihood algorithms sometimes lead to inappropriate estimates because the parameter estimates are gained from solutions of complex constraint optimization problem. Therefore, Bayesian algorithms may be more reliable to use, when compared with marginal maximum likelihood methods.

The monotonicity constraints in DCMs are important but have been difficult to employ in general MCMC sampling languages such as JAGS or Stan. In addition, a DCM MCMC method uses Metropolis Hastings-type MCMC algorithms with rejection sampling to satisfy parameter constraints. This study proposed a Gibbs sampling algorithm and simulation study showed the algorithm could recover true parameter values, and the parameter autocorrelations of the complete monotonicity-constrained algorithm decreased within a few iterations. Moreover, the complete monotonicity-constrained algorithm converged well with a reasonable number of iterations, likely because it prevented label switching in MCMC iterations.

One drawback of our algorithm is that the parameters, after MCMC sampling, would have to be converted to LCDM or G-DINA parameters. In addition, to get MCMC samples of sub-models of the general DCMs, Eq. (3) would need to be changed for each type of model. Finally, our R code was fully written in R language. This means that the computational speed is not as fast as other computer languages such as C++.

In addition, when the number of attributes measured by an item is increased and the number of parameters is increased, the MCMC sampler may have difficulty moving across the posterior distributions of the parameters. We note that this weakness is not limited to our algorithm but also is true in some general MCMC procedures.

Another limitation of our framework is that the current model formulation is not directly applicable for additive-DCMs or the R-RUM because the parameters in such main effects type DCMs are not represented in equivalent constraint in our frameworks. Expanded applicability of this framework is a consideration for future studies. However, the model parameterization in this study can represent various sub-models of DCMs such as the DINA, DINO, or many other equality constrained DCMs.

However, the proposed algorithm is flexible to apply to another sub-models of the saturated DCM and extended DCMs such as longitudinal type DCMs (e.g., Madison & Bradshaw, 2018) or DCMs with covariates (e.g., Park & Lee, 2014). The Gibbs sampling algorithm with monotonicity constraints will extend to more complex models. The longitudinal type DCMs are special cases of hidden Markov models and they are singular models (e.g., Watanabe, 2018). This means that the maximum likelihood estimates are sometimes unstable and usual regular asymptotic theory does not hold. In such situations, the Bayesian methods are more appropriate than the maximum likelihood methods.

Funding This work was supported by JSPS Grant-in-Aid for JSPS Research Fellow 18J01312 and JSPS KAKANHI 20H01720. Jonathan Templin was supported by grants from the National Science Foundation (DRL-1813760) and the Institute of Education Sciences (R305A190079).

Declarations

Conflict of Interest The authors declare no competing interests.

References

Brooks, S. P., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics, 7*, 434–455. https://doi.org/10.2307/1390675.

Brooks, S., Gelman, A., Jones, G. L., & Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan : a probabilistic programming language. *Journal of Statistical Software, 76*. https://doi.org/10.18637/jss.v076.i01.

Chen, Y., Culpepper, S. A., Chen, Y., & Douglas, J. (2018). Baysean estimation of the the DINA Q matrix. *Psychometrika, 83*, 89–108. https://doi.org/10.1007/s11336-017-9579-4.

Chen, Y., Culpepper, S., & Liang, F. (2020). A sparse latent class model for cognitive diagnosis. *Psychometrika, 85*(1), 121–153. https://doi.org/10.1007/s11336-019-09693-2.

Chen, J., & de la Torre, J. (2014). A procedure for diagnostically modeling extant large-scale assessment data : the case of the programme for international student assessment in reading. *Psychology, 5*, 1967–1978. https://doi.org/10.4236/psych.2014.518200.

Chiu, C. Y., & Douglas, J. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. *Journal of Classification, 30*, 225–250. https://doi.org/10.1007/s00357-013-9132-9.

Chung, M. (2019). A Gibbs sampling algorithm that estimates the Q-matrix for the DINA model. *Journal of Mathematical Psychology, 93*, 102275. https://doi.org/10.1016/j.jmp.2019.07.002.

Culpepper, S. A. (2015). Bayesian estimation of the DINA model with Gibbs sampling. *Journal of Educational and Behavioral Statistics, 40*, 454–476. https://doi.org/10.3102/1076998615595403.

Culpepper, S. A. (2019). Estimating the cognitive diagnosis Q matrix with expert knowledge: application to the fraction-subtraction dataset. *Psychometrika, 84*, 333–357. https://doi.org/10.1007/s11336-018-9643-8.

