2016

Some Issues In Markov Chain Monte Carlo Estimation For Item Response Theory

Han Kil Lee
University of South Carolina

Some Issues In Markov Chain Monte Carlo Estimation For Item Response Theory

by

Han Kil Lee

Bachelor of Science
University of Illinois 2009

Master of Science
Michigan State University 2011

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Statistics

College of Arts and Sciences

University of South Carolina

2016

Accepted by:

Brian Habing, Major Professor

John Grego, Committee Member

Tim Hanson, Committee Member

Christine DiStefano, Committee Member

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Acknowledgments

Even though I walk through the valley of the shadow of death,

I will fear no evil, for you are with me; your rod and your staff, they comfort me.

Pslam 23: 4

This dissertation is the result of a challenging journey, upon which many people have contributed and given their support.

First and foremost, I thank my God, my good Father, for lending this humble servant wisdom and peace through all the difficulties. I have experienced your guidance day by day. Thank you, Lord.

I would like to express my deepest gratitude to my mentor and friend, Dr. Brian Habing, for his excellent guidance, genuine caring and concern, and his unwavering faith in me. I am truly grateful to him for instilling in me the qualities of being a good researcher with integrity.

A good support system is important to surviving and staying sane in grad school. I thank my loving parents who always had faith in me. I thank my aunt and uncle who have provided support and wisdom of life from the first day I set foot in the United States. I thank my new family-in-law who welcomed me into the household warmheartedly and sent me the most sage wife.

Finally, special recognition and appreciation goes to my loving wife, forever life partner, mother of our new expected child, and lifelong comfort.

내가 사망의 음침한 골짜기로 다닐찌라도 해를 두려워하지 않을 것은
주께서 나와 함께 하심이라. 주의 지팡이와 막대기가 나를 안위하시나이다.
시편 23편 4절

먼저 능력없는 종에게 마음의 평안을 주시고 능력에 넘치는 지혜를 주셔서 매 순간을 인도해주신 하나님아버지께 감사 드립니다.

이 논문에 가장 큰 도움을 주었고 매 순간마다 친구처럼, 때로는 선생님으로, 또 때로는 부모의 마음으로 도와준 Dr. Habing에게 감사드립니다. Dr. Habing 지도없이는 이 논문의 시작도 끝도 할수 없었기에 더더욱 감사드립니다.

부족한 아들을 항상 믿어주시고 지지해주신 부모님과 미국에 온 첫날부터 지금까지 삶의 지혜를 보여주신 고모님 부부에게 감사 드립니다.

부족한 사위에게 세상에서 가장 현명한 아내를 보내주셔서 지치지 않고 포기하지 않게 해주신 장인어른, 장모님에게도 감사를 드립니다.

마지막으로 영원한 동반자이며 새로이 생길 가정에 어머니이고 평생 마음의 위로가 될 아내 정한나에게 이 모든 공을 돌립니다.

Abstract

Both the marginalized Bayesian modal estimation (MBME) and Metropolis-Hasting within Gibbs (MH/Gibbs) are the popular estimation methods for Item Response Theory (IRT). However, predictions from MBME and MH/Gibbs are not directly comparable because of two problems. First, the examinees with the same response pattern do not produce the same ability estimates from MH/Gibbs while MBME provides identical estimates. This problem can be handled by updating each response pattern instead of updating each examinee. Second, standard errors from MBME are smaller than standard error estimates from MH/Gibbs. This pattern occurs because of two speculated reasons; correlation between item parameter estimations and ability estimations even after thinning and the absence of a procedure which centers ability for every iteration which existed in MBME. In-chain centered MCMC mitigates the correlation by using a linking procedure to adjust candidates of item parameters at the middle of every iteration and can provide item parameter estimations which are closer the item parameter estimations from MBME. Two methods of in-chain centered MCMC are introduced - one using extra sets of abilities simulated in the proposal step, and one using the abilities generated previously. The simulation results demonstrate that both in-chain centered MCMC using extra simulated sets of abilities and in-chain centered MCMC with previous abilities can generate item parameter estimations with comparable standard errors without losing the accuracy of the point estimates. Therefore, in-chained MCMC can generate whole posterior distributions for item parameters which are comparable to the point estimates and standard errors from MBME.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Item response theory (IRT; e.g. Lord, 1980) is one of the most popular methods in Education and Psychology for measuring latent traits of examinees (e.g. ability, attitude, etc.) and item properties (e.g. difficulty, probability of answering correctly without related knowledge). IRT models are classified based on a variety of factors such as types of scoring used, dimensionality of latent traits, number of item parameters, etc. For example, a test that measures mathematical knowledge could be considered as a multidimensional IRT model (MIRT; Reckase, 1985) when the latent traits for this model are algebra, geometry, and trigonometry. However, it will be considered to be a unidimensional model when the only latent trait for this model is mathematical knowledge. Based on the assumptions, the same test can be analyzed with different IRT models.

Each IRT model has a unique item characteristic curve (ICC) which shows the probability of correctly answering the item depending on ability level, as a function of the given item parameters. For instance, the ICC of the Rasch (or 1PL; Rasch, 1960) model can be written as:

$$P\left(U_{ij}=1|\theta_j\right)=\frac{1}{1+e^{-1.7a(\theta_j-b_i)}} \tag{1.1}$$

where U_{ij} is an indicator of correctness for examinee j on item i and θ_j is the ability level of examinee j. There are two types of item parameters in the Rasch model: discrimination (constant across all items) and difficulty (one parameter for each item). In some formulations of the model, the discrimination parameter is a fixed constant and the standard deviation of the examinee ability distribution is estimated. The 1.7

(to put it on a normal CDF scale) is also commonly left out. In equation (1.1), a is the item discrimination parameter and b_i is the item difficulty parameter for item i.

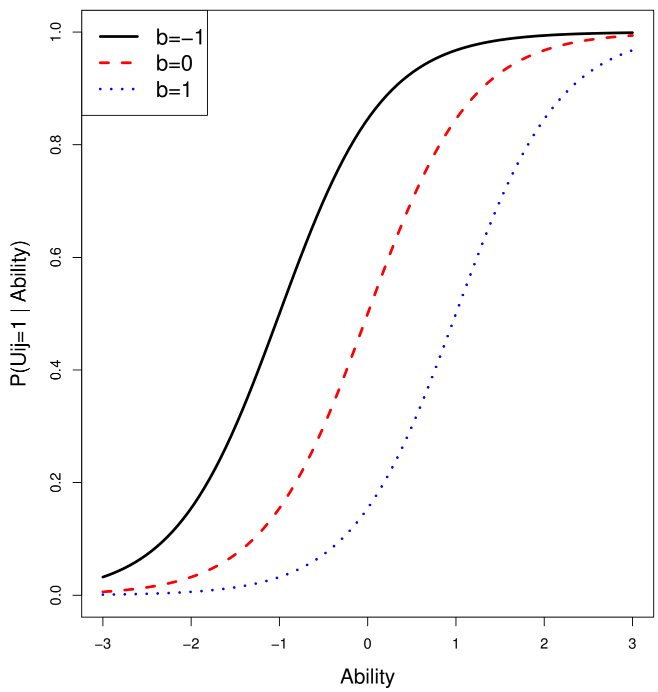

Figure 1.1 ICC for Rasch with $a = 1$ for all items and $b = -1$, $b = 0$, and $b = 1$

When each item has a different discrimination parameter, the model is called the two parameter logistic (2PL) model and the ICC is as follows:

$$P\left(U_{ij} = 1|\theta_j\right) = \frac{1}{1 + e^{-1.7a_i(\theta_j - b_i)}} \tag{1.2}$$

The Rasch model can be considered a special case of the 2PL model which has the same item discrimination parameter across all times. Difference in the item discrimination parameter causes different shapes of the ICC. Figure 1.2 shows different ICCs for different item discrimination parameters with the same item difficulty parameter ($b = 0$). Because the item difficulty parameters for all three ICC are same, all three ICC have the same inflection point.

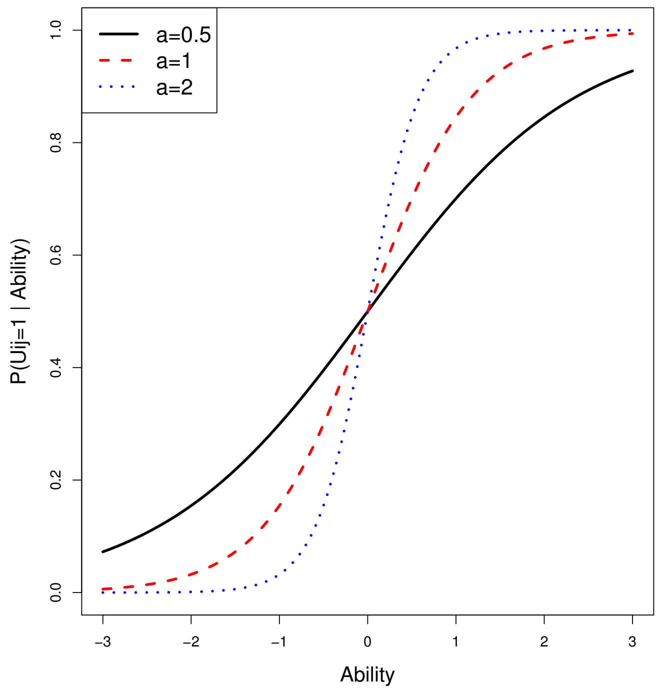

Figure 1.2 ICC for 2PL with $b = 0$ for all items and $a = 0.5$, $a = 1$, and $a = 2$

The 2PL model can be extended further by adding one more item parameter called the item guessing parameter. The item guessing parameter represents the probability of answering correctly without any related knowledge (e.g. $\theta \to -\infty$). For example, when the question is a multiple choice question with 5 choices, the probability to answer correctly for an examinee without knowledge will be 20%. Then, the item guessing parameter for this item will be 0.20 and ICC for this item will have a lower asymptote (Figure 1.3). The ICC for the 3PL model is:

$$P\left(U_{ij} = 1|\theta_j\right) = c_i + \frac{1 - c_i}{1 + e^{-1.7a_i(\theta_j - b_i)}} \tag{1.3}$$

where c_i is the guessing parameter for item i.

All 1PL (Rasch), 2PL, and 3PL models follow the Monotone Homogeneity Model (MHM; Mokken, 1971) which has three criteria:

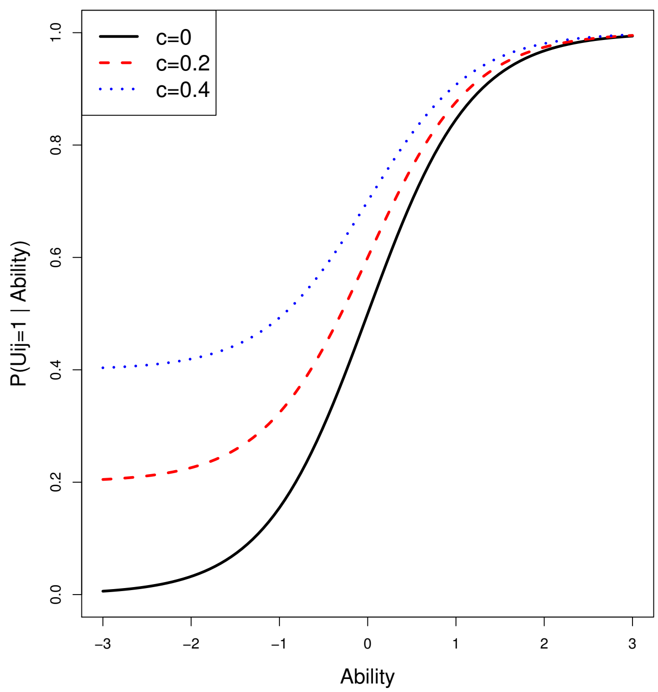

Figure 1.3 ICC for 3PL with $a = 1$, $b = 0$ for all items and $c = 0$, $c = 0.2$, and $c = 0.4$

1. Unidimensional latent trait : Only one latent trait is related with the probability to answer correctly.

2. Local Independence: The item responses are independent, given examinee ability.

3. Monotonicity: An examinee with more ability will have a higher probability of answering correctly.

With known item parameters, there are several common methods to estimate the ability of examinees; maximum likelihood estimation (MLE), maximum a posteriori estimation (MAP), and expected a posteriori estimation (EAP). The MLE maximizes the likelihood $L(\theta_j)$ where

$$L(\theta_j) = \prod_{i=1}^{N} P_i(\theta_j)^{U_{ij}} \left(1 - P_i(\theta_j)\right)^{1-U_{ij}} \tag{1.4}$$

Maximum likelihood is a traditionally popular method in many fields for computing point estimators, but there are some weaknesses to it employed with IRT models. For an examinee who answers all items correctly (or incorrectly), the MLE for the examinee equals positive (or negative) infinity as shown in Figure (1.4), which represents someone who has unlimited knowledge for a certain subject which is unrealistic.

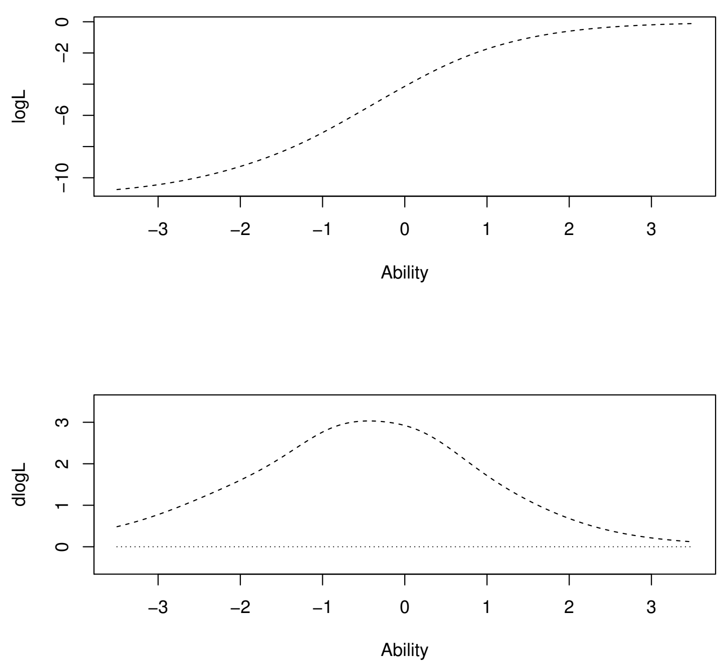

Figure 1.4 Log likelihood and derivative of the log likelihood for an examinee who answers all items correctly

In addition to the MLE, there are two Bayesian approaches, maximum a posteriori estimation (MAP) and expected a posteriori estimation (EAP), that are widely used in IRT to estimate the ability of examinees. Both Bayesian approaches apply a prior for the ability of examinee and the prior restricts the maximum and minimum estimates of ability.

$$L_{Bayes}\left(\theta_j\right) \propto \prod_{i=1}^{N} P_i\left(\theta_j\right)^{U_{ij}}\left(1-P_i\left(\theta_j\right)\right)^{1-U_{ij}}\phi\left(\theta_j\right) \tag{1.5}$$

Since the ability scale is arbitrary, it is commonly assumed to have mean zero and variance one, and imagined to be normal. Equation (1.5) is an example that applies the standard normal prior. Since most of the population of the standard normal distribution are between -3 and 3, the posterior distribution of the ability for an examinee who answers all items correctly will have a finite value, θ_j. To emphasize the connection with maximum likelihood, we use L_{Bayes} to represent the prior times the likelihood (which is proportional to the posterior).

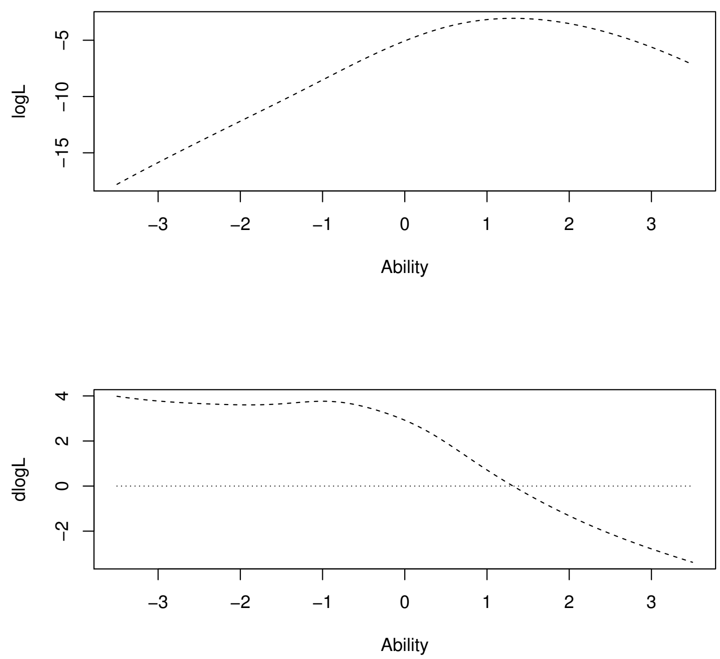

Figure 1.5 Likelihood and derivative of likelihood for an examinee who answers all items correctly after applying a standard normal prior

Figures (1.4) and (1.5) are log of the L_{Bayes} and derivative for an examinee with a perfect score, but MAP has a finite estimate. As explained above, it is very straightforward to estimate an examinee's ability with known items. It is not directly computable, however, for unknown item parameters because both item parameters and abilities should be estimated simultaneously.

As an early attempt, Birnbaum (1968) developed a joint maximum likelihood estimation (JMLE) procedure to estimate item parameters. However, the JMLE procedure had numerous difficulties (Baker and Kim, 2004). Presently, the marginal maximum likelihood estimation with expectation maximization algorithm (MML/EM) is considered to be a more general approach than JMLE. By adapting the EM algorithm to IRT models, the abilities are integrated out and only the estimation of item parameters needs to be computed. More details about MML/EM will be discussed in Chapter 2.

A more recent estimation method is Markov chain Monte Carlo (MCMC). The Metropolis-Hasting algorithm and Gibbs sampler are the most popular algorithms for MCMC, but Metropolis-Hasting within Gibbs Sampler (MH/Gibbs) is the common algorithm for IRT models because of the convenience of application. MCMC is especially needed when whole posterior distributions are required for each item and examinee. The reason that the MH algorithm and Gibbs sampler are not popular when used singly in IRT, and the strength of MH/Gibbs in IRT will be mentioned in Chapter 3.

Although both MML/EM and MH/Gibbs algorithm are widely used in IRT models, there are significant difference in the properties of their estimates. For example, ability estimates for examinees with exactly the same response should be the same regardless of the method of estimation. However, the standard MCMC implementation

Table 1.1 ability estimates for 2PL model with the same response patterns

	Response Pattern	MBME estimation	MCMC estimation
Examinee 11	11111111111111101001	1.740948	1.829128
Examinee 184	11111111111111101001	1.740948	1.794808
Examinee 248	11111111111111101001	1.740948	1.828383
Examinee 645	11111111111111101001	1.740948	1.876770

does not provide identical estimates for examinees with the same responses because of the randomness in generating candidates for each iteration. Also, the standard error for item parameters and exmainees' abilities from MCMC is greater than the standard error from MML/EM. This research focuses on modifying the MCMC estimation method to make comparable estimates to MML/EM for unidimensional dichotomous logistic models .

In Chapter 2, the MML/EM parameter estimation method will be discussed in detail including a well known IRT software package BILOG-MG (Zimowski et al., 2003). An introduction to Markov Chain Monte Carlo and different MCMC procedures in IRT will be given in Chapter 3. Chapter 4 discusses the proposed MCMC posterior estimation method, in-chain centered MCMC. The motivation for adjusting the classic MCMC approach and benefits of the new method will also be discussed in Chapter 4. To compare the performance of MML/EM, classic MCMC and in-chain centered MCMC, a simulation will be employed. Details about the simulation and results are presented in Chapter 5. Finally, a conclusion and future study will be disccussed in Chapter 6.

Chapter 2

MML/EM algorithm

Before the expectation and maximization algorithm (EM algorithm; Dempster et al., 1977), the most popular method to estimate item parameters was the joint maximum likelihood estimation (JMLE; Birnbaum, 1968) method. However, the JMLE method is known to have theoretical and practical problems which are discussed in many papers. Even before the JMLE was applied to IRT, Neyman and Scott (1948) indicated a potentially problematic situation in which estimates were inconsistent when structural parameters (e.g. ability of examinees) are estimated with incidental parameters (e.g. item parameters) simultaneously. After the JMLE was adopted, Wingersky (1983) indicated the necessity of employing an upper limit to the discrimination estimator to handle the divergence problem in the LOGIST program (Wingersky et al., 1982) which applies the JMLE procedure to estimated item parameters and examinees ability parameters. Recently, Baker and Kim (2004) noted that the discrimination estimates based on the JMLE procedure can be very large for certain data sets, implying higher ability estimates for some examinees who answer that question correctly. After successive iterations, both discrimination and ability estimates increase and eventually become infinity. Because of these flaws of JMLE, marginal maximum likelihood estimation (MMLE) with expectation and maximization algorithms is more widely used in the item parameters estimation.

The initial marginal maximum likelihood estimation (MMLE) was developed by Bock and Lieberman (1970). However, this initial MMLE approach to estimate item parameters was a computationally intensive process and only used for short tests

(Baker and Kim, 2004). To resolve the problem of the initial MMLE approach, Bock and Aitkin (1981) reformulated the MMLE and applied the EM algorithm which is computationally more feasible.

2.1 Initial Marginal Maximum Likelihood Estimation

Bock and Lieberman (1970) designed the initial marginal maximum likelihood solution for the item parameters from the marginal likelihood function as follows:

$$\begin{aligned} L &= \prod_{j=1}^{N} P\left(u_j\right) \\ &= \prod_{j=1}^{N} P\left(u_j|\xi,\tau\right) \\ &= \prod_{j=1}^{N} \int P\left(u_j|\theta_j,\xi\right) g\left(\theta_j|\tau\right) d\theta_j \end{aligned} \tag{2.1}$$

By taking the partial derivative, the solution is given as follows:

$$a_i : \left(1-c_i\right) \sum_{j=1}^{N} \int \left[u_{ij} - P_i\left(\theta_j\right)\right] W_{ij} \left(\theta_j - b_i\right) \left[P\left(\theta_j|u_j,\xi,\tau\right)\right] d\theta_j = 0 \tag{2.2}$$

$$b_i : -a_i \left(1-c_i\right) \sum_{j=1}^{N} \int \left[u_{ij} - P_i\left(\theta_j\right)\right] W_{ij} \left[P\left(\theta_j|u_j,\xi,\tau\right)\right] d\theta_j = 0 \tag{2.3}$$

$$c_i : \frac{1}{1-c_i} \sum_{j=1}^{N} \int \left[\frac{u_{ij} - P_i\left(\theta_j\right)}{P_i\left(\theta_j\right)}\right] \left[P\left(\theta_j|u_j,\xi,\tau\right)\right] d\theta_j = 0 \tag{2.4}$$

where

$$W_{ij} = \frac{P_i^*\left(\theta_j\right) Q_i^*\left(\theta_j\right)}{P_i\left(\theta_j\right) Q_i\left(\theta_j\right)}$$

for

$$P_i^*\left(\theta_j\right) = \frac{e^{a_i\left(\theta_j - b_i\right)}}{1+e^{a_i\left(\theta_j - b_i\right)}} \quad \text{and} \quad Q_i^*\left(\theta_j\right) = 1 - \frac{e^{a_i\left(\theta_j - b_i\right)}}{1+e^{a_i\left(\theta_j - b_i\right)}}$$

To simplify the iteration, a quadrature distribution is used instead. Then,

$$a_i : (1 - c_i) \sum_{j=1}^{N} \sum_{k=1}^{q} [u_{ij} - P_i(X_k)] W_{ik} (X_k - b_i) [P(X_k | u_j, \xi, \tau)] = 0 \tag{2.5}$$

$$b_i : -a_i (1 - c_i) \sum_{j=1}^{N} \sum_{k=1}^{q} [u_{ij} - P_i(X_k)] W_{ik} [P(X_k | u_j, \xi, \tau)] = 0 \tag{2.6}$$

$$c_i : \frac{1}{1 - c_i} \sum_{j=1}^{N} \sum_{k=1}^{q} \left[\frac{u_{ij} - P_i(X_k)}{P_i(X_k)} \right] [P(X_k | u_j, \xi, \tau)] = 0 \tag{2.7}$$

To solve 2.5-2.7, Bock and Lieberman (1970) used Fisher's method of scoring which required a $3n \times 3n$ inverse matrix which was computationally too intensive. Therefore, the initial MMLE approach was used for small number of items only.

2.2 Expectation and Maximization Algorithm

The EM algorithm is a well-known iterative method for computation of maximum-likelihood estimates composed of an expectation step and a maximization step. It is employed for maximum likelihood estimation for incomplete datasets (Dempster et al., 1977). In IRT, latent traits of examinees were treated as missing values (Bock and Aitkin, 1981), so the unobservable complete data set is (U, θ) and the joint probability density function of complete data is $f(U, \theta | \xi)$ where θ represents the ability of examinees and ξ is the item parameter. So, the E-step and M-step in the EM algorithm for pth iteration are as follows:

E-Step: Compute $E[log(U, \theta | \xi^p) | U, \xi^{p-1}]$

M-Step: Choose ξ^p which maximize $E[log(U, \theta | \xi^p) | U, \xi^{p-1}]$

To apply the EM algoritm, Bock and Aitkin (1981) reformulated the initial MML equations using two new notations; $\overline{f}_{ik}$ and $\overline{r}_{ik}$, and discretizing the ability distribution (Baker and Kim, 2004).

$$
\begin{aligned}
\overline{f}_{ik} &= \sum_{j=1}^{N} P\left(X_k | u_j, \xi, \tau\right) \\
&= \sum_{j=1}^{N} \left[\frac{\prod_{i=1}^{n} P_i\left(X_k\right)^{u_{ij}} Q_i\left(X_k\right)^{1-u_{ij}} A\left(X_k\right)}{\sum_{k=1}^{q} \prod_{i=1}^{n} P_i\left(X_k\right)^{u_{ij}} Q_i\left(X_k\right)^{1-u_{ij}} A\left(X_k\right)} \right]
\end{aligned} \tag{2.8}
$$

$$
\begin{aligned}
\overline{r}_{ik} &= \sum_{j=1}^{N} u_{ij} P\left(X_k | u_j, \xi, \tau\right) \\
&= \sum_{j=1}^{N} \left[\frac{\prod_{i=1}^{n} u_{ij} P_i\left(X_k\right)^{u_{ij}} Q_i\left(X_k\right)^{1-u_{ij}} A\left(X_k\right)}{\sum_{k=1}^{q} \prod_{i=1}^{n} P_i\left(X_k\right)^{u_{ij}} Q_i\left(X_k\right)^{1-u_{ij}} A\left(X_k\right)} \right]
\end{aligned} \tag{2.9}
$$

$\overline{f}_{ik}$ represents expected number of examine at X_k and $\overline{r}_{ik}$ represents expected number of correct responses at X_k. Then, 2.5-2.7 are modified to the following:

$$
a_i : \left(1 - c_i\right) \sum_{k=1}^{q} \left(X_k - b_i\right) \left[\overline{r}_{ik} - \overline{f}_{ik} P_i\left(X_k\right)\right] W_{ik} = 0 \tag{2.10}
$$

$$
b_i : -a_i \left(1 - c_i\right) \sum_{k=1}^{q} \left[\overline{r}_{ik} - \overline{f}_{ik} P_i\left(X_k\right)\right] W_{ik} = 0 \tag{2.11}
$$

$$
c_i : \frac{1}{1 - c_i} \sum_{k=1}^{q} \left[\frac{\overline{r}_{ik} - \overline{f}_{ik} P_i\left(X_k\right)}{P_i\left(X_k\right)} \right] = 0 \tag{2.12}
$$

The EM algorithm then becomes:

E-Step: Compute $\overline{f}_{ik}$ and $\overline{r}_{ik}$ using (2.8) and (2.9)

M-Step: Solve equations (2.10)-(2.12) using $\overline{f}_{ik}$ and $\overline{r}_{ik}$

Continue to process the E-step and M-step iteratively until all item parameters are converged.

2.3 BILOG-MG

When joint maximum likelihood estimation (JMLE) was widely used to estimate item parameters for IRT, LOGIST (Lord, 1980; Mislevy and Stocking, 1989; Wingersky et al., 1982) was the most popular software package (an estimation program based on JMLE). As MML/EM algorithm was becoming the more popular estimation method because of theoretical and practical problems of JMLE, the alternative software BILOG became a common estimation program which uses the MML/EM algorithm (Mislevy and Bock, 1983). Yen demonstrated this superiority of BILOG over LOGIST by comparing estimates of item parameters (1987). For this research, BILOG-MG version 3.0 (Zimowski et al., 2003) is used for the estimation of item parameters and examinees' ability levels.

Among many estimation programs such as MULTILOG, PARSCALE (Du Toit, 2003), and IRTPRO (Cai et al., 2011), BILOG is one of the most popular parameter estimation tool for unidimensional dichotomous IRT models (Rupp, 2003) because of its application of MML/EM in a Bayesian framework (Bock and Aitkin, 1981; Mislevy, 1986) and practical strengths of flexibility of reading multiple formats of raw data, well-written manuals, convenient R package for BILOG-MG, etc... (Mead et al., 2007).

BILOG-MG uses default priors for the ability and item parameters such that:

$$\theta \sim \text{Normal}\,(0, 1) \tag{2.13}$$

$$a \sim \text{Lognormal}\,(0, 0.5) \tag{2.14}$$

$$b \sim \text{Normal}\,(0, 2) \tag{2.15}$$

$$c \sim \text{Beta}\,(5, 17) \tag{2.16}$$

BILOG-MG computes the item parameters and the ability of examinee separately. By using the response matrix, U, BILOG estimated item parameters first depending on the desired model. Then, the ability of examinee is estimated by treating the estimated item parameters as fixed.

Chapter 3

Markov chain Monte Carlo estimation

To compute point estimators with standard errors for item parameters and examinees' ability parameters, the MML/EM algorithm is a suitable method. When the whole posterior distribution is desired, however, one needs more than the MML/EM algorithm. For instance, the three parameter logistic model (3PL; Birnbaum, 1968) has three item parameters for each item and ability parameters for each examinee which makes it infeasible to have a closed form solution for the joint posterior distribution. Instead of developing the closed form equations for the posterior distributions, Markov chain Monte Carlo (MCMC) can be a good alternative.

MCMC (Chib and Greenberg, 1995; Gelman et al., 1995; Smith and Roberts, 1993; Tierney, 1994) is one of the most popular methods to generate a marginal posterior distribution for a high dimensional distribution or a distribution in which it is infeasible to have the closed form solution. Although MCMC is a computationally intense and time consuming process, it is widely used in IRT as well as other fields for estimating posterior distributions (Albert, 1992; Béguin and Glas, 2001; Bradlow et al., 1999; Patz and Junker, 1999a,b; Sinharay, 2004) because of its ease of application to complex distributions.

The core objective of the MCMC is to generate the Markov chain $M_0, M_1, ...$ with $M_t = (\theta^t, \xi^t)$ where M_t will converge to stationary distribution, $\pi(\theta, \xi)$ for the examinee and item parameters, under suitable conditions (Tierney, 1994). In Bayesian analyses, converged stationary distributions are often of high dimension and the marginal posterior distribution is represented by the distribution of sampled values.

To generate the Markov chain, different transition kernels are defined for each algorithm. The transition kernel, $k\left(M_t, M_{t+1}\right)$, is the probability of accepting a new state, M_{t+1}, when the current state of the chain, M_t, is given.

$$k\left(M_t, M_{t+1}\right) = P\left[M_{t+1}|M_t\right] \tag{3.1}$$

In practice, there are two procedures are needed to generated samples for posterior distribution from MCMC. First procedure is burn-in which is elimination step of beginning of the chain to remove stable samples of chain. Second procedure is thinning which is elimination of interval step to generate independent samples. Samples from MCMC are highly dependent from previous sample. By keeping one sample out of certain amount of sample, dependency can be removed. Burn-in and thinning are necessary processes but this makes MCMC is highly computational procedure.

Three popular algorithms to generate Markov chains are: Metropolis-Hastings, Gibbs sampling, and Metropolis-Hastings in Gibbs. Strength and weakness of each algorithm will be discussed in this chapter.

3.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH; Metropolis et al., 1953; Hastings, 1970) algorithm generates a Markov chain using a certain probability called the acceptance probability which represents the probability of accepting the new candidate of samples from the proposal distribution. When the stationary distribution is π and the proposal distribution is $q\left(x, y\right)$, where x represents the current value and y represent a candidate value for the next state, the following steps represent the MH algorithm:

1. Define current value $x = M_t$

2. Generate a candidate value, y, from the proposal desnsity, $q\left(x, y\right)$

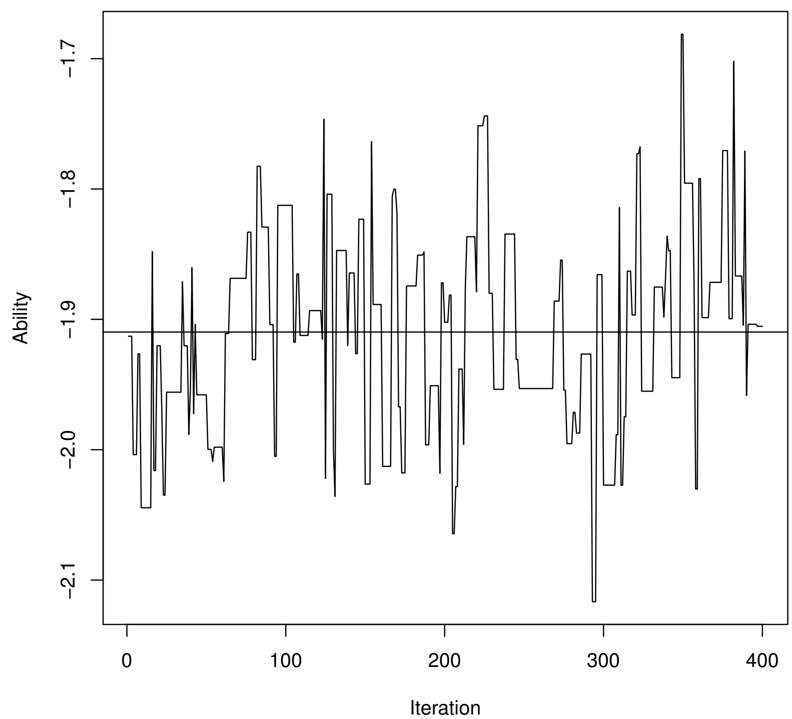

Figure 3.1 Iteration of difficulty parameter (MH with acceptance rate=.0072)

3. Accept y with probability $\alpha(x, y) = \min\left(\frac{\pi(y)q(x,y)}{\pi(x)q(y,x)}, 1\right)$. Then, $M_{t+1} = y$. Otherwise, $M_{t+1} = x = M_t$.

4. Repeat Steps 2 and 3 until t reaches the desired number of iterations.

The strength of the MH algorithm lies in the flexibility of the proposal distribution, $q(x, y)$. When a candidate, y, for the next chain is sampled, one can choose a distribution which is easy to sample from. Symmetric distributions are frequently chosen for q to simplify the acceptance rate, $\alpha(x, y)$:

$$\alpha(x, y) = \min\left(\frac{\pi(y)\, q(x, y)}{\pi(x)\, q(y, x)}, 1\right) = \min\left(\frac{\pi(y)}{\pi(x)}, 1\right) \tag{3.2}$$

but, $q(x, y)$ does not have to be symmetric and can be any distribution.

Although the choice of the proposed distribution is flexible, the proposed distribution, q, is selected very carefully because the proposal distribution affects how quickly

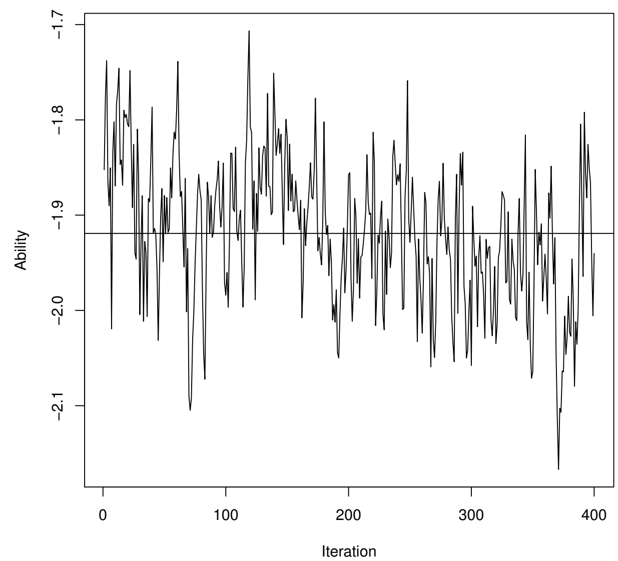

Figure 3.2 Iteration of difficulty parameter (MH with acceptance rate=.9511)

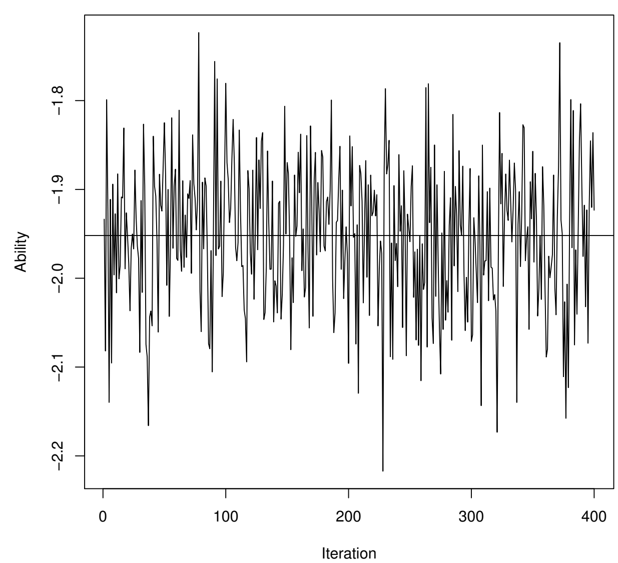

Figure 3.3 Iteration of difficulty parameter (MH with acceptance rate=.2033)

the Markov chain converges to the stationary distribution, π (Chib and Greenberg, 1995). If the acceptance rate is extremely low for some proposal distributions, the generated chain will be stuck in one value and this cannot generate a stationary distribution (Figure 3.1). On the other hand, when the acceptance rate is extremely high, the chain will move with every iteration which requires more iterations to achieve a stationary distribution (Figure 3.2). There have been many studies that have investigated the effective acceptance rate for the MH algorithm (Müller, 1991; Gelman et al., 1996; Roberts et al., 1997) which vary depending on many factors such as number of parameters, shape of the targeting distribution, etc. But, as a rule of thumb, a value of approximately around 0.3 is considered a good acceptance rate; the rate should fall between 0.2 and 0.7 (Vanem et al., 2013; see Figure 3.3). However, handling the proposal distribution to maintain the acceptance rate to be within certain interval is difficult when the number of parameters increases (Müller, 1991). For IRT model, number of parameters are depends on size of exam and number of examinees but easily more than 100 parameters should be estimated which limits the application of MH algorithm to most IRT models.

3.2 Gibbs Sampler

The Gibbs sampler (Geman and Geman, 1984) is a sequential sampling procedure for more than one set of parameters. At the core of the Gibbs sampler is the complete conditional distributions of each set of parameters. For example, when the joint distribution, $\pi(\theta, \beta) = p(\theta, \beta | U)$, is the desired distribution, two complete conditional distributions are needed for the Gibbs sampler such that

$$p(\theta | \beta, U) = \frac{p(U | \theta, \beta)\, p(\theta, \beta)}{\int p(U | \theta, \beta)\, p(\theta, \beta)\, d\theta} \tag{3.3}$$

and

$$p(\beta | \theta, U) = \frac{p(U | \theta, \beta)\, p(\theta, \beta)}{\int p(U | \theta, \beta)\, p(\theta, \beta)\, d\beta}. \tag{3.4}$$

Then, the Gibbs sampler is composed of two transition steps:

1. Sample θ from $p\left(\theta | \beta^k, U\right)$ and update $\theta^{k+1} = \theta$

2. Sample β from $p\left(\beta|\theta^{k+1}, U\right)$ and update $\beta^{k+1} = \beta$

Repeatedly, proceed with Steps 1 and 2 up to the desired number of iterations.

