The Impact of Markov Chain Convergence on Estimation of Mixture IRT Model Parameters

Educational and Psychological

Measurement

2020, Vol. 80(5) 975–994

© The Author(s) 2020

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/0013164419898228

journals.sagepub.com/home/epm

Yoonsun Jang¹ and Allan S. Cohen²

Abstract

A nonconverged Markov chain can potentially lead to invalid inferences about model parameters. The purpose of this study was to assess the effect of a nonconverged Markov chain on the estimation of parameters for mixture item response theory models using a Markov chain Monte Carlo algorithm. A simulation study was conducted to investigate the accuracy of model parameters estimated with different degree of convergence. Results indicated the accuracy of the estimated model parameters for the mixture item response theory models decreased as the number of iterations of the Markov chain decreased. In particular, increasing the number of burn-in iterations resulted in more accurate estimation of mixture IRT model parameters. In addition, the different methods for monitoring convergence of a Markov chain resulted in different degrees of convergence despite almost identical accuracy of estimation.

Keywords

mixture item response theory models, MCMC algorithm, convergence diagnostics

Introduction

A mixture item response theory (MixIRT) model (Mislevy & Verhelst, 1990; Rost, 1990), which is a combination model merging latent class analysis with an IRT

¹National Youth Policy Institute, Sejong-si, Korea

²University of Georgia, Athens, GA, USA

Corresponding Author:

Yoonsun Jang, National Youth Policy Institute, Social Policy Building, Sejong National Research Complex, 370, Sejong-daero, Sejong-si, 30147, Korea.

Email: ysjang@nypi.re.kr

model, has been proposed for modeling heterogeneous populations. MixIRT models have been applied in a variety of educational assessments. They can provide useful information about nuisance factors that might affect the item and ability parameter estimates. For example, time limits of tests might affect examinees' performance at the end of tests (Bolt, Cohen, & Wollack, 2002). Additionally, MixIRT models have been used to help identify latent classes that represent different item response patterns (e.g., Mittelhaeuser, Béguin, & Sijtsma, 2015). MixIRT models have also been applied to detect differential item functioning (DIF) items (e.g., Cohen & Bolt, 2005; Samuelsen, 2005), differences in performance on computerized adaptive tests (e.g., Jiao, Macready, Liu, & Cho, 2012), and different relationships of multiple latent traits (e.g., Bacci, Bartolucci, Grilli, & Rampichini, 2017; Huang, 2017). Furthermore, MixIRT models have been used for comparing performance levels for standard setting (e.g., Jiao, Lissits, Macready, Wang, & Liang, 2011) and for detecting different response styles, such as extreme response style (e.g., Huang, 2016), inattentive response (e.g., Jin, Chen, & Wang, 2017), and performance decline (e.g., Jin & Wang, 2014; List, Robitzsch, Lüdtke, Köller, & Nagy, 2017).

There are two popular methods for estimation of MixIRT models: one is maximum likelihood estimation (MLE) and the other is Markov chain Monte Carlo (MCMC) estimation. MCMC estimation is useful, when fitting complex models, because it does not require integration of a high-dimensional likelihood function, although running the MCMC algorithm can be very time consuming. MLE, on the other hand, does not require as much time for estimation as MCMC. MLE, however, can lead to a local maximum rather than a global maximum likelihood solution (Finch & French, 2012). For these reasons, a number of studies have used an MCMC algorithm to estimate MixIRT models (e.g., Bolt et al., 2002; Cohen & Bolt, 2005; Huang, 2016, 2017; Jin et al., 2017; Jin & Wang, 2014; Kadengye, Ceulemans, & Van den Noortgate, 2013; Mittelhaeuser et al., 2015).

MCMC estimation is a convenient and flexible approach for fitting complex statistical models, however, it is difficult to answer the question at "what point is it reasonable to believe that the samples are truly representative of the underlying stationary distribution of the Markov chain?" (Cowles & Carlin, 1996, p. 883). Sahlin (2011) has noted that an MCMC algorithm is commonly initiated at a random point when uninformative prior distributions of parameters are applied. This starting point might or might not be located far from the high-density region of the posterior distribution. Consequently, the sample values at the beginning of the simulation might not be close to the true distribution. These invalid samples at the beginning of the simulation are not a major problem, because these initial parts of the chain, known as the burn-in period, are discarded. Determining the numbers of iterations for the burn-in and post-burn-in period, however, is necessary in order to minimize the effects of the burn-in period on the samples from the converged part of the chain.

A number of methods were proposed to assess a Markov chain convergence but there is little consensus regarding a single method. In a comparative review by Cowles and Carlin (1996), 13 methods for assessing convergence were compared

and classified according to a number of criteria. The first criterion was that each method assess the convergence of univariate of quantities or of the full joint distribution. The nature of outcome for each method (quantitative or graphical) was the second criterion. The third is whether each method mainly focuses on a bias from a non-representative chain or a variance to ensure converged chain. Additionally, the easiness of use and applicability were also considered to classify diagnostics. Based on the classification by Cowles and Carlin (1996), Gelman and Rubin's method, Raftery and Lewis's method, Geweke's method, and Heidelberger and Welch's method are applicable for any MCMC sampler and are easy to use.

In addition, Cowles and Carlin (1996) empirically applied these 13 diagnostics for monitoring convergence of a Markov chain. The results of empirical application revealed the disagreement among many of diagnostics. To be specific, the Geweke's method and Heidelberger and Welch's method were relatively lenient than the Gelman and Rubin's method and Raftery and Lewis's method. The results from the Roberts' method and the Gibbs stopper were extremely varied, and the result from the Zellner and Min's method varied depending on the conditions. The Robert's and the Jonson's method indicated clear convergence, while the Liu et al.'s method indicated the failure to converge. Cowles and Carlin (1996) assumed that these disagreements might result from their different definition about 'convergence' and recommended taking a variety of diagnostics to assess convergence of a Markov chain.

Brooks and Roberts (1998) also reviewed 16 methods based on their theoretical and statistical features, and then compared them in terms of easiness of computation, applicability and interpretability. Among 16 methods, the methods proposed by Raftery and Lewis, Geweke, and Heidelberger and Welch are relatively easy to compute and interpret. And these methods are available for any sampler. However, Brooks and Roberts (1998) emphasized that it would be worth considering more than one method to avoid misleading because each method has both strengths and weaknesses.