Culpepper, S. A., & Hudson, A. (2018). An improved strategy for Bayesian estimation of the reduced reparameterized unified model. *Applied Psychological Measurement, 42*, 99–115. https://doi.org/10.1177/0146621617707511.

de la Torre, J. (2011). The generalized DINA model framework. *Psychometrika, 76*, 179–199. https://doi.org/10.1007/S11336-011-9207-7.

de la Torre, J., & Douglas, J. A. (2004). Higher-order latent trait models for cognitive diagnosis. *Psychometrika, 69*, 333–353. https://doi.org/10.1007/BF02295640.

DeCarlo, L. T. (2012). Recognizing uncertainty in the Q-matrix via a Bayesian extension of the DINA model. *Applied Psychological Measurement, 36*, 447–468. https://doi.org/10.1177/0146621612449069.

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Erlbaum.

Fang, G., Liu, J., & Ying, Z. (2019). On the identifiability of diagnostic classification models. *Psychometrika, 84*(1), 19–40. https://doi.org/10.1007/s11336-018-09658-x.

Gelman, A., Meng, X.-L., & Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica, 6*(4), 733–760 https://www.jstor.org/stable/24306036.

George, A. C., Robitzsch, A., Kiefer, T., Groß, J., & Ünlü, A. (2016). The R package CDM for cognitive diagnosis models. *Journal of Statistical Software, 74*. https://doi.org/10.18637/jss.v074.i02.

Gu, Y., & Xu, G. (2019). The sufficient and necessary condition for the identifiability and estimability of the DINA model. *Psychometrika, 84*(2), 468–483. https://doi.org/10.1007/s11336-018-9619-8.

Gu, Y., & Xu, G. (2020). Partial identifiability of restricted latent class models. *Annals of Statistics, 48*(4), 2082–2107. https://doi.org/10.1214/19-AOS1878.

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

Hartz, S., & Roussos, L. (2008). The fusion model for skills diagnosis: Blending theory with practice. *ETS Research Report Series, 08–71*, 1–57 Retrieved from https://www.ets.org/Media/Research/pdf/RR-08-71.pdf.

Henson, R. A., Templin, J. L., & Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika, 74*, 191–210. https://doi.org/10.1007/S11336-008.

Hoijtink, H. (1998). Constrained latent class analysis using the Gibbs sampler and posterior predictive P-values: applications to educational testing. *Statistica Sinica, 8*, 691–711.

Hong, C.-Y., Chang, Y.-W., & Tsai, R.-C. (2016). Estimation of generalized DINA model with order restrictions. *Journal of Classification, 33*, 460–484. https://doi.org/10.1007/s0035.

Hu, B., & Templin, J. (2019). Using diagnostic classification models to validate attribute hierarchies and evaluate model fit in Bayesian networks. *Multivariate Behavioral Research, 55*, 300–311. https://doi.org/10.1080/00273171.2019.1632165.

Jiang, Z., & Carter, R. (2018). Using Hamiltonian Monte Carlo to estimate the log-linear cognitive diagnosis model via Stan, (2014).

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement, 25*, 258–272. https://doi.org/10.1177/01466210122032064.

Laudy, O., Boom, J., & Hoijtink, H. (2004). Bayesian computational methods for inequality constrained latent class analysis. *New Developments in Categorical Data Analysis for the Social and Behavioral Sciences*, 52–69. https://doi.org/10.4324/9781410612021

Leighton, J. P., & Gierl, M. J. (Eds.). (2007). *Cognitive diagnostic assessment for education: theory and applications*. Cambridge University Press.

Li, F., Cohen, A., Bottge, B., & Templin, J. (2016). A latent transition analysis model for assessing change in cognitive skills. *Educational and Psychological Measurement, 76*, 181–204. https://doi.org/10.1177/0013164415588946.

Li, H., Hunter, C. V., & Lei, P.-W. (2016). The selection of cognitive diagnostic models for a reading comprehension test. *Language Testing, 33*, 1–35. https://doi.org/10.1177/0265532215590848.

Liu, X., & Johnson, M. S. (2019). Estimating CDMs using MCMC. In M. von Davier & Y.-S. Lee (Eds.), *Handbook of Diagnostic Classification Models* (pp. 629–649). Chem. https://doi.org/10.1007/978-3-030-05584-4_31.

Liu, J., Xu, G., & Ying, Z. (2013). Theory of self-learning Q-matrix. *Bernoulli, 19*(5 A), 1790–1817. https://doi.org/10.3150/12-BEJ430

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). *WinBUGS—a Bayesian modelling framework: concepts, structure, and extensibility*, 325–337.

Macready, G. B., & Dayton, C. M. (1977). The use of probabilistic models in the assessment of mastery. *Journal of Educational Statistics, 2*, 99–120.

Madison, M. J., & Bradshaw, L. P. (2018). Assessing growth in a diagnostic classification model framework. *Psychometrika, 83*, 963–990. https://doi.org/10.1007/s11336-018-9638-5.

