Measurement and Research Department Reports 2011-2

A Conditional Composition Algorithm for Latent Regression

Maarten Marsman
Gunter Maris
Timo M. Bechger
Cees A.W. Glas

Measurement and Research Department Reports 2011-2

A Conditional Composition Algorithm for Latent Regression

Maarten Marsman, Cito

Gunter Maris, Cito & University of Amsterdam

Timo M. Bechger, Cito

Cees A.W. Glas, University of Twente

Cito
Arnhem, 2011

Abstract

The article describes a general Gibbs sampler for structural measurement models consisting of an item response model for item responses conditional on ability, and a structural model for ability. We focus on an algorithm to simulate from the posterior distribution of ability. The algorithm, which we call the conditional-composition algorithm, can be used with any item response model and can be used in large scale applications in educational measurement.

1. Introduction

As a rule, large-scale applications in educational measurement use structural measurement models for data analysis. A *structural measurement model*, as defined by Adams, Wilson, and Wu (1997), consists of two parts: a monotone, unidimensional *item response theory* (IRT) model for the conditional distribution $P(\mathbf{x}|\theta)$ of the item responses $\mathbf{x}$ as a function of an ability θ and a *structural model* for the distribution of ability $f(\theta|\mathbf{y})$ as a function of measured characteristics of the persons $\mathbf{y}$. Conditional on ability, the responses are assumed to be independent of $\mathbf{y}$, and together, the IRT model and the structural model determine the statistical model for the distribution of the data

$$P(\mathbf{x}|\mathbf{y}) = \int_{\mathcal{R}} P(\mathbf{x}, \theta|\mathbf{y}) d\theta = \int_{\mathcal{R}} P(\mathbf{x}|\theta) f(\theta|\mathbf{y}) d\theta. \tag{1}$$

The class of structural measurement models encompasses a wide range of models including the multi-level latent regression models that are used in projects such as PISA or NAEP to deal with data involving a hierarchical nesting of persons, e.g., students within schools. There is general agreement that a Bayesian approach is useful to handle the complexity of these models. In this paper we present a *Markov Chain Monte Carlo* (MCMC) algorithm that can be used for Bayesian inferences in structural measurement models when the IRT model is known. Specifically, we developed a Gibbs sampler for drawing a sample from the joint posterior distribution of ability and the parameters of a multi-level regression model.

The Gibbs sampler (Geman & Geman, 1984) is a well-known abstract *divide-and-concquer* algorithm for generating a *dependent* sample from a complex multivariate distribution. The interested reader is referred to Casella and George (1992) or Tanner (1996, 6.1) for a general introduction to the Gibbs sampler and to Fox (2010) for a survey of applications in educational measurement. Formally, the Gibbs

sampler generates a *Markov chain* for which the (posterior) distribution from which a sample is desired is the invariant distribution. In each iteration, a sample is drawn from so-called *full conditional distributions*: i.e., distributions of one (set of) variable(s) conditionally on *all* the other variable(s). Here we wish to sample from the posterior $f(\theta, \Gamma|\mathbf{x}, \mathbf{y})$, where Γ denotes the parameters of the structural model. In each iteration *it* of the Gibbs sampler, a sample is drawn from two full-conditional distributions:

1. Abilities $\theta^{(it+1)} = \left(\theta_1^{(it+1)}, \ldots, \theta_p^{(it+1)}\right)$ are sampled from

$$f\left(\theta|\mathbf{x}, \mathbf{y}, \Gamma^{(it)}\right) \propto P(\mathbf{x}|\theta) f\left(\theta|\mathbf{y}, \Gamma^{(it)}\right), \tag{2}$$

   the posterior distribution of ability.

2. With these abilities we sample $\Gamma^{(it+1)}$ from

$$f\left(\Gamma|\mathbf{x}, \mathbf{y}, \theta^{(it+1)}\right) = f\left(\Gamma|\mathbf{y}, \theta^{(it+1)}\right). \tag{3}$$

Values drawn from the posterior distribution of ability are commonly called *plausible values* (PVs; Mislevy, 1991). In this paper, we present a generic algorithm to generate the PVs which we call the *conditional composition* (CC) algorithm. The second sampling problem is of less concern since it is a routine matter for the normal regression models used in educational measurement (see Gelman, Carlin, Stern, & Rubin, 2004; Fox, 2010). In our discussion of the sampling of PVs, we will provide the details of a Gibbs sampler for multilevel linear latent regression. Simulation will be used to illustrate that this Gibbs sampler can be used to deal with real problems in real time.

PVs are commonly calculated in large-scale educational surveys to provide a complete data set that can be used for secondary analyses. Two options are currently available. The first is to use an existing MCMC algorithm for IRT models,

assuming that the item parameters are known. Most MCMC algorithms for IRT models have been inspired by Albert (1992) who proposed a Gibbs sampler for a random-effects 2-parameter normal-ogive model with a normal distribution for ability (see Tanner, 1996, 6.2.5). This is a very simple algorithm due to the clever use of latent item responses as auxilliary variables. All related samplers use this form of *data augmentation* (Tanner & Wong, 1987), although it causes a positive dependence between subsequent PVs which slows down convergence of the Gibbs sampler (e.g., Fox, 2010, p. 77). Albert's work has been generalized to estimate more complex structural models by Béguin and Glas (2001) and Fox and Glas (2001) assuming normal-ogive IRT models. Gibbs samplers for logistic IRT models (e.g., the Rasch (1960) or the 2PL model) have been developed by, e.g., Maris and Maris (2002) and Maris and Bechger (2005). Each of these Gibbs samplers is specific for a particular model. To obtain a generic algorithm that works for all IRT models, Patz and Junker (1999) proposed using Metropolis within Gibbs. Maier (2001) applied this idea to develop a sampler for a random-effects Rasch model with a multilevel latent regression model.

The second option is to sample from an approximate posterior. Mislevy (1991) directly simulated PVs using a discrete approximation to the posterior distribution. Thomas and Gan (1997) used a normal approximation and refined this approximation using a *Sampling Importance Re-sampling algorithm* (SIR; Rubin, 1987). The advantages of the present algorithm are that it can be applied to any IRT model, that it does not use data augmentation and hence generates a Markov chain with (much) less autocorrelation, and that it provides a sample from the exact posterior.

In the following section, a simple example is presented to illustrate the CC algorithm. This sets the stage for a formal introduction of the algorithm. We then discuss how the algorithm can be made more efficient when dealing with large data

sets. The greatest efficiency is obtained when the IRT model belongs to the exponential family: e.g., the Rasch model that is used in PISA. Our assumption that the IRT model is known is a mild one for exponential family IRT models which can be fitted independently of the structural model using conditional likelihood methods. Finally, a Gibbs sampler for latent multilevel regression is introduced. Throughout, small simulation studies will be presented to illustrate the behaviour and efficiency of the algorithms.