Sinharay (2004) focused on application of several convergence diagnostics for the IRT models. Two empirical data sets were fitted a three-parameter logistic (3PL) IRT model and a testlet 3PL IRT model, respectively. The convergence was assessed by the three graphical methods (running mean, time series, and autocorrelation plots), and five quantitative method (Geweke, Heidelberger–Welch, Gelman–Rubin, Brooks–Gelman, and Raftery–Lewis methods), which are relatively popular methods. The results of empirical studies showed that the chance to pass the convergence assessment by each method tended to be increased as a total number of iterations increased. Moreover, the Gelman and Rubin's method and Brooks and Gelman's method were less rigorous than other methods. The Raftery and Lewis method was the most conservative among five diagnostics as in Cowles and Carlin (1996).

Susanto et al. (2018) also conducted an empirical study to compare the performance of four convergence diagnostics, including Brooks–Gelman–Rubin, Geweke, Raftery–Lewis, and Heidelberger–Welch. The finite mixture model was applied to fit the empirical data set that followed a multimodal distribution. The results of Susanto et al. (2018) mostly agreed with those reported by Cowles and Carlin (1996)

and Sinharay (2004). In particular, the results by Raftery and Lewis's method were insensitive to the number of iterations. To sum up, the results of convergence test can be different depending on the diagnostics, in particular the Raftery and Lewis method tends to disagree with other methods. Furthermore, applying multiple methods is recommended rather than monitoring convergence of a Markov chain by single method (Brooks & Roberts, 1998; Cowles & Carlin, 1996; Lesaffre & Lawson, 2012; Sinharay, 2004).

As described at the beginning of the introduction, the application of MixIRT models via MCMC algorithm has increased, but there is little information about the effect of the degree of convergence on the accuracy of estimates of model parameters. For these reasons, this study focused on the impact of convergence of Markov chain on the estimation MixIRT models. Indeed, the convergence of a Markov chain is not a special issue just in MixIRT model parameters. This study shows how a nonconvergent Markov chain affects the accuracy of estimates of MixIRT model parameters, in particular for dichotomous MixIRT models, as an illustrative example.

In this study, we investigated whether the findings reported in the previous research are also observed in MixIRT models. That is, we investigated how convergent rates differed depending on the methods for monitoring convergence. Additionally, we examined how the degree of convergence affects the accuracy of estimated MixIRT model parameters. Finally, the sensitivity of each method to the number of iterations was also explored. A simulation study was conducted to address these questions. In the following sections, we present brief descriptions of an MCMC algorithm, MixIRT models, and methods for monitoring convergence of a Markov chain.

The MCMC Algorithm

Drawing statistically consistent samples from a distribution is an important aspect of an effective simulation. However, such a simple task often cannot be done for high dimensional distributions without integrals or summations that are necessary for calculating the expectation and expected values of a distribution. MCMC provides an answer to the difficult problem of creating a simulation from a high dimensional distribution (Gamerman & Lopes, 2006). MCMC consists of two parts: the Markov chain design and the Monte Carlo integration. Notations used in this study follow those used by Gilks, Richardson, and Spiegelhalter (1998).

Suppose $\mathbf{X}$ be a vector of k random variables from $\pi(\cdot)$ the expectation can be defined and approximated by Monte Carlo integration by drawing samples $\{X_t, t = 1, \dots, n\}$ from $\pi(\cdot)$ as in the following equation:

$$E[f(\mathbf{X})] = \frac{\int f(x)\pi(\cdot)dx}{\int \pi(x)dx} \approx \frac{1}{n} \sum_{t=1}^n f(X_t). \quad (1)$$

From the law of large numbers, as n increases, the confidence in the estimates obtained also increases, when the samples are independent. Using Monte Carlo

integration, the integrals for expectation can be replaced by summations, which are simpler and easier to implement.

A Markov chain is one way to draw samples from $\pi(\cdot)$. A Markov chain is a sequence of random variables, $\{X_0, X_1, X_2, \dots\}$, from a distribution $P(X_{t+1}|X_t)$ that depends only on the current state of the chain. That is, given the present state, the conditional probability distribution of future states of the chain does not depend on the past. In addition, the first state (i.e., X_0) does not affect X_t under regularity conditions because the chain will eventually converge to a unique stationary distribution, $\phi(\cdot)$. Thus, after sufficient “burn-in” iterations, m , samples $\{X_t, t = m+1, \dots, n\}$, become dependent on samples from $\phi(\cdot)$. The expectation $E[f(X)]$ in Equation (1) is an estimate obtained using only post burn-in samples from the Markov chain:

$$\bar{f} = \frac{1}{n-m} \sum_{t=m+1}^n f(X_t) \quad (2)$$

By combining features of Monte Carlo integration and a Markov chain, a converged Markov chain generates statistically consistent samples from the target distribution, $\pi(\cdot)$.

MixIRT Models

The main features of MixIRT models are that the examinees are from a heterogeneous population consisting of multiple discrete latent classes and that an IRT model can address each latent class (Jiao et al., 2012). That is, a different set of item parameters is allowed for each latent class. The probability of a correct response using a mixture 3PL (Mix3PL) model is defined as follows;

$$P(y_{ij} = 1|\theta_j) = \sum_{g=1}^G \pi_g \times \left[c_{ig} + (1 - c_{ig}) \times \frac{1}{1 + \exp\{a_{ig}(-\theta_j + b_{ig})\}} \right], \quad (3)$$

where y_{ij} is the response to item i of examinee j , θ_j is the latent trait of examinee j , g is latent class, π_g is the probability of membership in latent class g with the constraint that $\sum_{g=1}^G \pi_g = 1$; c_{ig} , b_{ig} , and a_{ig} are the guessing parameter, item difficulty, and item discrimination of item i in latent class g , respectively. A mixture 2PL (Mix2PL) model can be defined by fixing the guessing parameter, c_{ig} , to zero, and a mixture Rasch (MixRasch) model can be obtained with the additional constraint that $a_{ig} = 1$.