The Gibbs sampler can be considered a special case of the MH algorithm by setting the proposal distribution, q, as the complete conditional distribution with the acceptance rate, α, equaling 1. For the above example, there are only two steps, but this procedure can be generalized for high dimensional distributions.

Albert (1992) applied the Gibbs sampler to IRT for the normal ogive (where the ICC is the normal cumulative distribution function) with two parameters and compared the results from the Gibbs sampler and the EM algorithm using MATLAB. For the same model (two parameter normal ogive model), Fortran was also used later by Baker (1998). Application of the Gibbs sampler was successful, but there were some practical difficulties.

First, the Gibbs sampler simplifies the sampling steps using a complete conditional distribution and can sample one or two variables at a time. However, if it is infeasible to compute the complete conditional distributions for each parameter, the Gibbs sampler cannot be used. Second, when there are many parameters to estimate, additional steps for each parameter will be needed for the Gibbs sampler and this slows down convergence (Liu et al., 1994). Although Albert and Baker used Gibbs sampler for IRT with limited condition such as normal ogive with two item parameters, it is difficult to apply Gibbs sampler for more general IRT models and this is the reasons that Metropolis-Hasting within Gibbs algorithm (MH/Gibbs; Tierney, 1994) becomes a general approach IRT models.

3.3 Metropolis-Hastings within Gibbs Algorithm

It is easy to sample candidates with the MH algorithm because of the flexibility of the proposal distribution. However, it is hard to control the acceptance rate for IRT models with large numbers of examinees and items. Gibbs sampling does not requires controlling the acceptance rate and is easily used by sampling from complete conditional distributions. However, it is often difficult to develop the complete conditional distribution for most IRT models. By combining strategies from the Gibbs and MH algorithms, generating the Markov chain for IRT models could become easier (Patz and Junker, 1999a,b). This combination of the Gibbs sampler and MH algorithm is called Metroplis-Hasting within Gibbs (MH/Gibbs) and this algorithm was explained by Chib and Greenberg (1995).

The basic structure of the MH/Gibbs algorithm is similar to the Gibbs sampler which samples one or a couple parameters at a time based on the complete conditional distribution. However, the MH/Gibbs algorithm adapts the MH algorithm to overcome the difficulty of computing the complete conditional distribution. Then, the transition steps for the MH/Gibbs for IRT model are as follows:

1. Sample θ_j^{k+1} from $p\left(\theta_j|\theta_j^k, \xi^k, U\right)$ step for $1 \leq j \leq N$
   a) Draw a candidate, θ_j^*, for θ_j^{k+1} from proposal distribution, $q_\theta\left(\theta_j, \theta_j^k\right)$.
   b) Update $\theta_j^{k+1} = \theta_j^*$ with acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$
      i. Compute acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$:
      $$\alpha\left(\theta_j^k, \theta_j^*\right) = \min\left\{\frac{p\left(U|\theta_j^*, \xi^k\right) p\left(\theta_j^*, \xi^k\right) q_\theta\left(\theta_j^*, \theta_j^k\right)}{p\left(U|\theta_j^k, \xi^k\right) p\left(\theta_j^k, \xi^k\right) q_\theta\left(\theta_j^k, \theta_j^*\right)}, 1\right\} \tag{3.5}$$
      ii. Draw a random number, u, from uniform(0,1)
      iii. If $u \leq \alpha\left(\theta^k, \theta^*\right)$, then $\theta_j^{k+1} = \theta_j^*$. Otherwise, $\theta_j^{k+1} = \theta_j^k$.
2. Sample ξ_i^{k+1} from $p\left(\xi_i|\xi_i^k, \theta^{k+1}, U\right)$ step for $1 \leq i \leq I$
   a) Draw a candidate, ξ_i^*, for ξ_i^{k+1} from proposal distribution, $q_\xi\left(\xi_i, \xi_i^k\right)$.
   b) Update $\xi_i^{k+1} = \xi_i^*$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$
      i. Compute acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$:
      $$\alpha\left(\xi_i^k, \xi_i^*\right) = \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\} \tag{3.6}$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(\xi_i^k, \xi_i^*\right)$, then $\xi_i^{k+1} = \xi_i^*$. Otherwise, $\xi_i^{k+1} = \xi_i^k$.

3. Repeat Steps 1 and 2 until k reaches desired iteration size.

When the MH/Gibbs algorithm is used for IRT models, a symmetric distribution is a popular choice for the proposal distribution due to simplicity. Then, the equation for the acceptance rate is as follows:

$$\begin{aligned}\alpha\left(\theta_j^k, \theta_j^*\right) &= \min\left\{\frac{p\left(U|\theta_j^*, \xi^k\right)p\left(\theta_j^*, \xi^k\right)q_\theta\left(\theta_j^*, \theta_j^k\right)}{p\left(U|\theta_j^k, \xi^k\right)p\left(\theta_j^k, \xi^k\right)q_\theta\left(\theta_j^k, \theta_j^*\right)}, 1\right\} \\ &= \min\left\{\frac{p\left(U|\theta_j^*, \xi^k\right)p\left(\theta_j^*, \xi^k\right)}{p\left(U|\theta_j^k, \xi^k\right)p\left(\theta_j^k, \xi^k\right)}, 1\right\}. \end{aligned} \tag{3.7}$$

Steps 1 and 2 contain a sampling process like the Gibbs sampler, and the sub-steps in Steps 1 and 2, (a) and (b), have the same structure as the single iteration MH algorithm. Those sub-steps generate the complete conditional distribution by proposal distributions, q_θ and q_ξ which do not have any restriction. This makes it easier to apply the MH/Gibbs algorithm for IRT models than the Gibbs sampler. Also, even with a single iteration of the MH algorithm, the result of the MH/Gibbs algorithm converges to a stationary distribution π (Tierney, 1994).

Chapter 4

Comparison of Item and Ability Estimates between MBME and MH/Gibbs

4.1 Motivation

The two most popular methods to estimate item parameters for IRT models are the marginalized Bayesian modal estimation (MBME), which is the Bayesian implementation of the marginal maximum likelihood with expectation and maximization (MML/EM), and the Metropolis-Hasting within Gibbs algorithm. When the whole posterior distribution is not required, the MBME algorithm will be used because it is faster than MCMC and more easily employed through the use of off-the-shelf software such as BILOG-MG. Although the MBME algorithm can compute point estimates for item parameters and examinee abilities, this method cannot provide the entire posterior distributions. Thus, the MH/Gibbs algorithm is considered to estimate the entirety of the posterior distributions. Kim and Bolt (2007) demonstrated the similarity in item parameter estimate for the 2PL model between the MBME algorithm and MH/Gibbs algorithm. However, there are two problems in comparing item parameters and ability estimates between MBME and MH/Gibbs.

First, MH/Gibbs estimates have random difference for examinees who have identical responses or have the same number of items answered correctly for the Rasch model. When two examinees have exactly the same response for all items, the estimation of level of ability should be the same. Ability estimate from MCMC, however, are different for examinees even with identical responses because of the generation of a random candidate for the Metropolis-Hasting algorithm (Table 4.1). The Rasch model has an even stronger characteristic where examinees with the same number of correct answers should have a same ability because the number of correct answers is a sufficient statistic for the Rasch model (Fischer and Molenaar, 2012). This characteristic of the Rasch model is well demonstrated in Figure 4.1 which has only 21

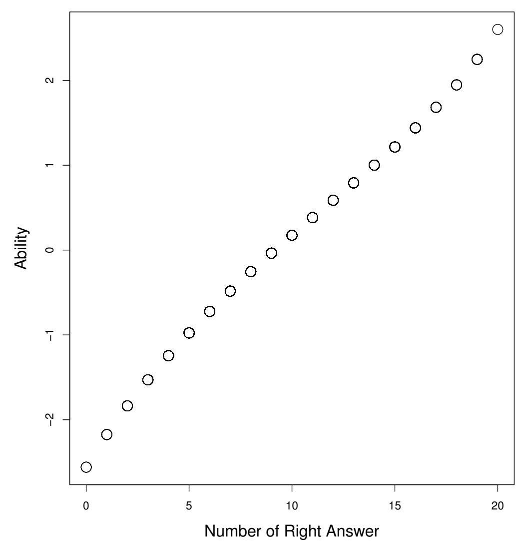

Figure 4.1 ability estimate vs. number correct for Rasch model from MBME

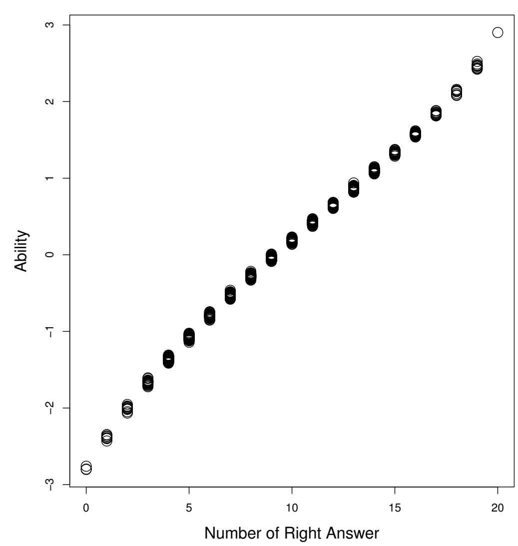

Figure 4.2 ability estimate vs. number correct for Rasch model from MCMC

different ability estimate levels corresponding to the number of correct answers. On the other hand, ability estimates from MCMC vary for examinees even with same number of correct answers (Figure 4.2).

Table 4.1 ability estimates for 2PL model with the identical response patterns

	Response Pattern	MBME estimation	MCMC estimation
Examinee 11	11111111111111101001	1.740948	1.829128
Examinee 184	11111111111111101001	1.740948	1.794808
Examinee 248	11111111111111101001	1.740948	1.828383
Examinee 645	11111111111111101001	1.740948	1.876770

Second, the pattern of standard errors between estimates from the MBME algorithm and estimates from the MH/Gibbs algorithm are inconsistent. Kim and Bolt (2007) compared item parameter estimates for 36 items. For all 36 items, estimates of both the discrimination parameter, a, and the difficulty parameter, b, from MCMC had a greater or equal standard error compared to those based on MBME. The difference in standard errors between the MBME and MCMC estimations are also mentioned in Hendrix's study (2011). She presents a method to estimate the approximate posterior distribution for IRT item parameters based on the MBME algorithm, called expectation and maximization plus expectation (EM+E; Hendrix, 2011) and the standard errors for item estimates from the MH/Gibbs algorithm are greater than the standard errors from the EM+E method which is based on MBME.

4.2 Supporting Evidence and Reasoning

Before the MBME algorithm was widely used, the JMLE was employed for IRT models. However, the ability estimate and discrimination estimates from the JMLE can be diverge for certain data sets. For instance, when the estimate of ability is bigger, the estimates of item discrimination. Then, the estimate of ability will be greater because of the item discrimination parameters. For many iteration steps, both estimations of ability and item discrimination will be greater repetitively and this can cause a divergence problem (Baker and Kim, 2004).

Similar trends are observed in the MCMC iteration between the ability and item difficulty. The mean of item difficulties is increased when the mean of abilities in the previous iteration are large and vice versa (Figure 4.3 and Figure 4.4). The ability of examinees and difficulty parameters of items do not diverge in the MH/Gibbs algorithm like the JMLE because of prior distributions, but this pattern could cause greater standard errors than it supposed to be which was mentioned by Kim and Bolt (2007). By regulating this pattern, the standard errors for item parameters from the MCMC become close to the standard errors from the MBME algorithm.

4.3 Item Linking between MBME and MH/Gibbs

When estimates from the two methods are compared, the linking of item and ability estimate is needed (von Davier, 2010). Since estimation of IRT models are based on arbitrary scales, a simple comparison between estimates from different methods

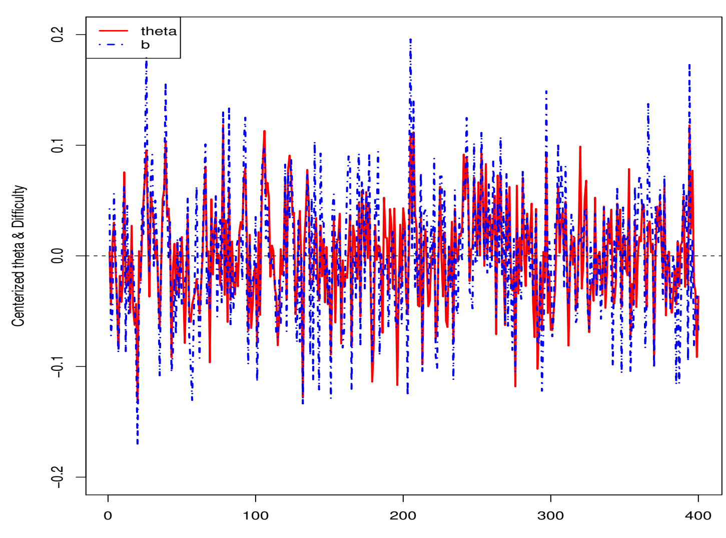

Figure 4.3 MCMC trend of mean ability and mean difficulty for the 1PL model

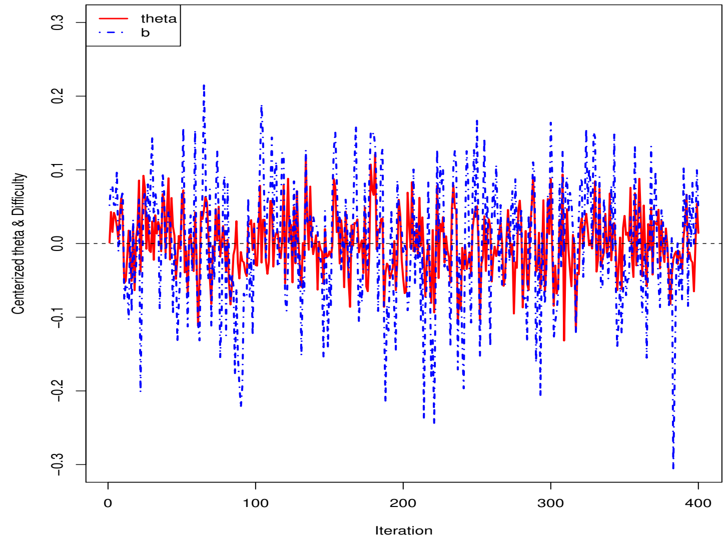

Figure 4.4 MCMC trend of mean ability and mean difficulty for the 2PL model

cannot be meaningful. Test of English as a Foreign Language (TOEFL) scores are an example of this problem. There are three different scoring methods for TOEFL: paper-based test(PBT), computer-based test(CBT), and internet-based test(IBT) scores. An examinee with a 119 IBT score cannot compare this score with an examinee with a 647 PBT score because these two scores have different scales. So, for a fair comparison, one of the scores should be transformed to the other scale first.

To link item parameter estimates from the MH/Gibbs to MBME, two properties will be used. First, it is assumed the two procedures are recovering the same underlying model, but with an undetermined scale and location. Since the strong linear relationship between the ability estimate from BILOG and MCMC are observed (Figure 4.5), the linear transformation between BILOG and MCMC for the ability exists as follows:

$$\theta_{BILOG} = A \cdot \theta_{MCMC} + B \tag{4.1}$$

To solve equation (4.1), use the mean and standard deviation of the ability from

MCMC and BILOG. Then,

$$E\left(\theta_{BILOG}\right) = A \cdot E\left(\theta_{MCMC}\right) + B \tag{4.2}$$

$$S\left(\theta_{BILOG}\right) = A \cdot S\left(\theta_{MCMC}\right) \tag{4.3}$$

By solving the above equations, the equation (4.1) becomes

$$\begin{aligned} \theta_{BILOG} &= A \cdot \theta_{MCMC} + B \\ &= \frac{S\left(\theta_{BILOG}\right)}{S\left(\theta_{MCMC}\right)} \cdot \theta_{MCMC} + E\left(\theta_{BILOG}\right) - \frac{S\left(\theta_{BILOG}\right)}{S\left(\theta_{MCMC}\right)} E\left(\theta_{MCMC}\right) \end{aligned} \tag{4.4}$$

Especially, when the ability estimates from BILOG are approximately nomrally distributed with a mean of zero and a standard deviation one, equation (4.4) will be simplified such that

$$\theta_{BILOG} = \frac{\theta_{MCMC} - E\left(\theta_{MCMC}\right)}{S\left(\theta_{MCMC}\right)} \tag{4.5}$$

Second, the fact when the same examinee solves the same question, the probability of a correct answer would be same regardless of the way ability and item parameters

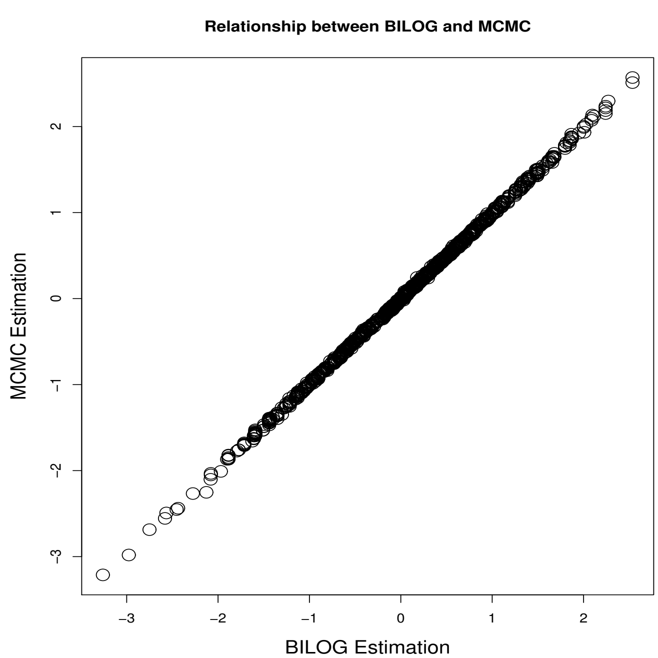

Figure 4.5 Linear relationship between MCMC and BILOG for the ability

are estimated. This relationship gives the second equation to link the item parameters from MH/Gibbs to MBME. For example, the ICC for 2PL models is as follows

$$P\left(U_{ij}=1|\theta_{j,BILOG},\xi_{i,BILOG}\right)=\frac{1}{1+e^{\left[-1.7a_{i,BILOG}\left(\theta_{j,BILOG}-b_{i,BILOG}\right)\right]}} \quad (4.6)$$

$$=\frac{1}{1+e^{\left[-1.7a_{i,MCMC}\left(\theta_{j,MCMC}-b_{i,MCMC}\right)\right]}} \quad (4.7)$$

$$=P\left(U_{ij}=1|\theta_{j,MCMC},\xi_{i,MCMC}\right)$$

From equations (4.6) and (4.7), the relationship between estimates from BILOG-MG and estimates from MCMC is defined as follows:

$$a_{MCMC}\left(\theta_{MCMC}-b_{MCMC}\right) \quad (4.8)$$

$$=a_{BILOG}\left(\theta_{BILOG}-b_{BILOG}\right)$$

$$=a_{BILOG}\left(\frac{\theta_{MCMC}-E\left(\theta_{MCMC}\right)}{S\left(\theta_{MCMC}\right)}-b_{BILOG}\right)$$

$$=\frac{a_{BILOG}}{S\left(\theta_{MCMC}\right)}\left(\theta_{MCMC}-E\left(\theta_{MCMC}\right)-S\left(\theta_{MCMC}\right)*b_{BILOG}\right) \quad (4.9)$$

Then, the transitions for item parameters will be

$$a_{BILOG}=S\left(\theta_{MCMC}\right)*a_{MCMC} \quad (4.10)$$

$$b_{BILOG}=\frac{b_{MCMC}-E\left(\theta_{MCMC}\right)}{S\left(\theta_{MCMC}\right)} \quad (4.11)$$

4.4 In-Chain Centered MCMC

As explained in Section 4.3, item linking between the estimates from the two different methods can allow for reasonable comparison of point estimates. However, the difference of standard errors for item parameters between MBME and MH/Gibbs does not improve with item linking. As mentioned in Section 4.2, Hendrix (2011) predicted a plausible reason of the greater standard errors in MH/Gibbs: a high correlation between item difficulty and the exmainee's ability. She suggested a shift and rescaling of the ability distribution at every update of the MCMC. As she was using OpenBugs this was impossible to implement, so she use a post-hoc centering in her study. This was done by separately recentering each iteration of the chain after entire MCMC procedure has finished. In this research, we examine rescaling the item parameters at the every iteration to mimic BILOG's setting of the posterior ability distribution to have mean zero and standard deviation one.

Instead of linking item parameters at the end of the iterations in MCMC, a linking procedure can be implemented between each iteration (in-chain) to regulate item parameters. Similar to applying the prior for ability parameter in EM algorithm, the item parameters can be centered by a linking process for every iteration and this can prevent a high correlation between the item difficulty parameters and the mean of ability estimates. We call this type of application in-chain centered and this application can make the standard errors from MCMC to better proximate the standard errors from MBME asymptotically. Also, in-chain centered MCMC can remove the chasing pattern (Figure 4.3) which causes over-estimation of standard errors.

In-Chain Centered MCMC using ability from previous iteration

There are two different approaches to centering item parameters. The first method is to use the abilities from the previous iteration (Figure 4.6). This algorithm required almost no extra computational time compared to MH/Gibbs and the algorithm is as follows:

1. Sample θ_j^{k+1} from $p\left(\theta_j|\theta_j^k, \xi^k, U\right)$ step for $1 \leq j \leq N$
    a) Draw a candidate, θ_j^*, for θ_j^{k+1} from proposal distribution, $q_\theta\left(\theta_j, \theta_j^k\right)$.
    b) Update $\theta_j^{k+1} = \theta_j^*$ with acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$

i. Compute acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$:

$$\alpha\left(\theta_j^k, \theta_j^*\right) = \min\left\{\frac{p\left(U|\theta_j^*, \xi^k\right) p\left(\theta_j^*, \xi^k\right) q_\theta\left(\theta_j^*, \theta_j^k\right)}{p\left(U|\theta_j^k, \xi^k\right) p\left(\theta_j^k, \xi^k\right) q_\theta\left(\theta_j^k, \theta_j^*\right)}, 1\right\} \tag{4.12}$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(\theta^k, \theta^*\right)$, then $\theta_j^{k+1} = \theta_j^*$. Otherwise, $\theta_j^{k+1} = \theta_j^k$.

2. Sample ξ_i^{k+1} from $p\left(\xi_i|\xi_i^k, \theta^{k+1}, U\right)$ step for $1 \leq i \leq I$

a) Draw a candidate, $\xi_i^{'}$, for ξ_i^{k+1} from proposal distribution, $q_\xi\left(\xi, \xi_i^k\right)$.

b) Adjust $\xi_i^{'}$ by linking such that

$$\xi_{i,1}^* = S_{\theta^{k+1}} * \xi_{i,1}^{'} \ \textbf{ and } \ \xi_{i,2}^* = \frac{1}{S_{\theta^{k+1}}}\left(\xi_{i,2}^{'} - \overline{\theta^{k+1}}\right)$$

c) Update $\xi_i^{k+1} = \xi_i^*$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$

i. Compute acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$:

$$\alpha\left(\xi_i^k, \xi_i^*\right) = \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\} \tag{4.13}$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(\xi_i^k, \xi_i^*\right)$, then $\xi_i^{k+1} = \xi_i^*$. Otherwise, $\xi_i^{k+1} = \xi_i^k$.

3. Repeat Steps 1 and 2 until k desired iteration size is reached.

In-Chain Centered MCMC using additional abilities generated by candidate of ξ_i^{k+1}

The second method is to cetneter the item parameters using new abilities which are generated from ξ_i^{k+1} (Figure 4.7). This algorithm requires an extra step to generate additional abilities which demands extra time and the algorithm is as follows:

1. Sample θ_j^{k+1} from $p\left(\theta_j|\theta_j^k, \xi^k, U\right)$ step for $1 \leq j \leq N$

a) Draw a candidate, θ_j^*, for θ_j^{k+1} from proposal distribution, $q_\theta\left(\theta_j, \theta_j^k\right)$.

b) Update $\theta_j^{k+1} = \theta_j^*$ with acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$

i. Compute acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$:

$$\alpha\left(\theta_j^k, \theta_j^*\right) = \min\left\{\frac{p\left(U|\theta_j^*, \xi^k\right) p\left(\theta_j^*, \xi^k\right) q_\theta\left(\theta_j^*, \theta_j^k\right)}{p\left(U|\theta_j^k, \xi^k\right) p\left(\theta_j^k, \xi^k\right) q_\theta\left(\theta_j^k, \theta_j^*\right)}, 1\right\} \tag{4.14}$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \le \alpha\left(\theta^k, \theta^*\right)$, then $\theta_j^{k+1} = \theta_j^*$. Otherwise, $\theta_j^{k+1} = \theta_j^k$.

2. Sample ξ_i^{k+1} from $p\left(\xi_i|\xi_i^k, \theta^{k+1}, U\right)$ step for $1 \le i \le I$

a) Draw a candidate, $\xi_i^{'}$, for ξ_i^{k+1} from proposal distribution, $q_\xi\left(\xi_i, \xi_i^k\right)$.

b) Sample $\theta_{j,b}^{**}$ from $p\left(\theta_j|\theta_j^{k+1}, \xi^{'}, U\right)$ for $1 \le j \le N$ and $1 \le b \le B$ where B is a desired size of replication.

c) Adjust $\xi_i^{'}$ by linking such that

$$\xi_{i,1}^* = S_{\theta^{**}} * \xi_{i,1}^{'} \ \textbf{and}\ \ \xi_{i,2}^* = \frac{1}{S_{\theta^{**}}}\left(\xi_{i,2}^{'} - \overline{\theta^{**}}\right)$$

d) Update $\xi_i^{k+1} = \xi_i^*$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$

i. Compute acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$:

$$\alpha\left(\xi_i^k, \xi_i^*\right) = \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\} \tag{4.15}$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \le \alpha\left(\xi_i^k, \xi_i^*\right)$, then $\xi_i^{k+1} = \xi_i^*$. Otherwise, $\xi_i^{k+1} = \xi_i^k$.

3. Repeat Steps 1 and 2 until k desired iteration size is reached.

By adding the item linking step, item parameters will be centered iteratively and the standard deviation of the item parameters should be decreased compared to those from MH/Gibbs.

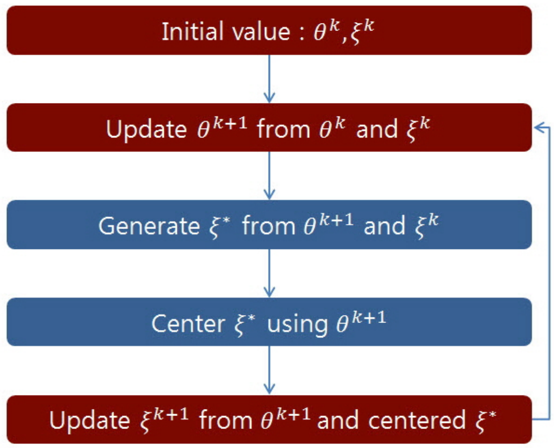

Figure 4.6 Flow chart for in-chain centered MCMC using the ability from previous iteration

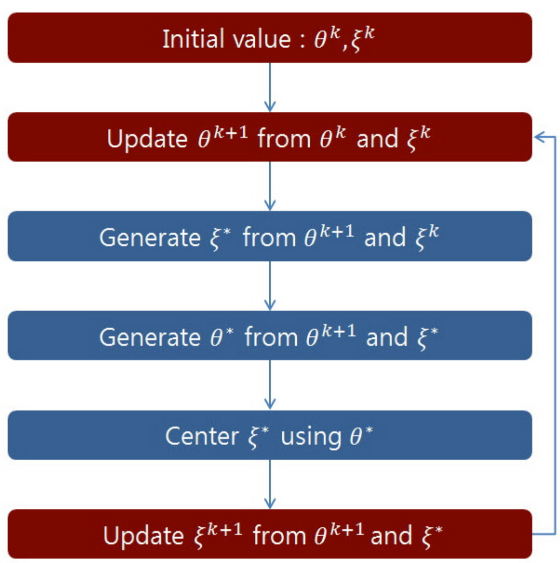

Figure 4.7 Flow chart for in-chain centered MCMC using the new set of ability

Both in-chain centered MCMC can be expressed in alternative algorithms which modify the proposal distribution instead of adding adjust steps. For this study, the proposal distributions for all item parameters (q_a, q_b, and q_c) are normal distributions. So, the item linking step for item discrimination parameters is as follows:

a) Draw a candidate, $a_i^{'}$, for a_i^{k+1} from proposal distribution $N\left(a_i^k, \sigma_a\right)$

b) Adjust $a_i^{'}$ by linking such that

$$a_i^* = S * a_i^{'}$$

where $S = S_{\theta^{k+1}}$ for in-chain centered MCMC using ability from previous iteration and $S = S_{\theta^{**}}$ for in-chain centered MCMC using additional abilities generated by candidate of ξ_i^{k+1}

c) Update $a_i^{k+1} = a_i^*$ with acceptance rate, $\alpha\left(a_i^k, a_i^*\right)$

However, Step a) and b) can be integrated by modifying the proposal distribution. Then,

a) Draw a candidate, $a_i^{'}$, for a_i^{k+1} from proposal distribution $N\left(S \cdot a_i^k, S \cdot \sigma_a\right)$

b) Update $a_i^{k+1} = a_i^*$ with acceptance rate, $\alpha\left(a_i^k, a_i^*\right)$

which makes in-chain centered MCMC identical to MH/Gibbs. The item linking step for the item difficulty parameters also can be integrated. The item linking step for item difficulty parameters is as follows:

a) Draw a candidate, $b_i^{'}$, for b_i^{k+1} from proposal distribution $N\left(b_i^k, \sigma_b\right)$

b) Adjust $b_i^{'}$ by linking such that

$$b_i^* = \frac{1}{S}\left(b_i^{'} - \overline{\theta}\right)$$

where $S = S_{\theta^{k+1}}$ and $\overline{\theta} = \overline{\theta^{k+1}}$ for in-chain centered MCMC using ability from previous iteration and $S = S_{\theta^{**}}$ and $\overline{\theta} = \overline{\theta^{**}}$ for in-chain centered MCMC using additional abilities generated by candidate of ξ_i^{k+1}

c) Update $b_i^{k+1} = b_i^*$ with acceptance rate, $\alpha\left(b_i^k, b_i^*\right)$

and Step a) and b) can be integrated similar procedure with the item discrimination parameter. Then,

a) Draw a candidate, $b_i^{'}$, for b_i^{k+1} from proposal distribution $N\left(\frac{b_i^k - \overline{\theta}}{S}, \frac{\sigma_b}{S}\right)$

b) Update $b_i^{k+1} = b_i^*$ with acceptance rate, $\alpha\left(b_i^k, b_i^*\right)$

Since both adjusting steps for the item discrimination parameter and the item difficulty parameter can be expressed in the Mh/Gibbs algorithm by modifying the proposal distributions, the generated chain from in-chain centered MCMC is an embedded ergodic Markov chain.

4.5 Same Response Treatment

The ability difference problem for the examinees with identical responses can be caused during Step (1a) in the MH/Gibbs procedure which generates a random candidate from the $q_\theta\left(\theta_j^k, \theta_j\right)$ step. So, by holding θ^{k+1} consistent across examinees with identical responses, the difference of ability estimate for MH/Gibbs algorithm can be solved.

1. Sample θ_r^{k+1} from $p\left(\theta_r | \theta_r^k, \xi^k, U\right)$ step for $1 \leq r \leq R$ where R is the number of different response patterns and θ_r is the ability of examinee with r^{th} response pattern.

    a) Draw a candidate, θ_r^*, for θ_r^{k+1} from proposal distribution, $q_\theta\left(\theta_r, \theta_r^k\right)$.

    b) Update $\theta_r^{k+1} = \theta_r^*$ with acceptance rate, $\alpha\left(\theta_r^k, \theta_r^*\right)$

i. Compute acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$:

$$\begin{aligned}
\alpha\left(\theta_r^k, \theta_r^*\right) &= \min\left\{\frac{p\left(U|\theta_r^*, \xi^k\right) p\left(\theta_r^*, \xi^k\right) q_\theta\left(\theta_r^*, \theta_r^k\right)}{p\left(U|\theta_r^k, \xi^k\right) p\left(\theta_r^k, \xi^k\right) q_\theta\left(\theta_r^k, \theta_r^*\right)}, 1\right\} \\
&= \min\left\{\frac{p\left(U|\theta_r^*, \xi^k\right) \phi\left(\theta_r^*\right)}{p\left(U|\theta_r^k, \xi^k\right) \phi\left(\theta_r^k\right)}, 1\right\} \\
&= \min\left\{\frac{\prod_{i=1}^I \left[P_i\left(\theta_r^*\right)^{U_{ir}} Q_i\left(\theta_r^*\right)^{1-U_{ir}}\right] \phi\left(\theta_r^*\right)}{\prod_{i=1}^I \left[P_i\left(\theta_r^k\right)^{U_{ir}} Q_i\left(\theta_r^k\right)^{1-U_{ir}}\right] \phi\left(\theta_r^k\right)}, 1\right\} \qquad (4.16)
\end{aligned}$$

where U_{ir} represents a response of r^{th} response pattern for item i.

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(\theta_r^k, \theta_r^*\right)$, then $\theta_r^{k+1} = \theta_r^*$. Otherwise, $\theta_r^{k+1} = \theta_r^k$.

2. Sample ξ_i^{k+1} from $p\left(\xi_i^k|\theta^{k+1}, U\right)$ step for $1 \leq i \leq I$

a) Draw a candidate, ξ_i^*, for ξ_i^{k+1} from proposal distribution, $q_\xi\left(\xi_i, \xi_i^k\right)$.

b) Update $\xi_i^{k+1} = \xi_i^*$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$

i. Compute acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$:

$$\alpha\left(\xi_i^k, \xi_i^*\right) \qquad (4.17)$$

$$= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\}$$

$$= \min\left\{\frac{\prod_r^R \left[P_i^*\left(\theta_r^{k+1}\right)^{U_{ir}} Q_i^*\left(\theta_r^{k+1}\right)^{1-U_{ir}}\right]^{n_r} p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{\prod_r^R \left[P_i\left(\theta_r^{k+1}\right)^{U_{ir}} Q_i\left(\theta_r^{k+1}\right)^{1-U_{ir}}\right]^{n_r} p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\} \qquad (4.18)$$

where n_r is the number of examinees with the r^{th} response pattern with $\sum_{r=1}^R n_r = N$

ii. Draw a random number, u, from uniform (0,1)

iii. If $u \leq \alpha\left(\xi_i^k, \xi_i^*\right)$, then $\xi_i^{k+1} = \xi_i^*$. Otherwise, $\xi_i^{k+1} = \xi_i^k$.

3. Repeat Steps 1 and 2 until k reaches desired iteration size.

This algorithm gives item estimates for all I items but only provides R ability estimates where $R \leq N$ for each response pattern. Then the examinees with the identical responses, U_r, will have the same ability estimate and iteration chain corresponding to θ_r.

Chapter 5

Simulation Design and Result

5.1 Simulation Design

For this study, three popular dichotomous responses with unidimensional ability models are considered: Rasch (i.e. 1PL: Rasch, 1960), 2PL (Lord, 1952), and 3PL (Birnbaum, 1968). The simulation design in detail is presented in Table (5.1) and all simulations are replicated once at each setting.

Table 5.1 Simulation design

Variable	Setting
IRT Model	1PL (Rasch), 2PL, 3PL
Number of Items	15,30
Number of Examinees	500, 1000
Simulated Parameter for U and Prior Distribution	$\theta \sim \text{Normal}(0,1)$ $a \sim \text{Lognormal}(0,0.5)$ $b \sim \text{Normal}(0,2)$ $c \sim \text{Beta}(5,17)$
Proposal Distributions	$\theta, a, b, c \sim$ Normal
Algorithm	MBME (BILOG-MG) MH within Gibbs In-chain centered MCMC using previous ability In-chain centered MCMC using extra ability Post-Hoc Linking

Since MH/Gibbs has flexibility of proposal distribution, any distributions can be chosen to generate candidate observation for the next iterations. However, symmetric distributions are recommended because of simplicity of acceptance rate computation. Normal distribution was used for a proposal distribution because normal distribution is the most popular symmetric distribution. Another possible candidate will be uniform distribution.

Because MCMC is a heavily computational process, Fortran is used for all MCMC processes instead of R. BILOG-MG, which is based on MBME algorithm, is used for the estimate of item parameters and the ability of examinees. For comparison purposes, the default priors (Du Toit, 2003; Partchev et al., 2008) of BILOG-MG are adapted to both MH within Gibbs and in-chain centered MCMC.

The biggest difficult for this simulation was resources. Because of size of simulation is large, it takes a lot of time even though I used Fortran for MCMC. Also, there are memory problems in R.

5.2 MH within Gibbs Algorithm Programming

In this simulation study, three logistic models with different number of item parameters are compared. Since each model has different numbers of item parameters and dependent relationships between item parameters, the application of MCMC is slightly different for each model.

The Rasch model has I item difficulty parameters but has only one discrimination parameter which is constant across all items. Thus, there is a single update process needed for the item discrimination parameter, a. For the 2PL model, both the item discrimination and difficulty parameters are different for each item. And there is one extra item parameter which is guessing parameter for 3PL model. Because of this difference among the three models, there are some practical differences in applying MH/Gibbs algorithm. As mentioned in the previous chapter, the MH/Gibbs algorithm is composed of 3 steps as follows:

1. Sample θ_j^{k+1} from $p\left(\theta_j|\theta_j^k, \xi^k, U\right)$ step for $1 \leq j \leq N$
    a) Draw a candidate, θ_j^*, for θ_j^{k+1} from the proposal distribution, $N\left(\theta_j^k, \sigma_\theta\right)$.
    b) Update $\theta_j^{k+1} = \theta_j^*$ with acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$

i. Compute acceptance rate, $\alpha\left(\theta_j^k, \theta_j^*\right)$:

$$\alpha\left(\theta_j^k, \theta_j^*\right) = \min\left\{\frac{p\left(U|\theta_j^*, \xi^k\right) p\left(\theta_j^*, \xi^k\right) q_\theta\left(\theta_j^*, \theta_j^k\right)}{p\left(U|\theta_j^k, \xi^k\right) p\left(\theta_j^k, \xi^k\right) q_\theta\left(\theta_j^k, \theta_j^*\right)}, 1\right\} \tag{5.1}$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(\theta_j^k, \theta_j^*\right)$, then $\theta_j^{k+1} = \theta_j^*$. Otherwise, $\theta_j^{k+1} = \theta_j^k$.