2. A Random-Effects Rasch Model

Assume that a sample of n persons respond to k Rasch items out of the $I \geq k$ items in the study. Let x_{pi} denotes the response of person p to the ith item such that

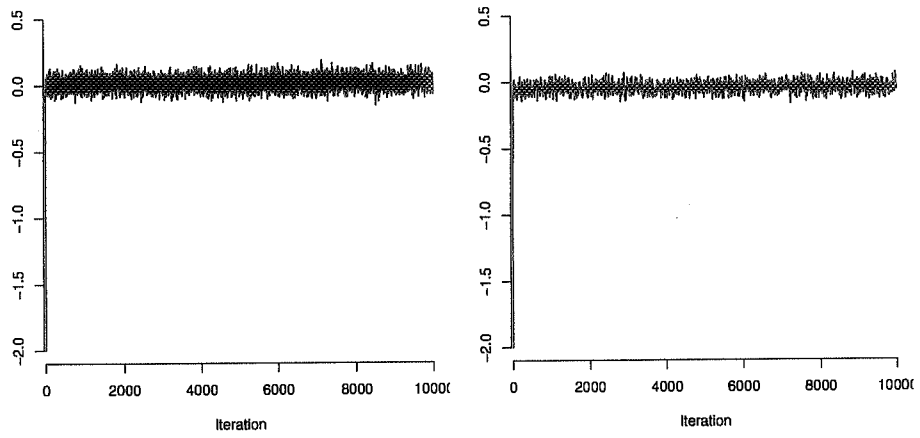
$$P(x_{pi} = 1|\theta_p, \delta_i) = \frac{\exp(\theta_p - \delta_i)}{1 + \exp(\theta_p - \delta_i)}, \tag{4}$$

where θ_p denotes the ability of person p and δ_i is interpreted as the difficulty of item i. We assume a normal ability distribution with mean μ and variance σ^2. Thus, we have a random-effects Rasch model with a normal distribution for person effects. This is a simple but non-trivial example of a structural measurement model. Assuming that we have the correct Rasch model, our interest is in obtaining a sample from the joint posterior of σ^2 and μ: $f(\mu, \sigma^2|\mathbf{x})$. To this end, we set up a Gibbs sampler as described in the Introduction.

In iteration it of the Gibbs sampler, we began by generating a PV from the posterior of each person:

$$f(\theta_p|x_{p+}, \mu^{(it)}, \sigma^{(it)}, \delta) \propto P(x_{p+}|\theta_p, \delta) f(\theta_p|\mu^{(it)}, \sigma^{(it)}) \tag{5}$$

which depends only on the data through the number of correct answers $x_{p+} = \sum_i x_{pi}$ which is sufficient for θ_p in the Rasch model. To do this, we first generated a sample



(a) $n = 5000$ and $k = 2$. (b) $n = 1000$ and $k = 10$.

FIGURE 1.

Trace plots of μ.

(θ^*, x_+^*) from the joint distribution $f(\theta, x_+|\mu^{(it)}, \sigma^{(it)}, \delta) = P(x_+|\theta, \delta) f(\theta|\mu^{(it)}, \sigma^{(it)})$ using what Tanner (1996) calls *the method of composition*: i.e., we generated an ability and with that ability a number correct score. The posterior is the distribution of ability conditional on the event that $x_+^* = x_{p+}$. To ensure that this condition holds, we simply ignored draws where $x_+^* \neq x_{p+}$. Thus, we repeatedly generated an ability from the structural model and a response from the IRT model until the generated score equals x_{p+}. We set $\theta_p^{(it+1)} = \theta^*$ when a response was generated that equalled the observed response of person p. In pseudo code:

Repeat :

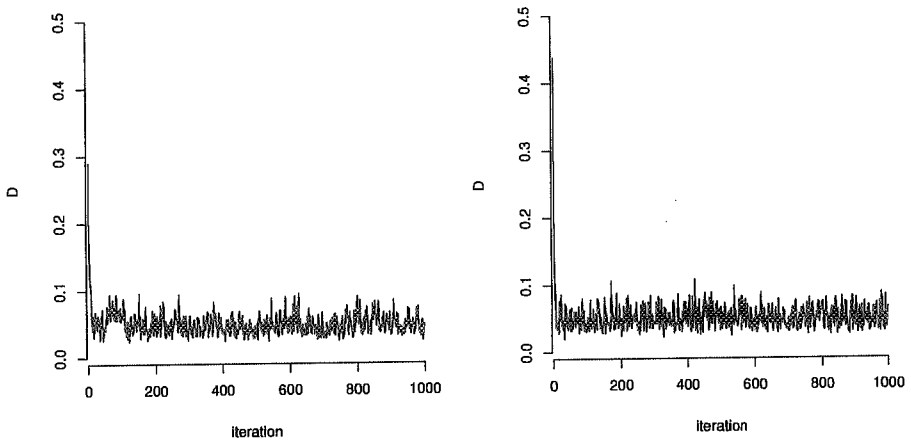
Generate θ^* from $N(\mu^{(it)}, \sigma^{(it)})$

Generate x* from the Rasch model with δ and $\theta = \theta^*$

Until : $x_+^* = x_{p+}$

$\theta_p^{(it)} = \theta^*$

This is an example of the *CC algorithm for exponential family IRT models* which will be discussed in detail below. Second, we drew μ and σ^2 from $f(\mu, \sigma^2|\theta^{(it)})$,



(a) $n = 5000$ and $k = 2$.

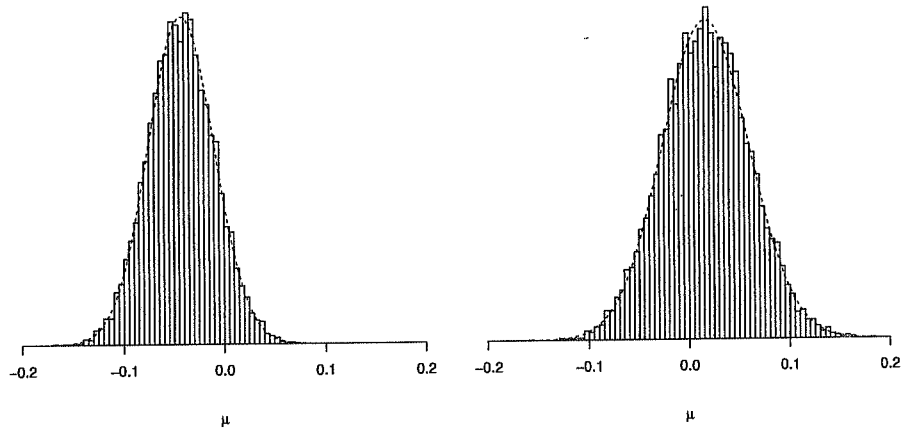
(b) $n = 1000$ and $k = 10$.

FIGURE 2.

Kolmogorov-Smirnov test statistic for 500 Markov Chains comparing the distribution of the 500 draws from the full conditional of μ at each iteration compared to iteration $1,000$

the joint posterior of the mean and variance of normal data, a case that has been studied extensively. If we assume independent non-informative priors for the hyper-parameters μ and σ^2, a sample from the posterior is obtained as follows. First, we drew σ^2 from an Inv-$\chi^2(n-1, s^2)$ distribution, then we drew μ from $N(m, \sigma^2)$, where m and s^2 denote the sample mean and variance of $\theta^{(it)}$ (see, e.g., Gelman et al., 2004, 3.2). A script for GNU-R (R Development Core Team, 2010) is provided in the Appendix.