Because MixIRT models can provide information about latent classes with a different set of item parameters, MixIRT models are suitable to explore both qualitative differences in examinees characterized by different latent class memberships and quantitative differences represented by a level of ability parameters. For instance, MixIRT models can be applied to explore the response styles, including extreme response style, acquiescent response style, or midpoint response style when research attitudes are measured by a questionnaire with Likert-type scale. Moreover, these models can be useful, for example, to improve the accuracy of item parameter

estimates by detecting and removing unmotivated or randomly guessing examinees' responses.

Monitoring Convergence

In this section, we present a brief overview of the methods for monitoring convergence of a Markov chain. In this article, we restricted our choices for convergence diagnostics to ones implemented in OpenBugs, PROC MCMC in SAS, Mplus, and R because of the popularity of these software programs. Moreover, the methods, which can be used in these software programs, (i.e., Gelman and Rubin's method, Geweke's method, Heidelberger and Welch's method, and Raftery and Lewis's method) are considered the easiest diagnostics to compute and interpret in the literature review. Two of the R packages are Bayesian Output Analysis (BOA: Smith, 2007) and CODA (Plummer, Best, Cowles, & Vines, 2006). The methods by Gelman and Rubin, Geweke, Heidelberger and Welch, Raftery and Lewis, as well as autocorrelation and trace plots are available in BOA, CODA, and SAS (PROC MCMC). In Mplus, potential scale reduction (PSR) is used to determine the convergence of a Markov chain (Asparouhov & Muthén, 2010). Also, there are a number of informative materials about how to assess convergence of a Markov chain when using these programs.

A single Markov chain has been commonly applied as in previous research (e.g., Cohen & Bolt, 2005; Huang, 2016; Jin & Wang, 2014; Kadengye et al., 2013; Sen, Cohen, & Kim, 2016). Although there are balanced arguments for using a very long chain (Geyer, 1992; Raftery & Lewis, 1992a) versus multiple chains (Gelman & Rubin, 1992), in this study, monitoring convergence focused on a very long chain. The graphical approaches, such as checking a trace or autocorrelation plot, were not considered because it is laborious to check manually these plots across a large number of replications in a simulation study. Consequently, four methods for checking convergence were applied in this study. Three of them are the Heidelberger and Welch method, the Geweke method, and the Raftery and Lewis method. The last method is the ratio of the Monte Carlo (MC) error with the posterior standard deviation (SD). Although this method does not have a theoretical basis, this is often suggested as a rule of thumb (e.g., Lesaffre & Lawson, 2012; Lunn, Jackson, Best, Thomas, & Spiegelhalter, 2013; Woodward, 2011) because it can be calculated easily. Thus, the MC error–SD ratio was compared with other three methods in this study.

Heidelberger and Welch's Method

The Heidelberger and Welch (1983) method assesses the convergence through two steps. The first step is to check stationarity based on the distance between the cumulative sum and the sampled posterior mean of the parameter. This step starts by comparing the cumulative sum of the first 10% of the Markov chain with the sampled

posterior mean. If the first 10% of the chain passes the stationarity test, the whole chain is taken into the second step. If not, the first 10% of the chain is discarded and an additional 10% of the chain is tested until it passes. This step is repeated until the stationarity test is passed, or until more than 50% of the chain is discarded. The latter indicates a nonconverged chain. The second step is to determine the accuracy of the chain. In this step, the credibility interval of the mean of each estimated parameter is generated and tested as to whether the interval meets a desired accuracy.

Geweke's Method

The Geweke (1992) method is also based on the sampled posterior mean for monitoring convergence. This method assesses whether the means of the initial and the last part of Markov chain are significantly different. It is assumed that the distribution of differences between the means from two parts of chain divided by the asymptotic standard error of the difference follows a Z distribution as the total number of iterations increases. Thus, the convergence of a chain can be tested by applying an unpaired Z -test. Geweke suggested using the first 10% and the last 50% of the chain to create the two samples.

Raftery and Lewis's Method

The Raftery and Lewis (1992b) method uses a specific posterior quantile instead of the sampled posterior means to monitor the convergence of a Markov chain. In essence, this method estimates the number of iterations required to estimate a particular quantile, $\hat{\theta}_q$, with a cumulative probability q within a desired accuracy r with a probability s . That is, the Raftery and Lewis method finds the minimum number of iterations needed to satisfy $(\hat{P}_q \in (q - r, q + r)) = s$, where $\hat{P}_q = P(\theta < \hat{\theta}_q)$. In the R package CODA, the default is $q = 0.025$, $r = 0.005$, and $s = 0.95$. This method provides the number of burn-in iterations (M), total number of iterations (N), and the lower bound of iterations ($N_{\min}$) based on the pre-specified q , r , and s . When the dependence factor defined as the ratio of $N + M$ and $N_{\min}$ is larger than 5, the chain is not long enough to converge.

MC Error–SD Ratio

The three methods described above are based on the sampled values from every iteration commonly called “coda” file. A coda file can be produced easily in OpenBUGS, although it can be laborious to save coda files for all model parameters as the models become more complex. For this reason, several researchers (e.g., Lesaffre & Lawson, 2012; Lunn et al., 2013; Woodward, 2011) provide a more practical guideline to monitor convergence of a Markov chain based on summary statistics of MCMC estimation. These are the ratio of the MC error with the posterior standard deviation. As a typical rule of thumb, an MC error smaller than 5% of the

standard deviation of the parameter from post burn-in iterations indicates convergence of the Markov chain.

Method

Simulation Design

A simulation study manipulating conditions for the degree of convergence was conducted to explore the relationship between Markov chain convergence and the estimation of model parameters in MixIRT models. To manipulate degree of convergence, three different levels of lengths of chains were simulated. Each chain length was paired with itself and the other two chain lengths to simulate both burn-in and post-burn-in chains. Consequently, there were nine conditions with varying lengths of burn-in and post burn-in iterations. For a short length, a total of 500 iterations was used to represent relatively few iterations as in Cowles and Carlin (1996). The number of iterations for a medium and long length were determined to consider the number of iterations commonly used in the previous research. In previous research (e.g., Cohen & Bolt, 2005; De la Torre & Hong, 2010; Huang, 2017; Jin & Wang, 2014; Sen et al., 2016), the number of iterations generally ranged from 10,000 to 30,000. Accordingly, 5,000 and 10,000 iterations were used for the medium and long length, respectively, so that a total number of iterations considered in the simulation study ranged from 1,000 (500 burn-in iterations and 500 post burn-in iterations) to 20,000 (10,000 burn-in iterations and 10,000 post burn-in iterations).