2. Sample ξ_i^{k+1} from $p\left(\xi_i^k|\theta^{k+1}, U\right)$ step for $1 \leq i \leq n$

a) Draw a candidate, ξ_i^*, for $\xi_{i,1}^{k+1}$ from the proposal distribution, $q_\xi\left(\xi_i, \xi_i^k\right)$.

b) Update $\xi_i^{k+1} = \xi_i^*$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$

i. Compute acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$:

$$\alpha\left(\xi_i^k, \xi_i^*\right) = \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\} \tag{5.2}$$

ii. Draw a random number, u, from uniform(0,1)

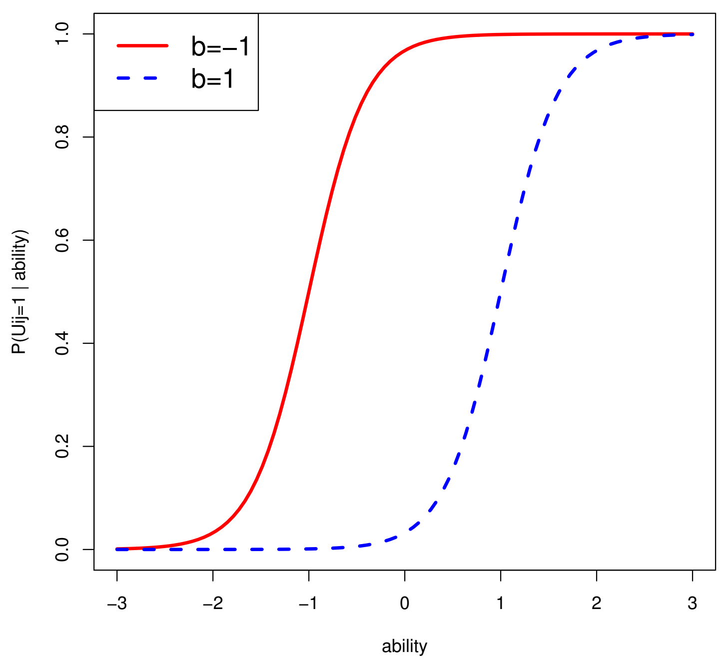

Figure 5.1 ICC curve for $b = 1$ and $b = -1$

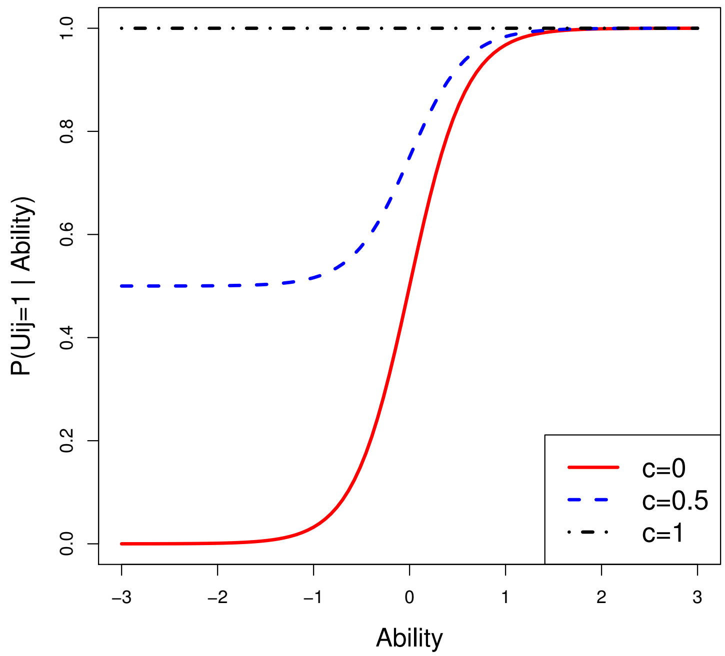

Figure 5.2 ICC curve for $c = 0$, $c = 0.5$, and $c = 1$

iii. If $u \leq \alpha\left(\xi_i^k, \xi_i^*\right)$, then $\xi_i^{k+1} = \xi_i^*$. Otherwise, $\xi_i^{k+1} = \xi_i^k$.

3. Repeat Steps 1 and 2 until k reaches desired iteration size.

For all algorithms in the Rasch, 2PL, and 3PL models, Step 1 and Step 3 are identical but Step 2 is slightly different across the three tested algorithms. For the Rasch model, there is one common discrimination parameter for all items and I difficulty parameters for each item. Step 2 for the 1PL model is separated into two substeps as follows:

2. Sample ξ_i^{k+1} from $p\left(\xi_i^k | \theta^{k+1}, U\right)$ step for $1 \leq i \leq n$

2-1. Sample a^{k+1} from $p\left(a^k | \theta^{k+1}, U\right)$ step for common a

(a) Draw a candidate, a^*, for a^{k+1}

from the proposal distribution, $N\left(a^k, \sigma_{1PL,a}\right)$.

(b) Update $a^{k+1} = a^*$ with acceptance rate, $\alpha\left(a^k, a^*\right)$

i. Compute acceptance rate, $\alpha\left(a^k, a^*\right)$, where $\xi_i^* = \left(a^*, b_i^k\right)$ and $\xi_i^k = \left(a^k, b_i^k\right)$:

$$
\begin{aligned}
&\alpha\left(a^k, a^*\right) \qquad (5.3)\\
&= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi^*\right) p\left(\theta^{k+1}, \xi^*\right) q_a\left(a^*, a^k\right)}{p\left(U|\theta^{k+1}, \xi^k\right) p\left(\theta^{k+1}, \xi^k\right) q_a\left(a^k, a^*\right)}, 1\right\}\\
&= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi^*\right) p\left(\theta^{k+1}, \xi^*\right)}{p\left(U|\theta^{k+1}, \xi^k\right) p\left(\theta^{k+1}, \xi^k\right)}, 1\right\}\\
&= \min\left\{\frac{\prod_{i=1}^{I}\prod_{j=1}^{N}\left[P\left(\theta_j^{k+1}, \xi_i^*\right)^{U_{ij}}\left(1 - P\left(\theta_j^{k+1}, \xi_i^*\right)\right)^{1-U_{ij}}\right] f_a\left(a^*\right)}{\prod_{i=1}^{I}\prod_{j=1}^{N}\left[P\left(\theta_j^{k+1}, \xi_i^k\right)^{U_{ij}}\left(1 - P\left(\theta_j^{k+1}, \xi_i^k\right)\right)^{1-U_{ij}}\right] f_a\left(a^k\right)}, 1\right\}
\end{aligned} \tag{5.4}
$$

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(a^k, a^*\right)$, then $a^{k+1} = a^*$. Otherwise, $a^{k+1} = a^k$.

2-2. Sample b_i^{k+1} from $p\left(b_i^k|\theta^{k+1}, U\right)$ step for $1 \leq i \leq n$

(a) Draw a candidate, b_i^*, for b_i^{k+1}

from the proposal distribution, $N\left(b_i^k, \sigma_{1PL,b}\right)$.

(b) Update $b_i^{k+1} = b_i^*$ with acceptance rate, $\alpha\left(b_i^k, b_i^*\right)$

i. Compute acceptance rate, $\alpha\left(a^k, a^*\right)$, where $\xi_i^* = \left(a^{k+1}, b_i^*\right)$ and $\xi_i^k = \left(a^{k+1}, b_i^k\right)$:

$$
\begin{aligned}
\alpha\left(b_i^k, b_i^*\right) &= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_b\left(b_i^*, b_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_b\left(b_i^k, b_i^*\right)}, 1\right\}\\
&= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right)}, 1\right\}\\
&= \min\left\{\frac{\prod_{j=1}^{N}\left[P\left(\theta_j^{k+1}, \xi_i^*\right)^{U_{ij}}\left(1 - P\left(\theta_j^{k+1}, \xi_i^*\right)\right)^{1-U_{ij}}\right] f_b\left(b_i^*\right)}{\prod_{j=1}^{N}\left[P\left(\theta_j^{k+1}, \xi_i^k\right)^{U_{ij}}\left(1 - P\left(\theta_j^{k+1}, \xi_i^k\right)\right)^{1-U_{ij}}\right] f_b\left(b_i^k\right)}, 1\right\}
\end{aligned} \tag{5.5}
$$

because the proposal distribution is normal and thus, symmetric.

ii. Draw a random number, u, from uniform(0,1)

iii. If $u \leq \alpha\left(b_i^k, b_i^*\right)$, then $b_i^{k+1} = b_i^*$. Otherwise, $b_i^{k+1} = b_i^k$.

For the 2PL model, item discrimination parameters and item difficulty parameters can be updated separately in the MH/Gibbs algorithm. However, updating both parameters simultaneously can decrease computation time.

2. Sample ξ_i^{k+1} from $p\left(\xi_i^k|\theta^{k+1}, U\right)$ step for $1 \leq i \leq n$

    a) Draw a candidate, a_i^*, for $\xi_{i,1}^{k+1}$
    from the proposal distribution, $N\left(\xi_{i,1}^k, \sigma_{2PL,a}\right)$.

    b) Draw a candidate, b_i^*, for $\xi_{i,2}^{k+1}$
    from the proposal distribution, $N\left(\xi_{i,2}^k, \sigma_{2PL,b}\right)$.

    c) Update $\xi_i^{k+1} = \xi_i^*$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$ where $\xi_i^k = \left(a_i^k, b_i^k\right)$ and $\xi_i^* = \left(a_i^*, b_i^*\right)$

        i. Compute acceptance rate, $\alpha\left(\xi_i^k, \xi_i^*\right)$:

$$\alpha\left(\xi_i^k, \xi_i^*\right) \tag{5.6}$$

$$
\begin{aligned}
&= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right) q_\xi\left(\xi_i^*, \xi_i^k\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right) q_\xi\left(\xi_i^k, \xi_i^*\right)}, 1\right\} \\
&= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) p\left(\theta^{k+1}, \xi_i^*\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) p\left(\theta^{k+1}, \xi_i^k\right)}, 1\right\} \\
&= \min\left\{\frac{p\left(U|\theta^{k+1}, \xi_i^*\right) f_a\left(a_i^*\right) f_b\left(b_i^*\right)}{p\left(U|\theta^{k+1}, \xi_i^k\right) f_a\left(a_i^k\right) f_b\left(b_i^k\right)}, 1\right\} \\
&= \min\left\{\frac{\prod_{j=1}^N \left[P\left(\theta_j^{k+1}, \xi_i^*\right)^{U_{ij}} \left(1 - P\left(\theta_j^{k+1}, \xi_i^*\right)\right)^{1-U_{ij}}\right] f_a\left(a_i^*\right) f_b\left(b_i^*\right)}{\prod_{j=1}^N \left[P\left(\theta_j^{k+1}, \xi_i^k\right)^{U_{ij}} \left(1 - P\left(\theta_j^{k+1}, \xi_i^k\right)\right)^{1-U_{ij}}\right] f_a\left(a_i^k\right) f_b\left(b_i^k\right)}, 1\right\}
\end{aligned} \tag{5.7}
$$

        ii. Draw a random number, u, from uniform(0,1)

        iii. If $u \leq \alpha\left(\xi_i^k, \xi_i^*\right)$, then $\xi_i^{k+1} = \xi_i^*$. Otherwise, $\xi_i^{k+1} = \xi_i^k$.

The algorithm for the 3PL model is the same as that of the 2PL model except for one extra step for the guessing parameters. As previously explained, it is not easy to compute the joint distribution of the three parameters when two of them are not independent from each other. An easier way is to add an additional step for the guessing parameter from the algorithm for the 2PL model.

5.3 In-Chain Centered MCMC Programming

As mentioned on Chapter 4, in-chain centered MCMC has one extra step from MH/Gibbs algorithm to adjust item parameters.

$$a_i^{**} = S_{\theta_{k+1}} * a_i^* \tag{5.8}$$

$$b_i^{**} = \frac{1}{S_{\theta_{k+1}}} \left(b_i^* - \overline{\theta^{k+1}} \right) \tag{5.9}$$

By adjusting generated candidates a^* and b^* using item linking, the new candidates , a^{**} and b^{**}, which are candidates under the assumption that the ability of examinees, are normally distributed with a mean of zero and standard deviation of one will replace a^* and b^*. For example, an algorithm for the 2PL model will have the following procedure:

2. Sample ξ_i^{k+1} from $p\left(\xi_i^k | \theta^{k+1}, U\right)$ step for $1 \le i \le n$

   a) Draw a candidate, a_i^*, for $\xi_{i,1}^{k+1}$ from the proposal distribution, $N\left(\xi_{i,1}^k, \sigma_{2PL,a}\right)$.

      (a) adjust Step - **Let** $a_i^{**} = S_{\theta_{k+1}} * a_i^*$

   b) Draw a candidate, b_i^*, for $\xi_{i,2}^{k+1}$ from the proposal distribution, $N\left(\xi_{i,2}^k, \sigma_{2PL,b}\right)$.

      (b) adjust Step - **Let** $b_i^{**} = \frac{1}{S_{\theta_{k+1}}} \left(b_i^* - \overline{\theta^{k+1}} \right)$

   c) Update $\xi_i^{k+1} = \xi_i^{**}$ with acceptance rate, $\alpha\left(\xi_i^k, \xi_i^{**}\right)$ where $\xi_i^k = \left(a_i^k, b_i^k\right)$ and $\xi_i^* = (a_i^{**}, b_i^{**})$

Table 5.2 Legend for Graphs

Legend	Explanation
MBME	Estimations from Bilog-MG
MHGibbs	Estimations from MH/Gibbs
Resp	Estimations from MH/Gibbs with response treatment
InChainPrv	In-Chain centered MCMC using previous ability with resp. treatment

Table 5.3 Response matrices

	Number of Examinees	Number of Items	Model 1PL	2PL	3PL
Num. of Score Pattern	500	15	14 (1∼14)	16 (0∼15)	13 (2∼14)
	500	30	26 (4∼29)	26 (3∼27, 29)	24 (2∼15)
	1000	15	15 (0∼14)	15 (0∼13, 15)	14 (2∼15)
	1000	30	30 (0, 2∼30)	26 (1∼26)	26 (5∼30)
Num. of Resp. Pattern	500	15	190	163	346
	500	30	482	477	491
	1000	15	342	219	572
	1000	30	584	857	985

5.4 Simulation Result

The simulation study begins with generating the response matrix, U, which differs depending on the generating model. For this research, three different response matrices are simulated for 1PL, 2PL, and 3PL models. Each simulated response have different ranges of scores and number of response patterns.

Figures (5.3) and (5.4) show comparisons of ability estimates and standard errors between MH/Gibbs and MH/Gibbs after the same score treatment in the sample simulation for Rasch model. By adjusting the ability of examinees with same number of correct answers, the ability estimates and standard errors between MBME and MCMC have a one to one relationship without losing accuracy in the ability estimate. On the other hand, standard errors for item difficulty are increased because of the same score treatment procedure (Figure 5.5) although item difficulty estimates are almost same between MH/Gibbs with and without same score treatment procedure. Amount of standard errors increase for item difficulty of Rasch is significantly greater than 2PL and 3PL case because of sufficient statistics. For Rasch model, sufficient statistic is number of right answer and which decrease variety of ability. This property

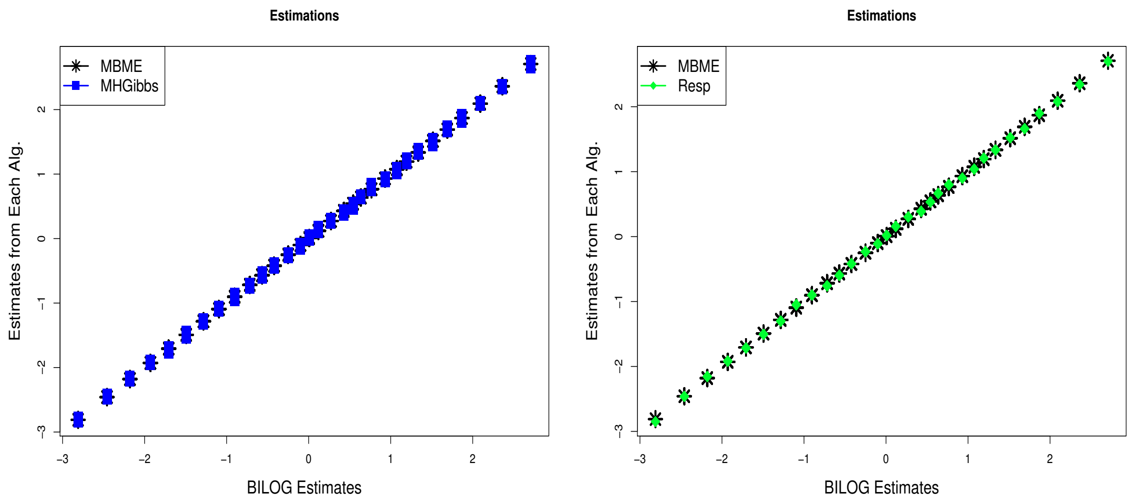

Figure 5.3 Comparison of estimates for ability between MHGibbs and MHGibbs with response treatment (Rasch model with 1000 examinees and 30 items)

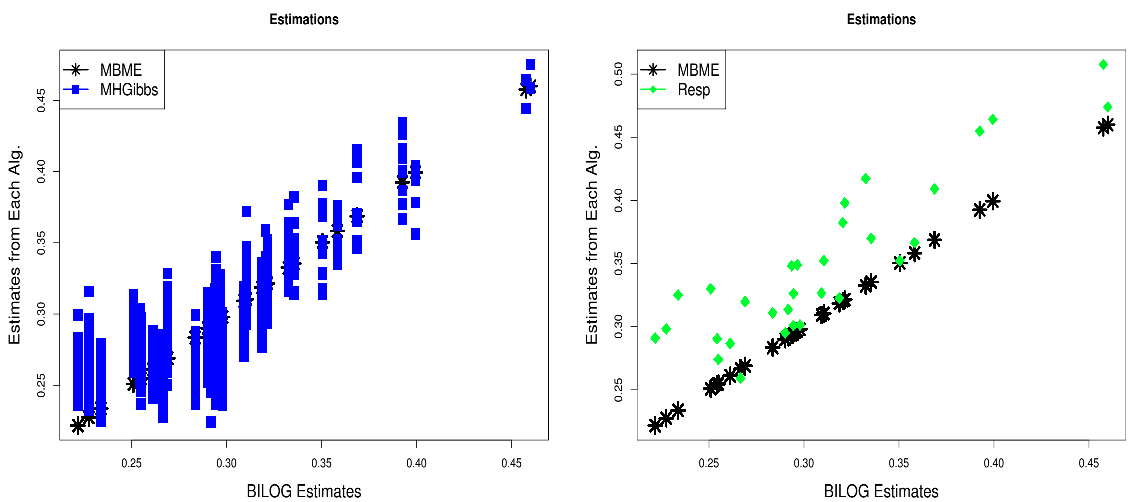

Figure 5.4 Comparison of SE for ability between MHGibbs and MHGibbs with response treatment (Rasch model with 1000 examinees and 30 items)

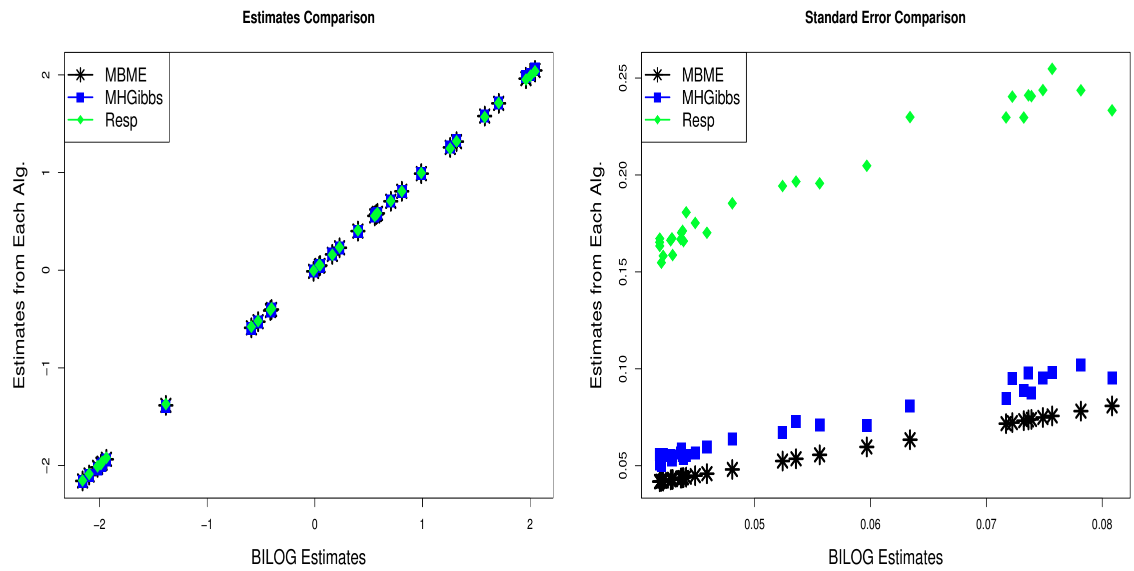

Figure 5.5 Comparison of estimates and SE for the item difficulty between MHGibbs and MHGibbs with response treatment (Rasch model with 1000 examinees and 30 items)

of Rasch model affects greater standard errors.

The two parameter logistic model (2PL) does not have to have the same ability estimates for the different responses with same number of right answers. However, it is rational to assume that the examinees with identical responses have identical ability estimates. Table 5.3 shows number of response patterns for each simulations. For example, there are 190 different response patterns for the simulated response among 1000 examinees with 30 items which implies that there are 310 examinees who have at least one more examinee with the identical response pattern. However, it is hard to distinguish the difference before and after forcing the ability of the examinees with the identical response pattern to be equal through the algorithm noted in Section 4.5 (Figure 5.6 and 5.7). Item discrimination parameters and difficulty parameters showed very similar patterns even after adjusting the ability of examinees with identical response patterns (Figure 5.8 and 5.9).

The identical response treatment does not affect the 3PL model either. For all ability, discrimination, difficulty, and guessing parameters, there is no significant difference between with and without same response treatment. (Figure 5.10, 5.11, 5.12, 5.13, and 5.14).

Based on the simulations, the identical response treatment (same score treatment for the Rasch model) does not cause any significant difference in estimates and standard errors of all parameters for 2PL and 3PL models. However, for the Rasch model, the standard errors for the item difficulty parameter are increased when the ability of the examinees with the same score are forced to be identical.

Same Response Treatment with In-Chain Centered MCMC

The correlation between the mean of examinee abilities and the mean of the item difficulty parameters is examined in Section 4.2. For the first 200 iterations after burn in and thinning from the MH/Gibbs algorithm for the Rasch and 2PL models, the mean ability estimates correlated with the mean of item difficulty parameters (Figure 5.15 and 5.21). The decreasing correlation between the mean of ability and the mean of item difficulty is observed by applying an in-chain centered procedure for both 1PL (Rasch) and 2PL models (Figure 5.16 $\sim$ 5.19 and 5.22 $\sim$ 5.25). Although the iteration patterns for 3PL is not mentioned in Section 4.2, 3PL also shows a the similar weaker shrinkage pattern similar to those of 1PL and 2PL (Figure 5.28 $\sim$ 5.31).

When the expected reason for the greater standard errors from MH/Gibbs is the corresponding relationship between the ability of examinee and the item difficulty estimate, the vibration pattern of iterations by applying an in-chain centered procedure to MH/Gibbs algorithm will reduce the standard errors for item parameters and the standard error reduction of item difficulty parameters will decrease the standard error for the ability of the examinee.

Figure 5.33 shows a significant reduction in the standard error by applying the in-chain centered procedure into the same score treatment. Because the sufficient

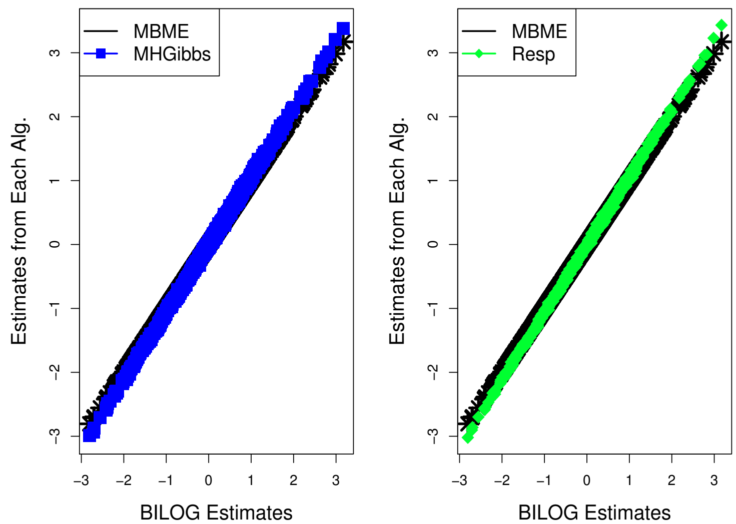

Figure 5.6 Comparison of estimates for ability between MHGibbs and MHGibbs with response treatment (2PL model with 1000 examinees and 30 items)

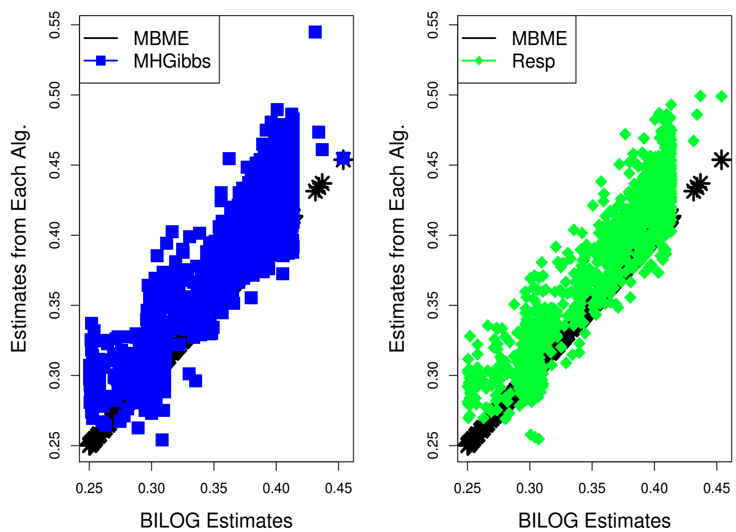

Figure 5.7 Comparison of SE for ability between MHGibbs and MHGibbs with response treatment (2PL model with 1000 examinees and 30 items)

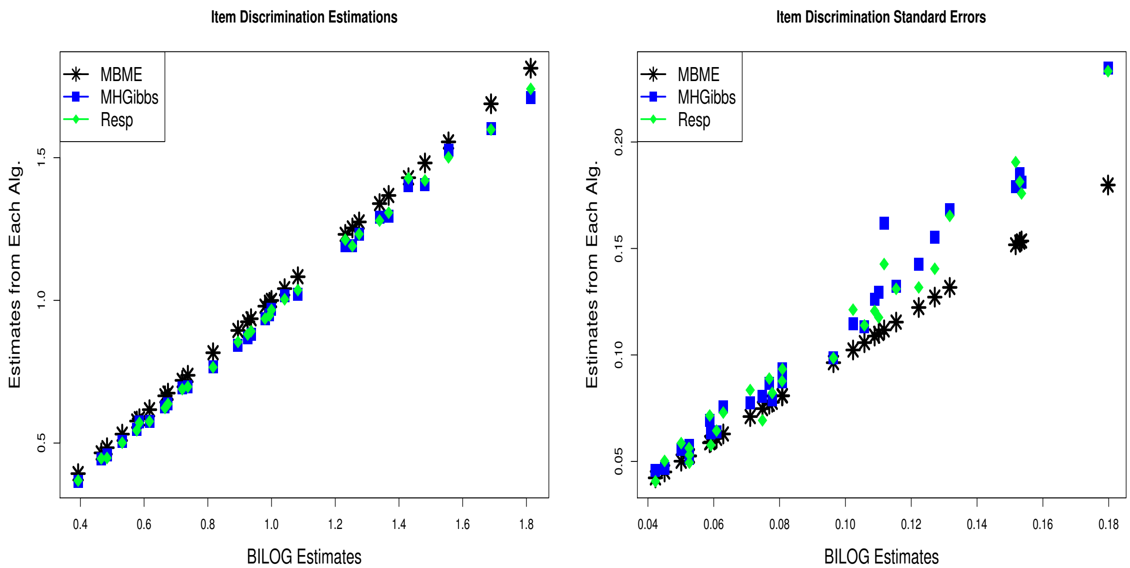

Figure 5.8 Comparison of estimates and SE for the item discrimination between MHGibbs and MHGibbs with response treatment (2PL model with 1000 examinees and 30 items)

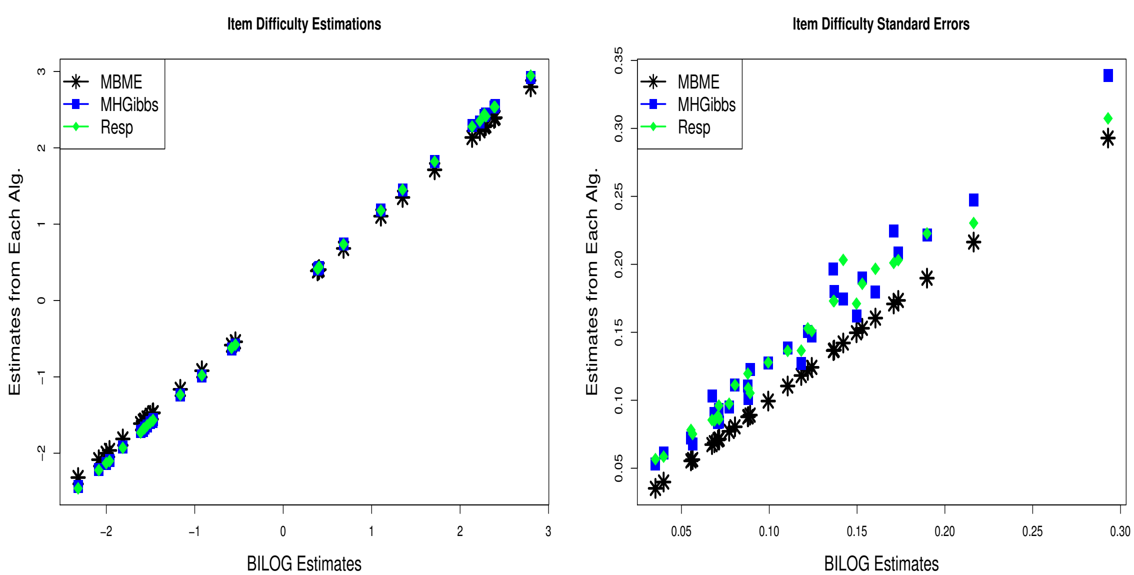

Figure 5.9 Comparison of estimates and SE for the item difficulty between MHGibbs and MHGibbs with response treatment (2PL model with 1000 examinees and 30 items)

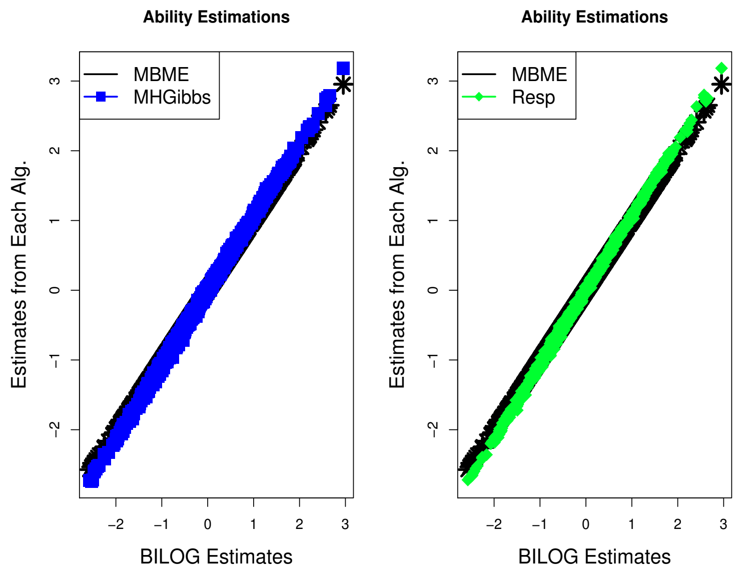

Figure 5.10 Comparison of estimates for ability between MHGibbs and MHGibbs with response treatment (3PL model with 1000 examinees and 30 items)

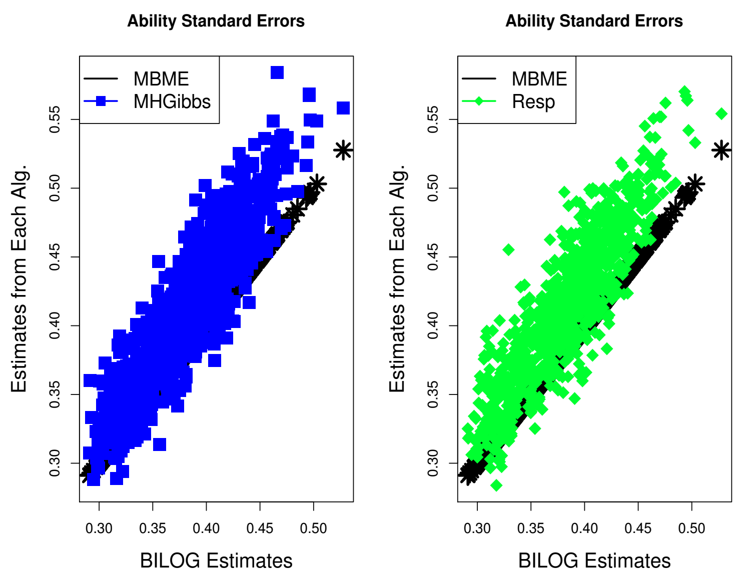

Figure 5.11 Comparison of SE for ability between MHGibbs and MHGibbs with response treatment (3PL model with 1000 examinees and 30 items)

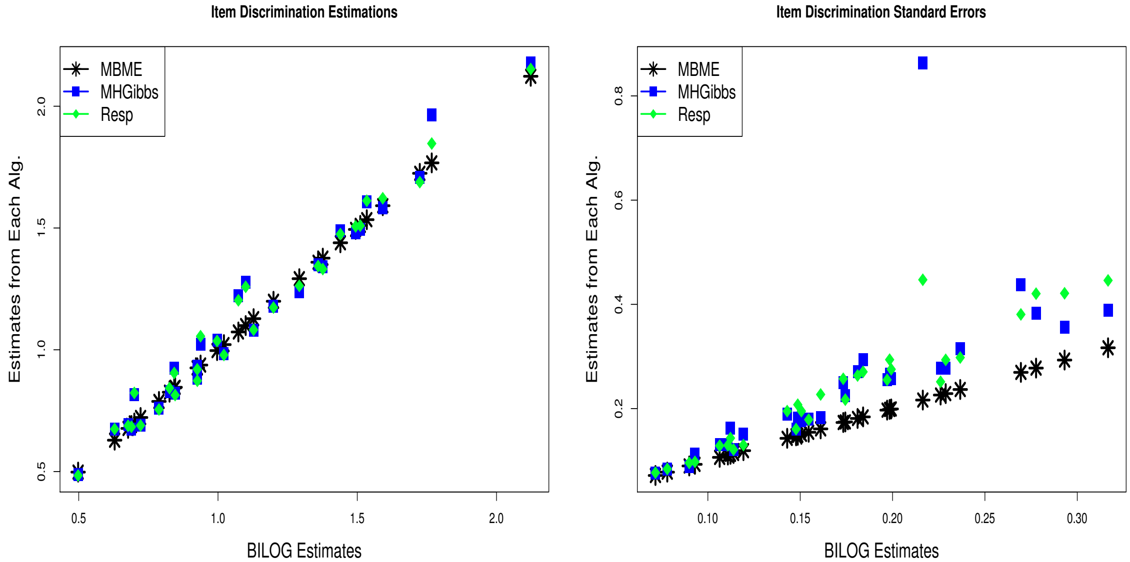

Figure 5.12 Comparison of estimates and SE for the item discrimination between MHGibbs and MHGibbs with response treatment (3PL model with 1000 examinees and 30 items)

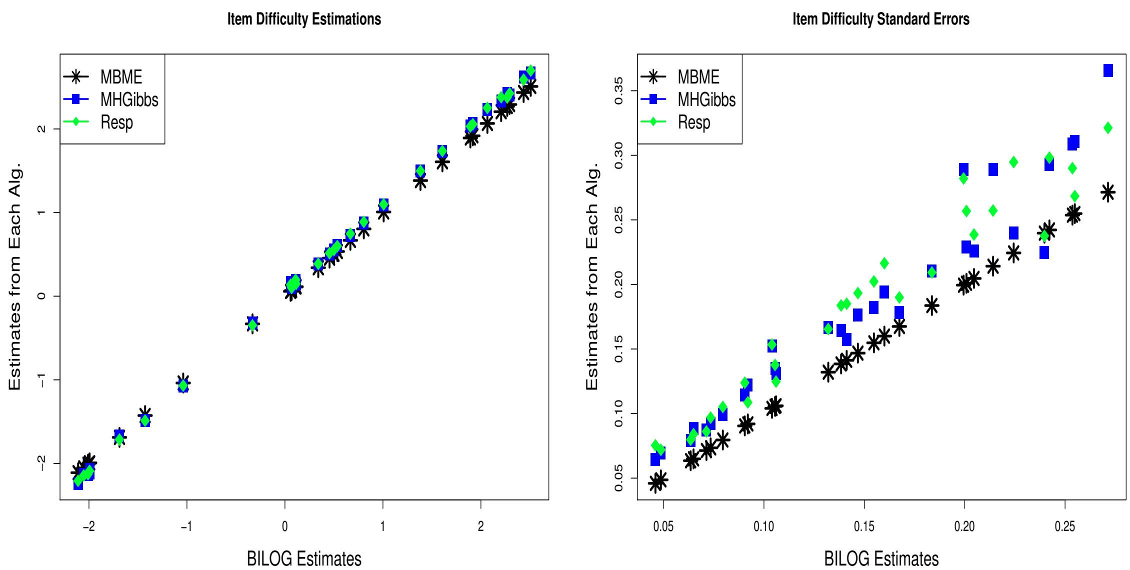

Figure 5.13 Comparison of estimates and SE for the item difficulty between MHGibbs and MHGibbs with response treatment (3PL model with 1000 examinees and 30 items)

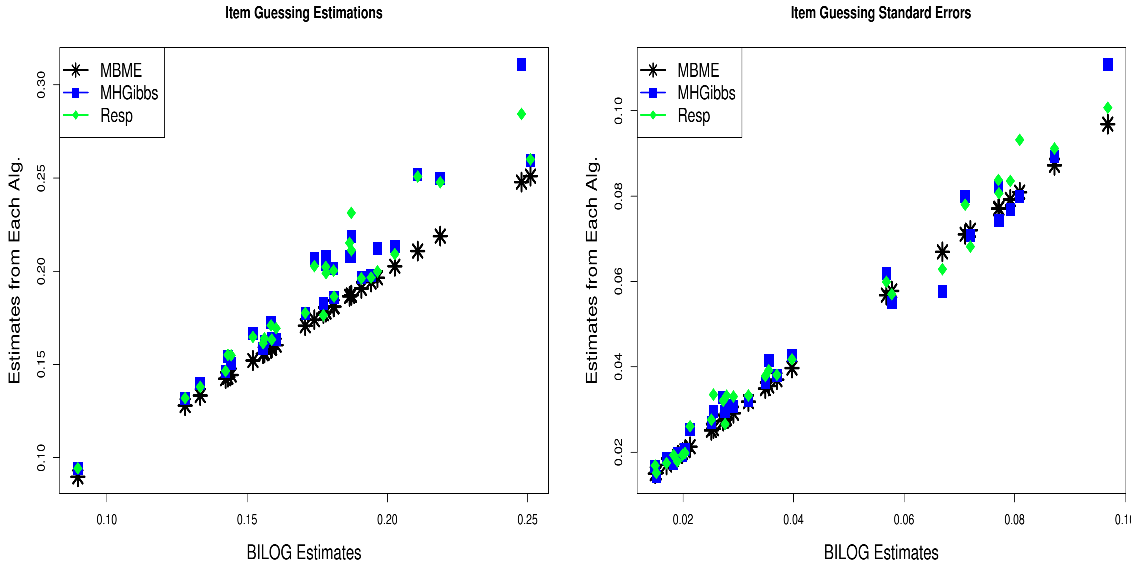

Figure 5.14 Comparison of estimates and SE for the item guessing between MHGibbs and MHGibbs with response treatment (3PL model with 1000 examinees and 30 items)

statistics for Rasch model is number of correct answer, there are only estimated abilities. For all 30 estimated abilities for the examinee, standard errors were closer to the standard errors from MBME without losing accuracy of ability estimate. In-chain centered MCMC improves estimating standard error for not only the ability of examinee but also the item parameters (Figure 5.34). However, the standard errors from MH/Gibbs with the same score treatment after the in-chain centered procedure are not as close to the standard errors from MBME as much as the standard errors from MH/Gibbs (Figure 5.35). This result shows that applying the same score treatment to MH/Gibbs will decrease accuracy of of standard errors for item difficulty, but, by applying an in-chain centered procedure, the appropriate estimation can be computed even after applying the same score treatment for the Rasch model.