We applied this Gibbs sampler to simulated datasets where $n = 5,000$ ($n = 1,000$) persons were randomly assigned $k = 2$ ($k = 10$) items out of the $I = 100$ Rasch items in the study. The item parameters were equally spaced between $[-4, 4]$ and abilities are drawn from a standard normal distribution. The Gibbs sampler in each simulation was run for $10,000$ iterations, starting from $\mu^{(0)} = -2$ and $(\sigma^2)^{(0)} = 2$. Figure 1 shows trace plots of $\mu^{(it)}$ against it showing that convergence was almost immediate in both simulations. To test for convergence of μ we started 500 Markov Chains running $1,000$ iterations for both simulations. We compared the distribution



(a) $n = 5000$ and $k = 2$. (b) $n = 1000$ and $k = 10$.

FIGURE 3.

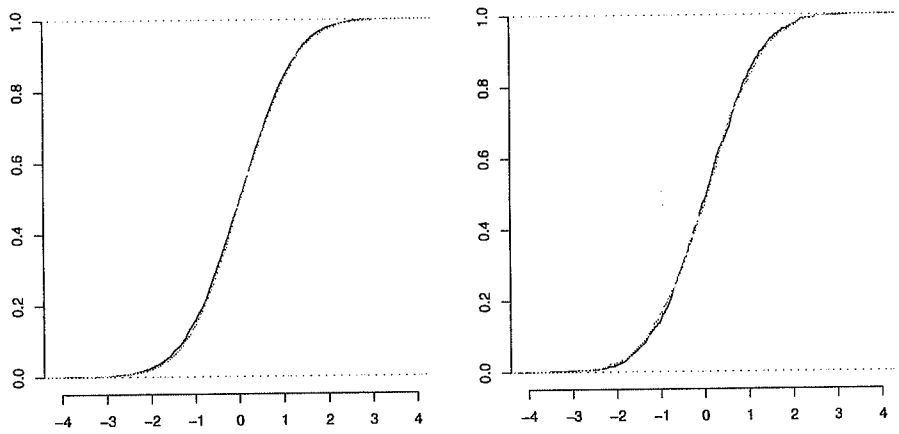
Posterior density of μ.

of the 500 draws from the full conditional of μ at each iteration with that at the last iteration using a Kolmogorov-Smirnov test. Figure 2 shows the test statistic against iterations and confirms that convergence to the stationary distribution is almost immediate. Figure 3 shows two density plots of the posterior of μ. Figure 4 shows the PV distribution and the true ability distribution and confirms that both are the same in the simulations. Figure 5 shows that PVs introduce no autocorrelation to the Markov Chain. This is because the PVs for a person in each iteration are independent conditional on the model parameters.

Note that the posterior variance in Figure 3.(b) is larger than that in Figure 3.(a), while the total number of observations $N = n \times k$ is the same in both cases. This shows that it is statistically more efficient to have many persons responding to a few of the items than the other way around (Lord & Novick, 1968, 11). To see this, consider the average of the PVs

$$\bar{\theta}^* = \frac{1}{n} \sum_{p=1}^{n} \theta_p^*,$$

which is an unbiased estimator of the average ability in the population. Since the



(a) $n = 5000$ and $k = 2$. (b) $n = 1000$ and $k = 10$.

FIGURE 4.
Cumulative density plots of θ (black) and θ^* (gray)

observations for different persons are independent,

$$Var\left(\bar{\theta}^*\right) = \frac{1}{n^2}\sum_{p=1}^{n} Var\left(\theta_p^*\right). \tag{6}$$

Using the variance decomposition formula, the variance of θ_p^* is

$$Var\left(\theta_p^*\right) = Var\left(E\left[\theta_p^*|\mathbf{X}_p\right]\right) + E\left(Var\left[\theta_p^*|\mathbf{X}_p\right]\right).$$

Plugging this result in Eq. (6) gives

$$Var\left(\bar{\theta}^*\right) = \frac{1}{n^2}\sum_{p=1}^{n} Var\left(E\left[\theta_p^*|\mathbf{X}_p\right]\right) + \frac{1}{n^2}\sum_{p=1}^{n} E\left(Var\left[\theta_p^*|\mathbf{X}_p\right]\right), \tag{7}$$

where $Var\left(E\left[\theta_p^*|\mathbf{X}_p\right]\right)$ is the variance of the posterior means and $E\left(Var\left[\theta_p^*|\mathbf{X}_p\right]\right)$ is the expected posterior variance of person p, both over repeated collection of data. The posterior variance is smaller for persons responding to many items, while the variance of the posterior means is approximately independent of the number of items. This can be shown by approximating (7), assuming that $E\left[\theta_p^*|\mathbf{X}_p\right] = \theta_p$ and $E\left(Var\left[\theta_p^*|\mathbf{X}_p\right]\right) \approx \sigma^2/k$ for all persons p. This implies $Var\left(E\left[\theta_p^*|\mathbf{X}_p\right]\right) = Var\left(\theta\right)$

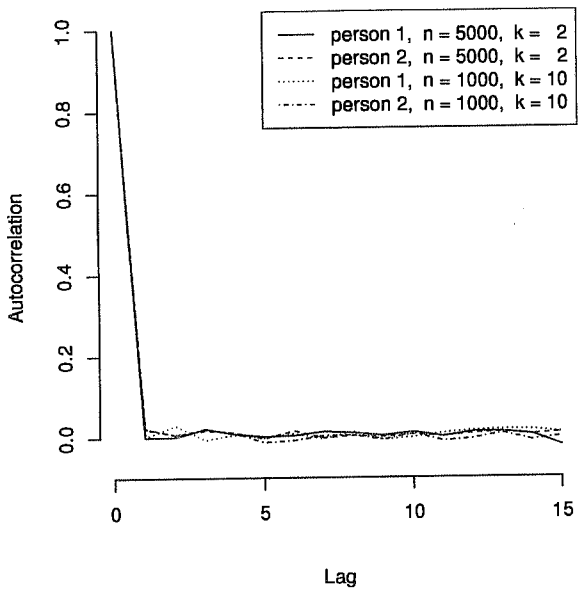


FIGURE 5.

Autocorrelation of PVs for two persons in the two simulations.

and

$$Var\left(\bar{\theta}^*\right) \approx \frac{Var\left(\theta\right)}{n} + \frac{\sigma^2}{n \times k}.$$

Note that the second term is not influenced by any assignment of n or k since their product always equals N. The first term, however, is smallest whenever $n = N$, and $k = 1$. Thus, if we are interested in studying the distribution of θ, we want to maximize the number of persons in the sample and not necessarily require that each person responds to many items.

3. The Conditional-Composition Algorithm

The idea is to generate a sample from $f(\mathbf{x}_p, \theta|\mathbf{y}_p, \mathbb{\Gamma}^{(it)})$ using composition, i.e., a candidate ability θ^* for person p is sampled from the structural model $f(\theta_p|\mathbf{y}_p, \mathbb{\Gamma}^{(it)})$ and used to sample a *response pattern* $\mathbf{x}^*$ from the IRT model $p(\mathbf{x}_p|\theta^*)$. Repeating these two simulations provides a set of ordered pairs $\{\theta_i^*, \mathbf{x}_i^*\}$. If we then ignore

all pairs where $\mathbf{x}_i^* \neq \mathbf{x}_p$, we obtain a sample from the posterior $f(\theta|\mathbf{x}_p, \mathbf{y}_p)$. The algorithm in pseudo code is then

Repeat:

Generate θ^* from $f(\theta|\mathbf{y}_p)$

Generate $\mathbf{x}^*$ from $p(\mathbf{x}|\theta^*)$

Until: $\mathbf{x}^* = \mathbf{x}_p$

$\theta_p = \theta^*$

We call this the *Conditional Composition (CC) algorithm* because it uses the method of composition to sample from a conditional distribution.