Response data were generated for three dichotomous MixIRT models: a MixRasch, a Mix2PL, and a Mix3PL. The sample size, test length, and the number of latent classes were fixed at 2,000 examinees, 40 items, and 2 latent classes, respectively. A large sample size (2,000 examinees) and long test length (40 items) were selected based on the previous research with MixIRT models (e.g., Jin & Wang, 2014; Sen et al., 2016) to avoid the confounding effects of degree of convergence and presence of nuisance factors. Jin and Wang (2014) used a sample size of 2,000, with 25 items (20 items were dichotomous and five items were polytomous) for both Mix3PL and a Mix3PL general partial credit models. Similarly, Sen et al. (2016) generated 2,000 examinees' responses to 28 dichotomous items and analyzed a Mix3PL model. De la Torre and Hong (2010) generated 1,000 examinees and 20 items for the higher-order IRT model. The mixing proportions of the two latent classes were equal (i.e., 50% and 50%), and 100 replications were generated for each model, for a total of 300 simulated data sets.

Generating item parameters for the Mix3PL model were modified from Li, Cohen, Kim, and Cho (2009). The first ten items were used as anchor items, meaning there was no difference in the generating item parameters between the two latent classes. For one latent class, Items 11 to 20 were assumed to be easy items, Items 21 to 30 were assumed to be of average difficulty, and Items 31 to 40 were assumed to be difficult items. For the other latent class, the three item groups (i.e., Items 11 to 20,

Figure 1. Patterns of generating item parameters for Mix3PL model.
Note. X axis indicates the items. 3PL = three-parameter logistic.

Items 21 to 30, and Items 31 to 40) were assumed as average, difficult, and easy items, respectively. The patterns of generated item parameters are shown in Figure 1.

The same item discrimination and difficulty parameters used for generating data for the Mix3PL model were used to generate response data for the Mix2PLmodel. Likewise, the same difficulty parameters were used to generate response data for the MixRasch model. Ability parameters were simulated following a standard normal distribution, $N(0, 1)$, for each latent class. This resulted in 100 data sets generated for each MixIRT model. A given set of response data was analyzed nine times with each of the different lengths of burn-in and post burn-in iterations. This resulted in conducting a total of 2,700 (100 replications $\times$ 3 MixIRT models $\times$ 9 lengths of iterations) calibrations.

Model Estimations

A single Markov chain was estimated for of the replications for each condition for each of the three MixIRT models using the MCMC algorithm implemented in the

OpenBugs software (Spiegelhalter, Thomas, Best, & Lunn, 2014). The OpenBugs codes are available from the first author. Based on the previous research (e.g., Li et al., 2009; Sen et al., 2016), the following priors were specified for the Mix3PL model: $\beta_{ig} \sim N(0, .5)$, $\alpha_{ig} \sim N(0, .5)I(0,)$, $c_{ig} \sim \text{Beta}(5, 17)$, $\theta_j \sim N(\mu_g, 1)$, $\mu_2 \sim N(0, 1)$, $g_j \sim \text{Bernoulli}(\pi_1, \pi_2)$, $(\pi_1, \pi_2) \sim \text{Dirichlet}(.5, .5)$. The code $I(0,)$ in the prior for the item discriminations (i.e., α_{ig}) is a restriction for drawing only positive values. For model identification, the mean ability for the first latent class was fixed to zero (i.e., $\mu_1 = 0$). The same priors and constraints were used for the relevant parameters of Mix2PL and MixRasch models. As described above, the numbers of burn-in and post burn-in iterations is the major feature of the simulation study. Therefore, each model was estimated repeatedly with nine different replications for the same response data.

Checking Convergence

As noted above, we considered only an MCMC algorithm with a single chain for this study. Thus, three methods for monitoring convergence of a single chain (i.e., Heidelberger and Welch, Raftery and Lewis, and Geweke) were used as implemented in the R package CODA. In addition, the ratio of the MC error with the posterior SD was examined. The percentages of model parameters that converged based on each of the four different convergence diagnostics, respectively, across 100 replications of each condition were used as the rates of convergence. For example, when the number of converged item discrimination parameter estimates based on the Heidelberger and Welch method was 7,700 under the Mix2PL model with 5,000 burn-in iterations and 10,000 post burn-in iterations, the rate of convergence was calculated as .96 ($= 7,700/8,000$) where the number of estimated item discrimination parameters was 8,000 ($= 40 \text{ items} \times 2 \text{ classes} \times 100 \text{ replications}$).

Recovery Analysis

To examine the effect of convergence on the estimation of the MixIRT model parameters, the root mean square error (RMSE) values were compared. RMSE was computed as follows:

$$\text{RMSE}(\hat{\beta}) = \sqrt{E(\hat{\beta} - \beta)^2} = \sqrt{\frac{\sum_{i=1}^I \sum_{r=1}^R (\hat{\beta}_{ir} - \beta_i)^2}{RI}}, \quad (4)$$

where $\hat{\beta}$ is the estimated parameter (i.e., the posterior mean of β drawn from the converged chain via MCMC) for the generating parameter β , and $E(\cdot)$ is the expected value. $\hat{\beta}_{ir}$ is the estimated parameter for item i ($i = 1, \dots, 40$) at the replication r ($r = 1, \dots, 100$), and β_i is the generating parameter for item i . In addition to estimating model parameters, the percentages of simulated cases that were correctly classified into the correct class (i.e., the generating class) were calculated.

Prior to calculating RMSEs and the percentages of correct membership cases, label switching was checked by comparing the generating parameters and estimated parameters across replications. In a MixIRT model, the meaning of each class may differ because classes are latent, not manifest. To be specific, it is possible that the characteristics of Class 1 in the first replication might be observed as the characteristics of Class 2 in a second replication. When the patterns of estimated parameters were opposite to those of the generating parameters, the latent classes were relabeled. After correcting label switching, the estimated parameters should be placed on the same scale as the generating parameters, because the estimated parameters are on their own scale. Thus, the estimated parameters were transformed to a common metric using the mean and sigma method.