The standard error difference for item difficulty parameters are only observed in the Rasch model, while the 2PL and 3PL models give similar standard errors for all ability and item parameters regardless of applying identical response treatment. So,

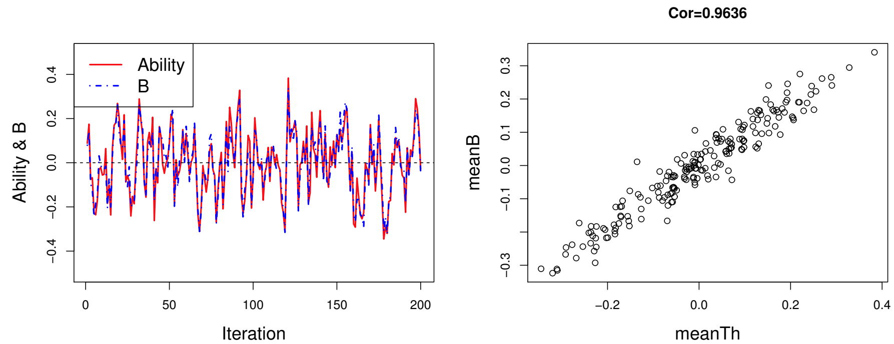

Figure 5.15 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for MH/Gibbs (Rasch model with 1000 examinees and 30 items)

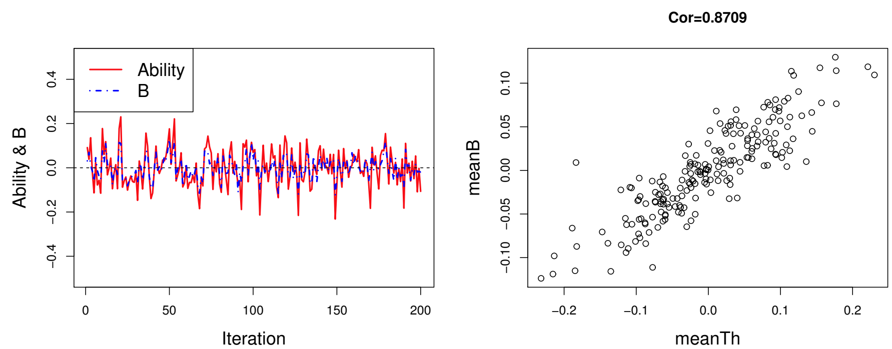

Figure 5.16 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC using previous ability (Rasch model with 1000 examinees and 30 items)

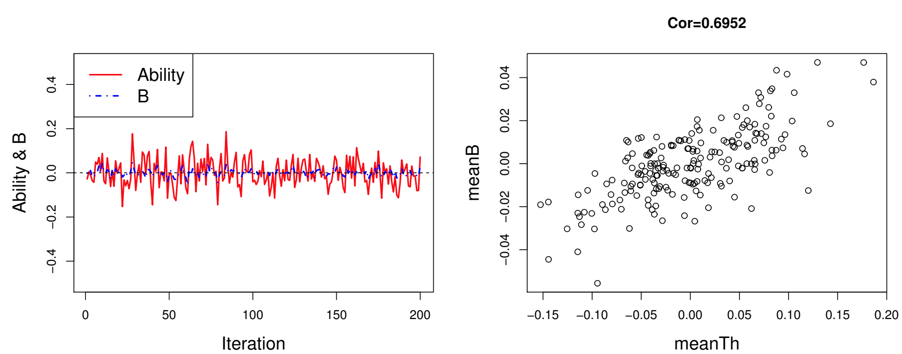

Figure 5.17 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC one extra ability set (Rasch model with 1000 examinees and 30 items)

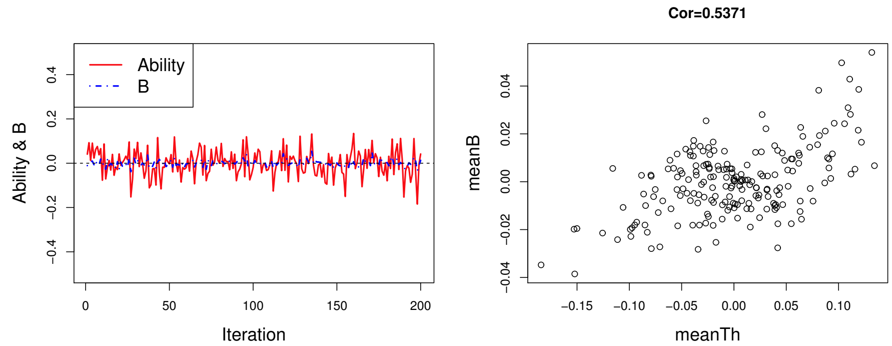

Figure 5.18 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC three extra ability sets (Rasch model with 1000 examinees and 30 items)

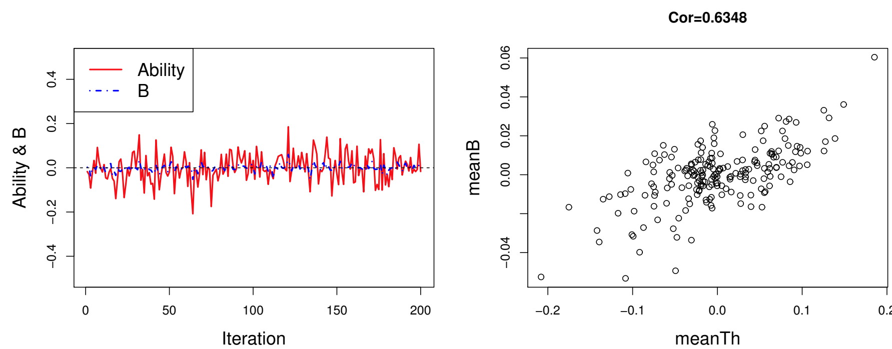

Figure 5.19 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC five extra ability sets (Rasch model with 1000 examinees and 30 items)

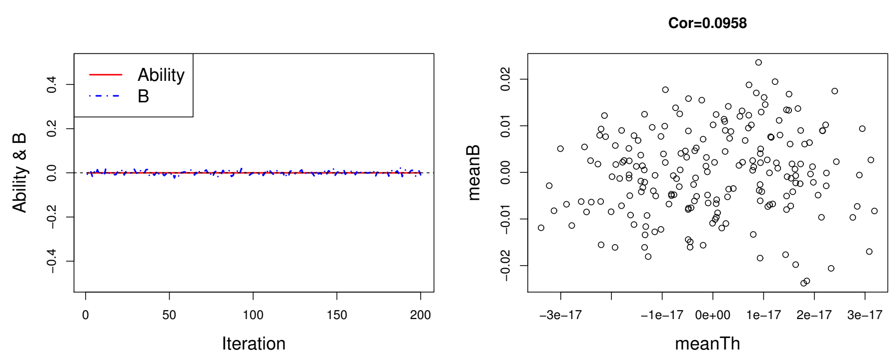

Figure 5.20 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for post-hoc linking (Rasch model with 1000 examinees and 30 items)

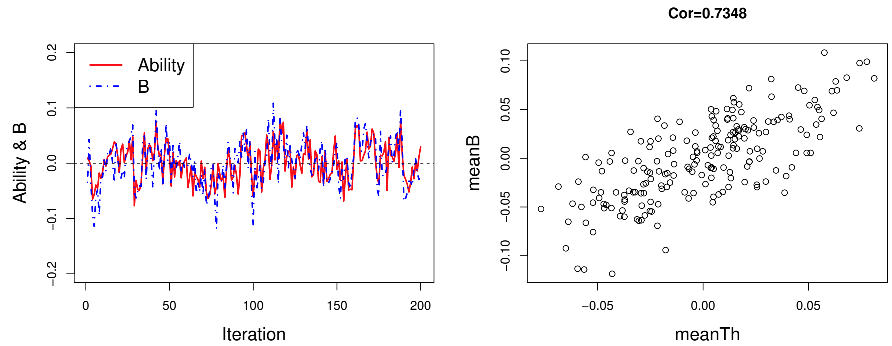

Figure 5.21 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for MH/Gibbs (2PL model with 1000 examinees and 30 items)

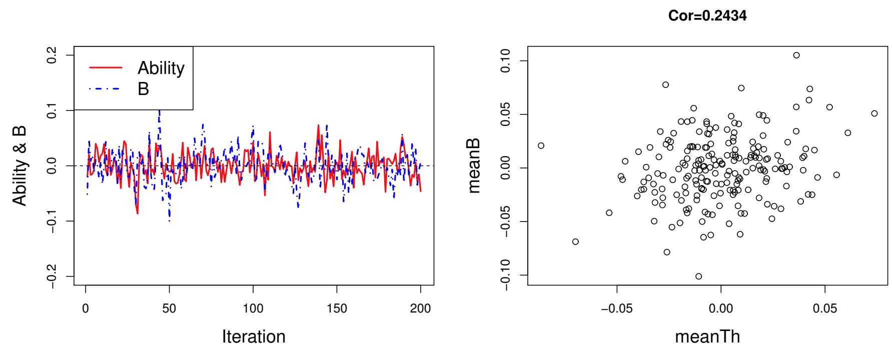

Figure 5.22 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC using previous ability (2PL model with 1000 examinees and 30 items)

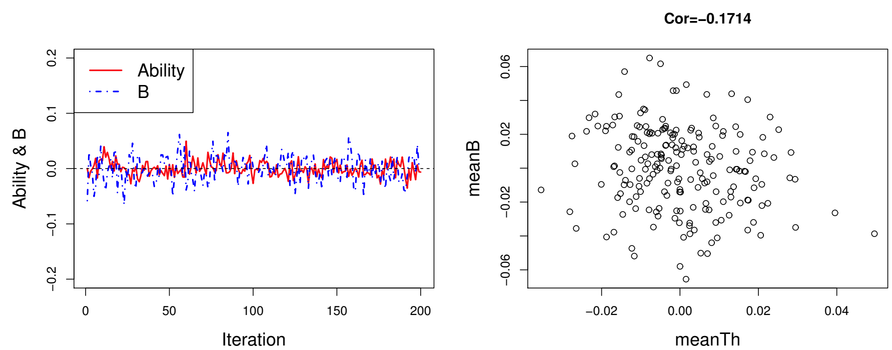

Figure 5.23 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC using one extra ability set (2PL model with 1000 examinees and 30 items)

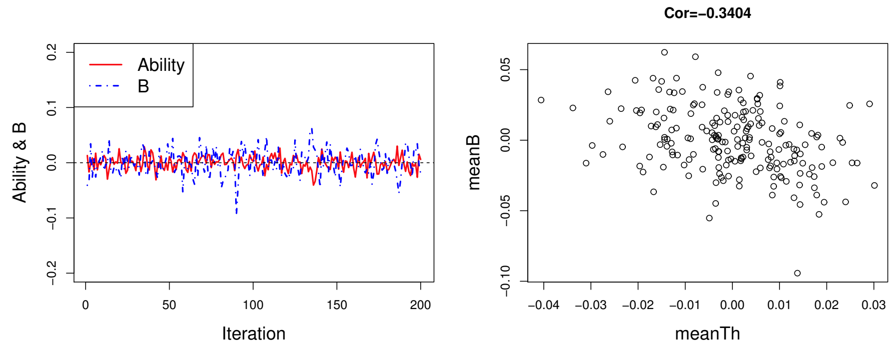

Figure 5.24 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC three extra ability sets (2PL model with 1000 examinees and 30 items)

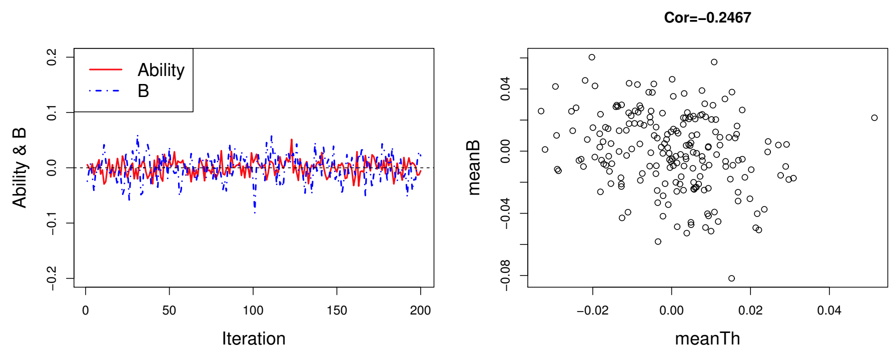

Figure 5.25 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC five extra ability sets (2PL model with 1000 examinees and 30 items)

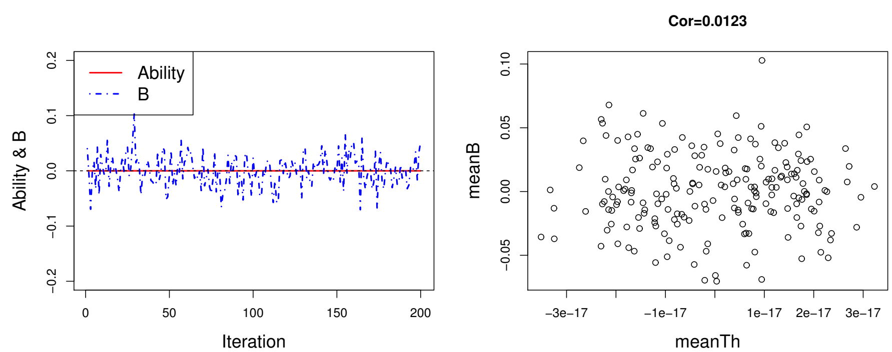

Figure 5.26 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for post-hoc linking (2PL model with 1000 examinees and 30 items)

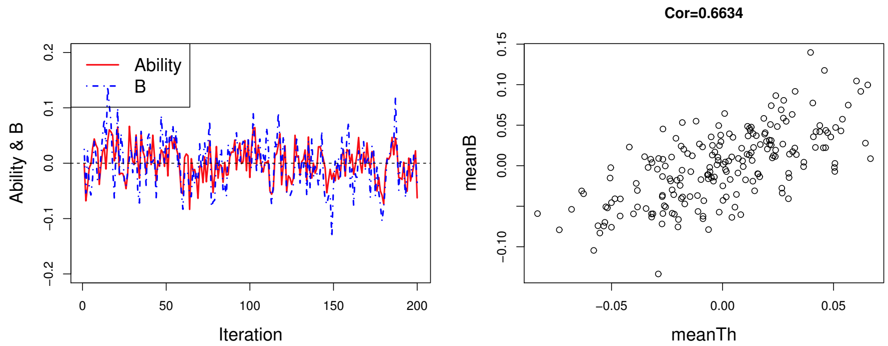

Figure 5.27 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for MH/Gibbs (3PL model with 1000 examinees and 30 items)

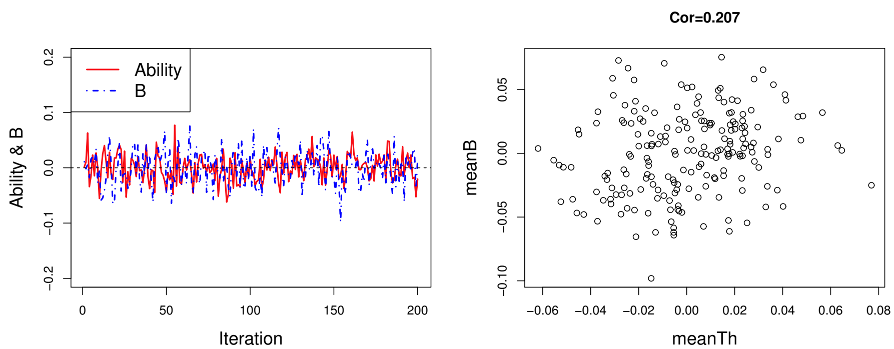

Figure 5.28 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC using previous ability (3PL model with 1000 examinees and 30 items)

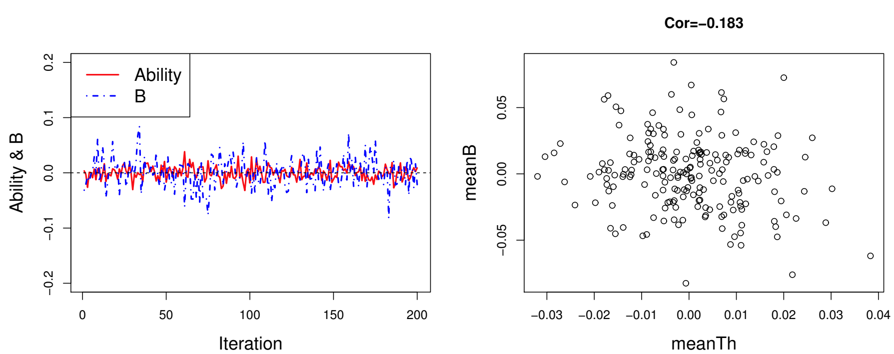

Figure 5.29 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC using one extra ability set (3PL model with 1000 examinees and 30 items)

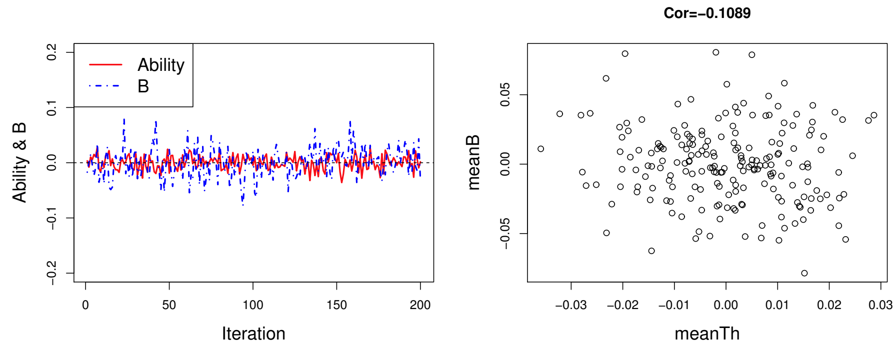

Figure 5.30 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC three extra ability sets (3PL model with 1000 examinees and 30 items)

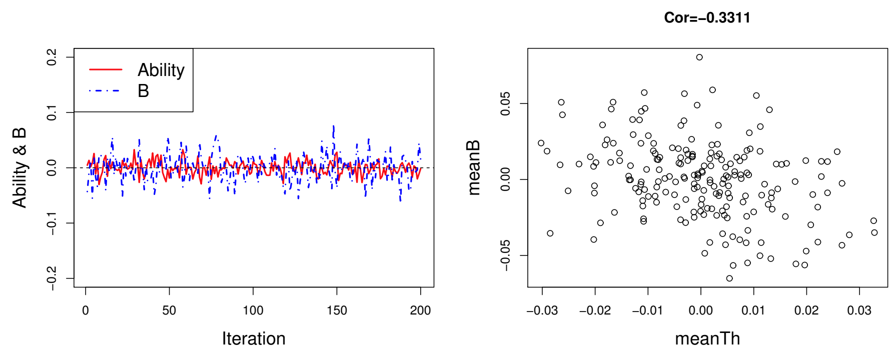

Figure 5.31 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for in-chain centered MCMC five extra ability sets (3PL model with 1000 examinees and 30 items)

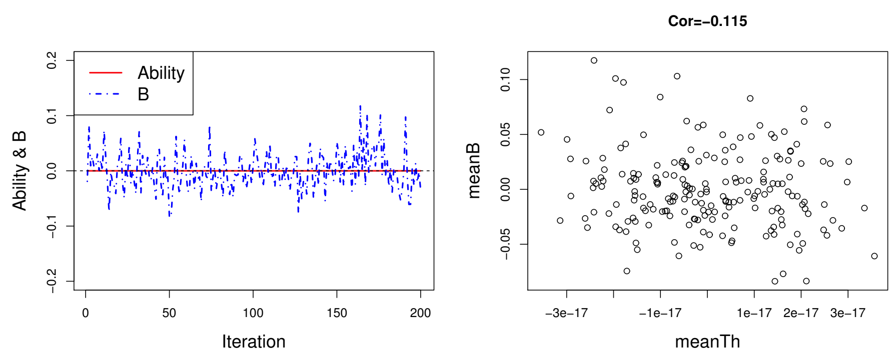

Figure 5.32 Iteration trends of the mean ability and the mean difficulty for MH/Gibbs and dotplot between mean ability and mean item difficulty for post-hoc linking (3PL model with 1000 examinees and 30 items)

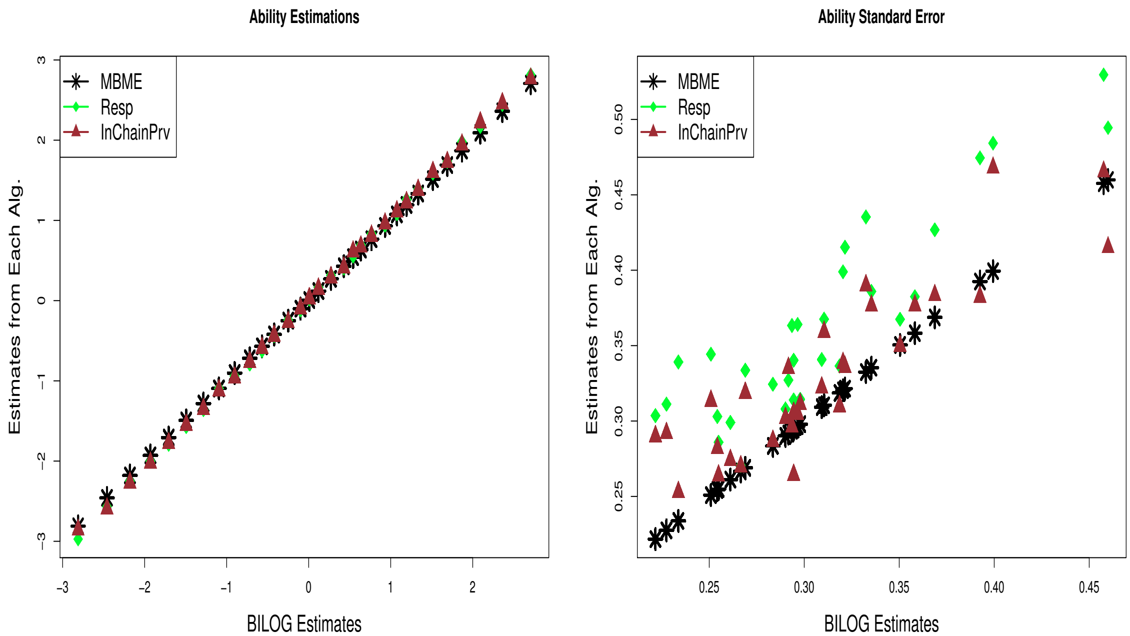

Figure 5.33 Comparison of estimates and SE for the ability of examinee from MHGibbs with response treatment between before and after in-chain centered procedure (Rasch model)

in-chain centered MCMC will be more effective to reduce standard errors for the Rasch model than 2PL and 3PL models. Estimates and standard errors from the MH/Gibbs, MH/Gibbs with an identical response treatment, and MH/Gibbs with an identical response treatment and in-chain centered procedure are shown in Figure 5.36. There is no significant difference that can be detected without numerical comparison. On the other hand, significant improvements are observed for the standard errors of item discrimination and difficulty parameters (Figure 5.37 and 5.38). However, in-chain centered MCMC does not provide clearly better result for the 3PL model at present. All three different algorithms provide the similar estimates and similar standard errors for ability, item discrimination, item difficulty, and item guessing parameters.

Additional simulation results are in appendices which include all item parameters comparison graphs and tables for 1000 examinees with 30 items, 1000 examinees with 15 items, 500 examinees with 30 items, and 500 examinees with 15 items.

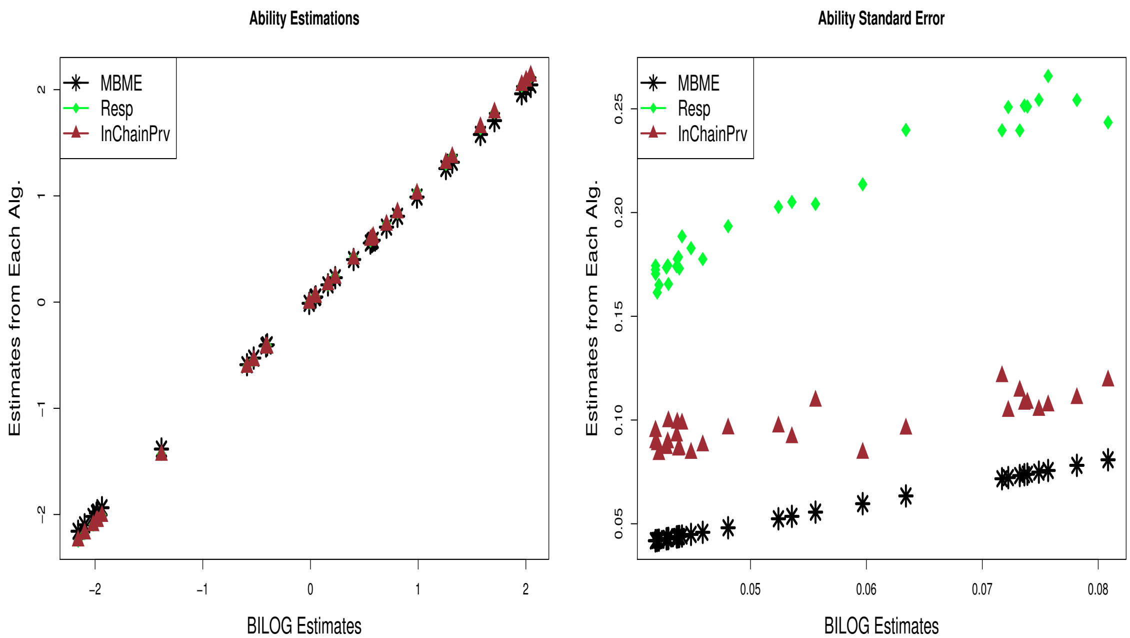

Figure 5.34 Comparison of estimates and SE for the item difficulty from MHGibbs with response treatment between before and after in-chain centered procedure (Rasch model)

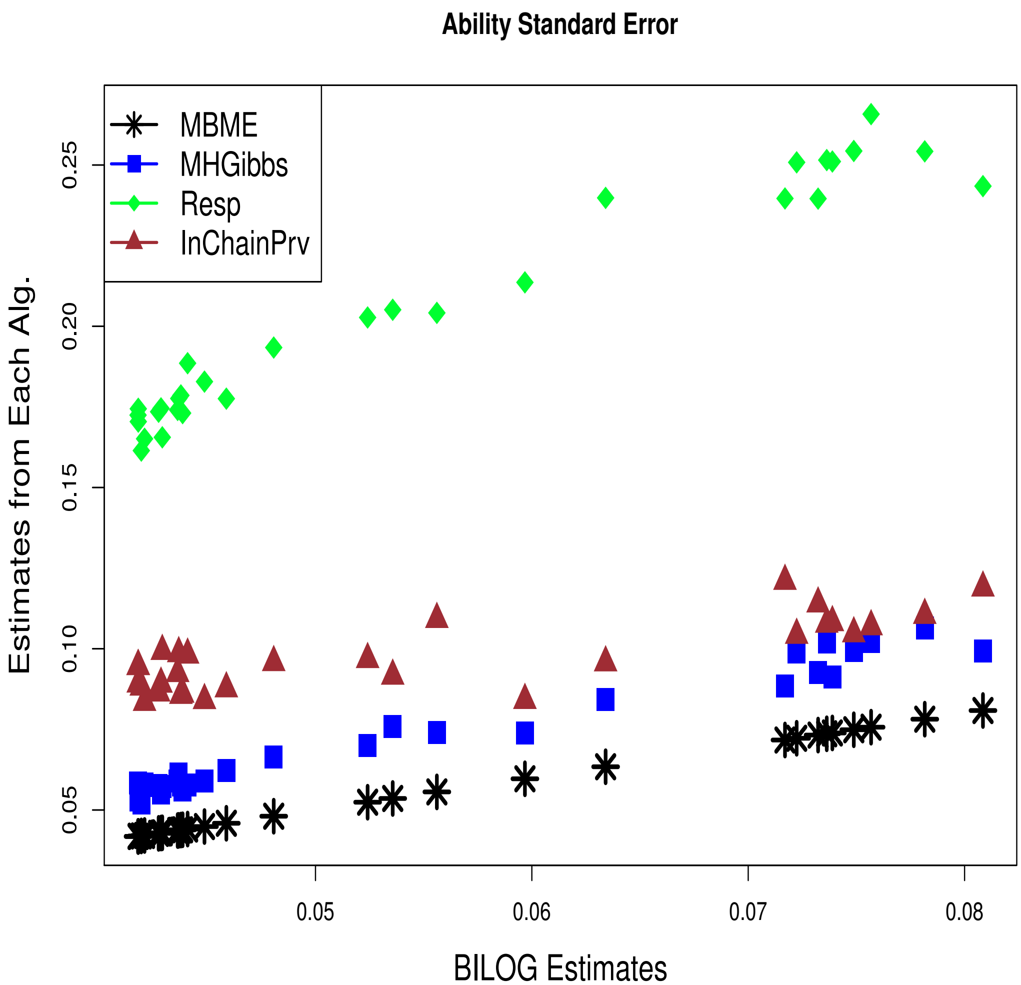

Figure 5.35 Comparison of SE for the item difficulty among MHGibbs with response treatment between before and after in-chain centered procedure and MH/Gibbs (Rasch model)

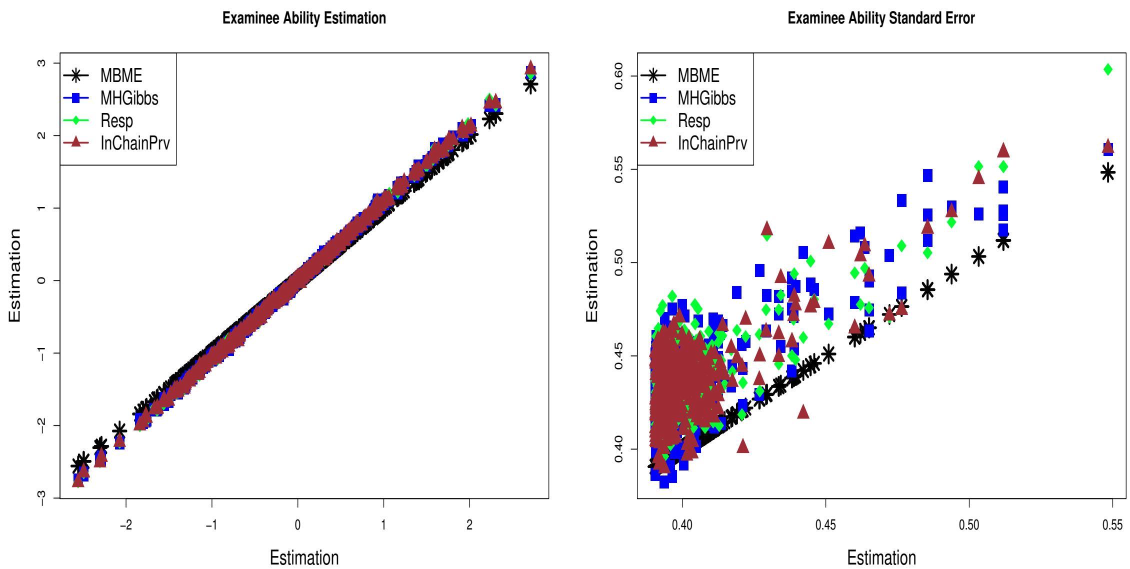

Figure 5.36 Comparison of estimates and SE for the ability of examinee from MHGibbs with response treatment between before and after in-chain centered procedure and MH/Gibbs (2PL model)

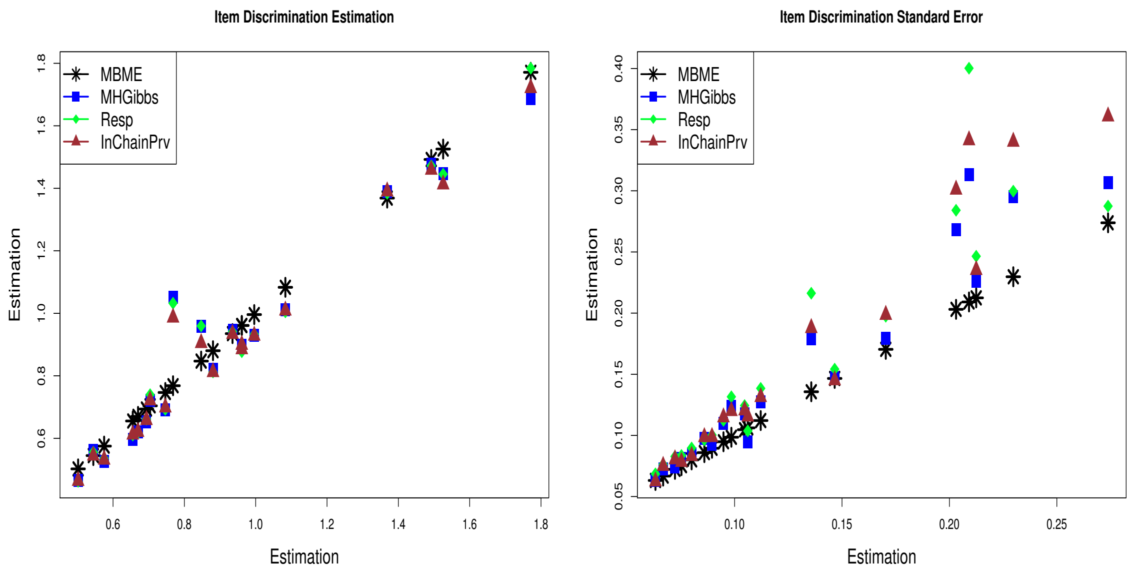

Figure 5.37 Comparison of estimates and SE for the item discrimination from MHGibbs with response treatment between before and after in-chain centered procedure, and MH/Gibbs (2PL model)

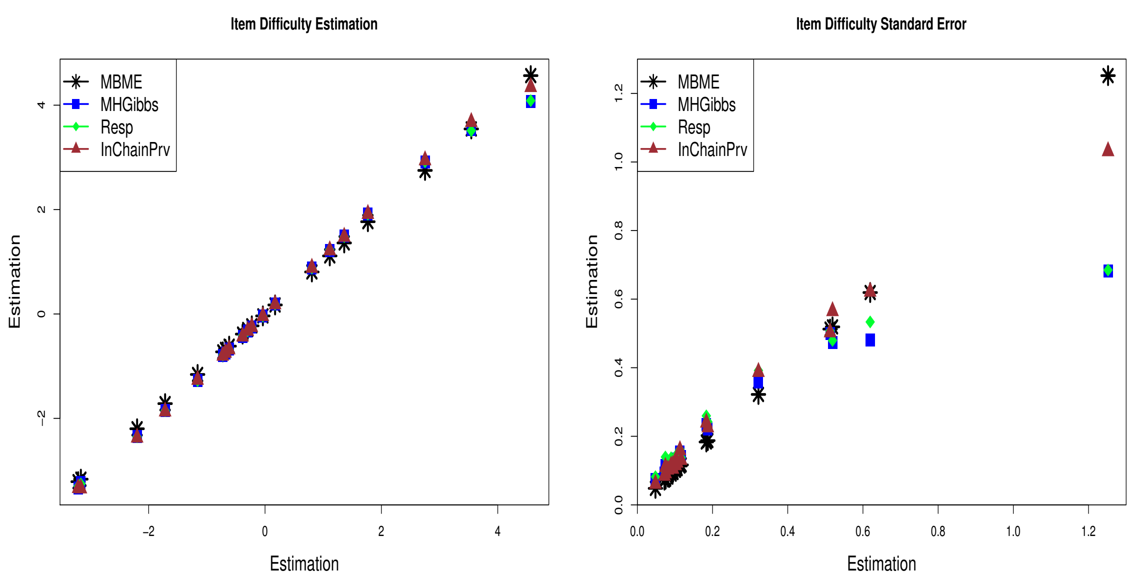

Figure 5.38 Comparison of estimates and SE for the item difficulty from MHGibbs with response treatment between before and after in-chain centered procedure, and MH/Gibbs (2PL model)

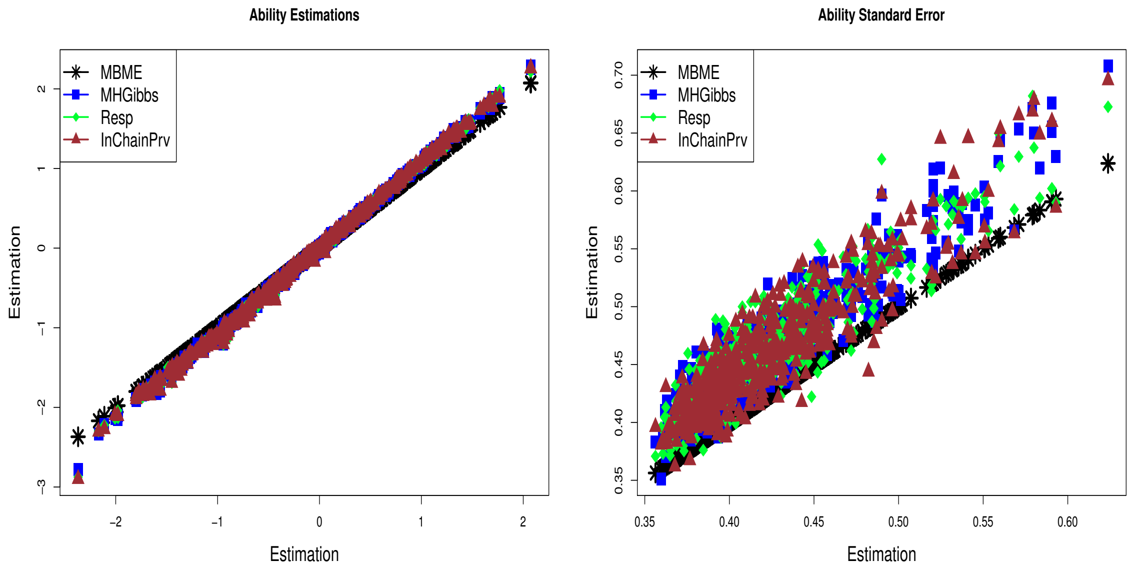

Figure 5.39 Comparison of estimates and SE for the ability of examinee from MHGibbs with response treatment between before and after in-chain centered procedure and MH/Gibbs (3PL model)

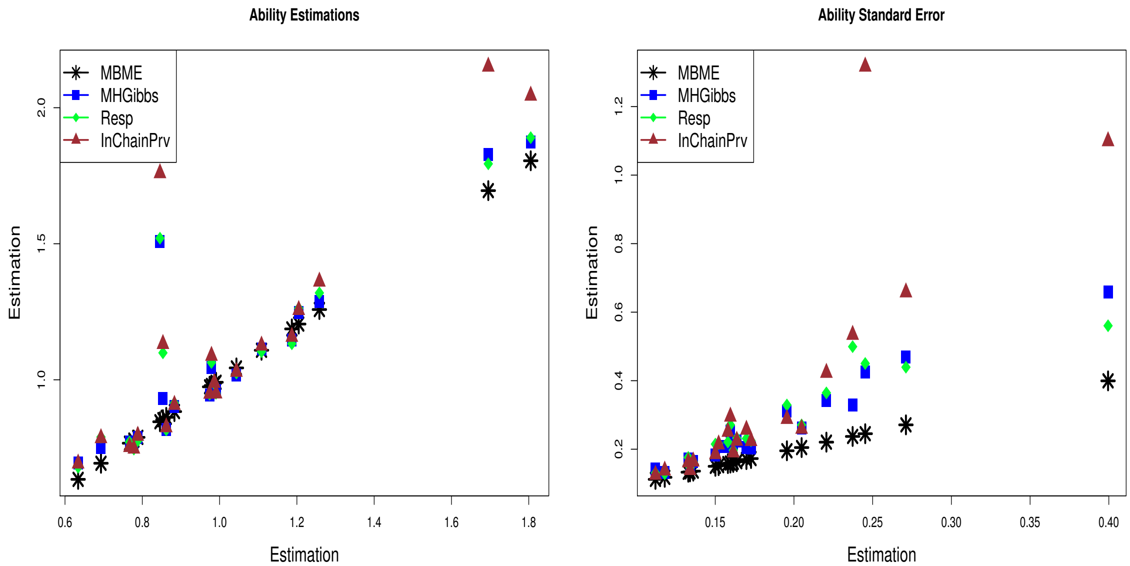

Figure 5.40 Comparison of estimates and SE for the item discrimination from MHGibbs with response treatment between before and after in-chain centered procedure, and MH/Gibbs (3PL model)

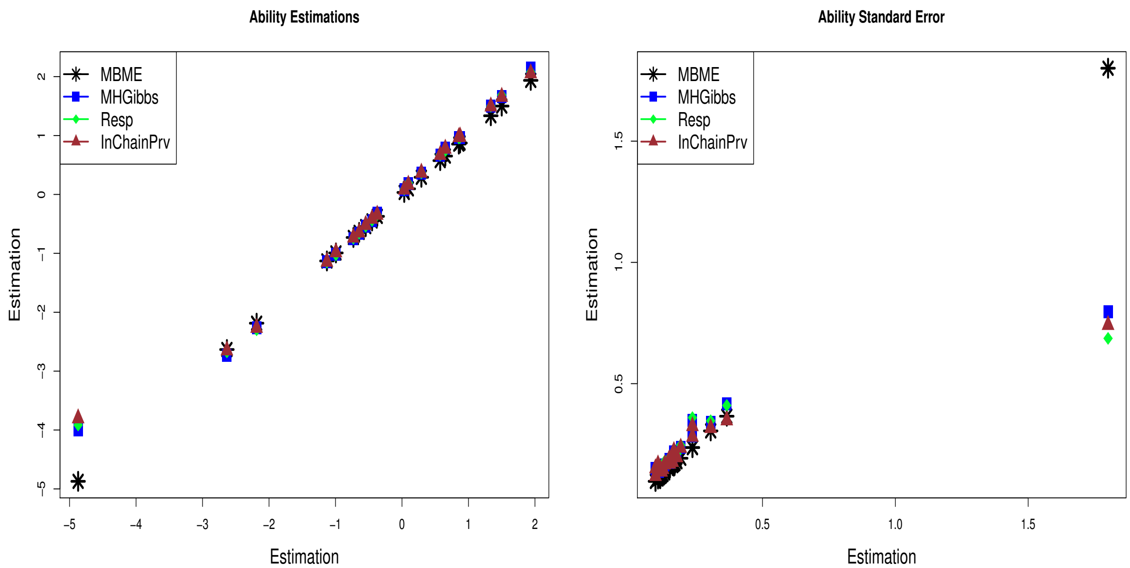

Figure 5.41 Comparison of estimates and SE for the item difficulty from MHGibbs with response treatment between before and after in-chain centered procedure, and MH/Gibbs (3PL model)

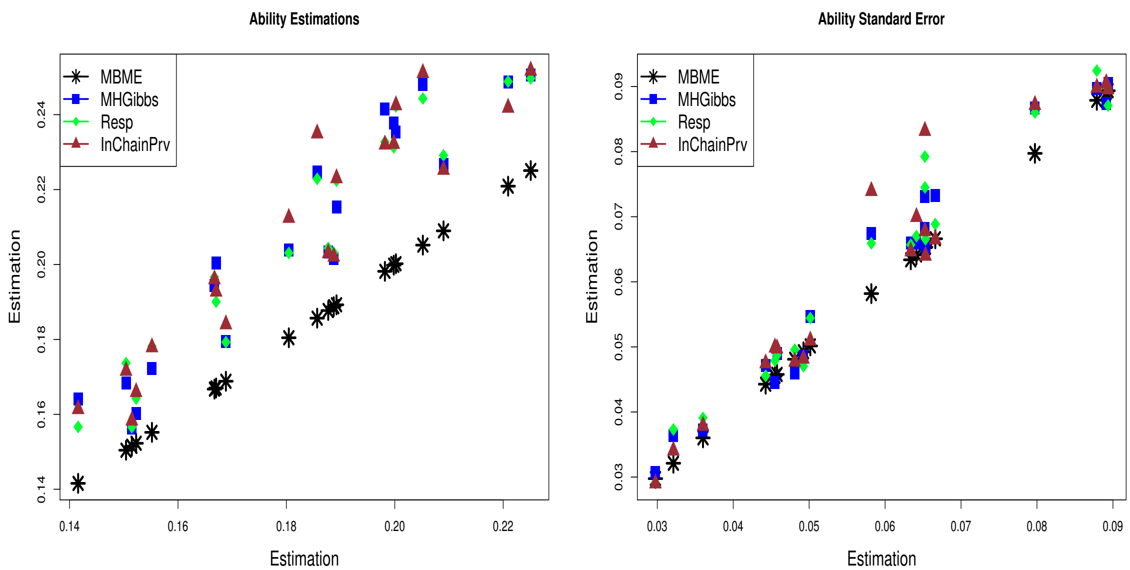

Figure 5.42 Comparison of estimates and SE for the item guessing from MHGibbs with response treatment between before and after in-chain centered procedure, and MH/Gibbs (3PL model)

Chapter 6

Conclusion and Future Study

The Marginalized Bayes modal estimation (MBME) has been widely used for IRT because of several reasons: MBME does not require long computation time to estimate point estimates for abilities of examinees and item parameters, it is very easy to apply because of the availability of off-the-shelf softwares (e.g. BILOG-MG). However, MBME can provide only point estimators instead of the whole distribution and this is one reason that the MH/Gibbs algorithm has become more popular. Even though MH/Gibbs become more popular method, there are some problems which makes hard to compare estimates between MBME and MH/Gibbs. First, MH/Gibbs provides different ability estimates for examinees with identical responses while MBME gives identical ability estimates. Second, the standard errors from MBME is smaller than those from MH/Gibbs.