An algorithm is only an algorithm when it is guaranteed to stop at a certain moment. We therefore assume that item responses are discrete and there is a positive probability of generating the observed response pattern. Nevertheless, it is clear that, for any test of nontrivial length, there are many possible score patterns and it may take a long time for the CC algorithm to produce a PV. However, the CC algorithm is feasible when each person responds to a small number of items. In the next section, we discuss ways to make the algorithm more efficient when the number of persons taking a test *and/or* the number of items is increased.

4. Gaining Computational Efficiency

When the purpose is to make inferences about the structural model and we are not interested in the abilities of individual students, it is more efficient in a statistical sense to have many persons respond to a few items each than to have few persons responding to many items. An illustration was given earlier. Thus, our first priority was to adapt the CC algorithm to deal with many persons, this is discussed in the following section. We then describe the CC algorithm for the situation where the IRT model is a member of the exponential family. The advantage of an exponential family

IRT model is that posteriors depend only on the data via the sufficient statistic for ability: e.g., the number of correct answers in the Rasch model. There are many response patterns that lead to the same sufficient statistic and the CC algorithm runs until it finds one of them.

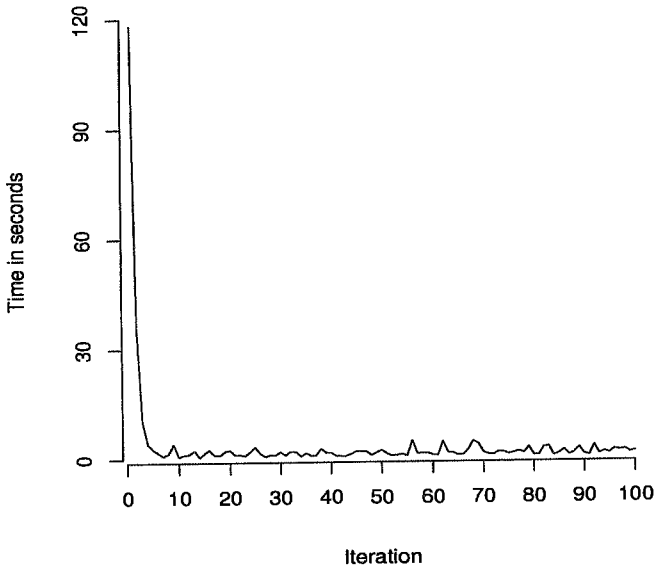


FIGURE 6.

Number of seconds for 100 consecutive iterations Gibbs Sampler

4.1. Many Persons: Recycling

When the basic CC algorithm is used to generate a PV for person p, it discards all candidate abilities that generate a score pattern that does not match that of person p. A simple way to make the algorithm more efficient is to make use of these intermediate candidate values. We grouped persons responding to the same items according to their value on the background variables $\mathbf{y}$. Intermediate candidate values θ^* that generate a response pattern $\mathbf{x}^*$ can be assigned to any person p in the same group whose response pattern matches $\mathbf{x}^*$. Thus, instead of sampling a PV for each person, we generated $n_{\mathbf{x}}$ PVs for each response pattern. In pseudo code, this

leads to the following algorithm which we call *the CC-R algorithm*:

Repeat:

Generate θ^* from $f(\theta|\mathbf{y})$

Generate $\mathbf{x}^*$ from $p(\mathbf{x}|\theta^*)$

If $n_{\mathbf{x}^*} > 0$, then $R_{n_{\mathbf{x}^*}, c_{\mathbf{x}^*}} = \theta^*$ and $n_{\mathbf{x}^*} = n_{\mathbf{x}^*} - 1$

Until $n_{\mathbf{x}} = 0$, $\forall \mathbf{x}$

The PVs are stored in a matrix $\mathbb{R}$ and $c_{\mathbf{x}}$ denotes the column in $\mathbb{R}$ corresponding to response pattern $\mathbf{x}$. If necessary, the PVs can then be assigned to persons in the sample.

If there is only one person in each marginal distribution defined by $\mathbf{y}$, the CC-R algorithm reduces to the basic CC algorithm. Thus, recycling becomes more efficient when there are few groups with many persons in each group. The efficiency of recycling is further determined by the time it takes to find PVs for each observed response pattern in a marginal distribution. A nice aspect of the CC-R algorithm is that it is *self-weighting* in the sense that, if the model fits the data, the probability to generate each response pattern will match the proportions in the data. Hence, we expect iterations to become faster as the Gibbs sampler converges. A simulation is used to illustrate this. Data were generated using $k = 5$ Rasch items with item parameters equally spaced between $[-2, 2]$ and $n = 1,000$ persons sampled from a standard normal distribution. A Gibbs sampler was used to estimate μ and σ. The starting values $\mu^{(0)} = -2$ and $(\sigma^2)^{(0)} = .3$ are far off from the true parameters, and, consequently, the Gibbs sampler takes a great deal of time in the first step. In Figure 6, the time of each consecutive iteration is plotted for the first 100 iterations and confirms that time decreases significantly when the Markov Chain converges.

4.2. Many Items: Exponential Family IRT Models and the CC Algorithm

If the IRT model belongs to the EF, all information about ability is contained in the sufficient statistic. Specifically, $\mathbf{x}_p \perp\!\!\!\perp \theta_p | t_p$, where $t_p = t(\mathbf{x}_p)$ denotes the sufficient statistic for the ability of person p. It follows that $\theta_p \perp\!\!\!\perp \mathbf{x}_p | t_p$, according to Theorem 3.1 in Dawid (1979), implying

$$\begin{aligned} f(\theta_p | t_p, \mathbf{y}_p) &= f(\theta_p | \mathbf{x}_p, t_p, \mathbf{y}_p) \\ &= f(\theta_p | t_p, \mathbf{y}_p) \\ &\propto p(t_p | \theta_p) f(\theta_p | \mathbf{y}_p). \end{aligned}$$

That is, the posterior distribution of person p is characterized by t_p instead of $\mathbf{x}_p$. Thus, to sample a PV for person p, a candidate value θ^* is sampled from $f(\theta_p | \mathbf{y}_p)$ and t_p is sampled from $p(t_p | \theta^*)$ conditional on the candidate value. In pseudo code, the algorithm which we call *the CC-EF algorithm* is

Repeat:

Generate θ^* from $f(\theta | \mathbf{y}_p)$

Generate t^* from $p(t | \theta^*)$

Until: $t^* = t_p$

$\theta_p = \theta^*$

The efficiency of the algorithm is determined by the number of response patterns corresponding to the same value of the sufficient statistic. In the Rasch model, for example, the sufficient statistic is the number of correct score $\sum_i^k x_{ip} = x_{+p}$. With 10 binary items there are $2^{10} = 1024$ different response patterns to consider, but only 11 different scores.