The following are expectations for this study. First, the convergence rate would increase as the number of iterations increased. Second, the degree of improvement in the convergence rate would be different depending on the methods for checking convergence. Specifically, the convergence rate by the Raftery and Lewis method would be lower than by the Geweke method or the Heidelberger and Welch method as observed in the previous research (e.g., Cowles & Carlin, 1996; Sinharay, 2004). Third, under the same number of iterations, the convergence rate for a more complex model (e.g., Mix3PL model) would be lower than for a less complex model (e.g., MixRasch model). Finally, the low convergence rate would result in a large RMSE in item parameter estimates, which indicates a bad parameter recovery.

Results

Rate of Convergence

The rates of convergence based on the four different convergence diagnostics (i.e., Heidelberger and Welch, Geweke, Raftery and Lewis, and MC error–SD ratio) are presented in Table 1 and Figure 2. The simulation conditions are summarized in Table 1. The label for a simulation condition is a combination of the numbers of burn-in and post burn-in iterations. For instance, 500-500 indicates the condition with 500 burn-in iterations and 500 post burn-in iterations. Overall, the rates of convergence appear to differ between methods as expected. The Heidelberger and Welch method tended to be more lenient than the other methods with rates of convergence ranging from .673 to .997. These rates of convergence were the highest among the rates by other three methods, except for item difficulty for the Mix2PL model. For MixRasch and Mix2PL models, the rates of convergence appeared to increase as the numbers of burn-in and post burn-in iterations increased. In contrast, the rates of convergence decreased as the numbers of iterations increased for item difficulty for the Mix3PL model. However, this pattern was not evident for item discrimination and guessing parameter estimates for the Mix3PL model, when the number of burn-in iterations was larger than 5,000.

The MC error–SD ratio was the most conservative and insensitive. The rates of convergence based on this method were the lowest for all conditions and models,

Table 1. Rates of Convergence (Over 100 Replications for Each Condition)

		Condition									
Model	Parameter	Method	500-500	500-5000	500-10000	5000-500	5000-5000	5000-10000	10000-500	10000-5000	10000-10000
MixRasch	b	HW	.859	.967	.985	.916	.960	.983	.856	.961	.997
		G	.568	.810	.843	.559	.768	.794	.525	.772	.835
		RL	NA	.533	.501	NA	.533	.501	NA	.556	.501
Mix2PL	b	MC	.008	.500	.500	.012	.500	.500	.014	.500	.500
		HW	.837	.924	.969	.845	.911	.938	.855	.927	.948
		G	.535	.744	.774	.578	.735	.772	.547	.705	.780
		RL	NA	.584	.496	NA	.574	.494	NA	.572	.495
		MC	.025	.997	1.000	.027	.996	1.000	.026	.997	1.000
Mix3PL	b	HW	.982	.984	.985	.982	.988	.990	.980	.989	.988
		G	.857	.906	.921	.862	.904	.908	.862	.909	.904
		RL	NA	.975	.964	NA	.977	.963	NA	.978	.964
		MC	.002	.473	.500	.003	.476	.500	.001	.473	.500
		HW	.804	.724	.673	.901	.822	.740	.914	.822	.761
		G	.329	.267	.316	.576	.544	.463	.602	.639	.577
		RL	NA	.007	.003	NA	.008	.001	NA	.005	.000
		MC	.000	.000	.002	.000	.000	.006	.000	.000	.005
		HW	.844	.813	.809	.922	.935	.946	.929	.958	.962
		G	.378	.274	.317	.592	.681	.648	.623	.791	.819
	c	RL	NA	.014	.004	NA	.014	.003	NA	.010	.002
		MC	.000	.000	.000	.000	.000	.004	.000	.000	.004
		HW	.867	.845	.880	.933	.954	.962	.941	.973	.975
		G	.424	.291	.339	.609	.739	.724	.615	.817	.842
		RL	NA	.009	.001	NA	.003	.000	NA	.001	.000
		MC	.000	.000	.000	.000	.000	.004	.000	.000	.003

Note. When convergent rate is smaller than .5, the values are in boldface. 2PL = two-parameter logistic; 3PL = three-parameter logistic; a = item discrimination; b = item difficulty; c = guessing parameter; HW = Heidelberger and Welch; G = Geweke; RL = Raftery and Lewis; MC = MC error-SD (standard deviation) ratio; NA = not available.

Figure 2. Patterns of rates of convergence and RMSE for each model parameter.
Note. X axis indicates the simulation conditions combined by the number of burn-in and post burn-in iterations. Model parameters *a*, *b*, and *c* represent item discrimination, item difficulty, and guessing parameter, respectively. HW, G, RL, and MC represent Heidelberg and Welch, Geweke, Raftery and Lewis, and MC error–SD ratio, respectively. RMSE = root mean square error; SD = standard deviation.

except for item difficulty for the Mix2PL model. For post burn-in iterations of more than 500, the rates of convergence for item difficulty for the Mix2PL were almost 1 regardless of the number of burn-in iterations. Furthermore, the rate of convergence under the 500-5000 condition was .997, while those under the 5000-500 condition was .027.

The rates of convergence for Geweke’s method varied, ranging from .267 to .921. The patterns of rates were similar to those with Heidelberg and Welch’s method, although the rates in absolute terms were generally lower than those with the Heidelberg and Welch method. The rates for the Raftery and Lewis method were varied depending on the model parameters, rather than the number of iterations. The results for the Raftery and Lewis method were not available when the number of post burn-in iterations was 500. The rates for MixRasch and Mix3PL models for the Raftery and Lewis’ method were close to those with the MC error–SD ratio. For Mix2PL, however, the rates for item difficulty were almost .500, while those for item discrimination were near 1 regardless of the length of burn-in and post burn-in.