Meng, X.-L. (1994). Posterior predictive *p*-values. *The Annals of Statistics, 22*(3), 1142–1160. https://doi.org/10.1214/aos/1176348654.

Muthén, L. K., & Muthén, B. O. (1998-2017). *Mplus User's Guide* (8th ed.). Muthén & Muthén.

Papastamoulis, P. (2016). Label.switching: an R package for dealing with the label switching problem in MCMC outputs. *Journal of Statistical Software, 69*, 1–11. https://doi.org/10.18637/jss.v069.c01.

Park, Y. S., & Lee, Y.-S. (2014). An extension of the DINA model using covariates: examining factors affecting response probability and latent classification. *Applied Psychological Measurement, 38*, 376–390. https://doi.org/10.1177/0146621614523830.

Plummer, M. (2003). JAGS: a program for analysis of Bayesian graphical models using Gibbs sampling. *The 3rd International Workshop on Distributed Statistical Computing*, *124*, 1–8. Retrieved from http://www.ci.tuwien.ac.at/Conferences/DSC-2003/

Plummer, M. (2017). *JAGS Version 4.3.0 user manual.* Retrieved from https://people.stat.sc.edu/hansont/stat740/jags_user_manual.pdfPlummer,

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News, 6*, 7–11 Retrieved from https://journal.r-project.org/archive/.

R Core Team. (2019). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing URL: https://www.R-project.org/.

Rupp, A. A., & Templin, J. (2008). Unique characteristics of diagnostic classification models: a comprehensive review of the current state-of-the-art. *Measurement: Interdisciplinary Research & Perspective, 6*, 219–262. https://doi.org/10.1080/15366360802490866.

Rupp, A. A., Templin, J. L., & Henson, R. A. (2010). *Diagnostic measurement: theory, methods and applications*. Guilford Press.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 62*(4), 795–809.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement, 20*, 345–354. https://doi.org/10.1111/j.1745-3984.1983.tb00212.x

Tatsuoka, K. K., & Tatsuoka, M. M. (1997). Computerized cognitive diagnostic adaptive testing: effect on remedial instruction as empirical validation. *Journal of Educational Measurement, 34*, 3–20. https://doi.org/10.1111/j.1745-3984.1997.tb00504.x.

Templin, J., & Bradshaw, L. (2014). Hierarchical diagnostic classification models: a family of models for estimating and testing attribute hierarchies. *Psychometrika, 79*, 317–339. https://doi.org/10.1007/s11336-013-9362-0.

Templin, J. L., & Henson, R. a. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods, 11*, 287–305. https://doi.org/10.1037/1082-989X.11.3.287.

Templin, J., & Hoffman, L. (2013). Obtaining diagnostic classification model estimates using Mplus. *Educational Measurement: Issues and Practice, 32*, 37–50. https://doi.org/10.1111/emip.12010.

von Davier, M. (2008). A general diagnostic model applied to language testing data. *The British Journal of Mathematical and Statistical Psychology, 61*, 287–307. https://doi.org/10.1348/000711007X193957.

von Davier, M. (2014). The log-linear cognitive diagnostic model (LCDM) as a special case of the general diagnostic model (GDM). *ETS Research Report Series* (Vol. RR–14-40). Princeton, NJ. https://doi.org/10.1002/ets2.12043

Watanabe, S. (2018). *Mathematical theory of Bayesian statistics*. Chapman and Hall/CRC.

Xu, G. (2017). Identifiability of restricted latent class models with binary responses. *Annals of Statistics, 45*(2), 675–707. https://doi.org/10.1214/16-AOS1464.

Xu, G., & Shang, Z. (2018). Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association, 113*(523), 1284–1295. https://doi.org/10.1080/01621459.2017.1340889.

Xu, G., & Zhang, S. (2016). Identifiability of diagnostic classification models. *Psychometrika, 81*(3), 625–649. https://doi.org/10.1007/s11336-015-9471-z.

Yamaguchi, K., & Okada, K. (2018). Comparison among cognitive diagnostic models for the TIMSS 2007 fourth grade mathematics assessment. *PLoS One, 13*, e0188691. https://doi.org/10.1371/journal.pone.0188691.

Yamaguchi, K., & Okada, K. (2021). Variational Bayes inference algorithm for the saturated diagnostic classification model. *Psychometrika, 85*, 973–995. https://doi.org/10.1007/s11336-020-09739-w.

Zhan, P., Jiao, H., Man, K., & Wang, L. (2019). Using JAGS for Bayesian cognitive diagnosis modeling: a tutorial. *Journal of Educational and Behavioral Statistics, 44*, 473–503. https://doi.org/10.3102/1076998619826040.