4.3. Many Persons and Many Items: Recycling for EF IRT models

When an EF IRT is used to generate PVs for persons in the marginal distribution $f(\theta|\mathrm{y})$, the intermediate candidate abilities θ^* that generate a sufficient statistic t^* can be assigned to any person p in the marginal distribution (i.e., all persons p with $\mathrm{y}_p = \mathrm{y}$ that respond to the same items) for which $t^* = t_p$. Let n_t denote the the number of persons in the marginal distribution with statistic t. We generated an iid sample of size n_t from $f(\theta|t)$ for each statistic t. In pseudo code, this leads to the following algorithm which we call *the CC-R-EF algorithm*:

Repeat :

Generate θ^* from $f(\theta|\mathrm{y})$

Generate t^* from $p(t|\theta^*)$

If $n_{t^*} > 0$, then $R_{n_{t^*},c_{t^*}} = \theta^*$ and $n_{t^*} = n_{t^*} - 1$.

Until $n_t = 0$, $\forall t$

The PVs are stored in the matrix $\mathbb{R}$, and c_t denotes the column in $\mathbb{R}$ corresponding to the statistic t.

5. Efficiency of the Different CC Algorithms

Data for the random effects Rasch model were simulated and used to illustrate performance of the different CC algorithms: item responses of a *small* sample of $n = 1,000$ persons to $k = 2, 5, 10$ and 100 items, and a *large* sample of $n = 1,000,000$ persons to $k = 5$ and 10 items. We assumed that $k = I$, i.e., the number of items in the study equals the number of items taken by each person. Item parameters were sampled from a uniform distribution in the range $[-2, 2]$. Ability parameters were sampled from a standard normal population. The different CC algorithms were used to produce five PVs for each person and we averaged the number of trials needed to produce one PV for each person. The results are in Table 1.

TABLE 1.

Average number of trials

	$n = 1,000$			
k	CC	CC-R	CC-EF	CC-R-EF
2	3.970	1.087	2.937	1.030
5	34.864	4.487	5.793	1.170
10	1146.439	285.928	10.182	1.333
100	-	-	93.561	3.888
	$n = 1,000,000$			
k	CC	CC-R	CC-EF	CC-R-EF
5	-	1.089	5.997	1.004
10	-	-	-	1.304

Table 1 confirms that the CC algorithm ("CC") took longer to assign PVs to each person when k became larger. The number of trials was greatly reduced when we used the CC-R algorithm ("CC-R") and the sample size increased. However, the efficiency of both the CC and CC-R algorithm decreased when persons responded to more items. Table 1 confirms that compared to the CC and CC-R algorithms, the CC-EF algorithm is more efficient when there are more items. For instance, when the small sample responded to two items the CC-EF algorithm was $3.970/2.937 = 1.351$ times more efficient than the CC algorithm, but when they responded to ten items it was $1,146.439/10.182 = 112.594$ times more efficient. As expected, the results show that the number of trials required by the CC algorithm is approximately 2^k whereas the CC-EF algorithm requires approximately $k + 1$ trails. It is clear that the CC-R-EF algorithm ("CC-R-EF") is most efficient. Compared to the other algorithms, its efficiency increases when both the number of items and the number of persons increases.

6. A Gibbs sampler for multilevel IRT models

A general framework for multilevel regression models in the context of structural IRT models was introduced by Fox and Glas (2001) and Fox (2010). A multilevel model for school s including the effect of n_p student characteristics (level one predictors) on ability is

$$\theta_s = \mathbf{y}_s\beta_s + \mathbf{e}_s, \tag{8}$$

where $\mathbf{y}_s$ is a $n_s \times n_p$ matrix containing the student characteristics, $\mathbf{e}_s$ is a $n_p \times 1$ vector containing student-specific residuals and β_s is a $n_p \times 1$ vector of random effects for school s. The elements in $\mathbf{e}_s$ are *i.i.d.* Normal$(0, \sigma)$ variates. The random effects for school s are the outcome of a regression model defined at the school level including the effects of n_q school-specific covariates

$$\beta_s = \mathbf{w}_s\gamma + \mathbf{r}_s, \tag{9}$$

where $\mathbf{w}_s$ is a $n_p \times n_q$ matrix with school characteristics and γ is a $n_q \times 1$ vector of fixed effects. The matrix $\mathbf{w}_s$ contains the stacked vectors $\mathbf{w}_{js}^T$ for the jth random effect at school s relating to the fixed effects γ. That is,

$$\mathbf{w}_s = \begin{pmatrix} \mathbf{w}_{1s}^T & 0 & \dots & 0 \\ 0 & \mathbf{w}_{2s}^T & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{w}_{n_p s}^T \end{pmatrix}.$$

The $n_p \times 1$ vector $\mathbf{r}_s$ contains school-specific residuals. It is multivariate normally distributed with zero mean and variance-covariance matrix Σ_β.

6.1. Full Conditionals for the Multilevel Model

Given that we have a sample of PVs, the Gibbs sampler requires full conditional distributions for (sets of) parameters from Eqs. (8) and (9) considering the PVs as

data. The posterior distribution of σ, γ, β and Σ_β using prior independence of σ, γ and Σ_β is

$$f(\gamma,\beta,\Sigma_\beta,\sigma|\theta,\mathrm{y},\mathrm{w}) \propto \tag{10}$$
$$\left[\prod_s^S \prod_p^{n_s} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\left(\theta_{ps} - \mathrm{y}_{ps}\beta_s\right)^2\right)\right] f(\sigma) \left[\prod_q^{n_q} f(\gamma_q)\right] \left[\prod_s^S f(\beta_s|\Sigma_\beta)\right] f(\Sigma_\beta)$$

where $f(\sigma)$, $f(\gamma)$, $f(\beta_s|\Sigma_\beta)$ and $f(\Sigma_\beta)$ are prior distributions. The β_s are a priori exchangeable conditional on Σ_β. We derived the full conditional distributions from which to sample for the above model using specific choices for the prior distributions. Different prior distributions lead to different full conditional distributions and the reader is referred to Gelman et al. (2004) and Fox (2010) for other options.

Using Eqs. (10), (8) and (9) the full conditional distribution for β_s is proportional to

$$f(\beta_s|\mathrm{y},\mathrm{w},\theta,\sigma,\gamma,\Sigma_\beta) \propto$$
$$\exp\left(-\frac{1}{2}\left[\sigma^{-2}\left(\theta_s - \mathrm{y}_s\beta_s\right)^T\left(\theta_s - \mathrm{y}_s\beta_s\right) + \left(\beta_s - \mathrm{w}_s\gamma\right)^T\Sigma_\beta^{-1}\left(\beta_s - \mathrm{w}_s\gamma\right)\right]\right).$$

We see an exponent with two terms which are quadratic in the parameter vector of interest. To transform the full conditional into tractable form, we expanded the terms in the exponent and discarded terms not depending on β_s. Completing the square gives

$$f(\beta_s|\mathrm{y},\mathrm{w},\theta,\sigma,\gamma,\Sigma_\beta) \sim \tag{11}$$
$$N\left(\left(\sigma^{-2}\mathrm{y}_s^T\mathrm{y}_s + \Sigma_\beta^{-1}\right)^{-1}\left(\sigma^{-2}\mathrm{y}_s^T\theta_s + \Sigma_\beta^{-1}\mathrm{w}_s\gamma\right), \left(\sigma^{-2}\mathrm{y}_s^T\mathrm{y}_s + \Sigma_\beta^{-1}\right)^{-1}\right).$$