Table 2. RMSE and Percentage of Correct Detection of True Membership (Over 100 Replications for Each Condition)

Condition	RMSE						Correct detection of class membership (%)		
	MixRasch	Mix2PL		Mix3PL			MixRasch	Mix2PL	Mix3PL
	<i>b</i>	<i>b</i>	<i>a</i>	<i>b</i>	<i>a</i>	<i>c</i>			
500-500	7.530e-2	9.826e-2	12.121e-2	1.261	0.473	0.094	99.214	99.417	82.925
500-5000	7.522e-2	9.775e-2	12.076e-2	0.648	0.351	0.090	99.208	99.420	83.236
500-10000	7.521e-2	9.773e-2	12.077e-2	0.435	0.275	0.088	99.211	99.418	83.644
5000-500	7.531e-2	9.842e-2	12.105e-2	0.458	0.312	0.077	99.225	99.417	83.582
5000-5000	7.521e-2	9.784e-2	12.082e-2	0.328	0.244	0.076	99.214	99.417	83.758
5000-10000	7.521e-2	9.779e-2	12.078e-2	0.259	0.218	0.076	99.214	99.418	83.824
10000-500	7.533e-2	9.782e-2	12.109e-2	0.281	0.239	0.076	99.208	99.407	83.764
10000-5000	7.521e-2	9.779e-2	12.078e-2	0.220	0.207	0.075	99.210	99.416	83.829
10000-10000	7.520e-2	9.780e-2	12.076e-2	0.201	0.199	0.075	99.211	99.418	83.838

Note. RMSE = root mean square error; 2PL = two-parameter logistic; 3PL = three-parameter logistic; *a* = item discrimination; *b* = item difficulty; *c* = guessing parameter.

Recovery Analysis

To explore the impact of nonconverged Markov chain on the estimation of model parameters, the recovery of item parameters was assessed by computing RMSEs. As summarized in Table 2, RMSEs of the item difficulty for all simulation conditions were almost the same in spite of the evident differences in the rates of convergence based on the four methods. These values were about 0.075 and 0.098 for the MixRasch and the Mix2PL models, respectively. Similarly, RMSEs for item discrimination were equal (RMSE = 0.121) for two decimal places, regardless of the number of iterations. Unlike MixRasch and Mix2PL models, RMSEs of item parameters for Mix3PL differed depending on the number of burn-in and post burn-in iterations. RMSEs ranged from 0.201 to 1.261, from 0.199 to 0.473, and from 0.075 to 0.094 for the item difficulty, item discrimination, and guessing parameter, respectively. The differences in RMSEs for the item difficulty were the largest, and those for guessing parameter were the smallest. RMSEs of item difficulty were noticeably decreased as the number of burn-in and post burn-in iterations increased. RMSEs of item discrimination and guessing parameter also decreased as the length of burn-in and post burn-in iterations increased, although RMSEs were less variable than those of item difficulty. In addition, even though the Mix3PL produced noticeable decreases with an increase in burn-in and post-burn-in iterations, the smallest RMSE of each item parameter was still larger than those for the MixRasch and Mix2PL models. For example, the smallest RMSE of item discrimination for the Mix3PL model was 0.199 under the 10000-10000 condition, but the RMSE of item discrimination for the Mix2PL model was 0.121 disregarding the number of iterations.

In addition to RMSEs of the item parameters, the percentages of correct detections for class membership are summarized in Table 2. For MixRasch and Mix2PL models, the percentages of correct detections for class membership were close to 100%, ranging from 99.2% to 99.4%, regardless of the number of iterations. For the Mix3PL model, the percentages were smaller than those for the MixRasch and Mix2PL models, ranging from 82.9% to 83.8%. Although the percentages of correct detections for class membership slightly increased as the number of iterations increased, this difference was much smaller than the improvement observed in the recovery of item parameters. To sum up, the Mix3PL was much less likely to recover any of the item parameters or to detect the true class membership compared with the other two models.

Discussion

The MCMC algorithm is a popular method for estimating model parameters for MixIRT models because of its utility with complex models. Furthermore, MCMC estimation may be as accurate or at least almost identical to maximum likelihood estimation (e.g., Kieftenbeld & Natesan, 2012; J.-S. Kim & Bolt, 2007; S.-H. Kim, 2001). One of the major issues with an MCMC algorithm has to do with convergence of a Markov chain. It is obvious that MCMC estimation of complex models typically requires very large numbers of iterations to obtain convergence and provide accurate estimates. In terms of efficiency, however, some information about the effect of convergence of a Markov chain on the estimation of model parameters would be useful to determine the number of iterations needed. For this purpose, the present simulation study compared the recovery statistics and the rates of convergence for a Markov chain based on the four different methods.

The different methods for monitoring convergence of a Markov chain produced different patterns of rates of convergence and the tendency of convergence rate by each method was mostly consistent with the findings of previous research (e.g., Cowles & Carlin, 1996; Sinharay, 2004; Susanto et al., 2018). The rates of convergence assessed by the Heidelberger and Welch method were generally stable albeit more lenient than the other methods. The rates of convergence monitored by the Geweke method were most sensitive to the number of iterations in the chain. The Geweke method yielded constantly lower rates of convergence than did the Heidelberger and Welch method, although the pattern of rates was similar to the Heidelberger and Welch method.

The most conservative and insensitive method was the Raftery and Lewis method and the MC error–SD ratio. The rates of convergence by these methods were tended to be lower than other methods. In particular, the rates of convergence were almost zero for the Mix3PL regardless of type of model parameter and number of iterations. There are some studies that showed a similar concern on the Raftery and Lewis method. As in this study, Susanto et al. (2018) also showed that the convergence test failed regardless of the type of parameters and the number of iterations, when the

three parameters of the finite model were estimated. Jackman (2009) noted in chapter 6 (p. 234) that the conservative results of the Raftery and Lewis method could occur because the indicator to test convergence of a Markov chain is estimated based on a sequence of binary indicators, and some thinning would be required for approximation of this sequence to the Markov chain. Furthermore, Brooks and Roberts (1998) pointed out that this method provides information about the convergence of a chain depending on a prespecified q value, not of a full chain. Therefore, the estimated minimum number of burn-in iterations might be underestimated.