I proposed a new method named In-Chain centered MCMC with same response treatment and this refines MH/Gibbs methods to handle the two previously mentioned problems simultaneously. Based on the results of the simulation, In-Chain center MCMC with same response treatment reduced standard errors for item parameters. For instance, Figure D.1 and D.2 shows estimations and standard error for the item difficulty with 1000 examinees and 30 items. All nine estimations are identical for item difficulty. However, there are significant differences in standard errors. The standard errors from MH/Gibbs are greater than the those from MBME. The difference is even greater when the same response treatment is applied. By implementing a in-chain centered algorithm, differences between MCMC and MBME are significantly reduced. Among four in-chain centered MCMCs, standard errors from all three algorithms which used new sets of abilities to center the item parameters were more accurate than using previous abilities to center the item parameters. Similar patterns are observed for 2PL and 3PL models as well although there are some items which have even smaller standard errors than MBME. Overall, applying in-chain centered algorithm makes smaller standard error differences from MBME regardless of using new sets or previous abilities to center the item parameters.

Table 6.1 Legend for simulations

Legend	Explanation
MBME	Estimations from Bilog-MG
MHGibbs	Estimations from MH/Gibbs
Resp	Estimations from MH/Gibbs with response treatment
InChainPrv	In-Chain centered MCMC using previous ability
InChainNew1	In-Chain centered MCMC using new ability (1sets)
InChainNew3	In-Chain centered MCMC using new ability (3sets)
InChainNew5	In-Chain centered MCMC using new ability (5sets)
PostHocMHGibbs	Post Hoc estimates from MHGibbs estimates
PostHocResp	Post Hoc estimates from MHGibbs with response treatment

Based on the result of simulations, in-chain centered MCMC makes up for the weakness of MCMC to compare the estimates from MBME. Especially, in-chain centered MCMC works well for Rasch model. All MBME, MH/Gibbs, MH/Gibbs with response treatment, and in-chain centered MCMC gives identical estimates for item difficulty (6.1). However, there is significant improvement in SE even after adapting response treatment (6.2). After applying same response treatment to MH/Gibbs algorithm, standard errors for item difficulty are significantly increased. By implement in-chain centered MCMC to MH/Gibbs with response treatment, standard errors for item difficulty are very closed to standard error from MBME. When the size of the test and number of examinees are increased, in-chain centered MCMC work even better. However, for a small size test with a small number of examinees, in-chained centered MCMC using previous ability provides even bigger standard errors for the

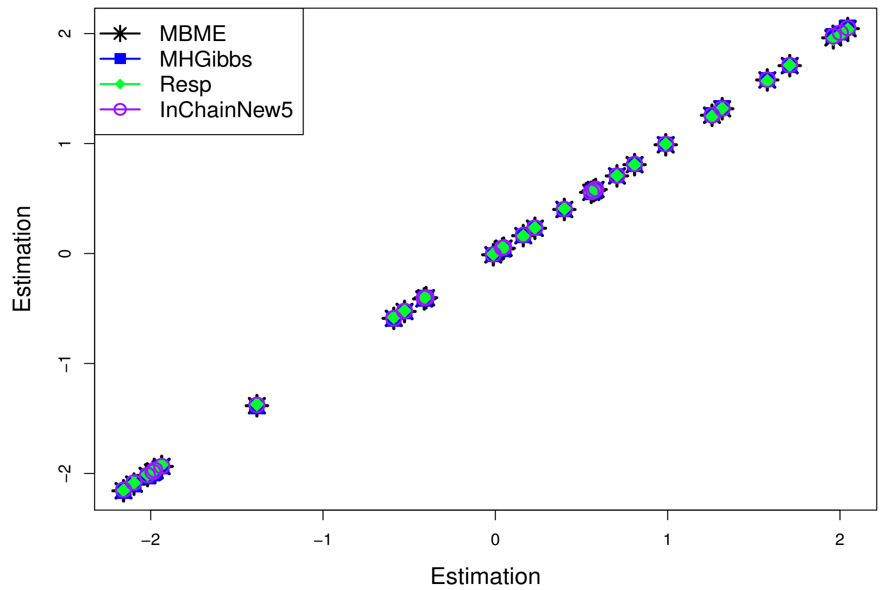

Figure 6.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items)

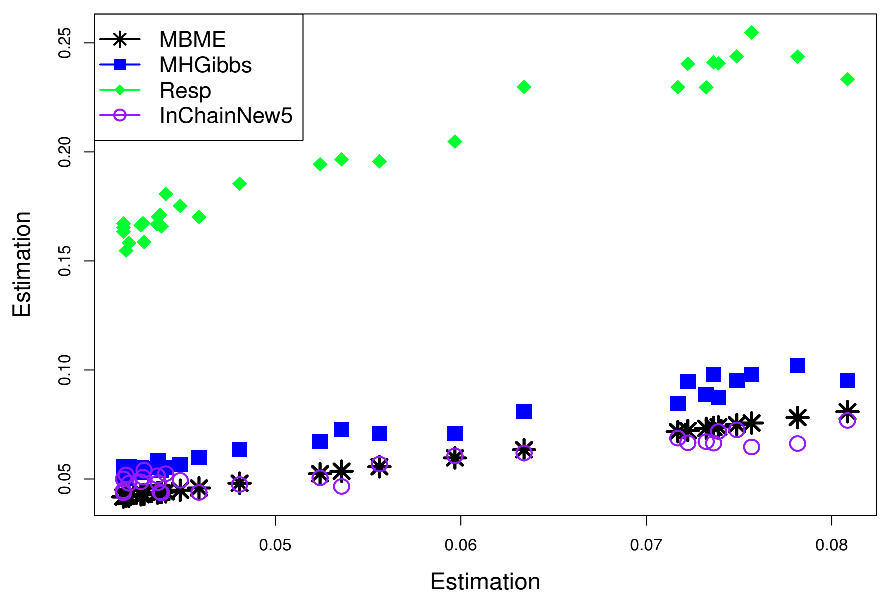

Figure 6.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items)

discrimination and difficulty parameters (Figure A.8 and A.10) for the 3PL model.

Despite some problematic conditions to in-chain centered MCMC (e.g. small size of tests), this algorithm provides whole posterior distributions for abilities and item parameters which are comparable with the standard error from MBME without losing accuracy of the point estimator. This advantage of in-chain centered MCMC makes it possible to compare inferences from two different statistical methods: MBME and MCMC.

For this dissertation, only BILOG was used for MBME but there are some other options such as PASCALE and MULTILOG (Du Toit, 2003) that could be examined. For the MCMC simulation, there is an alternative choice of software to Fortran; Stan is a probabilistic program language (Carpenter et al., 2016) that can generate the Markov chain for the item response models in a comparatively shorter time. The size of the MCMC simulation in this dissertation was limited because of resources, but if Stan can solve the resource problem, then a bigger size test with more examinees can be compared between MBME and MCMC. The possibility of increasing the size of tests can provide more understanding of how the size of items and number of examinees affect results between MBME and MCMC.

Bibliography

J. H. Albert. Bayesian estimation of normal ogive item response curves using gibbs sampling. *Journal of Educational and Behavioral Statistics*, 17(3):251–269, 1992.

F. B. Baker. An investigation of the item parameter recovery characteristics of a gibbs sampling procedure. *Applied Psychological Measurement*, 22(2):153–169, 1998.

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

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

A. Birnbaum. Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*, 1968.

R. D. Bock and M. Aitkin. Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, 46(4):443–459, 1981.

R. D. Bock and M. Lieberman. Fitting a response model forn dichotomously scored items. *Psychometrika*, 35(2):179–197, 1970.

E. T. Bradlow, H. Wainer, and X. Wang. A bayesian random effects model for testlets. *Psychometrika*, 64(2):153–168, 1999.

L. Cai, S. Du Toit, and D. Thissen. Irtpro: Flexible, multidimensional, multiple categorical irt modeling [computer software]. *Chicago, IL: Scientific Software International*, 2011.

B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *J Stat Softw*, 2016.

S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

M. Du Toit. *IRT from SSI: Bilog-MG, multilog, parscale, testfact*. Scientific Software International, 2003.

G. H. Fischer and I. W. Molenaar. *Rasch models: Foundations, recent developments, and applications*. Springer Science & Business Media, 2012.

A. Gelman, J. Carlin, H. S. Stern, and D. Rubin. Bayesian data analysis. *Chapman&Hall, London*, 1995.

A. Gelman, G. Roberts, and W. Gilks. Efficient metropolis jumping hules. *Bayesian statistics*, 5(599-608):42, 1996.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

L. Hendrix. *Fast EM Based Posterior Approximation for IRT Item Parameters*. PhD thesis, UNIVERSITY OF SOUTH CAROLINA, 2011.

J.-S. Kim and D. M. Bolt. Estimating item response theory models using markov chain monte carlo methods. *Educational Measurement: Issues and Practice*, 26 (4):38–51, 2007.

J. S. Liu, W. H. Wong, and A. Kong. Covariance structure of the gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81(1):27–40, 1994.

F. Lord. A theory of test scores. *Psychometric monographs*, 1952.

F. M. Lord. *Applications of item response theory to practical testing problems.* Routledge, 1980.

A. D. Mead, S. B. Morris, and D. L. Blitz. Open-source irt: A comparison of bilog-mg and icl features and item parameter recovery. *Unpublished manuscript]. Chicago: Institute of Psychology, Illinois Institute of Technology*, 2007.

N. Metropolis, A. W. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

R. Mislevy and R. Bock. Bilog: Item and test scoring with binary logistic models [computer program]. *Mooresville, IN: Scientific Software*, 1983.

R. J. Mislevy. Bayes modal estimation in item response models. *Psychometrika*, 51 (2):177–195, 1986.

R. J. Mislevy and M. L. Stocking. A consumer's guide to logist and bilog. *Applied psychological measurement*, 13(1):57–75, 1989.

R. J. Mokken. *A theory and procedure of scale analysis: With applications in political research*, volume 1. Walter de Gruyter, 1971.

P. Müller. *A generic approach to posterior integration and Gibbs sampling.* Purdue University, Department of Statistics, 1991.

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

I. Partchev, M. I. Partchev, and M. Suggests. The irtoys package, 2008.

R. J. Patz and B. W. Junker. Applications and extensions of mcmc in irt: Multiple item types, missing data, and rated responses. *Journal of educational and behavioral statistics*, 24(4):342–366, 1999a.

R. J. Patz and B. W. Junker. A straightforward approach to markov chain monte carlo methods for item response models. *Journal of educational and behavioral Statistics*, 24(2):146–178, 1999b.

G. Rasch. Probabilistic models for some intelligence and achievement tests. *Copenhagen: Danish Institute for Educational Research*, 1960.

M. D. Reckase. The difficulty of test items that measure more than one ability. *Applied Psychological Measurement*, 9(4):401–412, 1985.

G. O. Roberts, A. Gelman, W. R. Gilks, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1): 110–120, 1997.

A. A. Rupp. Item response modeling with bilog-mg and multilog for windows. *International Journal of Testing*, 3(4):365–384, 2003.

S. Sinharay. Experiences with markov chain monte carlo convergence assessment in two psychometric examples. *Journal of Educational and Behavioral Statistics*, 29 (4):461–488, 2004.

A. F. Smith and G. O. Roberts. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–23, 1993.

L. Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

E. Vanem, E. M. Bitner-Gregersen, and C. K. Wikle. *Bayesian Hierarchical Space-Time Models with Application to Significant Wave Height*. Springer, 2013.

A. von Davier. *Statistical models for test equating, scaling, and linking*. Springer Science & Business Media, 2010.

M. S. Wingersky. Logist: A program for computing maximum likelihood procedures for logistic test models. *Applications of item response theory*, pages 45–56, 1983.

M. S. Wingersky, M. A. Barton, and F. M. Lord. *LOGIST user's guide: LOGIST 5, version 1.0*. Educational Testing Service, 1982.

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

M. Zimowski, E. Muraki, R. Mislevy, and R. Bock. Bilog-mg [computer program]. lincolnwood, il: Scientific software international, 2003.

Appendix A

Result Tables and Graphs for 500 examinees and 15 items

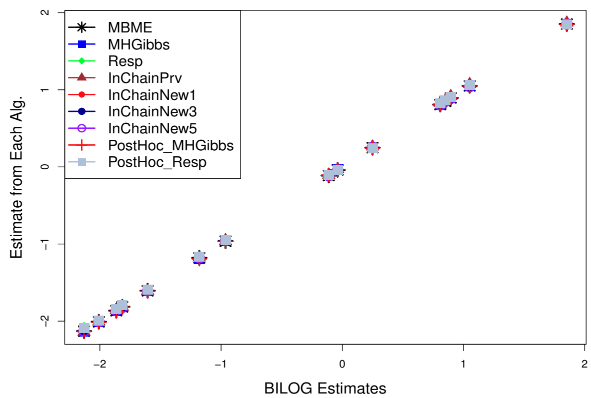

Figure A.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 500 examinees and 15 items)

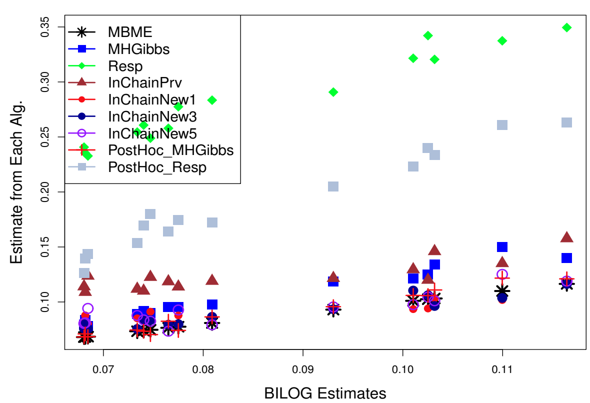

Figure A.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 500 examinees and 15 items)

Table A.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.851	0.863	0.856	0.862	0.831	0.841	0.850	0.863
2	-2.131	-2.133	-2.073	-2.140	-2.130	-2.129	-2.129	-2.133
3	-1.815	-1.817	-1.779	-1.815	-1.818	-1.827	-1.823	-1.818
4	-0.038	-0.034	-0.038	-0.030	-0.055	-0.041	-0.040	-0.034
5	1.051	1.050	1.056	1.073	1.035	1.042	1.040	1.050
6	0.249	0.243	0.239	0.262	0.230	0.248	0.257	0.243
7	0.895	0.894	0.898	0.918	0.882	0.883	0.886	0.894
8	0.808	0.806	0.821	0.830	0.796	0.796	0.802	0.806
9	-1.606	-1.612	-1.578	-1.608	-1.601	-1.613	-1.608	-1.612
10	-0.112	-0.107	-0.102	-0.104	-0.125	-0.123	-0.113	-0.107
11	-0.963	-0.964	-0.946	-0.965	-0.961	-0.965	-0.955	-0.964
12	-2.009	-2.015	-1.978	-2.007	-2.002	-2.014	-2.008	-2.014
13	-1.179	-1.194	-1.156	-1.176	-1.186	-1.185	-1.180	-1.194
14	1.852	1.845	1.832	1.869	1.825	1.846	1.858	1.845
15	-1.865	-1.873	-1.829	-1.858	-1.850	-1.859	-1.864	-1.873

Table A.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.074	0.092	0.261	0.110	0.085	0.083	0.084	0.073
2	0.116	0.140	0.349	0.158	0.116	0.116	0.119	0.121
3	0.101	0.121	0.322	0.130	0.094	0.110	0.098	0.106
4	0.068	0.080	0.241	0.114	0.082	0.076	0.080	0.068
5	0.078	0.095	0.277	0.114	0.088	0.080	0.093	0.074
6	0.068	0.078	0.233	0.124	0.080	0.076	0.094	0.069
7	0.075	0.090	0.249	0.123	0.091	0.084	0.082	0.070
8	0.073	0.089	0.254	0.112	0.085	0.075	0.088	0.075
9	0.093	0.118	0.291	0.122	0.093	0.093	0.095	0.096
10	0.068	0.084	0.235	0.109	0.088	0.080	0.081	0.071
11	0.077	0.095	0.258	0.119	0.078	0.079	0.073	0.082
12	0.110	0.150	0.337	0.135	0.102	0.104	0.125	0.122
13	0.081	0.098	0.283	0.119	0.085	0.087	0.079	0.086
14	0.103	0.125	0.342	0.120	0.094	0.106	0.104	0.106
15	0.103	0.134	0.321	0.146	0.104	0.096	0.101	0.111

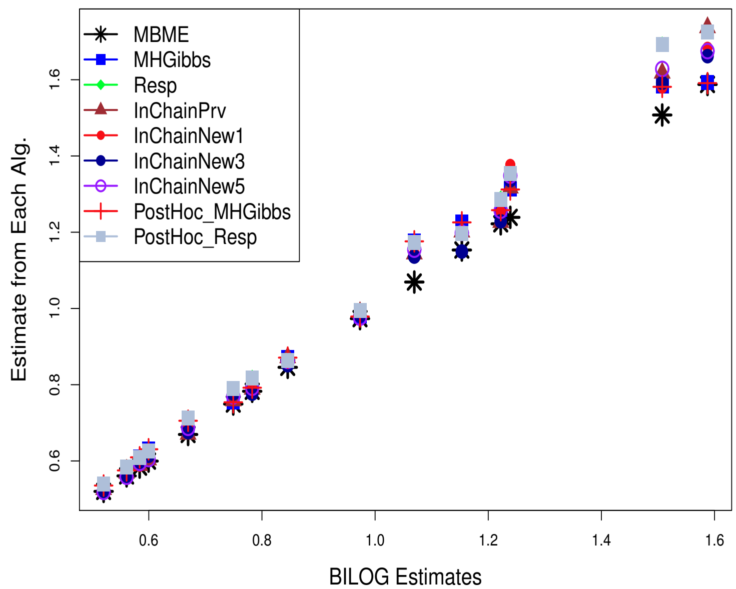

Figure A.3 Comparison of estimates for the item discrimination among algorithms (2PL model with 500 examinees and 15 items)

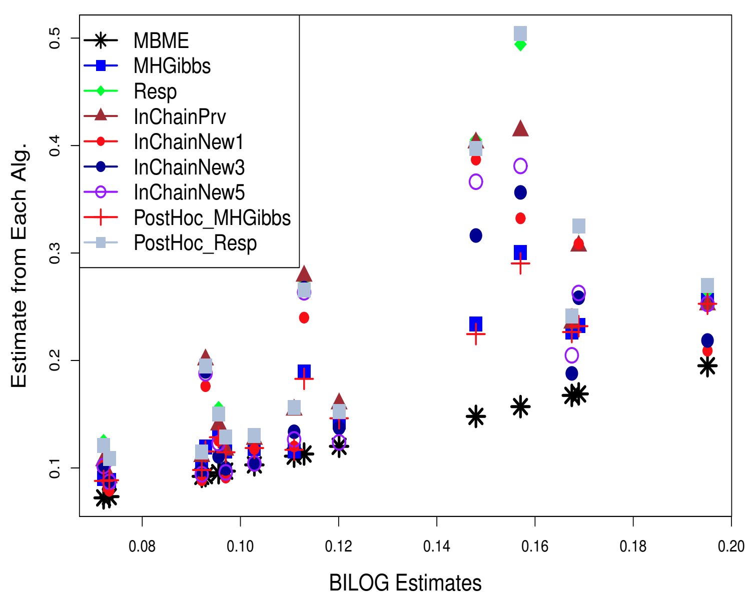

Figure A.4 Comparison of SE for the item discrimination among algorithms (2PL model with 500 examinees and 15 items)

Table A.3 Comparison of estimates for the item discrimination among algorithms (2PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	1.588	1.593	1.731	1.737	1.684	1.662	1.675	1.591
2	0.669	0.706	0.714	0.670	0.691	0.676	0.687	0.705
3	0.973	0.980	0.997	0.990	0.993	0.982	0.976	0.979
4	0.600	0.631	0.627	0.603	0.618	0.613	0.605	0.631
5	0.561	0.576	0.586	0.569	0.586	0.557	0.559	0.575
6	0.584	0.610	0.610	0.589	0.586	0.592	0.595	0.609
7	1.069	1.177	1.174	1.144	1.145	1.136	1.154	1.176
8	1.239	1.313	1.356	1.317	1.377	1.337	1.348	1.312
9	1.508	1.583	1.695	1.618	1.579	1.592	1.628	1.581
10	0.846	0.871	0.866	0.871	0.869	0.851	0.862	0.871
11	0.520	0.536	0.540	0.526	0.525	0.521	0.519	0.535
12	1.153	1.226	1.198	1.201	1.202	1.149	1.198	1.226
13	1.222	1.259	1.290	1.226	1.255	1.229	1.256	1.258
14	0.749	0.754	0.791	0.758	0.773	0.777	0.769	0.754
15	0.783	0.793	0.820	0.789	0.791	0.777	0.792	0.792

Table A.4 Comparison of SE for the item discrimination among algorithms (2PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.148	0.234	0.405	0.403	0.387	0.316	0.366	0.225
2	0.103	0.118	0.130	0.127	0.117	0.105	0.104	0.118
3	0.093	0.120	0.200	0.201	0.176	0.190	0.188	0.115
4	0.097	0.116	0.130	0.117	0.091	0.100	0.096	0.115
5	0.072	0.090	0.125	0.107	0.103	0.098	0.102	0.088
6	0.092	0.099	0.115	0.111	0.088	0.098	0.094	0.098
7	0.195	0.257	0.264	0.252	0.209	0.219	0.253	0.253
8	0.169	0.233	0.324	0.307	0.309	0.258	0.263	0.232
9	0.157	0.300	0.494	0.414	0.332	0.356	0.381	0.290
10	0.120	0.143	0.159	0.160	0.138	0.137	0.124	0.146
11	0.073	0.088	0.107	0.091	0.079	0.087	0.087	0.089
12	0.167	0.226	0.235	0.235	0.243	0.188	0.205	0.226
13	0.113	0.189	0.274	0.279	0.240	0.267	0.263	0.183
14	0.111	0.116	0.155	0.154	0.120	0.134	0.127	0.117
15	0.096	0.132	0.156	0.140	0.125	0.111	0.123	0.129

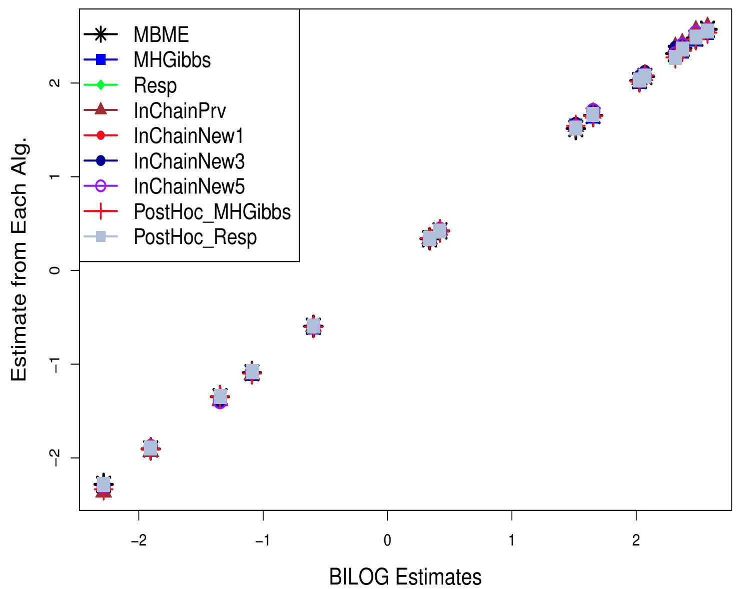

Figure A.5 Comparison of estimates for the item difficulty among algorithms (2PL model with 500 examinees and 15 items)

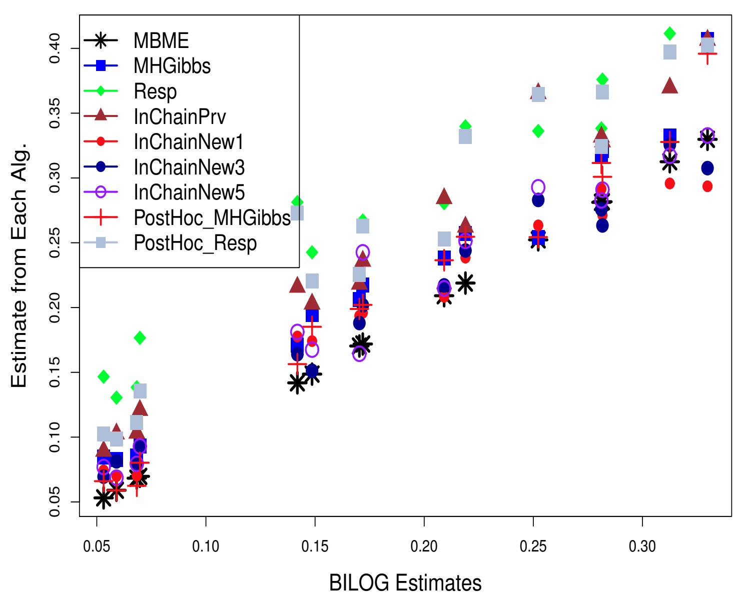

Figure A.6 Comparison of SE for the item difficulty among algorithms (2PL model with 500 examinees and 15 items)

Table A.5 Comparison of estimates for the item difficulty among algorithms (2PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-0.596	-0.603	-0.593	-0.594	-0.603	-0.595	-0.596	-0.603
2	2.314	2.273	2.267	2.380	2.321	2.372	2.304	2.274
3	0.338	0.345	0.342	0.339	0.338	0.348	0.339	0.345
4	2.479	2.467	2.484	2.557	2.469	2.498	2.531	2.467
5	-1.347	-1.346	-1.344	-1.388	-1.340	-1.387	-1.392	-1.346
6	2.370	2.343	2.358	2.418	2.409	2.399	2.398	2.344
7	2.573	2.538	2.552	2.592	2.545	2.564	2.564	2.537
8	-1.904	-1.904	-1.890	-1.930	-1.875	-1.881	-1.885	-1.904
9	-1.090	-1.098	-1.079	-1.108	-1.093	-1.099	-1.085	-1.098
10	2.070	2.080	2.075	2.082	2.086	2.116	2.077	2.081
11	1.653	1.649	1.664	1.680	1.694	1.699	1.699	1.649
12	2.026	2.016	2.031	2.038	2.031	2.066	2.033	2.017
13	0.422	0.422	0.423	0.431	0.436	0.430	0.431	0.422
14	-2.283	-2.333	-2.276	-2.362	-2.276	-2.290	-2.296	-2.334
15	1.514	1.543	1.521	1.538	1.534	1.562	1.527	1.543

Table A.6 Comparison of SE for the item difficulty among algorithms (2PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.053	0.085	0.147	0.089	0.074	0.069	0.077	0.066
2	0.281	0.318	0.338	0.332	0.292	0.276	0.283	0.312
3	0.068	0.085	0.138	0.103	0.070	0.080	0.079	0.062
4	0.330	0.407	0.407	0.406	0.294	0.308	0.333	0.396
5	0.172	0.217	0.267	0.236	0.196	0.202	0.243	0.202
6	0.313	0.332	0.411	0.370	0.296	0.326	0.317	0.328
7	0.282	0.321	0.376	0.328	0.271	0.263	0.291	0.301
8	0.142	0.171	0.281	0.216	0.178	0.164	0.181	0.156
9	0.070	0.093	0.177	0.121	0.095	0.093	0.093	0.080
10	0.209	0.238	0.281	0.284	0.208	0.217	0.215	0.237
11	0.219	0.257	0.340	0.262	0.239	0.244	0.251	0.255
12	0.170	0.206	0.230	0.218	0.194	0.188	0.164	0.199
13	0.059	0.083	0.131	0.103	0.070	0.081	0.069	0.059
14	0.252	0.254	0.336	0.365	0.263	0.283	0.293	0.254
15	0.149	0.195	0.243	0.203	0.174	0.151	0.168	0.185

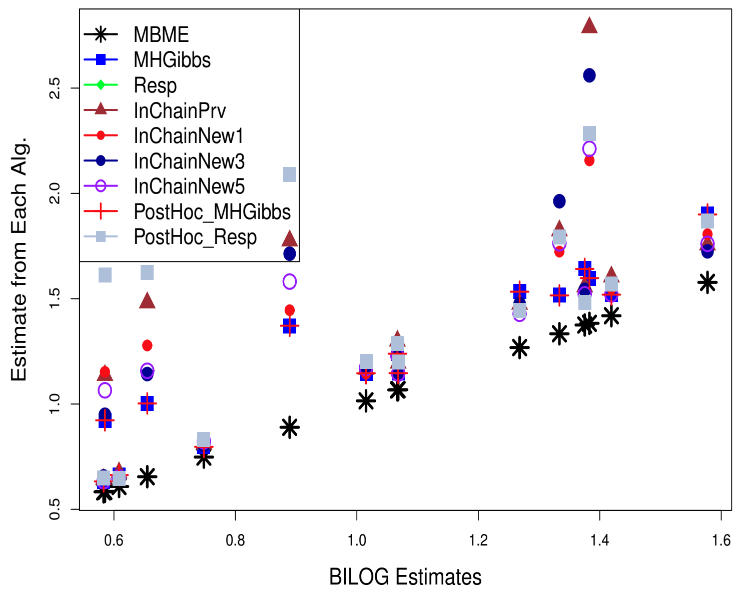

Figure A.7 Comparison of estimates for the item discrimination among algorithms (3PL model with 500 examinees and 15 items)

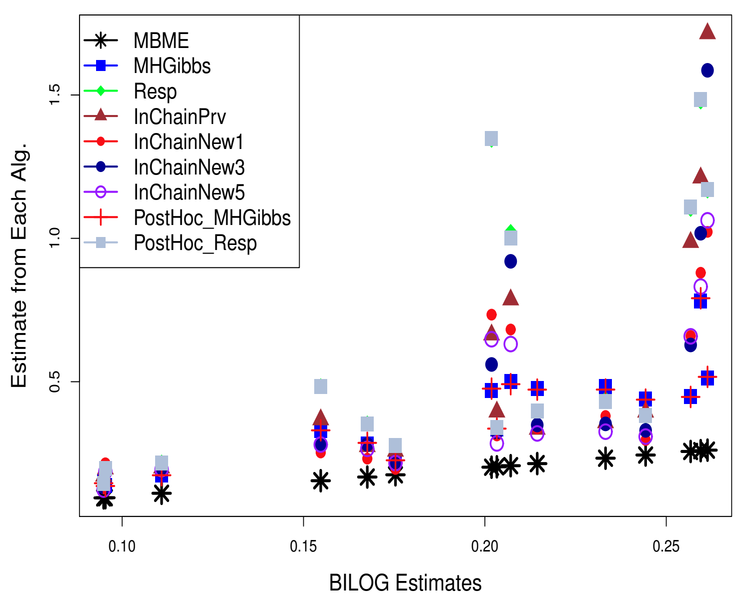

Figure A.8 Comparison of SE for the item discrimination among algorithms (3PL model with 500 examinees and 15 items)

Table A.7 Comparison of estimates for the item discrimination among algorithms (3PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	1.067	1.239	1.290	1.300	1.200	1.251	1.229	1.239
2	0.583	0.633	0.648	0.643	0.663	0.657	0.634	0.632
3	1.383	1.597	2.284	2.788	2.157	2.560	2.212	1.598
4	0.748	0.797	0.834	0.811	0.823	0.805	0.823	0.797
5	1.577	1.903	1.875	1.756	1.806	1.725	1.760	1.900
6	1.375	1.642	1.484	1.557	1.503	1.542	1.521	1.641
7	1.068	1.147	1.201	1.195	1.137	1.149	1.145	1.146
8	1.268	1.535	1.446	1.474	1.462	1.475	1.428	1.533
9	0.585	0.922	1.613	1.136	1.153	0.949	1.065	0.923
10	1.419	1.520	1.571	1.604	1.548	1.567	1.521	1.519
11	0.655	1.003	1.624	1.481	1.278	1.142	1.158	1.003
12	0.608	0.664	0.647	0.674	0.649	0.657	0.651	0.663
13	1.015	1.146	1.205	1.166	1.145	1.178	1.171	1.146
14	1.334	1.518	1.798	1.823	1.723	1.963	1.763	1.516
15	0.889	1.370	2.090	1.775	1.445	1.714	1.581	1.372

Table A.8 Comparison of SE for the item discrimination among algorithms (3PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.155	0.330	0.487	0.368	0.253	0.281	0.281	0.330
2	0.095	0.138	0.195	0.196	0.216	0.191	0.187	0.136
3	0.261	0.513	1.164	1.714	1.023	1.586	1.063	0.517
4	0.111	0.176	0.221	0.200	0.202	0.188	0.194	0.174
5	0.233	0.484	0.441	0.358	0.380	0.353	0.326	0.473
6	0.244	0.440	0.376	0.395	0.296	0.330	0.309	0.437
7	0.175	0.228	0.278	0.258	0.196	0.220	0.216	0.226
8	0.214	0.476	0.398	0.335	0.337	0.349	0.320	0.472
9	0.202	0.469	1.341	0.664	0.734	0.560	0.648	0.476
10	0.203	0.335	0.341	0.395	0.313	0.324	0.285	0.336
11	0.257	0.449	1.100	0.986	0.660	0.628	0.658	0.447
12	0.095	0.150	0.143	0.171	0.135	0.133	0.124	0.146
13	0.168	0.283	0.357	0.274	0.232	0.281	0.266	0.287
14	0.207	0.501	1.025	0.786	0.682	0.920	0.631	0.491
15	0.259	0.782	1.475	1.211	0.880	1.018	0.832	0.791

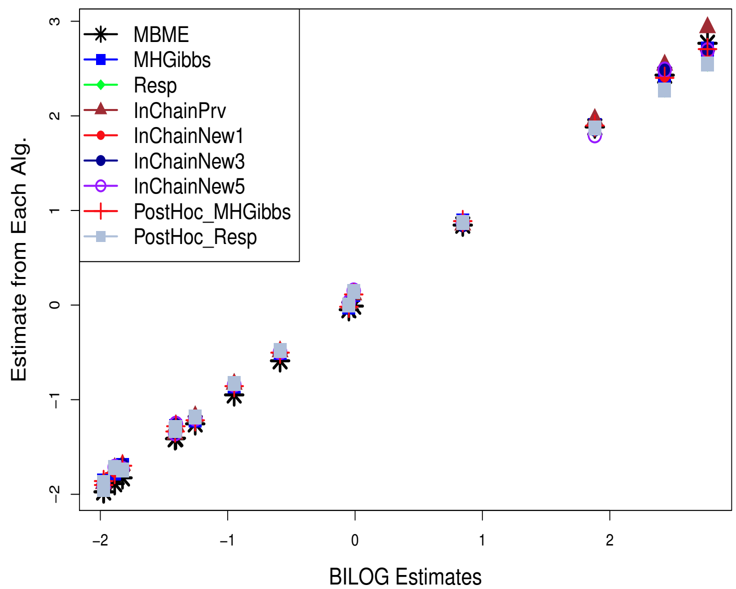

Figure A.9 Comparison of estimates for the item difficulty among algorithms (3PL model with 500 examinees and 15 items)