Using Eq. (10) and assuming an independent Normal prior with mean zero and

variance σ_γ^2 for the elements in γ results in the following full conditional for γ:

$$f(\gamma|\mathbf{y},\mathbf{w},\theta,\sigma,\beta,\Sigma_\beta,\sigma_\gamma) \propto$$
$$\exp\left(-\frac{1}{2}\left[\sum_s^S (\beta_s - \mathbf{w}_s\gamma)^T \Sigma_\beta^{-1} (\beta_s - \mathbf{w}_s\gamma) + \sigma_\gamma^{-2}\gamma^T\gamma\right]\right).$$

As before, we have an exponent with two terms which are quadratic in the parameter vector of interest. Completing the square gives

$$f(\gamma|\mathbf{w},\beta,\Sigma_\beta,\sigma_\gamma) \sim \tag{12}$$
$$N\left(\left(\sum_s^S \mathbf{w}_s^T\Sigma_\beta^{-1}\mathbf{w}_s + \sigma_\gamma^{-2}\mathbb{I}_{n_q}\right)^{-1}\sum_s^S \mathbf{w}_s^T\Sigma_\beta^{-1}\beta_s, \left(\sum_s^S \mathbf{w}_s^T\Sigma_\beta^{-1}\mathbf{w}_s + \sigma_\gamma^{-2}\mathbb{I}_{n_q}\right)^{-1}\right),$$

with $\mathbb{I}_{n_q}$ the $n_q \times n_q$ identity matrix.

Using Eq. (10) and assuming an Inverse Wishart prior distribution with a positive definite inverse scale matrix Σ_{β^*} and $n_{\beta^*} \geq n_p$ degrees of freedom, produces the following full conditional distribution for Σ_β:

$$f(\Sigma_\beta|\mathbf{y},\mathbf{w},\theta,\sigma,\gamma,\beta,\Sigma_{\beta^*},n_{\beta^*}) \propto$$
$$|\Sigma_\beta|^{-\frac{n_p+n_{\beta^*}}{2}} \exp\left(-\frac{1}{2}\left[\sum_s^S (\beta_s - \mathbf{w}_s\gamma)^T \Sigma_\beta^{-1} (\beta_s - \mathbf{w}_s\gamma) + \text{trace}\left(\Sigma_{\beta^*}\Sigma_\beta^{-1}\right)\right]\right),$$

since

$$\begin{aligned}(\beta_s - \mathbf{w}_s\gamma)^T \Sigma_\beta^{-1} (\beta_s - \mathbf{w}_s\gamma) &= \text{trace}\left((\beta_s - \mathbf{w}_s\gamma)^T \Sigma_\beta^{-1} (\beta_s - \mathbf{w}_s\gamma)\right) \\ &= \text{trace}\left((\beta_s - \mathbf{w}_s\gamma)(\beta_s - \mathbf{w}_s\gamma)^T \Sigma_\beta^{-1}\right),\end{aligned}$$

and

$$\begin{aligned}&\text{trace}\left((\beta_s - \mathbf{w}_s\gamma)(\beta_s - \mathbf{w}_s\gamma)^T \Sigma_\beta^{-1}\right) + \text{trace}\left(\Sigma_{\beta^*}\Sigma_\beta^{-1}\right) \\ &\qquad = \text{trace}\left(\left[(\beta_s - \mathbf{w}_s\gamma)(\beta_s - \mathbf{w}_s\gamma)^T + \Sigma_{\beta^*}\right]\Sigma_\beta^{-1}\right),\end{aligned}$$

it follows that the full conditional for Σ_β is proportional to an Inverse Wishart

distribution

$$f(\Sigma_\beta|\mathbf{w},\gamma,\beta,\Sigma_{\beta^*},n_{\beta^*}) \sim \tag{13}$$
$$\text{Inverse-Wishart}\left(\sum_s^S(\beta_s-\mathbf{w}_s\gamma)(\beta_s-\mathbf{w}_s\gamma)^T+\Sigma_{\beta^*},S+n_{\beta^*}\right).$$

Using Eq. (10) and assuming a Gamma prior distribution with parameters a_σ and b_σ for the precision $\frac{1}{\sigma^2}$, results in the following corresponding full conditional distribution:

$$f(\sigma^{-2}|\mathbf{y},\mathbf{w},\theta,\gamma,\beta,\Sigma_\beta,a_\sigma,b_\sigma) \propto$$
$$\left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+a_\sigma-1}\exp\left(-\frac{1}{2}\sum_s^S(\theta_s-\mathbf{y}_s\beta_s)^T(\theta_s-\mathbf{y}_s\beta_s)\left(\frac{1}{\sigma^2}\right)-b_\sigma\left(\frac{1}{\sigma^2}\right)\right).$$

in which we recognize the Gamma distribution

$$f(\sigma^{-2}|\mathbf{y},\theta,\beta,a_\sigma,b_\sigma) \sim \tag{14}$$
$$\text{Gamma}\left(\frac{n}{2}+a_\sigma,\frac{1}{2}\sum_s^S(\theta_s-\mathbf{y}_s\beta_s)^T(\theta_s-\mathbf{y}_s\beta_s)\ +b_\sigma\right).$$

Exactly the same Gibbs sampler is given in (Fox, 2010, Chapter 6.5).

6.2. Simulated Example

We simulated data using Eq. (8) as a structural model and the One-Parameter Logistic Model (OPLM; Verhelst & Glas, 1995) as the measurement model. The OPLM model is

$$P(x_{pi}=1|a_i,b_i,\theta_p)=\frac{\exp\left(a_i(\theta_p-b_i)\right)}{1+\exp\left(a_i(\theta_p-b_i)\right)}, \tag{15}$$

with $x_{pi}=1$ when person p gives a correct response to item i and $x_{pi}=0$ otherwise. Each item is characterized by a positive *integer-valued* discrimination parameter a_i and difficulty parameter b_i. The OPLM model is an EF IRT model with $t_p=\sum_i^k a_i x_{pi}$ sufficient for θ_p.

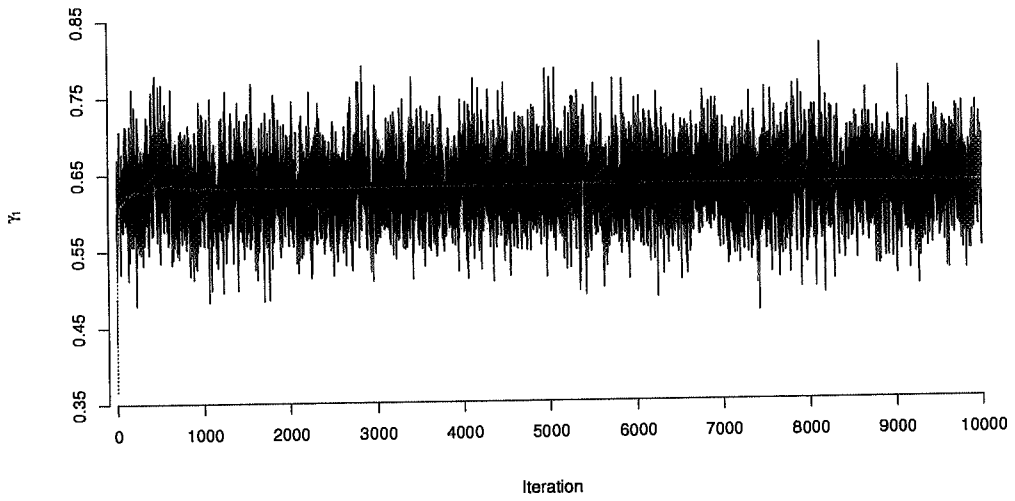


FIGURE 7.
Trace plot of γ_1.