As expected, the convergence results also differed depending on the complexity of models and the number of iterations. A more complex model yielded lower convergence rates as monitored by the Heidelberg and Welch and Geweke methods. Additionally, the rate of convergence assessed by the Heidelberg and Welch and Geweke methods with an increase in the number of post burn-in iterations. In particular, the degree of convergence for these methods appeared to be less affected by the number of burn-in iterations than by the number of post burn-in iterations. However, for the item difficulty of the Mix3PL model, the rate of convergence assessed by the Heidelberg and Welch and Geweke methods tended to decrease as the number of post burn-in iterations increased and to increase with an increase in burn-in iterations. These unexpected results might be caused by the slow convergence of the Mix3PL model than other models. The Heidelberg and Welch and Geweke methods basically assess a stationarity of a Markov chain based on the comparison of two parts of the chain. If a Markov chain converged slowly and the burn-in period was short, the non-converged part of the chain could be contained and used to assess convergence.

The results of a recovery analysis for MixRasch and Mix2PL models showed almost identical RMSEs for item parameters regardless of the difference rates of convergence resulted from the four methods. Additionally, the recovery statistics were less sensitive to the number of burn-in and post burn-in iterations than to the complexity of models. RMSEs increased as the complexity of the model increased, and this was consistent with previous research (e.g., Hulin, Lissak, & Drasgow, 1982; Sen et al., 2016; Li et al., 2009). For all simulation conditions, RMSEs reported in Table 2 for item difficulty were 0.075 (MixRasch) and 0.098 (Mix2PL), respectively, and that for item discrimination of the Mix2PL model was 0.121. These RMSEs are consistent with estimates of item parameters reported in previous research (e.g., Hulin et al., 1982; Stone, 1992). It appears that the shortest chain in this study, with 500 burn-in and 500 post burn-in iterations, may be sufficiently long enough to obtain precise item parameter estimates for MixRasch and Mix2PL models.

For the Mix3PL model, smaller numbers of iterations in the chain reduced the accuracy of item parameter estimates. RMSEs varied depending on the simulation conditions. The largest differences occurred in estimation of item difficulty. As expected, the largest RMSE was observed for the shortest chain length (i.e., the 500-500 condition); RMSEs decreased as the chain length increased. Furthermore, for the 500-5000 and 5000-500 conditions, the longer burn-in iterations resulted in the smaller RMSE. As an example, RMSE for estimated item difficulty for the Mix3PL

model was 0.648 under the 500-5000 condition, but was 0.458 under the 5000-500 condition, even when the total chain length for both conditions was the same, that is, 5,500 iterations. This pattern was similar to the pattern observed for item discrimination and guessing parameters generated by the Mix3PL model.

The findings of this study suggest some guidelines for monitoring convergence of a Markov chain. First, the results of convergence would differ depending on the convergence diagnostics, when MixIRT model parameters are estimated. This finding was consistent with the previous research that showed the similar results for other types of statistical models (e.g., finite latent class models, IRT models). The MC error–SD ratio was the most convenient approach to diagnosing convergence of a Markov chain because it is only based on the MC error and the SD from the posterior, which are easy to calculate. Nevertheless, the MC error–SD ratio was very conservative and insensitive. The Raftery and Lewis method also tended to yield relatively conservative results. Consequently, these two methods are not recommended as a dominant method to assess convergence of a Markov chain.

Second, as expected, the convergence rates by the Heidelberger and Welch method and the Geweke method would be more sensitive than those by the Raftery and Lewis method and the MC error–SD ratio to the number of iterations. However, the Heidelberger and Welch method tended to be too lenient. Therefore, when the Heidelberger and Welch method is used to monitor convergence of a Markov chain, the results should be interpreted conservatively. Based on the results of this study, the convergence rate by this method should be larger than 0.85 to ensure an acceptably small recovery statistics.

Furthermore, increasing the number of burn-in iterations appears to be more efficient than increasing the number of post burn-in iterations. Particularly, the Heidelberger and Welch method and the Geweke method could be affected by the number of burn-in iterations because of their basic statistical concepts. Thus, when a statistically complex model has slow convergence, a longer burn-in period might be effective to accelerate convergence of the Markov chain.

Although the current study demonstrated the different performance of the popular convergence diagnostics and provided some guideline based on the results of simulation study, there are some limitations that can have implications to generalize the findings. First, this study focused on methods for monitoring a single chain, but it is also common to use multiple chains. Research with multiple chains would be useful for understanding the relation between the estimates of model parameters and methods for monitoring convergence. Also, the convergence of a Markov chain may affect the model selection of latent class models such as a latent growth mixture model.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Yoonsun Jang <https://orcid.org/0000-0002-4994-848X>

References

- Asparouhov, T., & Muthén, B. (2010). *Bayesian analysis using Mplus: Technical implementation*. Los Angeles, LA: Muthén & Muthén.
- Bacci, S., Bartolucci, F., Grilli, L., & Rampichini, C. (2017). Evaluation of student performance through a multidimensional finite mixture IRT model. *Multivariate Behavioral Research, 52*, 732-746.
- Bolt, D. M., Cohen, A. S., & Wollack, J. A. (2002). Item parameter estimation under conditions of test speededness: Application of a mixture Rasch model with ordinal constraints. *Journal of Educational Measurement, 39*, 331-348.
- Brooks, S. P., & Roberts, G. O. (1998). Convergence assessment techniques for Markov chain Monte Carlo. *Statistics and Computing, 8*, 319-335.
- Cohen, A. S., & Bolt, D. M. (2005). A mixture model analysis of differential item functioning. *Journal of Educational Measurement, 42*, 133-148.
- Cowles, M. K., & Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association, 91*, 883-904.
- De la Torre, J., & Hong, Y. (2010). Parameter estimation with small sample size a higher-order IRT model approach. *Applied Psychological Measurement, 34*, 267-288.
- Finch, W. H., & French, B. F. (2012). Parameter estimation with mixture item response theory models: A Monte Carlo comparison of maximum likelihood and Bayesian methods. *Journal of Modern Applied Statistical Methods, 11*, 167-178.
- Gamerman, D., & Lopes, H. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference* (2nd ed.). Boca Raton, FL: Chapman & Hall/CRC Press.
- Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science, 7*, 457-472.
- Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In J. M. Bernardo, J. O. Berger, A. P. Dawid, & A. F. M. Smith (Eds.), (Eds.), *Bayesian statistics 4* (pp. 169-194). Oxford, England: Oxford University Press.
- Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science, 7*, 473-511.
- Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1998). *Markov chain Monte Carlo in practice*. Boca Raton, FL: Chapman & Hall/CRC Press.
- Heidelberger, P., & Welch, P. (1983). Simulation run length control in the presence of an initial transient. *Operations Research, 31*, 1109-1144.
- Huang, H.-Y. (2016). Mixture random-effect IRT models for controlling extreme response style on rating scales. *Frontiers in Psychology, 7*, 1706. doi:10.3389/fpsyg.2016.01706.
- Huang, H.-Y. (2017). Mixture IRT model with higher-order structure for latent traits. *Educational and Psychological Measurement, 77*, 275-304.