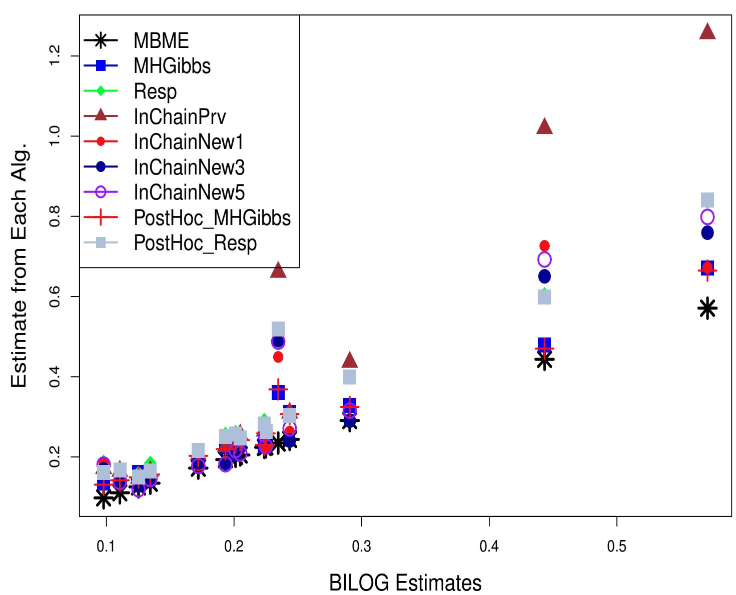

Figure A.10 Comparison of SE for the item difficulty among algorithms (3PL model with 500 examinees and 15 items)

Table A.9 Comparison of estimates for the item difficulty among algorithms (3PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-0.949	-0.856	-0.827	-0.835	-0.841	-0.827	-0.838	-0.857
2	-0.011	0.113	0.143	0.110	0.141	0.131	0.152	0.112
3	0.846	0.887	0.873	0.848	0.872	0.846	0.875	0.887
4	-0.589	-0.503	-0.485	-0.491	-0.478	-0.492	-0.493	-0.503
5	-1.413	-1.336	-1.324	-1.363	-1.342	-1.363	-1.353	-1.336
6	-1.974	-1.859	-1.943	-1.884	-1.898	-1.909	-1.898	-1.860
7	-1.974	-1.901	-1.874	-1.880	-1.917	-1.902	-1.916	-1.902
8	-1.826	-1.698	-1.745	-1.691	-1.720	-1.693	-1.734	-1.698
9	2.431	2.402	2.269	2.538	2.415	2.489	2.487	2.402
10	-1.255	-1.217	-1.185	-1.178	-1.196	-1.193	-1.198	-1.217
11	2.767	2.705	2.542	2.934	2.618	2.713	2.690	2.706
12	-1.407	-1.281	-1.293	-1.271	-1.302	-1.270	-1.257	-1.281
13	-1.887	-1.773	-1.714	-1.725	-1.764	-1.714	-1.713	-1.775
14	-0.049	-0.018	-0.003	0.017	0.024	0.036	0.025	-0.018
15	1.882	1.896	1.868	1.964	1.896	1.858	1.797	1.897

Table A.10 Comparison of SE for the item difficulty among algorithms (3PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.172	0.202	0.215	0.206	0.179	0.186	0.177	0.203
2	0.224	0.228	0.292	0.265	0.258	0.262	0.256	0.230
3	0.098	0.135	0.188	0.171	0.185	0.169	0.181	0.131
4	0.205	0.240	0.259	0.255	0.219	0.225	0.209	0.242
5	0.125	0.161	0.160	0.146	0.141	0.129	0.119	0.149
6	0.193	0.225	0.257	0.226	0.183	0.185	0.182	0.219
7	0.225	0.260	0.266	0.239	0.221	0.244	0.226	0.259
8	0.201	0.249	0.262	0.227	0.221	0.206	0.217	0.237
9	0.443	0.479	0.605	1.019	0.726	0.650	0.692	0.470
10	0.134	0.165	0.185	0.164	0.141	0.139	0.145	0.156
11	0.571	0.671	0.837	1.256	0.672	0.759	0.798	0.665
12	0.291	0.329	0.399	0.436	0.294	0.290	0.315	0.325
13	0.244	0.310	0.307	0.310	0.263	0.241	0.271	0.307
14	0.111	0.148	0.171	0.165	0.142	0.140	0.137	0.142
15	0.235	0.360	0.506	0.661	0.449	0.491	0.488	0.369

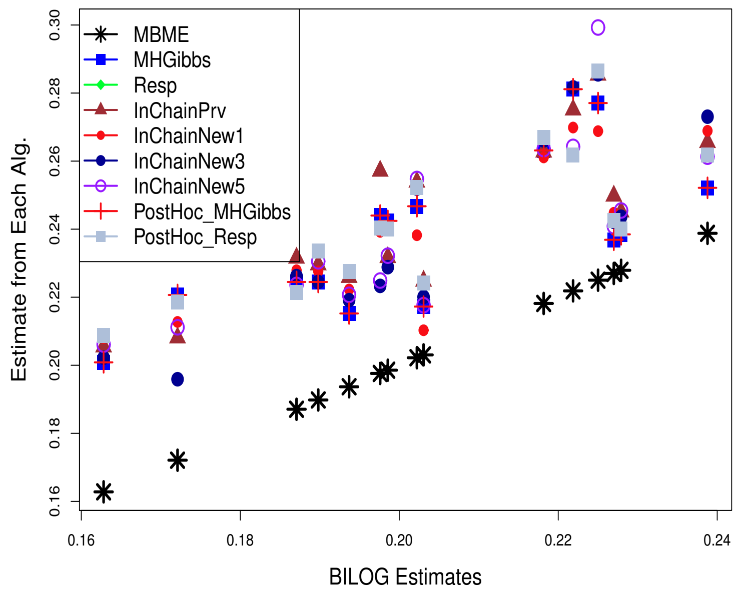

Figure A.11 Comparison of estimates for the item guessing among algorithms (3PL model with 500 examinees and 15 items)

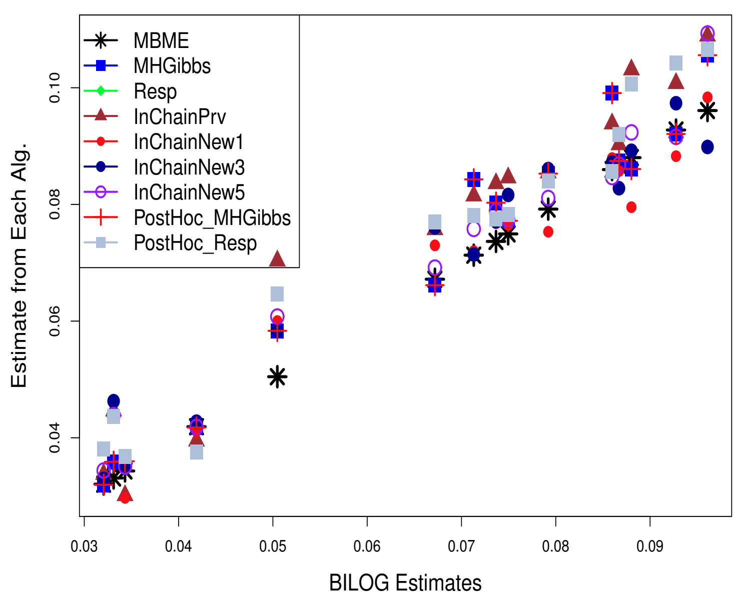

Figure A.12 Comparison of SE for the item guessing among algorithms (3PL model with 500 examinees and 15 items)

Table A.11 Comparison of estimates for the item guessing among algorithms (3PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.218	0.263	0.267	0.263	0.261	0.266	0.264	0.263
2	0.163	0.201	0.209	0.205	0.208	0.202	0.206	0.201
3	0.228	0.238	0.240	0.245	0.243	0.244	0.245	0.238
4	0.187	0.224	0.221	0.232	0.228	0.226	0.224	0.224
5	0.172	0.221	0.219	0.208	0.213	0.196	0.211	0.221
6	0.198	0.244	0.240	0.257	0.239	0.223	0.225	0.244
7	0.199	0.242	0.240	0.232	0.229	0.229	0.232	0.242
8	0.222	0.281	0.262	0.275	0.270	0.282	0.264	0.281
9	0.194	0.215	0.228	0.226	0.222	0.219	0.221	0.215
10	0.190	0.224	0.234	0.230	0.228	0.224	0.231	0.224
11	0.227	0.237	0.243	0.250	0.245	0.242	0.241	0.237
12	0.202	0.247	0.252	0.254	0.238	0.252	0.255	0.247
13	0.225	0.277	0.286	0.285	0.269	0.285	0.299	0.277
14	0.239	0.252	0.262	0.266	0.269	0.273	0.261	0.252
15	0.203	0.217	0.224	0.225	0.210	0.220	0.218	0.217

Table A.12 Comparison of SE for the item guessing among algorithms (3PL model with 500 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.079	0.085	0.084	0.085	0.075	0.086	0.081	0.085
2	0.067	0.066	0.077	0.076	0.073	0.076	0.069	0.066
3	0.033	0.036	0.044	0.045	0.044	0.046	0.044	0.036
4	0.075	0.077	0.078	0.085	0.076	0.082	0.077	0.077
5	0.071	0.084	0.078	0.081	0.072	0.071	0.076	0.084
6	0.086	0.099	0.086	0.094	0.088	0.087	0.085	0.099
7	0.087	0.087	0.092	0.090	0.086	0.083	0.087	0.087
8	0.093	0.092	0.104	0.101	0.088	0.097	0.092	0.092
9	0.042	0.042	0.038	0.039	0.041	0.043	0.042	0.042
10	0.074	0.080	0.077	0.084	0.077	0.077	0.079	0.080
11	0.034	0.036	0.037	0.030	0.030	0.035	0.035	0.036
12	0.088	0.086	0.101	0.103	0.080	0.089	0.092	0.086
13	0.096	0.106	0.107	0.109	0.098	0.090	0.109	0.106
14	0.050	0.058	0.065	0.070	0.060	0.058	0.061	0.058
15	0.032	0.032	0.038	0.034	0.033	0.033	0.034	0.032

Appendix B

Result Tables and Graphs for 500 examinees and 30 items

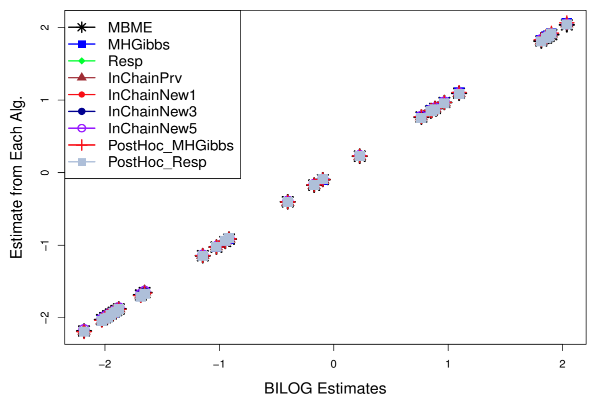

Figure B.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 500 examinees and 30 items)

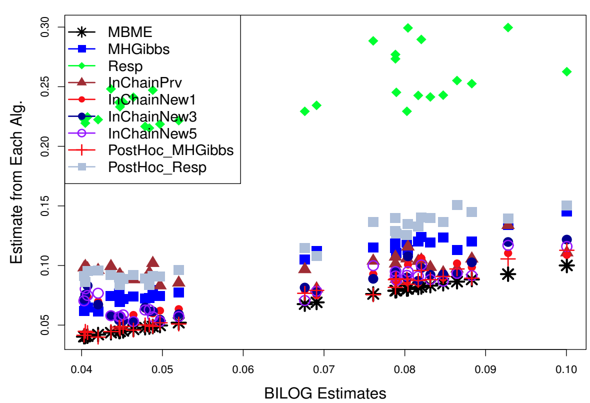

Figure B.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 500 examinees and 30 items)

Table B.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.766	0.770	0.754	0.772	0.767	0.765	0.778	0.770
2	-1.901	-1.906	-1.911	-1.888	-1.902	-1.900	-1.894	-1.906
3	-2.027	-2.035	-2.043	-2.038	-2.018	-2.029	-2.015	-2.035
4	-0.096	-0.097	-0.092	-0.098	-0.089	-0.086	-0.089	-0.097
5	1.095	1.106	1.088	1.096	1.097	1.097	1.097	1.106
6	0.227	0.229	0.229	0.225	0.231	0.236	0.236	0.229
7	0.967	0.969	0.953	0.966	0.964	0.967	0.974	0.968
8	0.885	0.889	0.872	0.882	0.886	0.887	0.893	0.889
9	-1.687	-1.695	-1.705	-1.682	-1.677	-1.678	-1.674	-1.695
10	-0.172	-0.169	-0.167	-0.172	-0.166	-0.170	-0.162	-0.169
11	-0.916	-0.913	-0.903	-0.911	-0.917	-0.916	-0.909	-0.913
12	-2.183	-2.193	-2.191	-2.181	-2.172	-2.196	-2.162	-2.193
13	-1.145	-1.145	-1.142	-1.134	-1.148	-1.149	-1.135	-1.145
14	1.857	1.874	1.858	1.866	1.852	1.852	1.874	1.874
15	-2.000	-2.007	-2.013	-1.988	-1.991	-1.995	-1.983	-2.007

Table B.2 Comparison of estimates for the item difficulty among algorithms (Rasch model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	-0.402	-0.404	-0.402	-0.403	-0.410	-0.392	-0.393	-0.404
17	1.814	1.832	1.807	1.817	1.812	1.806	1.835	1.832
18	-1.925	-1.934	-1.927	-1.911	-1.918	-1.919	-1.911	-1.935
19	0.849	0.858	0.842	0.852	0.846	0.845	0.858	0.858
20	-0.949	-0.945	-0.939	-0.937	-0.951	-0.944	-0.941	-0.945
21	-1.949	-1.955	-1.960	-1.950	-1.951	-1.943	-1.934	-1.955
22	0.856	0.855	0.850	0.851	0.857	0.856	0.867	0.855
23	-1.879	-1.883	-1.878	-1.873	-1.874	-1.883	-1.862	-1.884
24	-1.026	-1.025	-1.023	-1.018	-1.023	-1.025	-1.017	-1.025
25	-1.974	-1.979	-1.994	-1.970	-1.966	-1.969	-1.958	-1.979
26	-1.653	-1.660	-1.664	-1.640	-1.652	-1.650	-1.633	-1.660
27	1.879	1.891	1.877	1.890	1.876	1.867	1.889	1.891
28	1.857	1.874	1.856	1.871	1.849	1.849	1.872	1.875
29	2.037	2.066	2.040	2.054	2.030	2.035	2.051	2.066
30	1.902	1.923	1.916	1.914	1.899	1.897	1.918	1.923

Table B.3 Comparison of SE for the item difficulty among algorithms (Rasch model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.044	0.074	0.248	0.099	0.059	0.058	0.058	0.045
2	0.080	0.119	0.229	0.106	0.087	0.090	0.092	0.089
3	0.088	0.120	0.253	0.106	0.099	0.103	0.091	0.089
4	0.040	0.066	0.219	0.099	0.073	0.076	0.080	0.045
5	0.049	0.080	0.247	0.102	0.060	0.062	0.057	0.049
6	0.040	0.062	0.223	0.098	0.073	0.072	0.070	0.041
7	0.046	0.074	0.241	0.088	0.059	0.053	0.053	0.046
8	0.045	0.072	0.237	0.089	0.055	0.054	0.059	0.049
9	0.069	0.112	0.234	0.081	0.075	0.077	0.078	0.079
10	0.041	0.065	0.225	0.098	0.074	0.083	0.075	0.044
11	0.048	0.072	0.217	0.088	0.064	0.065	0.063	0.050
12	0.100	0.145	0.263	0.111	0.109	0.122	0.116	0.113
13	0.052	0.077	0.222	0.085	0.063	0.059	0.057	0.051
14	0.079	0.115	0.273	0.101	0.095	0.094	0.091	0.082
15	0.086	0.113	0.255	0.097	0.102	0.093	0.093	0.097

Table B.4 Comparison of SE for the item difficulty among algorithms (Rasch model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.042	0.062	0.222	0.096	0.070	0.067	0.077	0.040
17	0.076	0.115	0.288	0.104	0.093	0.089	0.100	0.075
18	0.082	0.120	0.243	0.104	0.085	0.087	0.091	0.087
19	0.045	0.076	0.237	0.092	0.053	0.054	0.057	0.048
20	0.048	0.073	0.215	0.094	0.061	0.062	0.063	0.050
21	0.083	0.119	0.241	0.099	0.091	0.092	0.088	0.088
22	0.045	0.069	0.233	0.092	0.058	0.057	0.057	0.047
23	0.079	0.110	0.245	0.089	0.086	0.095	0.088	0.089
24	0.050	0.074	0.219	0.083	0.062	0.055	0.053	0.052
25	0.085	0.124	0.243	0.094	0.090	0.092	0.087	0.091
26	0.068	0.105	0.229	0.097	0.079	0.082	0.071	0.077
27	0.080	0.113	0.299	0.116	0.101	0.108	0.095	0.084
28	0.079	0.118	0.277	0.107	0.095	0.093	0.095	0.088
29	0.093	0.134	0.300	0.134	0.110	0.120	0.117	0.106
30	0.082	0.124	0.290	0.105	0.106	0.098	0.100	0.096

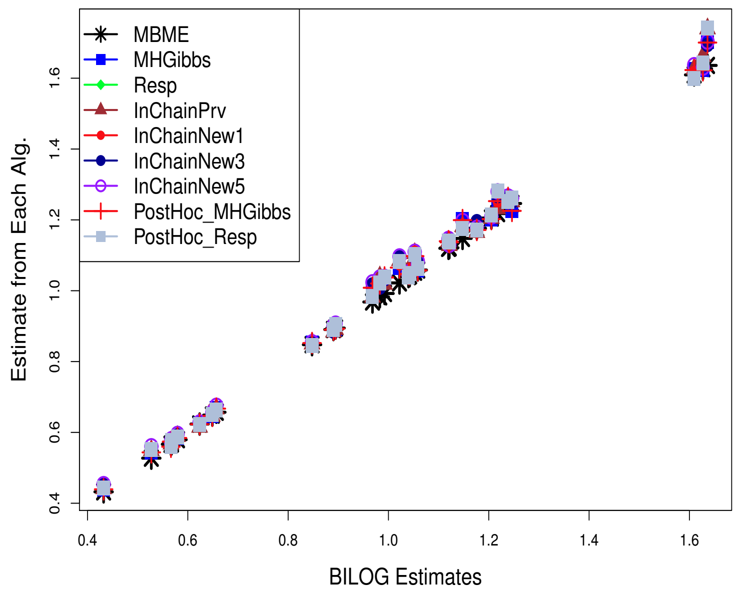

Figure B.3 Comparison of estimates for the item discrimination among algorithms (2PL model with 500 examinees and 30 items)

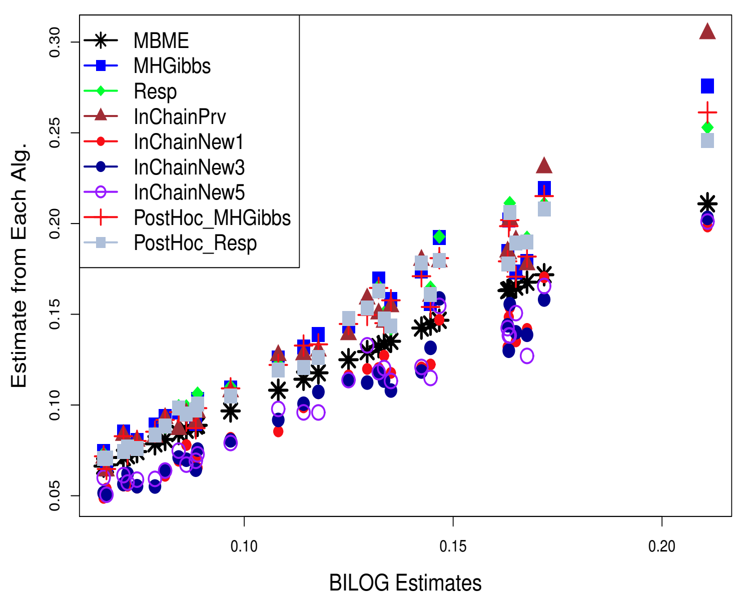

Figure B.4 Comparison of SE for the item discrimination among algorithms (2PL model with 500 examinees and 30 items)

Table B.5 Comparison of estimates for the item discrimination among algorithms (2PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.624	0.624	0.623	0.615	0.624	0.624	0.628	0.623
2	0.983	1.021	1.022	1.040	1.015	1.009	1.037	1.020
3	1.627	1.624	1.645	1.662	1.643	1.641	1.630	1.622
4	0.968	1.009	0.985	0.982	1.007	1.018	1.024	1.008
5	1.246	1.227	1.262	1.252	1.252	1.251	1.261	1.225
6	1.052	1.098	1.102	1.105	1.116	1.095	1.110	1.098
7	1.022	1.065	1.083	1.086	1.073	1.094	1.097	1.065
8	0.579	0.585	0.588	0.589	0.591	0.592	0.595	0.584
9	1.609	1.624	1.601	1.627	1.618	1.604	1.636	1.623
10	1.120	1.141	1.142	1.139	1.147	1.146	1.146	1.140
11	1.218	1.254	1.284	1.232	1.251	1.267	1.280	1.253
12	0.527	0.544	0.552	0.549	0.544	0.547	0.561	0.544
13	0.649	0.648	0.651	0.646	0.667	0.657	0.656	0.647
14	1.148	1.201	1.177	1.181	1.185	1.175	1.197	1.199
15	1.058	1.058	1.064	1.068	1.077	1.084	1.083	1.058

Table B.6 Comparison of estimates for the item discrimination among algorithms (2PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.568	0.574	0.577	0.574	0.580	0.577	0.581	0.574
17	1.042	1.048	1.049	1.055	1.045	1.038	1.053	1.047
18	0.566	0.560	0.561	0.568	0.567	0.566	0.574	0.560
19	1.176	1.173	1.172	1.165	1.182	1.196	1.177	1.172
20	1.238	1.262	1.252	1.263	1.260	1.266	1.266	1.261
21	1.121	1.138	1.138	1.128	1.132	1.137	1.130	1.137
22	0.432	0.440	0.443	0.442	0.448	0.454	0.454	0.439
23	1.040	1.048	1.039	1.057	1.051	1.044	1.047	1.048
24	0.992	1.038	1.040	1.019	1.027	1.032	1.039	1.037
25	1.205	1.203	1.215	1.205	1.220	1.213	1.210	1.202
26	0.657	0.668	0.665	0.667	0.668	0.664	0.675	0.667
27	0.895	0.894	0.906	0.891	0.901	0.900	0.907	0.893
28	1.636	1.704	1.743	1.740	1.735	1.694	1.704	1.701
29	0.890	0.891	0.892	0.884	0.900	0.885	0.901	0.890
30	0.848	0.853	0.846	0.850	0.852	0.850	0.854	0.852

Table B.7 Comparison of SE for the item discrimination among algorithms (2PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.071	0.085	0.075	0.083	0.056	0.056	0.061	0.083
2	0.142	0.174	0.178	0.180	0.121	0.118	0.121	0.171
3	0.147	0.192	0.193	0.179	0.147	0.159	0.155	0.181
4	0.145	0.156	0.165	0.160	0.122	0.132	0.115	0.154
5	0.129	0.154	0.155	0.159	0.120	0.112	0.133	0.150
6	0.165	0.173	0.191	0.191	0.135	0.140	0.151	0.171
7	0.168	0.179	0.192	0.177	0.142	0.139	0.127	0.182
8	0.079	0.089	0.085	0.082	0.056	0.055	0.059	0.085
9	0.172	0.219	0.211	0.231	0.170	0.158	0.166	0.215
10	0.133	0.149	0.151	0.146	0.127	0.113	0.120	0.145
11	0.164	0.205	0.211	0.202	0.154	0.156	0.138	0.202
12	0.081	0.094	0.088	0.093	0.061	0.064	0.064	0.092
13	0.072	0.077	0.078	0.077	0.055	0.062	0.057	0.073
14	0.163	0.184	0.180	0.184	0.132	0.144	0.142	0.179
15	0.114	0.132	0.125	0.128	0.099	0.101	0.096	0.133

Table B.8 Comparison of SE for the item discrimination among algorithms (2PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.074	0.081	0.080	0.080	0.054	0.055	0.059	0.080
17	0.108	0.126	0.123	0.127	0.085	0.092	0.098	0.122
18	0.066	0.075	0.073	0.071	0.049	0.051	0.060	0.072
19	0.118	0.139	0.129	0.130	0.106	0.107	0.096	0.133
20	0.163	0.202	0.183	0.183	0.148	0.130	0.138	0.199
21	0.132	0.169	0.165	0.150	0.120	0.118	0.118	0.165
22	0.067	0.065	0.070	0.064	0.054	0.051	0.051	0.065
23	0.097	0.109	0.110	0.107	0.082	0.081	0.079	0.109
24	0.135	0.158	0.142	0.154	0.118	0.108	0.113	0.158
25	0.125	0.143	0.146	0.139	0.116	0.114	0.114	0.145
26	0.088	0.090	0.101	0.090	0.071	0.064	0.069	0.087
27	0.089	0.103	0.106	0.096	0.065	0.075	0.073	0.098
28	0.211	0.276	0.253	0.305	0.198	0.203	0.201	0.261
29	0.086	0.097	0.099	0.095	0.078	0.070	0.067	0.091
30	0.084	0.096	0.099	0.087	0.069	0.071	0.075	0.096

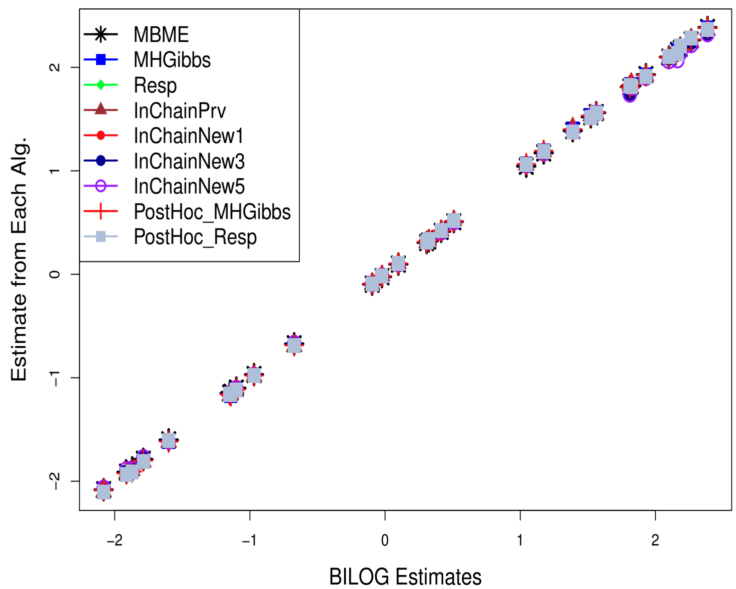

Figure B.5 Comparison of estimates for the item difficulty among algorithms (2PL model with 500 examinees and 30 items)

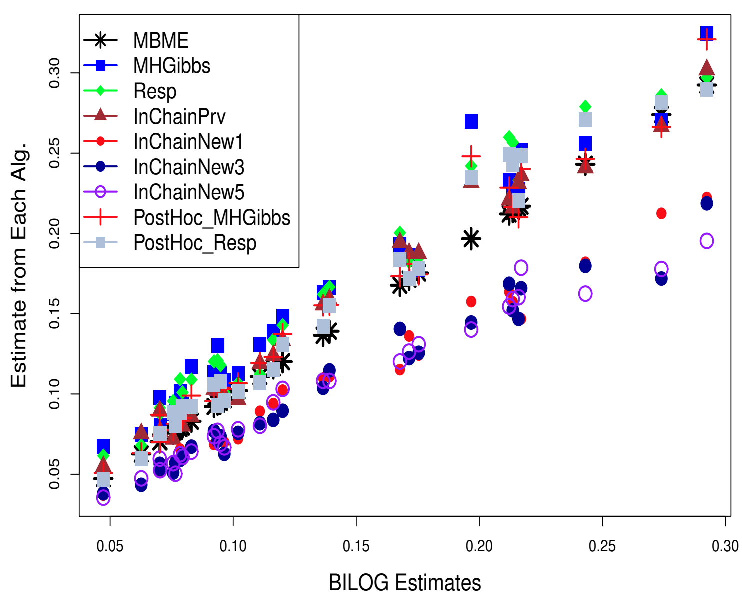

Figure B.6 Comparison of SE for the item difficulty among algorithms (2PL model with 500 examinees and 30 items)

Table B.9 Comparison of estimates for the item difficulty among algorithms (2PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-0.672	-0.686	-0.688	-0.694	-0.676	-0.673	-0.665	-0.686
2	2.100	2.099	2.103	2.084	2.081	2.086	2.061	2.099
3	0.307	0.313	0.312	0.309	0.312	0.318	0.313	0.312
4	2.184	2.173	2.208	2.194	2.153	2.152	2.140	2.173
5	-1.141	-1.171	-1.155	-1.156	-1.144	-1.147	-1.145	-1.171
6	2.264	2.266	2.285	2.246	2.228	2.238	2.217	2.266
7	2.385	2.382	2.366	2.361	2.342	2.320	2.322	2.382
8	-1.788	-1.800	-1.811	-1.812	-1.774	-1.760	-1.769	-1.800
9	-1.149	-1.159	-1.162	-1.170	-1.154	-1.160	-1.155	-1.160
10	1.522	1.534	1.517	1.533	1.518	1.527	1.525	1.534
11	1.822	1.834	1.815	1.849	1.821	1.807	1.804	1.834
12	2.163	2.160	2.132	2.144	2.132	2.135	2.070	2.160
13	0.415	0.421	0.424	0.421	0.414	0.419	0.405	0.421
14	-2.082	-2.082	-2.103	-2.100	-2.060	-2.094	-2.065	-2.082
15	1.173	1.193	1.188	1.185	1.166	1.171	1.169	1.194

Table B.10 Comparison of estimates for the item difficulty among algorithms (2PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	1.388	1.402	1.375	1.418	1.371	1.381	1.384	1.403
17	-1.100	-1.113	-1.118	-1.104	-1.101	-1.107	-1.100	-1.113
18	-0.095	-0.091	-0.092	-0.096	-0.090	-0.088	-0.095	-0.091
19	-0.969	-0.978	-0.979	-0.983	-0.970	-0.972	-0.972	-0.978
20	-1.871	-1.906	-1.910	-1.891	-1.885	-1.868	-1.881	-1.907
21	-1.601	-1.617	-1.606	-1.619	-1.597	-1.606	-1.614	-1.616
22	1.811	1.808	1.822	1.824	1.775	1.748	1.739	1.808
23	0.098	0.104	0.104	0.106	0.101	0.100	0.098	0.104
24	1.930	1.925	1.911	1.923	1.909	1.908	1.900	1.926
25	1.045	1.065	1.060	1.059	1.051	1.061	1.053	1.065
26	-1.913	-1.924	-1.937	-1.922	-1.903	-1.908	-1.886	-1.924
27	0.509	0.511	0.518	0.521	0.516	0.521	0.504	0.511
28	1.562	1.566	1.558	1.548	1.545	1.555	1.556	1.565
29	-0.023	-0.016	-0.018	-0.013	-0.014	-0.023	-0.012	-0.016
30	0.324	0.329	0.335	0.329	0.331	0.327	0.326	0.330

Table B.11 Comparison of SE for the item difficulty among algorithms (2PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.111	0.131	0.112	0.119	0.089	0.082	0.080	0.119
2	0.197	0.270	0.242	0.232	0.158	0.145	0.140	0.248
3	0.047	0.067	0.061	0.055	0.037	0.038	0.035	0.051
4	0.212	0.233	0.260	0.221	0.164	0.169	0.154	0.229
5	0.083	0.117	0.109	0.086	0.068	0.067	0.064	0.099
6	0.217	0.252	0.249	0.236	0.147	0.166	0.179	0.240
7	0.243	0.256	0.279	0.241	0.182	0.180	0.162	0.246
8	0.216	0.227	0.233	0.231	0.146	0.147	0.160	0.210
9	0.070	0.098	0.091	0.089	0.057	0.056	0.060	0.087
10	0.116	0.139	0.134	0.124	0.094	0.084	0.095	0.123
11	0.139	0.166	0.167	0.159	0.111	0.115	0.108	0.156
12	0.292	0.325	0.297	0.302	0.222	0.219	0.195	0.321
13	0.096	0.109	0.101	0.097	0.070	0.063	0.067	0.103
14	0.168	0.193	0.200	0.194	0.115	0.141	0.120	0.173
15	0.095	0.113	0.118	0.103	0.072	0.074	0.069	0.104

Table B.12 Comparison of SE for the item difficulty among algorithms (2PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.175	0.178	0.185	0.188	0.124	0.126	0.131	0.174
17	0.092	0.114	0.120	0.103	0.069	0.077	0.074	0.096
18	0.102	0.113	0.107	0.097	0.072	0.076	0.078	0.107
19	0.078	0.101	0.109	0.087	0.066	0.059	0.062	0.083
20	0.137	0.163	0.162	0.156	0.110	0.104	0.108	0.155
21	0.120	0.148	0.143	0.133	0.102	0.090	0.103	0.137
22	0.274	0.271	0.286	0.267	0.213	0.172	0.178	0.266
23	0.063	0.075	0.068	0.075	0.043	0.043	0.047	0.063
24	0.171	0.186	0.182	0.188	0.136	0.122	0.126	0.181
25	0.079	0.094	0.101	0.080	0.059	0.062	0.061	0.085
26	0.214	0.226	0.257	0.216	0.157	0.152	0.159	0.216
27	0.077	0.094	0.086	0.080	0.058	0.057	0.050	0.081
28	0.094	0.130	0.121	0.102	0.072	0.071	0.077	0.103
29	0.070	0.080	0.088	0.077	0.051	0.053	0.052	0.070
30	0.076	0.088	0.096	0.072	0.052	0.051	0.057	0.077

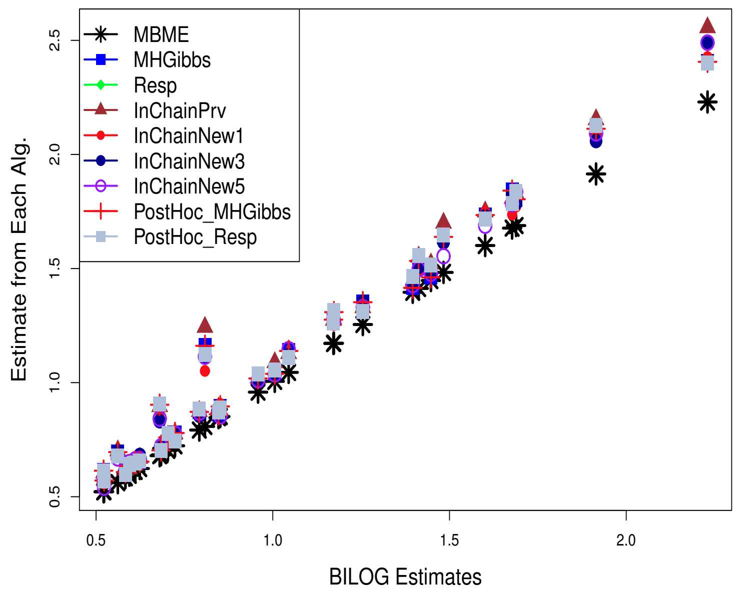

Figure B.7 Comparison of estimates for the item discrimination among algorithms (3PL model with 500 examinees and 30 items)

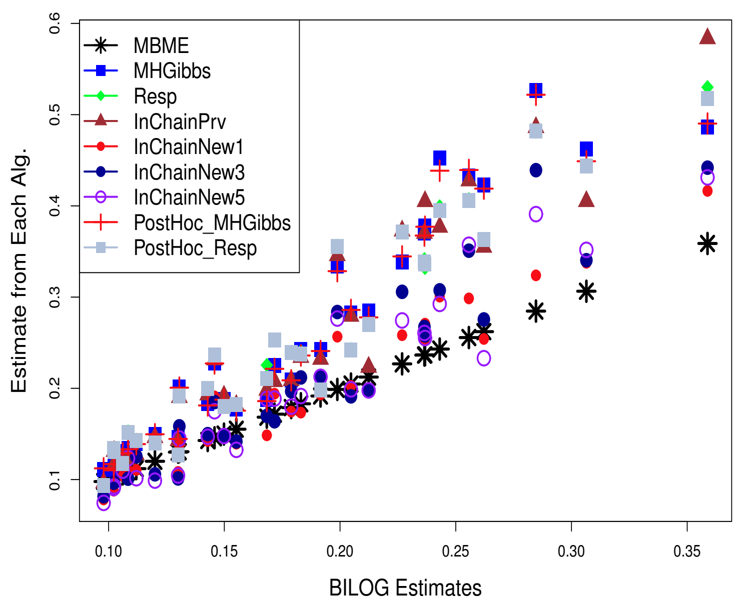

Figure B.8 Comparison of SE for the item discrimination among algorithms (3PL model with 500 examinees and 30 items)

Table B.13 Comparison of estimates for the item discrimination among algorithms (3PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	1.396	1.416	1.468	1.434	1.424	1.428	1.428	1.416
2	1.688	1.803	1.838	1.830	1.771	1.787	1.837	1.804
3	1.413	1.533	1.559	1.551	1.486	1.508	1.497	1.534
4	0.684	0.704	0.703	0.725	0.717	0.712	0.721	0.703
5	1.915	2.115	2.129	2.153	2.089	2.059	2.092	2.113
6	0.624	0.654	0.657	0.669	0.659	0.683	0.664	0.654
7	1.006	1.039	1.056	1.088	1.044	1.038	1.036	1.039
8	1.172	1.276	1.262	1.274	1.301	1.264	1.280	1.276
9	0.792	0.872	0.886	0.874	0.863	0.855	0.869	0.872
10	1.601	1.735	1.719	1.749	1.748	1.728	1.688	1.734
11	1.045	1.140	1.111	1.128	1.122	1.121	1.146	1.139
12	1.255	1.353	1.312	1.329	1.337	1.354	1.331	1.352
13	0.583	0.608	0.598	0.617	0.634	0.637	0.616	0.608
14	1.447	1.463	1.517	1.521	1.488	1.488	1.478	1.461
15	0.808	1.163	1.123	1.243	1.051	1.112	1.114	1.161

Table B.14 Comparison of estimates for the item discrimination among algorithms (3PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.521	0.614	0.613	0.606	0.579	0.579	0.574	0.614
17	2.230	2.407	2.405	2.556	2.427	2.486	2.490	2.406
18	0.523	0.571	0.568	0.575	0.557	0.573	0.540	0.571
19	1.483	1.641	1.648	1.699	1.611	1.615	1.554	1.639
20	0.680	0.904	0.906	0.896	0.869	0.831	0.842	0.903
21	0.562	0.696	0.678	0.702	0.683	0.698	0.668	0.696
22	1.677	1.844	1.788	1.809	1.736	1.825	1.783	1.841
23	0.703	0.761	0.777	0.766	0.753	0.745	0.747	0.760
24	1.173	1.310	1.316	1.278	1.252	1.282	1.269	1.309
25	0.852	0.896	0.890	0.878	0.907	0.875	0.853	0.896
26	0.958	1.019	1.038	1.021	1.021	1.005	1.011	1.018
27	0.846	0.862	0.872	0.890	0.880	0.888	0.877	0.861
28	0.611	0.652	0.649	0.640	0.658	0.653	0.664	0.652
29	0.594	0.636	0.642	0.623	0.629	0.628	0.651	0.636
30	0.723	0.779	0.746	0.777	0.774	0.771	0.776	0.779