A dataset was simulated consisting of $S = 1,000$ schools with $n_s = 15$ students each, responding to $k = I = 20$ OPLM items. The discrimination parameters, $\mathbf{a} = [a_i]$, were randomly chosen from $(1, 2, 3, 4)$. The difficulty parameters, $\mathbf{b} = [b_i]$, are random variates drawn from an Uniform$(-4, 4)$ distribution. The ability parameters were generated using the model in Eq. 8 with $n_p = 5$ random effects and $n_q = 10$ random effects. The first column of the matrix $\mathbf{y}$ was set to one, and the remaining $n_p - 1$ columns filled with random draws from a Bernoulli(0.5) distribution. The matrix $\mathbf{w}_s$ was an $n_p \times n_q$ diagonal matrix with diagonal elements $\mathbf{w}_{1s}^T = \mathbf{w}_{2s}^T = ... = \mathbf{w}_{n_p s}^T$, where $\mathbf{w}_{1s}$ was a 2×1 column vector of school characteristics, of which the first element was set to one and the second element was a random draw from a Bernoulli(0.5) distribution. Thus there was one school level covariate included for each random effect. The vector γ consisted of n_q random variates from a Normal$(0, 1)$ distribution. The vectors $\boldsymbol{\beta}_s$ were drawn from a Multivariate Normal$(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$ distribution. The level one residual variance σ was set to one, and the random effects variance-covariance matrix $\boldsymbol{\Sigma}_\beta$ had main diagonal elements set

to 0.5 and off diagonal elements to 0.1.

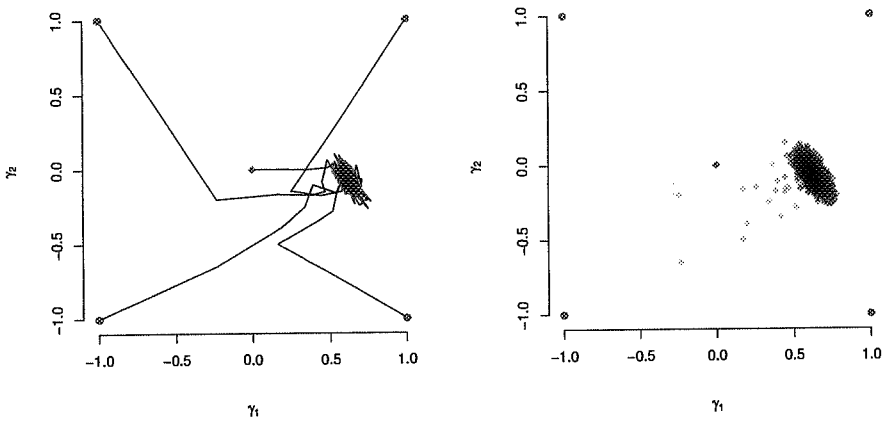
To implement the Gibbs sampler, prior parameters and starting values $\theta^{(0)}$, $\gamma^{(0)}$, $\beta^{(0)}$, $\Sigma_\beta^{(0)}$ and $\sigma^{(0)}$ had to be specified. The prior parameters for the level one residuals were set to $a_\sigma = 1$ and $b_\sigma = 1$. The prior variance σ_γ^2 parameter for γ was set to one. The prior scale matrix Σ_β was the $n_q \times n_q$ identity matrix and the corresponding degrees of freedom $n_{\beta^*} = n_q$. The elements in $\theta^{(0)}$ were generated from a standard normal distribution. The vectors $\gamma^{(0)}$ and $\beta_s^{(0)}$, for $s = 1, 2, ..., S$, were set to zero. The level one residual error standard deviation $\sigma^{(0)}$ was set to one, and the random effects variance-covariance matrix $\Sigma_\beta^{(0)}$ was the $n_q \times n_q$ identity matrix.

The Gibbs sampler proceeds by setting the iteration counter it to zero and repeating the following sequence in each of $10,000$ iterations:

1. Generate PVs $\theta^{(it)} \sim f\left(\theta|\mathbb{X}, \mathbf{a}, \mathbf{b}, \mathbf{y}, \beta^{(it-1)}, \sigma^{(it-1)}\right)$ from Eqs. (8) and (15) using the CC algorithm for EF IRT models.
2. Generate parameters of the structural model:
   (a) Generate $\beta_s^{(it)} \sim f\left(\beta_s|\mathbf{y}, \mathbf{w}, \theta^{(it)}, \sigma^{(it-1)}, \gamma^{(it-1)}, \Sigma_\beta^{(it-1)}\right)$ for $s = 1, 2, ..., S$ from Eq. (11).
   (b) Generate $\gamma^{(it)} \sim f\left(\gamma|\mathbf{w}, \beta^{(it)}, \Sigma_\beta^{(it-1)}, \sigma_\gamma\right)$ from Eq. (12).
   (c) Generate $\Sigma_\beta^{(it)} \sim f\left(\Sigma_\beta|\mathbf{w}, \theta^{(it)}, \gamma^{(it)}, \beta^{(it)}, n_{\beta^*}, \Sigma_{\beta^*}\right)$ from Eq. (13).
   (d) Generate $\sigma^{(it)} \sim f\left(\sigma^{-2}|\mathbf{y}, \theta^{(it)}, \beta^{(it)}, a_\sigma, b_\sigma\right)$ from Eq. (14).

We implemented the Gibbs sampler in R using a C++ routine to generate PVs [1]. Figure 7 shows a trace plot of γ_1 and shows that convergence was almost immediate. The green line depicts a running average updated at each iteration. Figure 8 shows

[1]The R code and C++ routine are available from the first author.



(a) Initial 100. (b) All 10,000.

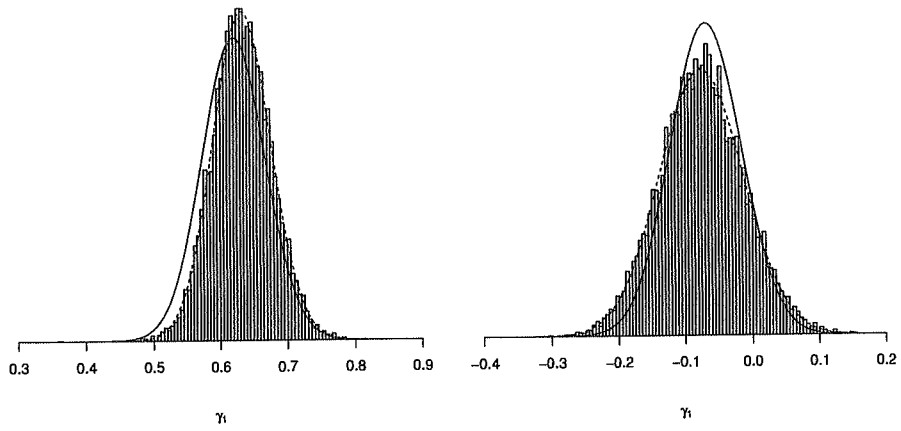
FIGURE 8.