- Hulin, C. L., Lissak, R. I., & Drasgow, F. (1982). Recovery of two- and three-parameter logistic item characteristic curves: A Monte Carlo study. *Applied Psychological Measurement*, 6, 249-260.
- Jackman, S. (2009). *Bayesian analysis for the social sciences*. Chichester, England: Wiley.
- Jiao, H., Lissits, B., Macready, G., Wang, S., & Liang, S. (2011). Exploring levels of performance using the mixture Rasch model for standard setting. *Psychological Test and Assessment*, 53, 499-522.
- Jiao, H., Macready, G., Liu, J., & Cho, Y. (2012). A mixture Rasch model-based computerized adaptive test for latent class identification. *Applied Psychological Measurement*, 36, 469-493.
- Jin, K.-Y., Chen, H.-F., & Wang, W.-C. (2017). Mixture item response models for inattentive responding behavior. *Organizational Research Methods*, 21, 197-225.
- Jin, K.-Y., & Wang, W.-C. (2014). Item response theory model for performance decline during testing. *Journal of Educational Measurement*, 51, 178-200.
- Kadengye, D. T., Ceulemans, E., & Van den Noortgate, W. (2013). A generalized longitudinal mixture IRT model for measuring differential growth in learning environments. *Behavioral Research*, 46, 823-840.
- Kieftenbeld, V., & Natesan, P. (2012). Recovery of graded response model parameters: A comparison of marginal maximum likelihood and Markov chain Monte Carlo estimation. *Applied Psychological Measurement*, 36, 399-419.
- Kim, J.-S., & Bolt, D. M. (2007). Estimating item response theory models using Markov chain Monte Carlo methods. *Educational Measurement: Issues and Practice*, 26(4), 38-51.
- Kim, S.-H. (2001). An evaluation of a Markov chain Monte Carlo method for the Rasch model. *Applied Psychological Measurement*, 25, 163-176.
- Lesaffre, E., & Lawson, A. B. (2012). *Bayesian biostatistics*. Chichester, England: Wiley.
- Li, F., Cohen, A. S., Kim, S.-H., & Cho, S.-J. (2009). Model selection methods for mixture dichotomous IRT models. *Applied Psychological Measurement*, 33, 353-373.
- List, M. K., Robitzsch, A., Lüdtke, O., Köller, O., & Nagy, G. (2017). Performance decline in low-stakes educational assessments: Different mixture modeling approaches. *Large-scale Assessments in Education*, 5, Article 15.
- Lunn, D., Jackson, C., Best, N., Thomas, A., & Spiegelhalter, D. (2013). *The BUGS book: A practical introduction to Bayesian analysis*. Boca Raton, FL: CRC Press.
- Mislevy, R. J., & Verhelst, N. (1990). Modeling item responses when different subjects employ different solution strategies. *Psychometrika*, 55, 195-215.
- Mittelhaeuser, M.-A., Béguin, A. A., & Sijtsma, K. (2015). The effect of differential motivation of IRT linking. *Journal of Educational Measurement*, 52, 339-358.
- Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnostics and output analysis for MCMC. *R News*, 6, 7-11.
- Raftery, A. E., & Lewis, S. M. (1992a). Comment: One long run with diagnostics: Implement strategies for Markov chain Monte Carlo. *Statistical Science*, 7, 493-497.
- Raftery, A. E., & Lewis, S. M. (1992b). How many iterations in the Gibbs sampler. In J. M. Bernardo, J. O. Berger, A. P. Dawid, & A. F. M. Smith (Eds.), *Bayesian statistics 4* (pp. 769-773). Oxford, England: Oxford University Press.
- Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement*, 14, 271-282.
- Sahlin, K. (2011). *Estimating convergence of Markov chain Monte Carlo simulations* (Unpublished master's thesis). Stockholm University, Stockholm, Sweden.

- Samuelsens, K. (2005). *Examining different item functioning from a latent class perspective* (Unpublished doctoral dissertation). University of Maryland, College Park.
- Sen, S., Cohen, A. S., & Kim, S.-H. (2016). The impact of non-normality on extraction of spurious latent classes in mixture IRT models. *Applied Psychological Measurement, 40*, 98-113.
- Sinharay, S. (2004). Experiences with Markov chain Monte Carlo convergence assessment in two psychometric examples. *Journal of Educational and Behavioral Statistics, 29*, 461-488.
- Smith, B. J. (2007). boa: An R package for MCMC output convergence assessment and posterior inference. *Journal of Statistical Software, 21*(i11), 1-37.
- Spiegelhalter, D. J., Thomas, A., Best, N. G., & Lunn, D. (2014). *OpenBUGS user manual, Version 3.2.3*. London, England: MRC Biostatistics Unit, Institute of Public Health and Department of Epidemiology and Public Health, Imperial College School of Medicine.
- Stone, C. A. (1992). Recovery of marginal maximum likelihood estimates in the two-parameter logistic response model: An evaluation of MULTILOG. *Applied Psychological Measurement, 16*, 1-16.
- Susanto, I., Iriawan N., Kuswanto, H., Suhartono, Fithriasari, K., Ulama, B. S. S., . . . Pravitarsari, A. A. (2018). On the Markov chain Monte Carlo convergence diagnostic of Bayesian finite mixture model for income distribution. *Journal of Physics: Conference Series, 1090*, Article 012014. doi:10.1088/1742-6596/1090/1/012014
- Woodward, P. (2011). *Bayesian analysis made simple: An Excel GUI for WinBUGS*. Boca Raton, FL: CRC Press.