Table B.15 Comparison of SE for the item discrimination among algorithms (3PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.168	0.188	0.225	0.198	0.149	0.170	0.189	0.186
2	0.256	0.433	0.407	0.428	0.299	0.351	0.357	0.439
3	0.227	0.338	0.372	0.373	0.258	0.306	0.274	0.345
4	0.102	0.120	0.132	0.131	0.097	0.097	0.093	0.117
5	0.306	0.462	0.445	0.405	0.338	0.340	0.352	0.449
6	0.102	0.111	0.134	0.116	0.088	0.097	0.091	0.110
7	0.155	0.177	0.184	0.181	0.139	0.142	0.133	0.176
8	0.205	0.283	0.243	0.279	0.203	0.191	0.199	0.286
9	0.179	0.209	0.239	0.209	0.175	0.197	0.180	0.209
10	0.236	0.370	0.342	0.370	0.253	0.267	0.261	0.367
11	0.212	0.285	0.271	0.223	0.200	0.198	0.198	0.278
12	0.192	0.243	0.196	0.232	0.193	0.213	0.213	0.241
13	0.098	0.111	0.093	0.096	0.078	0.082	0.074	0.112
14	0.183	0.243	0.241	0.234	0.173	0.212	0.192	0.236
15	0.285	0.526	0.481	0.486	0.324	0.439	0.391	0.522

Table B.16 Comparison of SE for the item discrimination among algorithms (3PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.131	0.201	0.191	0.190	0.137	0.158	0.145	0.201
17	0.359	0.486	0.530	0.583	0.416	0.442	0.431	0.490
18	0.112	0.139	0.140	0.127	0.111	0.125	0.101	0.139
19	0.262	0.423	0.364	0.355	0.254	0.276	0.233	0.419
20	0.237	0.378	0.332	0.405	0.271	0.258	0.255	0.377
21	0.146	0.228	0.236	0.231	0.191	0.185	0.175	0.227
22	0.243	0.453	0.399	0.377	0.300	0.307	0.293	0.439
23	0.143	0.184	0.199	0.193	0.142	0.150	0.147	0.181
24	0.199	0.334	0.356	0.346	0.257	0.284	0.276	0.328
25	0.150	0.188	0.181	0.192	0.150	0.147	0.148	0.188
26	0.172	0.225	0.253	0.208	0.194	0.164	0.189	0.221
27	0.130	0.146	0.128	0.143	0.108	0.101	0.105	0.145
28	0.106	0.131	0.123	0.123	0.104	0.117	0.110	0.129
29	0.108	0.134	0.153	0.122	0.110	0.101	0.123	0.133
30	0.120	0.149	0.143	0.146	0.106	0.106	0.099	0.150

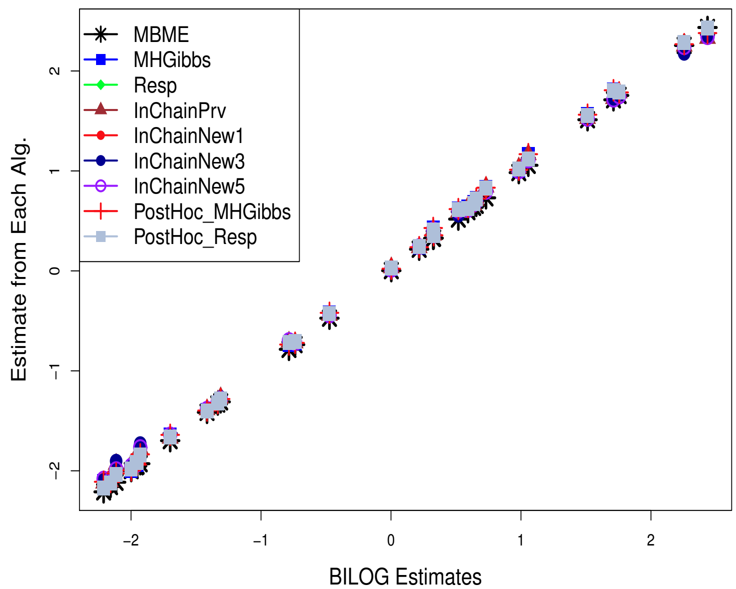

Figure B.9 Comparison of estimates for the item difficulty among algorithms (3PL model with 500 examinees and 30 items)

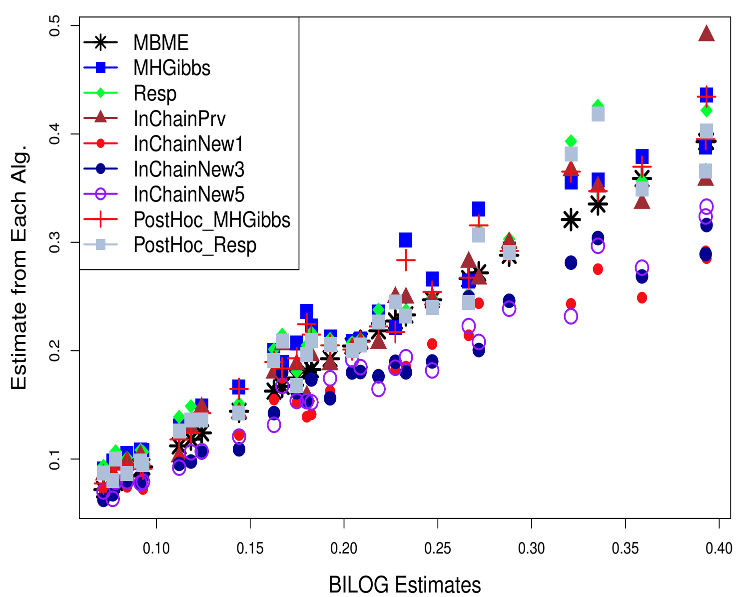

Figure B.10 Comparison of SE for the item difficulty among algorithms (3PL model with 500 examinees and 30 items)

Table B.17 Comparison of estimates for the item difficulty among algorithms (3PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-0.738	-0.721	-0.707	-0.718	-0.720	-0.724	-0.720	-0.720
2	0.217	0.237	0.244	0.234	0.244	0.233	0.244	0.237
3	0.636	0.685	0.675	0.661	0.656	0.659	0.669	0.685
4	-0.785	-0.738	-0.717	-0.711	-0.700	-0.700	-0.696	-0.737
5	-1.415	-1.397	-1.399	-1.387	-1.384	-1.402	-1.383	-1.397
6	-1.929	-1.835	-1.839	-1.776	-1.794	-1.727	-1.767	-1.835
7	-1.953	-1.939	-1.916	-1.859	-1.892	-1.907	-1.908	-1.939
8	-2.157	-2.112	-2.121	-2.107	-2.050	-2.085	-2.092	-2.112
9	1.512	1.563	1.555	1.527	1.536	1.539	1.524	1.562
10	-1.310	-1.282	-1.282	-1.268	-1.262	-1.267	-1.282	-1.282
11	1.755	1.786	1.789	1.783	1.758	1.764	1.749	1.785
12	-1.699	-1.641	-1.663	-1.673	-1.633	-1.643	-1.640	-1.641
13	-2.114	-2.025	-2.038	-1.933	-1.933	-1.903	-1.987	-2.026
14	0.002	0.021	0.031	0.033	0.024	0.017	0.015	0.021
15	2.435	2.379	2.426	2.328	2.363	2.336	2.342	2.379

Table B.18 Comparison of estimates for the item difficulty among algorithms (3PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	1.056	1.167	1.118	1.165	1.094	1.118	1.107	1.167
17	-1.332	-1.323	-1.322	-1.300	-1.313	-1.307	-1.311	-1.323
18	0.731	0.832	0.828	0.840	0.798	0.827	0.794	0.832
19	-1.959	-1.921	-1.915	-1.885	-1.916	-1.909	-1.939	-1.921
20	2.255	2.267	2.285	2.251	2.174	2.175	2.242	2.267
21	1.711	1.808	1.803	1.726	1.727	1.711	1.716	1.807
22	0.328	0.356	0.352	0.360	0.347	0.349	0.356	0.356
23	0.656	0.723	0.718	0.709	0.708	0.699	0.695	0.723
24	-0.473	-0.420	-0.426	-0.443	-0.447	-0.440	-0.442	-0.420
25	0.592	0.638	0.625	0.640	0.627	0.618	0.613	0.639
26	0.983	1.011	1.021	1.032	1.000	1.004	0.999	1.011
27	-1.998	-1.996	-1.981	-1.934	-1.931	-1.942	-1.952	-1.997
28	0.325	0.429	0.408	0.414	0.401	0.363	0.413	0.429
29	0.518	0.617	0.614	0.586	0.579	0.556	0.604	0.618
30	-2.211	-2.108	-2.174	-2.121	-2.076	-2.098	-2.082	-2.110

Table B.19 Comparison of SE for the item difficulty among algorithms (3PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.093	0.108	0.107	0.098	0.072	0.084	0.079	0.091
2	0.078	0.098	0.107	0.095	0.075	0.076	0.079	0.090
3	0.084	0.105	0.101	0.098	0.074	0.079	0.080	0.095
4	0.227	0.222	0.250	0.249	0.182	0.190	0.184	0.217
5	0.112	0.130	0.139	0.102	0.099	0.096	0.092	0.118
6	0.335	0.357	0.426	0.351	0.275	0.304	0.297	0.347
7	0.218	0.236	0.238	0.207	0.178	0.176	0.165	0.222
8	0.233	0.302	0.237	0.249	0.185	0.180	0.194	0.284
9	0.183	0.223	0.217	0.196	0.141	0.173	0.152	0.215
10	0.124	0.149	0.141	0.147	0.106	0.106	0.107	0.142
11	0.163	0.200	0.201	0.179	0.155	0.142	0.132	0.190
12	0.175	0.207	0.181	0.187	0.152	0.162	0.154	0.193
13	0.359	0.379	0.357	0.336	0.249	0.269	0.277	0.370
14	0.077	0.098	0.084	0.087	0.071	0.068	0.063	0.090
15	0.393	0.388	0.367	0.357	0.292	0.289	0.324	0.395

Table B.20 Comparison of SE for the item difficulty among algorithms (3PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.288	0.294	0.303	0.300	0.247	0.246	0.239	0.292
17	0.092	0.109	0.108	0.103	0.079	0.078	0.078	0.096
18	0.266	0.264	0.252	0.282	0.214	0.249	0.223	0.267
19	0.180	0.236	0.205	0.158	0.139	0.154	0.153	0.224
20	0.393	0.436	0.422	0.491	0.285	0.316	0.333	0.434
21	0.272	0.331	0.311	0.266	0.244	0.201	0.208	0.316
22	0.072	0.091	0.094	0.085	0.072	0.062	0.070	0.078
23	0.193	0.213	0.211	0.187	0.163	0.156	0.175	0.205
24	0.167	0.189	0.215	0.206	0.175	0.180	0.167	0.184
25	0.144	0.166	0.151	0.143	0.122	0.109	0.121	0.165
26	0.118	0.136	0.149	0.129	0.097	0.098	0.106	0.124
27	0.247	0.266	0.249	0.247	0.206	0.190	0.182	0.254
28	0.204	0.209	0.207	0.202	0.180	0.180	0.192	0.201
29	0.209	0.211	0.211	0.207	0.181	0.180	0.185	0.208
30	0.321	0.356	0.393	0.366	0.243	0.281	0.232	0.365

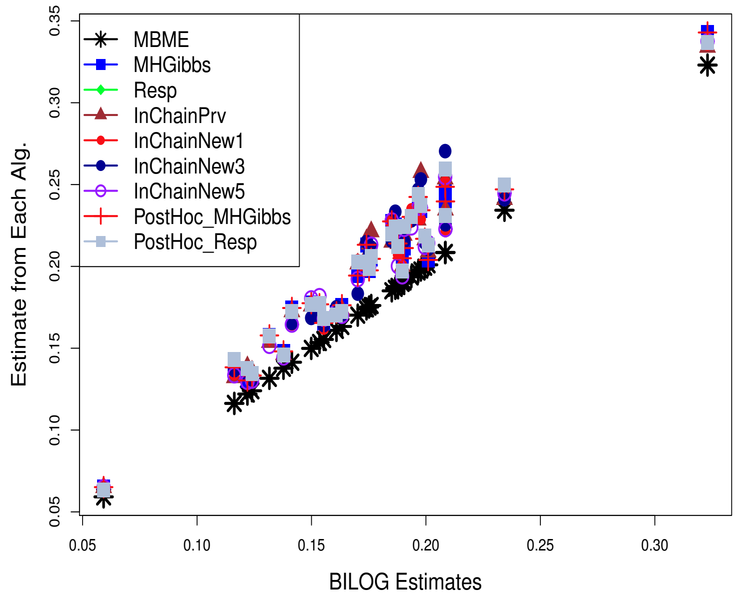

Figure B.11 Comparison of estimates for the item guessing among algorithms (3PL model with 500 examinees and 30 items)

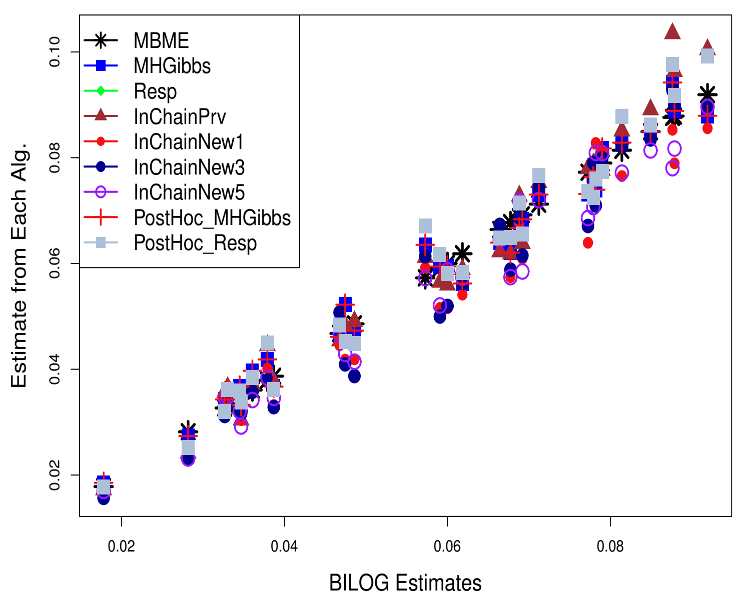

Figure B.12 Comparison of SE for the item guessing among algorithms (3PL model with 500 examinees and 30 items)

Table B.21 Comparison of estimates for the item guessing among algorithms (3PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.116	0.138	0.143	0.132	0.134	0.137	0.133	0.138
2	0.201	0.204	0.213	0.209	0.214	0.211	0.214	0.204
3	0.163	0.176	0.172	0.173	0.170	0.171	0.170	0.176
4	0.190	0.211	0.224	0.219	0.220	0.215	0.223	0.211
5	0.170	0.194	0.203	0.202	0.204	0.183	0.192	0.194
6	0.208	0.249	0.259	0.254	0.252	0.270	0.255	0.249
7	0.176	0.205	0.206	0.221	0.211	0.212	0.213	0.205
8	0.194	0.230	0.230	0.231	0.235	0.228	0.224	0.230
9	0.138	0.148	0.146	0.146	0.147	0.146	0.144	0.148
10	0.174	0.213	0.203	0.216	0.206	0.215	0.202	0.213
11	0.059	0.065	0.063	0.065	0.065	0.064	0.065	0.065
12	0.185	0.227	0.220	0.215	0.228	0.216	0.228	0.227
13	0.198	0.234	0.237	0.257	0.229	0.253	0.236	0.234
14	0.122	0.130	0.138	0.139	0.135	0.134	0.129	0.130
15	0.155	0.165	0.168	0.169	0.162	0.164	0.168	0.165

Table B.22 Comparison of estimates for the item guessing among algorithms (3PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.209	0.240	0.231	0.235	0.221	0.226	0.223	0.240
17	0.141	0.175	0.172	0.172	0.164	0.165	0.164	0.175
18	0.188	0.214	0.212	0.214	0.207	0.215	0.200	0.214
19	0.187	0.225	0.224	0.225	0.219	0.233	0.221	0.225
20	0.234	0.247	0.250	0.241	0.246	0.241	0.245	0.247
21	0.132	0.158	0.158	0.154	0.156	0.158	0.151	0.158
22	0.161	0.172	0.170	0.173	0.171	0.175	0.172	0.172
23	0.200	0.217	0.218	0.216	0.216	0.215	0.212	0.217
24	0.323	0.343	0.337	0.334	0.335	0.338	0.338	0.343
25	0.190	0.205	0.197	0.200	0.201	0.198	0.194	0.205
26	0.124	0.133	0.135	0.136	0.129	0.130	0.130	0.133
27	0.175	0.198	0.199	0.214	0.206	0.207	0.200	0.198
28	0.150	0.178	0.177	0.176	0.178	0.169	0.180	0.178
29	0.153	0.177	0.177	0.171	0.172	0.172	0.182	0.177
30	0.197	0.242	0.244	0.228	0.241	0.247	0.236	0.242

Table B.23 Comparison of SE for the item guessing among algorithms (3PL model with 500 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.047	0.046	0.048	0.045	0.045	0.051	0.047	0.046
2	0.038	0.042	0.045	0.044	0.040	0.038	0.038	0.042
3	0.033	0.034	0.032	0.036	0.033	0.031	0.035	0.034
4	0.077	0.073	0.074	0.078	0.064	0.067	0.069	0.073
5	0.066	0.064	0.065	0.062	0.067	0.067	0.065	0.064
6	0.092	0.088	0.099	0.100	0.086	0.089	0.090	0.088
7	0.078	0.074	0.076	0.079	0.083	0.071	0.081	0.074
8	0.085	0.085	0.086	0.089	0.086	0.083	0.081	0.085
9	0.035	0.033	0.034	0.030	0.030	0.032	0.029	0.033
10	0.069	0.067	0.071	0.073	0.068	0.069	0.072	0.067
11	0.018	0.019	0.018	0.017	0.018	0.016	0.017	0.019
12	0.079	0.082	0.077	0.078	0.081	0.080	0.081	0.082
13	0.088	0.089	0.092	0.096	0.079	0.090	0.082	0.089
14	0.036	0.040	0.038	0.037	0.036	0.036	0.034	0.040
15	0.028	0.027	0.025	0.024	0.024	0.023	0.023	0.027

Table B.24 Comparison of SE for the item guessing among algorithms (3PL model with 500 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.069	0.068	0.066	0.064	0.061	0.062	0.058	0.068
17	0.057	0.064	0.067	0.061	0.059	0.061	0.057	0.064
18	0.068	0.062	0.065	0.062	0.057	0.059	0.057	0.062
19	0.081	0.083	0.088	0.085	0.077	0.082	0.077	0.083
20	0.039	0.037	0.036	0.038	0.033	0.033	0.035	0.037
21	0.049	0.047	0.045	0.049	0.042	0.039	0.041	0.047
22	0.033	0.035	0.036	0.037	0.037	0.033	0.035	0.035
23	0.059	0.059	0.062	0.056	0.052	0.050	0.052	0.059
24	0.071	0.073	0.077	0.075	0.076	0.074	0.072	0.073
25	0.047	0.052	0.045	0.048	0.042	0.041	0.043	0.052
26	0.035	0.037	0.036	0.033	0.033	0.032	0.032	0.037
27	0.078	0.076	0.072	0.079	0.077	0.079	0.071	0.076
28	0.062	0.056	0.058	0.059	0.054	0.056	0.057	0.056
29	0.060	0.059	0.058	0.056	0.052	0.052	0.059	0.059
30	0.088	0.094	0.098	0.103	0.085	0.093	0.078	0.094

Appendix C

Result Tables and Graphs for 1000 examinees and 15 items

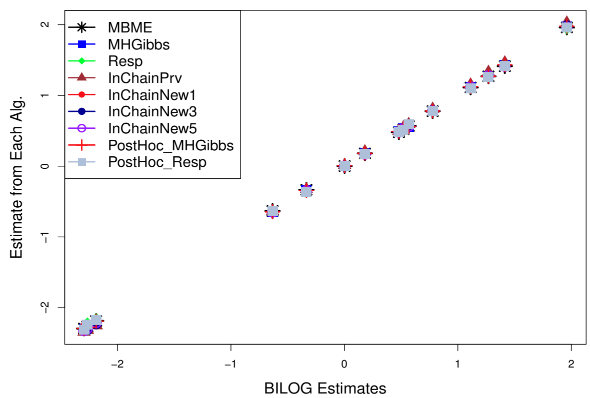

Figure C.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 1000 examinees and 15 items)

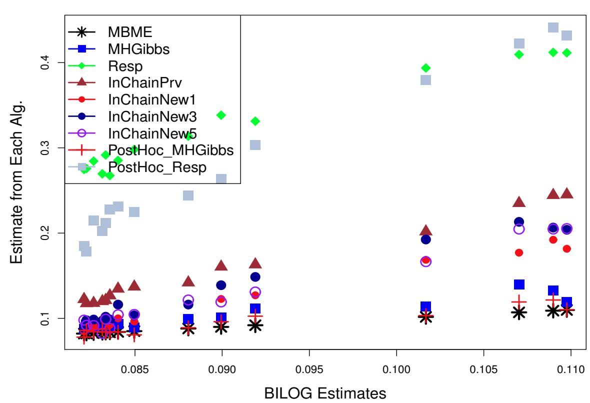

Figure C.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 1000 examinees and 15 items)

Table C.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-0.335	-0.343	-0.352	-0.330	-0.322	-0.363	-0.350	-0.343
2	1.112	1.124	1.086	1.151	1.110	1.124	1.117	1.124
3	-2.188	-2.190	-2.145	-2.253	-2.157	-2.238	-2.209	-2.190
4	0.002	0.006	0.003	0.014	0.013	-0.009	0.003	0.006
5	-2.265	-2.278	-2.206	-2.317	-2.231	-2.321	-2.293	-2.279
6	0.181	0.183	0.169	0.207	0.183	0.172	0.169	0.183
7	0.567	0.558	0.580	0.593	0.578	0.573	0.566	0.558
8	1.414	1.433	1.395	1.463	1.409	1.420	1.413	1.433
9	0.480	0.490	0.475	0.494	0.487	0.482	0.481	0.490
10	0.521	0.532	0.505	0.548	0.529	0.522	0.526	0.532
11	1.961	1.978	1.930	2.027	1.961	1.993	1.966	1.978
12	0.778	0.779	0.770	0.808	0.771	0.794	0.781	0.779
13	-0.633	-0.646	-0.626	-0.638	-0.614	-0.657	-0.647	-0.646
14	-2.295	-2.293	-2.274	-2.343	-2.262	-2.329	-2.325	-2.293
15	1.270	1.270	1.246	1.330	1.277	1.281	1.277	1.271

Table C.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.083	0.090	0.284	0.118	0.091	0.099	0.092	0.084
2	0.088	0.099	0.314	0.142	0.115	0.116	0.122	0.089
3	0.107	0.140	0.409	0.235	0.177	0.213	0.205	0.119
4	0.082	0.087	0.275	0.122	0.085	0.089	0.098	0.078
5	0.109	0.133	0.412	0.244	0.192	0.206	0.205	0.122
6	0.082	0.093	0.276	0.118	0.090	0.098	0.092	0.086
7	0.084	0.090	0.268	0.127	0.092	0.099	0.088	0.084
8	0.092	0.111	0.331	0.163	0.127	0.149	0.131	0.103
9	0.083	0.093	0.270	0.120	0.099	0.097	0.082	0.087
10	0.083	0.089	0.292	0.121	0.101	0.102	0.099	0.084
11	0.102	0.114	0.394	0.202	0.169	0.192	0.167	0.103
12	0.085	0.089	0.298	0.137	0.096	0.103	0.105	0.081
13	0.084	0.093	0.285	0.135	0.100	0.116	0.104	0.084
14	0.110	0.119	0.411	0.245	0.182	0.204	0.205	0.110
15	0.090	0.101	0.338	0.160	0.123	0.139	0.119	0.096

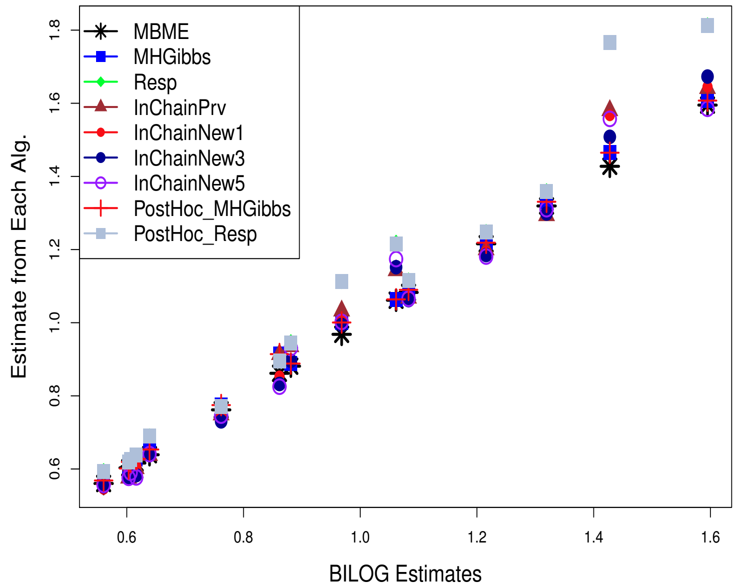

Figure C.3 Comparison of estimates for the item discrimination among algorithms (2PL model with 1000 examinees and 15 items)

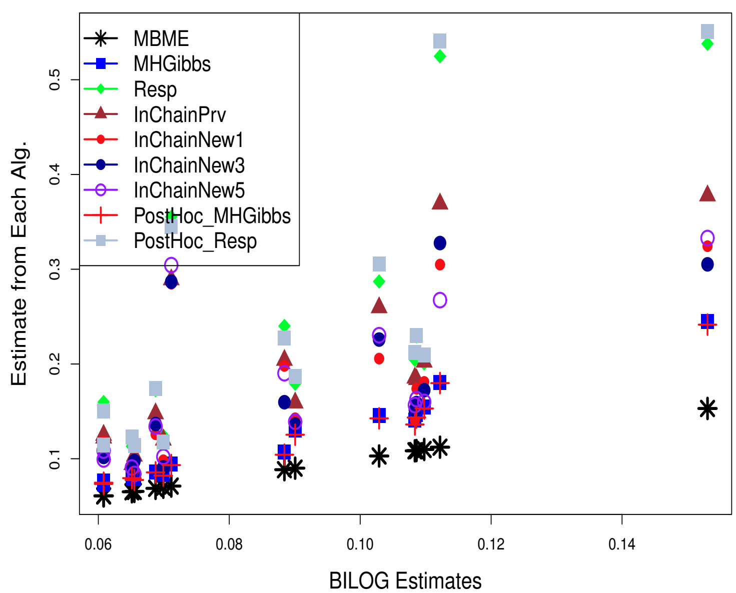

Figure C.4 Comparison of SE for the item discrimination among algorithms (2PL model with 1000 examinees and 15 items)

Table C.3 Comparison of estimates for the item discrimination among algorithms (2PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	1.216	1.219	1.250	1.200	1.194	1.185	1.181	1.219
2	0.762	0.775	0.773	0.748	0.743	0.730	0.746	0.775
3	0.560	0.569	0.597	0.572	0.546	0.553	0.557	0.569
4	1.595	1.608	1.815	1.641	1.645	1.673	1.585	1.607
5	1.061	1.064	1.221	1.143	1.154	1.152	1.174	1.064
6	0.616	0.628	0.639	0.602	0.608	0.583	0.577	0.628
7	0.639	0.653	0.692	0.637	0.636	0.641	0.641	0.653
8	0.603	0.602	0.620	0.576	0.588	0.574	0.576	0.601
9	0.862	0.915	0.896	0.913	0.853	0.832	0.825	0.914
10	1.428	1.466	1.767	1.580	1.567	1.508	1.557	1.465
11	0.607	0.620	0.628	0.609	0.586	0.579	0.590	0.619
12	0.968	1.001	1.112	1.032	1.009	0.994	1.002	1.001
13	1.319	1.331	1.362	1.293	1.296	1.299	1.310	1.331
14	0.881	0.889	0.949	0.935	0.942	0.896	0.931	0.888
15	1.083	1.091	1.118	1.068	1.071	1.062	1.065	1.090

Table C.4 Comparison of SE for the item discrimination among algorithms (2PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.108	0.141	0.204	0.185	0.141	0.152	0.157	0.136
2	0.070	0.083	0.124	0.120	0.098	0.091	0.102	0.082
3	0.061	0.073	0.160	0.125	0.118	0.112	0.109	0.073
4	0.112	0.180	0.525	0.369	0.305	0.328	0.267	0.180
5	0.071	0.095	0.355	0.289	0.285	0.287	0.304	0.093
6	0.065	0.080	0.113	0.094	0.090	0.088	0.091	0.079
7	0.069	0.086	0.172	0.148	0.126	0.137	0.134	0.086
8	0.066	0.079	0.110	0.103	0.088	0.097	0.084	0.078
9	0.110	0.154	0.201	0.202	0.181	0.172	0.160	0.153
10	0.153	0.245	0.538	0.378	0.324	0.305	0.333	0.241
11	0.061	0.077	0.116	0.122	0.104	0.102	0.099	0.075
12	0.103	0.146	0.287	0.260	0.206	0.226	0.230	0.143
13	0.109	0.147	0.229	0.184	0.174	0.159	0.163	0.143
14	0.088	0.107	0.240	0.204	0.198	0.160	0.190	0.104
15	0.090	0.130	0.179	0.159	0.143	0.136	0.139	0.125

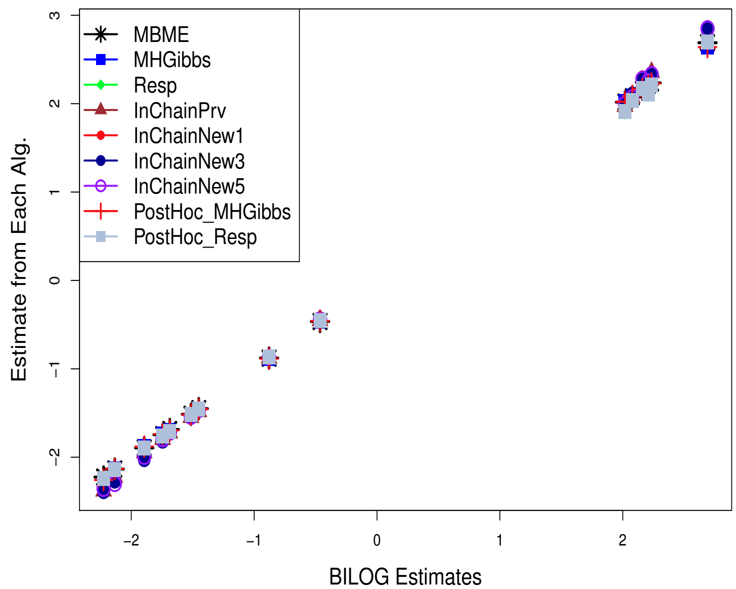

Figure C.5 Comparison of estimates for the item difficulty among algorithms (2PL model with 1000 examinees and 15 items)

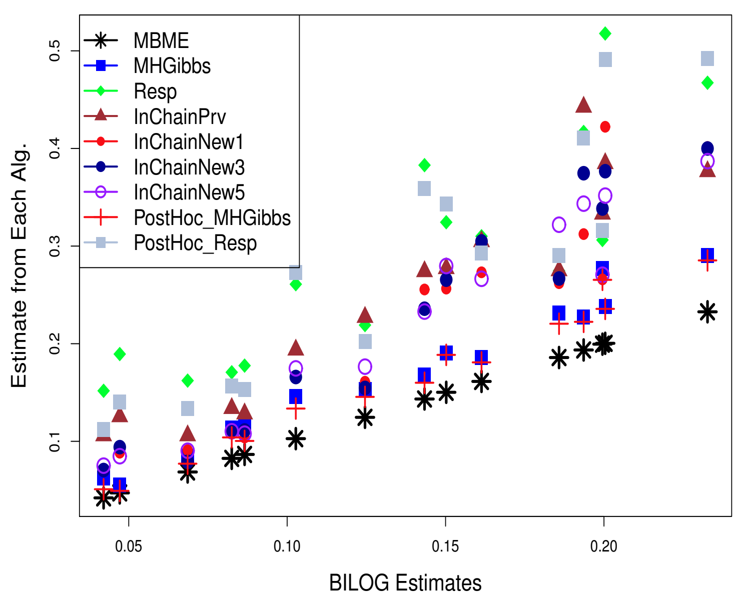

Figure C.6 Comparison of SE for the item difficulty among algorithms (2PL model with 1000 examinees and 15 items)

Table C.5 Comparison of estimates for the item difficulty among algorithms (2PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-1.687	-1.696	-1.708	-1.724	-1.716	-1.728	-1.720	-1.696
2	-1.745	-1.739	-1.761	-1.798	-1.803	-1.821	-1.794	-1.739
3	2.160	2.164	2.165	2.224	2.311	2.288	2.278	2.164
4	-0.879	-0.884	-0.866	-0.877	-0.876	-0.866	-0.869	-0.884
5	-0.464	-0.459	-0.457	-0.452	-0.454	-0.448	-0.441	-0.459
6	-2.134	-2.132	-2.135	-2.224	-2.218	-2.273	-2.301	-2.132
7	2.234	2.229	2.201	2.353	2.308	2.340	2.331	2.229
8	-2.225	-2.258	-2.237	-2.381	-2.334	-2.392	-2.356	-2.258
9	2.690	2.640	2.685	2.675	2.797	2.862	2.849	2.640
10	2.018	2.022	1.900	1.970	1.968	2.011	1.995	2.022
11	-1.895	-1.879	-1.897	-1.947	-2.007	-2.029	-1.994	-1.879
12	2.208	2.200	2.102	2.217	2.216	2.255	2.252	2.201
13	-1.451	-1.455	-1.454	-1.481	-1.483	-1.472	-1.464	-1.455
14	2.078	2.088	2.030	2.070	2.056	2.111	2.069	2.087
15	-1.514	-1.516	-1.517	-1.548	-1.538	-1.540	-1.541	-1.516

Table C.6 Comparison of SE for the item difficulty among algorithms (2PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.087	0.117	0.178	0.128	0.104	0.111	0.108	0.100
2	0.125	0.153	0.219	0.227	0.161	0.155	0.176	0.146
3	0.200	0.238	0.518	0.385	0.422	0.377	0.352	0.236
4	0.042	0.062	0.152	0.106	0.069	0.071	0.075	0.051
5	0.047	0.055	0.189	0.125	0.088	0.094	0.085	0.049
6	0.186	0.231	0.275	0.275	0.262	0.267	0.322	0.220
7	0.194	0.227	0.417	0.442	0.312	0.375	0.343	0.222
8	0.200	0.277	0.306	0.333	0.266	0.338	0.271	0.265
9	0.233	0.290	0.467	0.376	0.401	0.400	0.387	0.285
10	0.103	0.146	0.261	0.194	0.165	0.166	0.175	0.134
11	0.161	0.186	0.310	0.305	0.273	0.305	0.267	0.181
12	0.150	0.190	0.324	0.277	0.256	0.266	0.280	0.189
13	0.069	0.085	0.162	0.106	0.091	0.079	0.090	0.077
14	0.143	0.168	0.383	0.274	0.256	0.236	0.233	0.160
15	0.083	0.114	0.171	0.134	0.112	0.112	0.110	0.104

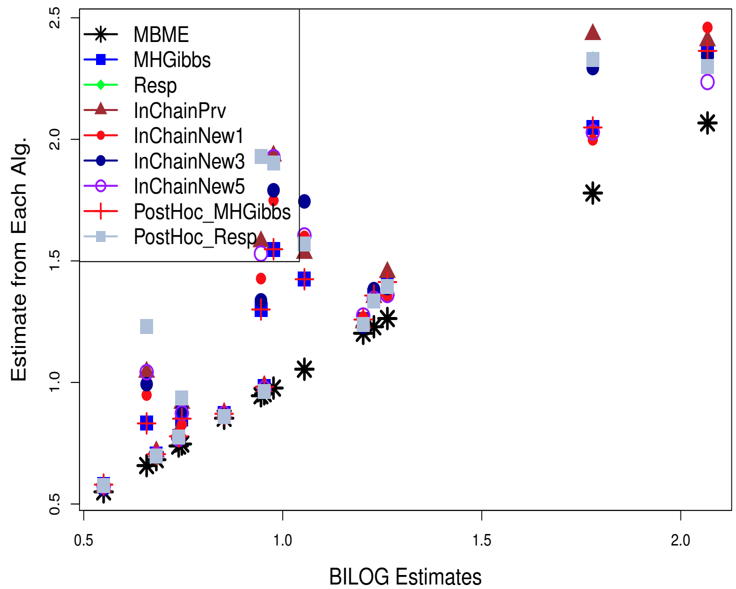

Figure C.7 Comparison of estimates for the item discrimination among algorithms (3PL model with 1000 examinees and 15 items)

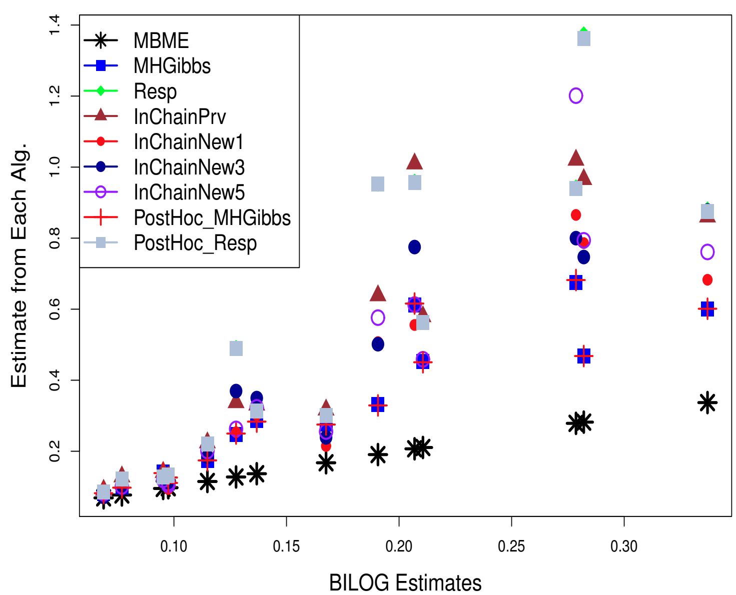

Figure C.8 Comparison of SE for the item discrimination among algorithms (3PL model with 1000 examinees and 15 items)

Table C.7 Comparison of estimates for the item discrimination among algorithms (3PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.945	1.300	1.932	1.579	1.427	1.338	1.530	1.300
2	0.853	0.871	0.860	0.872	0.864	0.870	0.870	0.871
3	0.682	0.705	0.698	0.715	0.710	0.698	0.703	0.705
4	0.954	0.982	0.964	0.984	0.963	0.975	0.975	0.982
5	1.054	1.425	1.573	1.532	1.601	1.745	1.605	1.425
6	0.550	0.580	0.577	0.569	0.573	0.561	0.574	0.580
7	0.747	0.850	0.938	0.915	0.827	0.891	0.874	0.851
8	0.739	0.778	0.776	0.797	0.789	0.767	0.768	0.778
9	1.229	1.358	1.334	1.350	1.359	1.384	1.344	1.357
10	0.658	0.832	1.231	1.043	0.948	0.993	1.042	0.832
11	1.263	1.413	1.397	1.453	1.357	1.388	1.360	1.413
12	2.067	2.364	2.306	2.406	2.460	2.341	2.236	2.364
13	1.202	1.259	1.241	1.246	1.270	1.230	1.275	1.259
14	0.977	1.547	1.905	1.932	1.749	1.790	1.926	1.548
15	1.779	2.049	2.332	2.432	1.998	2.293	2.029	2.049