Draws from the joint posterior of γ_1 and γ_2 for 5 separate Markov Chains. The red dots mark the starting points.

the path of 5 Markov Chains with different starting values through the joint posterior distribution of γ_1 and γ_2 in the first 100 iterations and shows a scatter plot of all draws for the joint posterior distribution for the five Markov Chains. The posterior density of γ_1 and γ_2 are shown in Figure 9.

7. Discussion

We have described a Gibbs sampler for structural measurement models and introduced the CC algorithm as a generic algorithm to sample PVs. The CC algorithm can be used with any IRT model for discrete item responses. It is most efficient for applications using an EF IRT model due to the presence of a sufficient statistic for ability. For non EF IRT models, the CC algorithm is useful when persons respond to few items, which is a good design for large scale educational surveys where interest is focused on population inference. When the sample is large, the CC algorithm becomes more efficient by recycling, i.e., by using intermediate candidate values. Furthermore, recycling makes the algorithm more efficient when the number of per-



(a) γ_1. (b) γ_2.

FIGURE 9.

Posterior densities of γ_1 and γ_2. The solid line shows the estimated density after 100 iterations and the dotted line after 10.000.

sons in the sample becomes larger. An important advantage of the CC algorithm is that it does not use data augmentation and generates Markov Chains that have much less autocorrelation than samplers that use data augmentation. Since adding persons does not introduce additional autocorrelation, convergence is not compromised. This makes the CC algorithm attractive for large scale applications such as educational surveys.

We have assumed that the IRT model was known a priori. For EF IRT models this is a mild condition since the model can be fitted independently of the structural model using conditional likelihood methods. A straightforward extension is to concurrently estimate the item- and structural parameters.

In closing we mention that the CC algorithm is not limited to unidimensional IRT models. It can be applied to multidimensional IRT models where PVs in each dimension are generated conditional on the other dimensions. We leave this as a topic for future research.

References

Adams, R., Wilson, M., & Wu, M. (1997). Multilevel item response models: An approach to errors in variables regression. *Journal of Educational and Behavioral Statistics*, *22*, 47-76.

Albert, J. (1992). Bayesian estimation of normal ogive item response curves using Gibbs sampling. *Journal of Educational Statistics*, *17*, 251-269.

Béguin, A., & Glas, C. (2001). MCMC estimation and some fit analysis of multidimensional IRT models. *Psychometrika*, *66*, 471-488.

Casella, G., & George, E. (1992). Explaining the Gibbs sampler. *The American Statistician*, *46*, 167-174.

Dawid, A. (1979). Conditional independence in statistical theory. *Journal of the Royal Statistical Society, Series B (Methodological)*, *41*, 1-31.

Fox, J. (2010). *Bayesian item response modeling.* Springer.

Fox, J., & Glas, C. (2001). Bayesian estimation of a multi-level IRT model using Gibbs sampling. *Psychometrika*, *66*, 271-288.

Gelman, A., Carlin, B., Stern, H., & Rubin, D. (2004). *Bayesian data analysis* (Second ed.). Chapman & Hall/CRC.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721-741.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores.* Reading, MA: Addison-Wesley.

Maier, K. (2001). A Rasch hierarchical measurement model. *Journal of Educational and Behavioral Statistics*, *26*, 307-330.

Maris, G., & Bechger, T. M. (2005). An introduction to the DA-T gibbs sampler for the two-parameter logistic (2pl) model and beyond. *Psicólogica*, *26*, 327-352.

Maris, G., & Maris, E. (2002). A MCMC-method for models with continuous latent responses. *Psychometrika*, *67*, 335-350.

Mislevy, R. (1991). Randomization-based inference about latent variables from complex samples. *Psychometrika*, *56*, 177-196.

Patz, R. J., & Junker, B. W. (1999). A straightforward approach to Markov Chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics*, *24*(2), 146-178.

R Development Core Team. (2010). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Available from `http://www.R-project.org/` (ISBN 3-900051-07-0)

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: The Danish Institute of Educational Research. (Expanded edition, 1980. Chicago, The University of Chicago Press)

Rubin, D. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, *82*, 543-546.

Tanner, M. A. (1996). *Tools for statistical inference* (Third ed.). New york: Springer-Verlag.

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, *82*, 528-540.

Thomas, N., & Gan, N. (1997). Generating multiple imputations for matrix sampling data analyzed with item response models. *Journal of Educational and Behavioral Statistics*, *22*, 425-445.

Verhelst, N. D., & Glas, C. A. W. (1995). The one parameter logistic model: OPLM. In G. H. Fischer & I. W. Molenaar (Eds.), *Rasch models: Foundations, recent developments and applications* (p. 215-238). New York: Springer Verlag.

8. Appendix

8.1. R-script for simulating scores from the Rasch model

Different approaches may be taken to simulate item scores from the Rasch model. Here we note that without loss of generality we may assume that there exists a latent response variable U such that

$$P(X_{pi} = 1|\delta_i, \theta_p) = P(U \leq \theta_p - \delta_i),$$

where U follows a standard logistic distribution and $P(U \leq \theta_p - \delta_i)$ denotes the cumulative logistic distribution. Note that

$$\begin{aligned} P(X_{pi} = 1|\delta_i, \theta_p) &= \int_{-\infty}^{\theta_p - \delta_i} \frac{\exp(-U)}{(1+\exp(-U))^2} dU \\ &= \frac{1}{1+\exp(-(\theta-\delta))} = \frac{\exp(\theta-\delta)}{1+\exp(\theta-\delta)}, \end{aligned}$$

the probability of providing a correct response to the Rasch model for item i by person p. We use this in our R function `Score`, where the sum is over the responses per individual:

```
Score=function(theta,delta)
{
  sum(1*(rlogis(length(delta),0,1)<=(theta-delta)))
}
```

8.2. R-script for the random-effects Rasch model

For the random-effects Rasch model we sample candidate abilities from a Normal(μ, σ^2) model and use the R function `Score` to generate test scores based on the Rasch model. The scores generated by persons are stored in the vector `score` where pth element corresponds to the score of person p. The full conditional distribution for μ is $N(m, \sigma^2)$, where m denotes the mean of $\theta^{(it)}$. The full conditional for σ^2 is the Inv-$\chi^2(n-1, s^2)$ distribution. This is equivalent to the Inv-$\Gamma\left(\frac{n-1}{2}, (n-1)s^2\right)$ distribution, from which we simulate with the R function `rgamma`. This is used to draw σ^2 in the following R script:

```
for(it in 1:nIter)
{
  for (p in 1:nPers){
    repeat
    {
      pv[p]      = rnorm(1,mu,sqrt(sigma))
      pscore     = Score(pv[p],delta)
      if (pscore==score[p]) break
    }
  }
  mu    = rnorm(1,mean(pv),sd=sqrt(sigma/nPers))
  sigma = 1/rgamma(1,shape=(nPers-1)/2,rate=.5*var(pv)*(nPers-1))
}
```