Table C.8 Comparison of SE for the item discrimination among algorithms (3PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.282	0.468	1.376	0.966	0.786	0.747	0.794	0.468
2	0.097	0.126	0.128	0.114	0.094	0.106	0.105	0.126
3	0.077	0.099	0.117	0.128	0.104	0.095	0.101	0.098
4	0.095	0.141	0.132	0.141	0.113	0.119	0.120	0.139
5	0.337	0.600	0.883	0.860	0.683	0.878	0.761	0.601
6	0.069	0.081	0.086	0.091	0.075	0.071	0.082	0.082
7	0.128	0.247	0.493	0.337	0.254	0.370	0.263	0.250
8	0.097	0.111	0.129	0.127	0.111	0.114	0.110	0.111
9	0.137	0.287	0.303	0.331	0.302	0.350	0.324	0.283
10	0.191	0.331	0.951	0.638	0.507	0.501	0.576	0.329
11	0.168	0.277	0.304	0.317	0.215	0.239	0.256	0.275
12	0.211	0.453	0.576	0.579	0.561	0.562	0.459	0.451
13	0.115	0.174	0.221	0.224	0.213	0.184	0.201	0.174
14	0.279	0.675	0.945	1.020	0.865	0.800	1.201	0.682
15	0.207	0.612	0.961	1.010	0.555	0.775	0.613	0.616

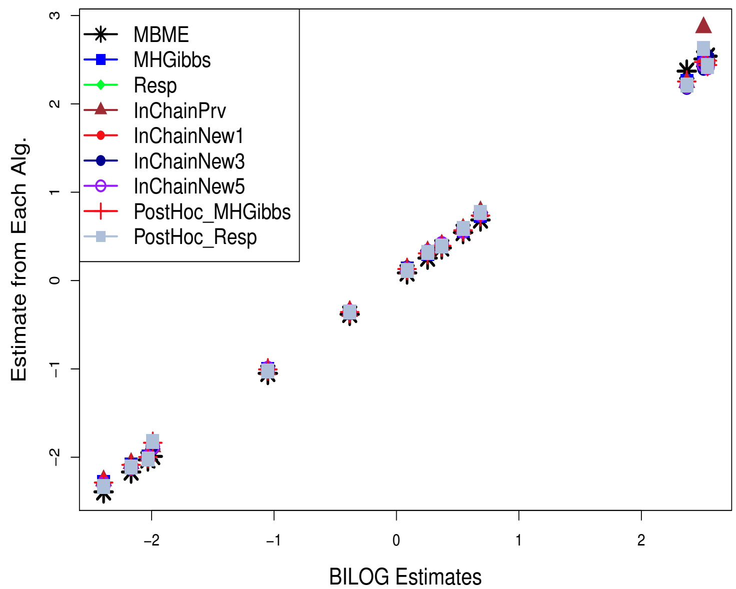

Figure C.9 Comparison of estimates for the item difficulty among algorithms (3PL model with 1000 examinees and 15 items)

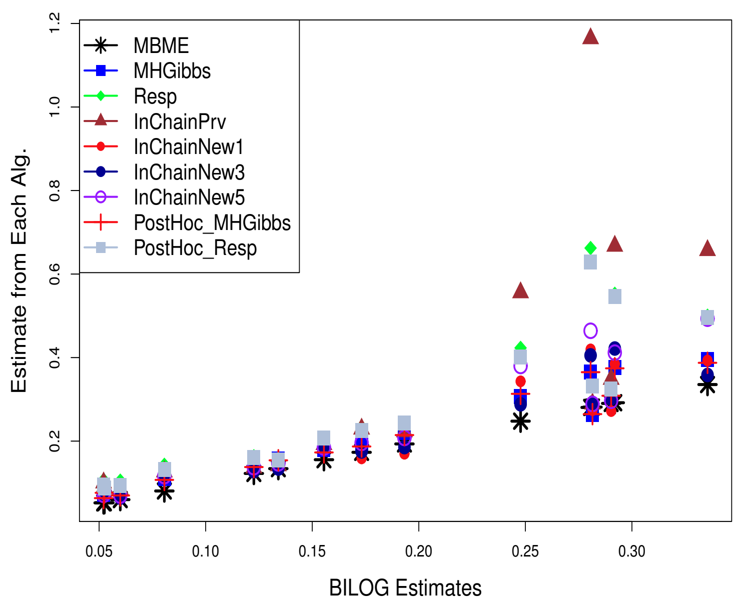

Figure C.10 Comparison of SE for the item difficulty among algorithms (3PL model with 1000 examinees and 15 items)

Table C.9 Comparison of estimates for the item difficulty among algorithms (3PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	2.540	2.439	2.423	2.555	2.471	2.535	2.417	2.439
2	-2.030	-1.998	-2.018	-1.985	-1.978	-1.993	-2.003	-1.998
3	0.254	0.309	0.320	0.328	0.330	0.313	0.323	0.308
4	-1.051	-1.005	-1.024	-1.007	-1.016	-0.998	-1.002	-1.005
5	2.508	2.473	2.631	2.865	2.439	2.400	2.450	2.473
6	-1.990	-1.836	-1.819	-1.879	-1.880	-1.895	-1.878	-1.836
7	0.686	0.737	0.768	0.781	0.744	0.745	0.757	0.737
8	-2.392	-2.286	-2.328	-2.264	-2.259	-2.325	-2.311	-2.286
9	0.086	0.131	0.112	0.134	0.140	0.145	0.124	0.131
10	2.539	2.490	2.434	2.524	2.482	2.444	2.445	2.490
11	-2.167	-2.086	-2.107	-2.069	-2.101	-2.092	-2.115	-2.086
12	-0.382	-0.356	-0.359	-0.346	-0.338	-0.356	-0.357	-0.357
13	0.370	0.394	0.390	0.402	0.398	0.401	0.410	0.394
14	2.371	2.252	2.214	2.244	2.238	2.184	2.206	2.254
15	0.544	0.569	0.587	0.583	0.580	0.593	0.567	0.569

Table C.10 Comparison of SE for the item difficulty among algorithms (3PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.292	0.376	0.553	0.667	0.385	0.422	0.412	0.374
2	0.193	0.220	0.239	0.223	0.170	0.186	0.208	0.214
3	0.123	0.140	0.164	0.136	0.126	0.128	0.138	0.138
4	0.134	0.157	0.159	0.141	0.144	0.134	0.147	0.154
5	0.281	0.366	0.662	1.164	0.420	0.405	0.464	0.365
6	0.290	0.313	0.329	0.348	0.272	0.302	0.297	0.309
7	0.155	0.180	0.203	0.192	0.193	0.189	0.184	0.173
8	0.281	0.263	0.329	0.289	0.274	0.290	0.289	0.265
9	0.081	0.110	0.143	0.127	0.110	0.119	0.116	0.107
10	0.335	0.395	0.500	0.657	0.393	0.358	0.493	0.388
11	0.173	0.195	0.234	0.229	0.159	0.175	0.196	0.187
12	0.052	0.079	0.096	0.091	0.072	0.073	0.078	0.076
13	0.060	0.082	0.105	0.085	0.070	0.065	0.070	0.070
14	0.248	0.306	0.423	0.556	0.343	0.288	0.380	0.313
15	0.052	0.072	0.107	0.100	0.068	0.070	0.071	0.063

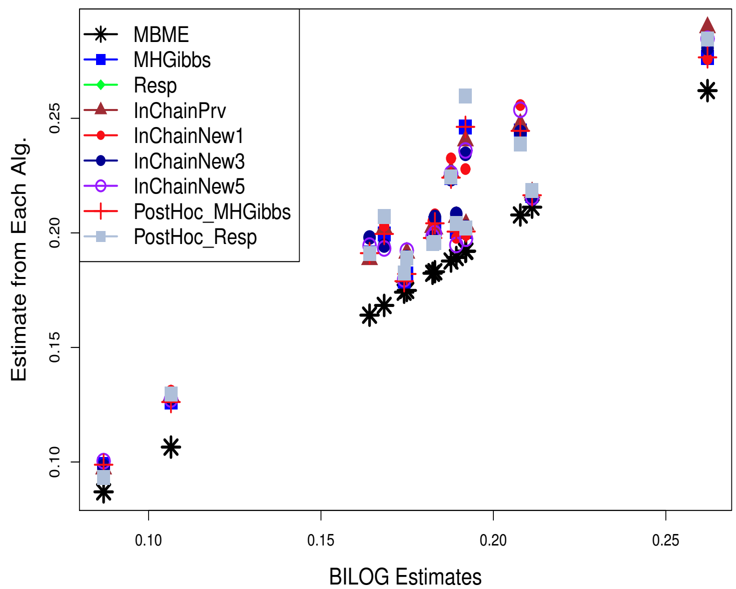

Figure C.11 Comparison of estimates for the item guessing among algorithms (3PL model with 1000 examinees and 15 items)

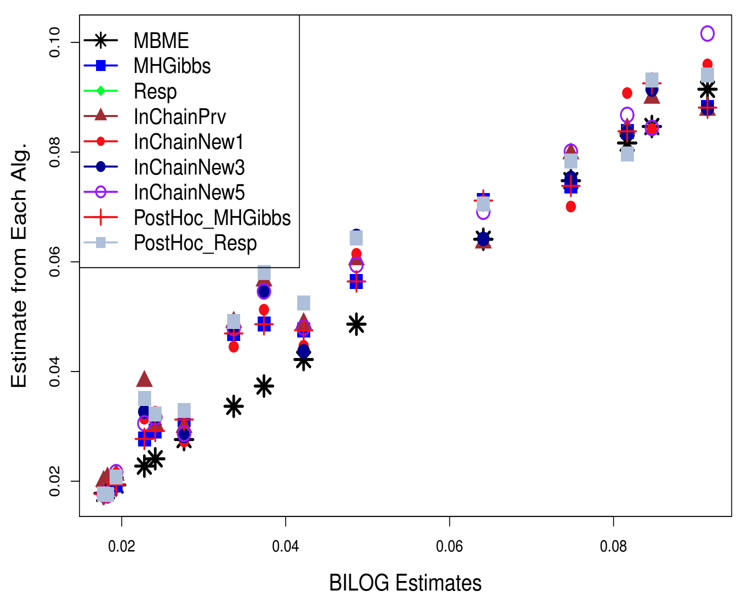

Figure C.12 Comparison of SE for the item guessing among algorithms (3PL model with 1000 examinees and 15 items)

Table C.11 Comparison of estimates for the item guessing among algorithms (3PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.174	0.179	0.182	0.181	0.181	0.180	0.180	0.179
2	0.168	0.200	0.207	0.202	0.203	0.194	0.193	0.200
3	0.106	0.126	0.130	0.129	0.131	0.127	0.127	0.126
4	0.164	0.191	0.191	0.188	0.192	0.198	0.194	0.191
5	0.211	0.216	0.218	0.218	0.217	0.215	0.216	0.216
6	0.192	0.246	0.260	0.240	0.228	0.235	0.236	0.246
7	0.262	0.277	0.285	0.290	0.276	0.281	0.285	0.277
8	0.208	0.245	0.239	0.247	0.256	0.244	0.254	0.245
9	0.183	0.204	0.196	0.202	0.208	0.207	0.200	0.204
10	0.175	0.182	0.189	0.191	0.189	0.189	0.192	0.182
11	0.188	0.224	0.224	0.227	0.233	0.223	0.226	0.224
12	0.182	0.198	0.195	0.202	0.205	0.199	0.198	0.198
13	0.087	0.099	0.093	0.097	0.098	0.098	0.100	0.099
14	0.192	0.201	0.202	0.204	0.199	0.203	0.197	0.201
15	0.189	0.201	0.204	0.207	0.198	0.208	0.195	0.201

Table C.12 Comparison of SE for the item guessing among algorithms (3PL model with 1000 examinees and 15 items)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.018	0.019	0.018	0.021	0.017	0.018	0.018	0.019
2	0.075	0.074	0.078	0.080	0.070	0.075	0.080	0.074
3	0.042	0.048	0.053	0.049	0.045	0.044	0.048	0.048
4	0.064	0.071	0.071	0.063	0.071	0.064	0.069	0.071
5	0.018	0.018	0.018	0.020	0.018	0.017	0.018	0.018
6	0.085	0.093	0.093	0.090	0.084	0.091	0.084	0.093
7	0.049	0.056	0.064	0.060	0.061	0.065	0.059	0.056
8	0.091	0.088	0.094	0.088	0.096	0.088	0.102	0.088
9	0.037	0.049	0.058	0.057	0.051	0.055	0.055	0.049
10	0.028	0.031	0.033	0.030	0.027	0.028	0.029	0.031
11	0.082	0.084	0.080	0.084	0.091	0.084	0.087	0.084
12	0.034	0.047	0.049	0.049	0.045	0.048	0.048	0.047
13	0.024	0.029	0.032	0.030	0.033	0.031	0.032	0.029
14	0.019	0.019	0.021	0.021	0.021	0.020	0.022	0.019
15	0.023	0.028	0.035	0.038	0.031	0.033	0.031	0.028

Appendix D

Result Tables and Graphs for 1000 examinees and 30 items

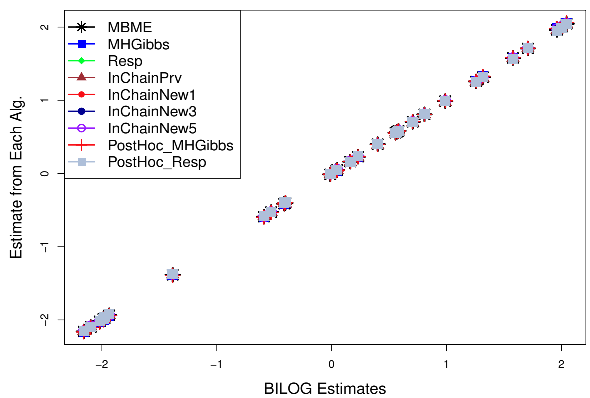

Figure D.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items)

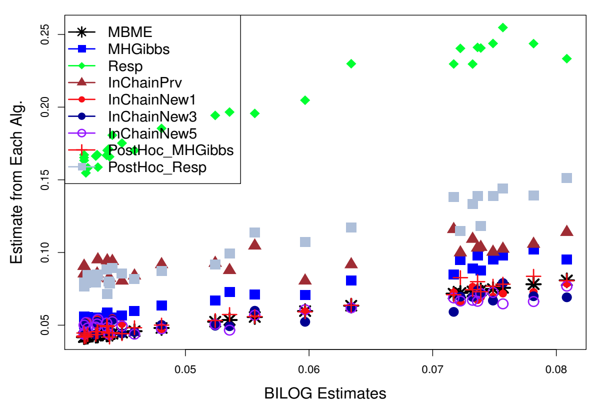

Figure D.2 Comparison of SE for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items)

Table D.1 Comparison of estimates for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-0.413	-0.416	-0.407	-0.407	-0.414	-0.400	-0.405	-0.416
2	0.989	0.993	0.996	0.976	0.992	0.990	0.995	0.993
3	-2.158	-2.164	-2.153	-2.154	-2.166	-2.149	-2.148	-2.164
4	0.042	0.037	0.055	0.044	0.037	0.053	0.051	0.037
5	-1.976	-1.982	-1.962	-1.977	-1.979	-1.960	-1.969	-1.982
6	0.162	0.164	0.158	0.157	0.167	0.162	0.168	0.164
7	0.582	0.580	0.584	0.585	0.587	0.588	0.587	0.580
8	1.257	1.262	1.246	1.253	1.270	1.250	1.256	1.262
9	0.400	0.398	0.409	0.390	0.402	0.398	0.404	0.398
10	0.556	0.553	0.556	0.559	0.557	0.556	0.559	0.553
11	1.961	1.982	1.953	1.956	1.975	1.963	1.976	1.982
12	0.706	0.709	0.709	0.696	0.712	0.710	0.708	0.709
13	-0.527	-0.532	-0.520	-0.527	-0.525	-0.521	-0.523	-0.532
14	-1.987	-2.002	-1.985	-1.979	-1.989	-1.971	-1.987	-2.002
15	1.316	1.328	1.320	1.306	1.329	1.322	1.317	1.328

Table D.2 Comparison of estimates for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	-2.097	-2.103	-2.087	-2.091	-2.100	-2.090	-2.083	-2.103
17	1.708	1.711	1.713	1.709	1.717	1.707	1.713	1.711
18	2.045	2.062	2.037	2.040	2.054	2.050	2.051	2.062
19	-0.590	-0.597	-0.579	-0.587	-0.590	-0.583	-0.589	-0.597
20	-1.936	-1.941	-1.921	-1.933	-1.941	-1.928	-1.933	-1.941
21	0.808	0.808	0.808	0.806	0.811	0.816	0.814	0.808
22	1.578	1.582	1.569	1.576	1.588	1.577	1.581	1.582
23	0.230	0.229	0.234	0.218	0.228	0.238	0.235	0.229
24	-2.018	-2.033	-2.009	-2.016	-2.026	-2.005	-2.015	-2.033
25	-0.012	-0.009	-0.005	-0.009	-0.011	-0.008	-0.008	-0.009
26	2.002	2.004	1.990	1.995	2.012	2.000	2.007	2.005
27	-0.402	-0.396	-0.391	-0.413	-0.402	-0.400	-0.399	-0.396
28	0.571	0.573	0.573	0.571	0.570	0.572	0.571	0.573
29	0.048	0.046	0.050	0.045	0.051	0.055	0.056	0.046
30	-1.384	-1.393	-1.371	-1.381	-1.381	-1.370	-1.377	-1.393

Table D.3 Comparison of SE for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.043	0.055	0.159	0.095	0.052	0.050	0.054	0.045
2	0.048	0.064	0.185	0.092	0.047	0.050	0.047	0.050
3	0.081	0.095	0.233	0.114	0.079	0.069	0.077	0.081
4	0.042	0.051	0.167	0.091	0.048	0.046	0.044	0.042
5	0.073	0.089	0.230	0.109	0.076	0.069	0.067	0.074
6	0.042	0.050	0.155	0.085	0.045	0.047	0.052	0.042
7	0.044	0.054	0.166	0.082	0.047	0.046	0.044	0.042
8	0.052	0.067	0.194	0.093	0.050	0.050	0.051	0.053
9	0.043	0.055	0.166	0.083	0.044	0.043	0.049	0.044
10	0.044	0.059	0.170	0.094	0.049	0.046	0.048	0.049
11	0.072	0.095	0.240	0.100	0.066	0.070	0.067	0.083
12	0.045	0.057	0.175	0.081	0.051	0.044	0.049	0.044
13	0.044	0.057	0.167	0.089	0.050	0.051	0.052	0.050
14	0.074	0.098	0.241	0.103	0.067	0.074	0.066	0.080
15	0.054	0.073	0.197	0.088	0.049	0.049	0.047	0.057

Table D.4 Comparison of SE for the item difficulty among algorithms (Rasch model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.078	0.102	0.244	0.106	0.073	0.070	0.066	0.084
17	0.063	0.081	0.230	0.092	0.061	0.062	0.062	0.064
18	0.076	0.098	0.255	0.102	0.072	0.079	0.065	0.078
19	0.044	0.055	0.181	0.094	0.055	0.054	0.052	0.045
20	0.072	0.085	0.230	0.116	0.073	0.059	0.069	0.071
21	0.046	0.060	0.170	0.084	0.045	0.045	0.044	0.048
22	0.060	0.071	0.205	0.081	0.060	0.052	0.061	0.059
23	0.042	0.056	0.158	0.080	0.053	0.045	0.048	0.044
24	0.075	0.095	0.244	0.100	0.070	0.067	0.073	0.077
25	0.042	0.056	0.165	0.091	0.050	0.050	0.045	0.045
26	0.074	0.087	0.241	0.104	0.069	0.071	0.072	0.073
27	0.043	0.053	0.167	0.085	0.054	0.052	0.051	0.046
28	0.044	0.055	0.171	0.082	0.047	0.046	0.044	0.045
29	0.042	0.051	0.163	0.085	0.051	0.051	0.050	0.043
30	0.056	0.071	0.196	0.105	0.058	0.060	0.057	0.056

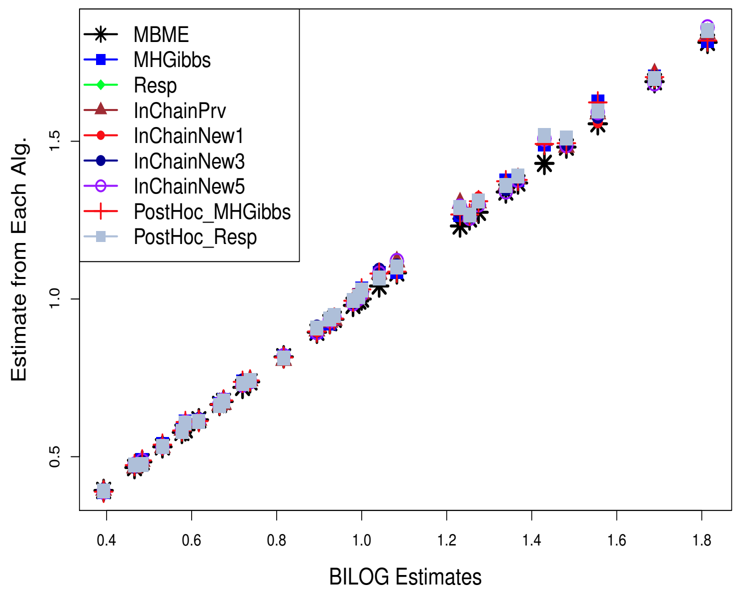

Figure D.3 Comparison of estimates for the item discrimination among algorithms (2PL model with 1000 examinees and 30 items)

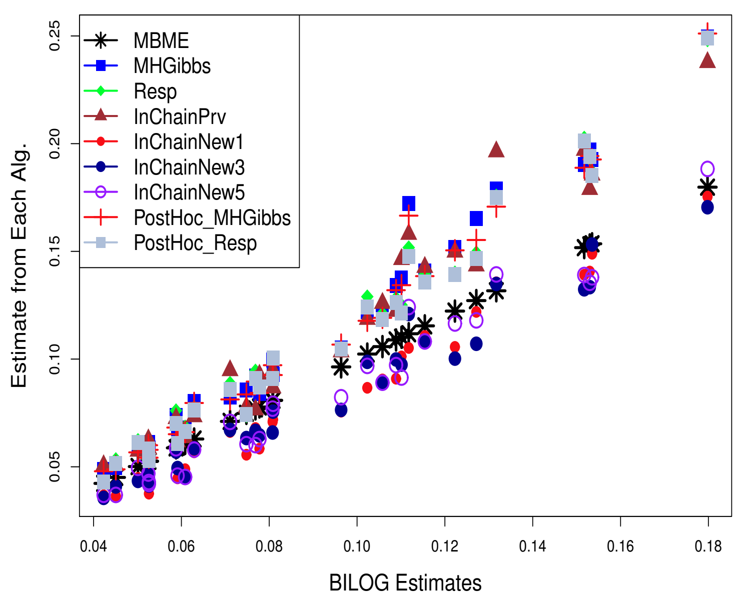

Figure D.4 Comparison of SE for the item discrimination among algorithms (2PL model with 1000 examinees and 30 items)

Table D.5 Comparison of estimates for the item discrimination among algorithms (2PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.531	0.538	0.533	0.529	0.534	0.538	0.533	0.538
2	1.555	1.624	1.597	1.587	1.559	1.579	1.593	1.623
3	1.041	1.081	1.068	1.080	1.089	1.092	1.080	1.081
4	1.814	1.820	1.852	1.841	1.843	1.850	1.861	1.819
5	1.689	1.704	1.700	1.713	1.694	1.703	1.684	1.703
6	0.720	0.737	0.733	0.734	0.732	0.741	0.729	0.737
7	1.275	1.310	1.311	1.305	1.323	1.309	1.300	1.310
8	0.577	0.583	0.579	0.584	0.579	0.578	0.583	0.583
9	1.430	1.492	1.519	1.503	1.504	1.505	1.508	1.491
10	1.000	1.031	1.028	1.017	1.031	1.028	1.018	1.030
11	0.675	0.679	0.678	0.666	0.682	0.678	0.678	0.678
12	1.083	1.087	1.102	1.118	1.098	1.106	1.120	1.087
13	0.980	0.995	0.995	0.992	0.979	0.987	0.987	0.994
14	1.231	1.268	1.290	1.304	1.289	1.274	1.292	1.268
15	1.482	1.495	1.511	1.482	1.497	1.490	1.488	1.494

Table D.6 Comparison of estimates for the item discrimination among algorithms (2PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.895	0.897	0.909	0.899	0.896	0.912	0.895	0.897
17	0.936	0.938	0.949	0.933	0.944	0.945	0.942	0.937
18	0.466	0.473	0.474	0.465	0.471	0.469	0.474	0.473
19	0.585	0.610	0.607	0.607	0.595	0.608	0.609	0.610
20	0.617	0.613	0.612	0.616	0.616	0.619	0.613	0.613
21	0.993	1.011	1.008	1.014	1.007	1.017	1.003	1.011
22	1.339	1.374	1.361	1.362	1.355	1.350	1.341	1.373
23	1.254	1.268	1.266	1.262	1.267	1.263	1.258	1.268
24	0.484	0.488	0.476	0.486	0.485	0.488	0.486	0.488
25	0.925	0.924	0.937	0.927	0.924	0.932	0.933	0.923
26	1.368	1.378	1.391	1.382	1.379	1.383	1.377	1.377
27	0.816	0.815	0.814	0.806	0.819	0.815	0.819	0.815
28	0.737	0.742	0.742	0.741	0.745	0.737	0.740	0.742
29	0.666	0.666	0.663	0.663	0.662	0.667	0.666	0.666
30	0.393	0.389	0.392	0.394	0.389	0.395	0.390	0.389

Table D.7 Comparison of SE for the item discrimination among algorithms (2PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.053	0.055	0.056	0.056	0.042	0.045	0.043	0.054
2	0.153	0.197	0.193	0.179	0.141	0.133	0.135	0.194
3	0.122	0.152	0.140	0.150	0.106	0.100	0.117	0.150
4	0.132	0.179	0.176	0.196	0.136	0.135	0.139	0.171
5	0.112	0.172	0.152	0.158	0.105	0.121	0.124	0.167
6	0.078	0.084	0.087	0.093	0.058	0.065	0.063	0.084
7	0.152	0.190	0.203	0.197	0.139	0.132	0.139	0.189
8	0.059	0.067	0.061	0.062	0.045	0.049	0.046	0.065
9	0.180	0.250	0.248	0.238	0.175	0.170	0.188	0.251
10	0.109	0.134	0.128	0.122	0.091	0.100	0.097	0.132
11	0.061	0.068	0.069	0.064	0.049	0.045	0.045	0.066
12	0.115	0.141	0.139	0.143	0.111	0.108	0.108	0.139
13	0.081	0.099	0.093	0.095	0.071	0.066	0.077	0.097
14	0.154	0.193	0.187	0.186	0.149	0.153	0.138	0.193
15	0.127	0.165	0.149	0.143	0.122	0.107	0.118	0.155

Table D.8 Comparison of SE for the item discrimination among algorithms (2PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.081	0.093	0.099	0.087	0.072	0.076	0.079	0.093
17	0.077	0.092	0.095	0.076	0.068	0.067	0.060	0.090
18	0.053	0.059	0.053	0.058	0.038	0.041	0.042	0.058
19	0.075	0.086	0.074	0.078	0.056	0.063	0.061	0.084
20	0.050	0.059	0.062	0.057	0.045	0.043	0.050	0.057
21	0.096	0.105	0.105	0.103	0.076	0.076	0.082	0.107
22	0.110	0.138	0.125	0.146	0.101	0.097	0.091	0.134
23	0.106	0.120	0.121	0.126	0.091	0.089	0.089	0.119
24	0.045	0.050	0.053	0.051	0.036	0.041	0.037	0.049
25	0.071	0.082	0.089	0.095	0.066	0.067	0.071	0.081
26	0.102	0.122	0.129	0.119	0.087	0.099	0.097	0.118
27	0.059	0.074	0.076	0.072	0.061	0.057	0.059	0.068
28	0.063	0.080	0.078	0.073	0.057	0.059	0.058	0.080
29	0.052	0.061	0.060	0.063	0.046	0.047	0.047	0.060
30	0.042	0.049	0.043	0.051	0.037	0.036	0.037	0.048

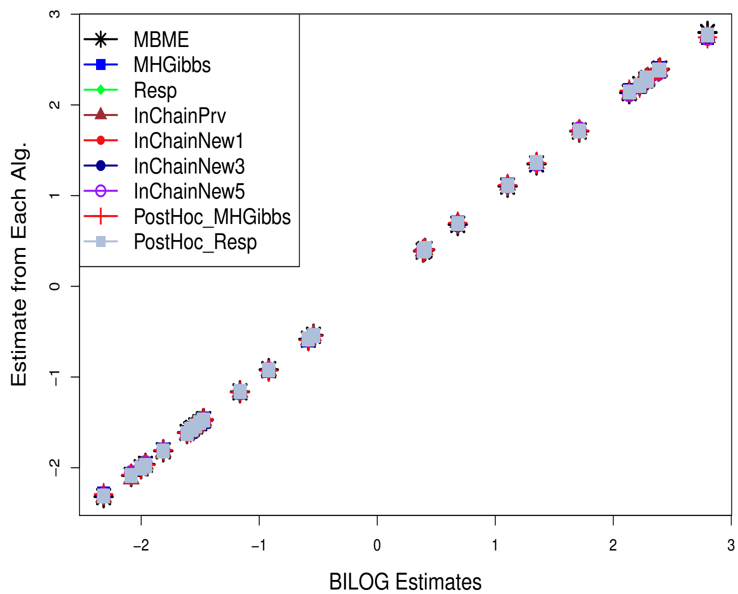

Figure D.5 Comparison of estimates for the item difficulty among algorithms (2PL model with 1000 examinees and 30 items)

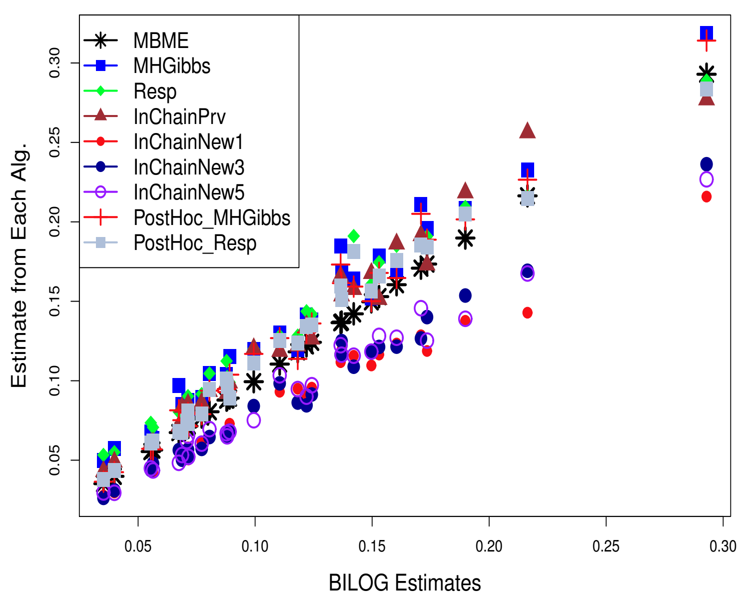

Figure D.6 Comparison of SE for the item difficulty among algorithms (2PL model with 1000 examinees and 30 items)

Table D.9 Comparison of estimates for the item difficulty among algorithms (2PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	-1.612	-1.611	-1.624	-1.629	-1.608	-1.613	-1.619	-1.611
2	-1.813	-1.805	-1.813	-1.823	-1.823	-1.813	-1.809	-1.805
3	2.384	2.385	2.384	2.370	2.354	2.344	2.357	2.385
4	-0.919	-0.930	-0.921	-0.927	-0.921	-0.920	-0.920	-0.930
5	-0.540	-0.544	-0.543	-0.541	-0.545	-0.543	-0.544	-0.544
6	-2.318	-2.291	-2.312	-2.321	-2.302	-2.288	-2.305	-2.291
7	2.275	2.279	2.296	2.289	2.260	2.269	2.283	2.279
8	-1.964	-1.969	-1.976	-1.959	-1.963	-1.979	-1.951	-1.969
9	2.291	2.297	2.269	2.281	2.275	2.266	2.268	2.297
10	2.224	2.201	2.212	2.234	2.201	2.203	2.212	2.201
11	-1.581	-1.602	-1.579	-1.603	-1.572	-1.580	-1.581	-1.601
12	2.135	2.158	2.143	2.119	2.146	2.130	2.121	2.158
13	-1.504	-1.503	-1.502	-1.497	-1.512	-1.509	-1.501	-1.503
14	2.395	2.400	2.385	2.379	2.369	2.377	2.371	2.400
15	-1.539	-1.547	-1.537	-1.552	-1.547	-1.543	-1.543	-1.546

Table D.10 Comparison of estimates for the item difficulty among algorithms (2PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	1.712	1.715	1.712	1.725	1.724	1.706	1.731	1.716
17	-1.475	-1.488	-1.476	-1.492	-1.482	-1.472	-1.477	-1.488
18	-2.085	-2.078	-2.087	-2.121	-2.078	-2.084	-2.061	-2.078
19	2.799	2.745	2.770	2.745	2.780	2.737	2.742	2.745
20	0.388	0.388	0.391	0.391	0.395	0.394	0.398	0.388
21	-2.001	-2.005	-2.002	-2.000	-2.001	-1.981	-1.999	-2.005
22	-1.470	-1.465	-1.472	-1.475	-1.472	-1.469	-1.477	-1.465
23	-1.567	-1.569	-1.573	-1.578	-1.572	-1.572	-1.573	-1.569
24	0.406	0.409	0.415	0.405	0.407	0.410	0.402	0.409
25	1.105	1.117	1.109	1.119	1.116	1.107	1.111	1.117
26	-1.163	-1.165	-1.160	-1.163	-1.167	-1.162	-1.167	-1.165
27	0.400	0.403	0.401	0.416	0.399	0.399	0.411	0.403
28	1.350	1.360	1.366	1.357	1.350	1.353	1.353	1.360
29	-0.584	-0.594	-0.583	-0.587	-0.582	-0.581	-0.584	-0.594
30	0.682	0.700	0.693	0.705	0.702	0.673	0.693	0.700

Table D.11 Comparison of SE for the item difficulty among algorithms (2PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.150	0.152	0.161	0.168	0.110	0.118	0.118	0.150
2	0.080	0.104	0.105	0.093	0.065	0.065	0.070	0.094
3	0.171	0.211	0.189	0.191	0.129	0.127	0.146	0.205
4	0.040	0.058	0.055	0.049	0.030	0.031	0.029	0.042
5	0.035	0.050	0.053	0.043	0.028	0.026	0.030	0.036
6	0.190	0.208	0.209	0.218	0.138	0.154	0.139	0.202
7	0.142	0.164	0.191	0.158	0.116	0.109	0.116	0.159
8	0.173	0.196	0.191	0.173	0.119	0.140	0.125	0.189
9	0.137	0.185	0.163	0.166	0.112	0.116	0.122	0.173
10	0.153	0.178	0.175	0.151	0.116	0.121	0.128	0.168
11	0.122	0.141	0.144	0.135	0.091	0.085	0.090	0.130
12	0.137	0.169	0.163	0.153	0.119	0.125	0.116	0.161
13	0.088	0.104	0.112	0.099	0.065	0.065	0.067	0.095
14	0.160	0.169	0.185	0.186	0.123	0.121	0.127	0.165
15	0.067	0.097	0.080	0.072	0.055	0.056	0.048	0.081

Table D.12 Comparison of SE for the item difficulty among algorithms (2PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.110	0.130	0.128	0.119	0.093	0.098	0.103	0.127
17	0.089	0.115	0.099	0.099	0.073	0.069	0.068	0.104
18	0.216	0.232	0.217	0.256	0.143	0.169	0.168	0.227
19	0.293	0.319	0.289	0.277	0.216	0.236	0.227	0.314
20	0.071	0.079	0.083	0.076	0.055	0.053	0.062	0.072
21	0.124	0.138	0.142	0.126	0.096	0.091	0.097	0.136
22	0.069	0.085	0.080	0.083	0.054	0.050	0.053	0.075
23	0.077	0.090	0.092	0.086	0.061	0.057	0.060	0.079
24	0.088	0.095	0.102	0.092	0.065	0.066	0.065	0.093
25	0.071	0.088	0.090	0.085	0.058	0.057	0.065	0.080
26	0.055	0.068	0.074	0.061	0.044	0.044	0.045	0.057
27	0.056	0.064	0.071	0.061	0.042	0.048	0.044	0.057
28	0.099	0.120	0.120	0.120	0.083	0.084	0.075	0.117
29	0.071	0.079	0.081	0.083	0.050	0.053	0.052	0.071
30	0.118	0.120	0.128	0.122	0.095	0.086	0.095	0.114

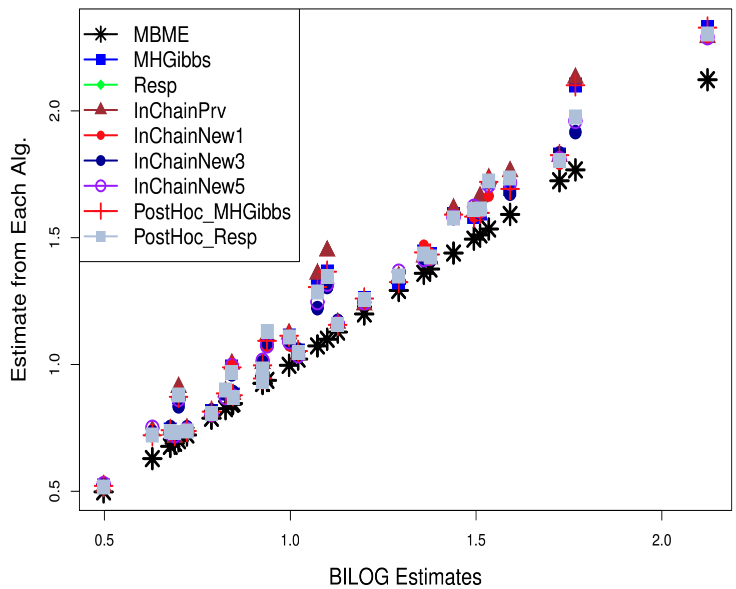

Figure D.7 Comparison of estimates for the item discrimination among algorithms (3PL model with 1000 examinees and 30 items)

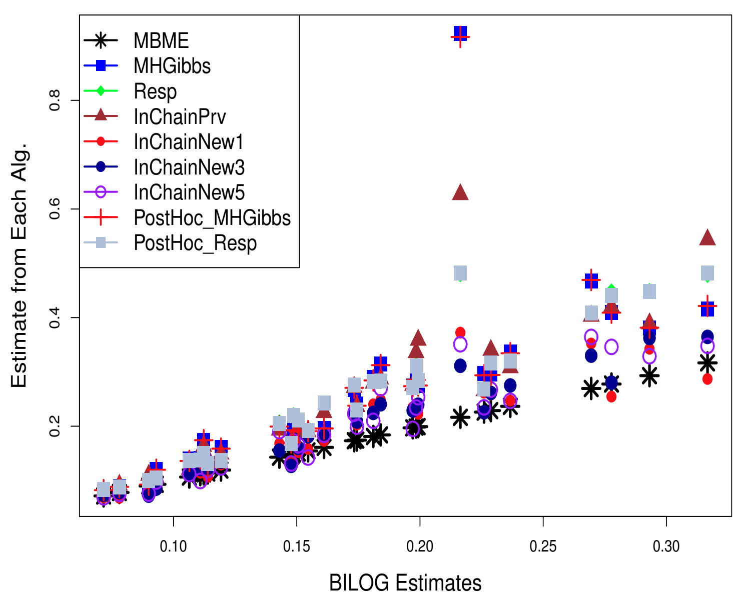

Figure D.8 Comparison of SE for the item discrimination among algorithms (3PL model with 1000 examinees and 30 items)

Table D.13 Comparison of estimates for the item discrimination among algorithms (3PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.700	0.873	0.880	0.910	0.853	0.835	0.864	0.872
2	1.494	1.583	1.612	1.603	1.582	1.607	1.621	1.583
3	0.498	0.521	0.518	0.519	0.529	0.524	0.528	0.521
4	0.722	0.738	0.738	0.747	0.741	0.752	0.742	0.738
5	1.073	1.306	1.287	1.357	1.216	1.222	1.246	1.305
6	0.788	0.814	0.808	0.814	0.802	0.819	0.805	0.814
7	1.725	1.826	1.806	1.821	1.790	1.833	1.809	1.825
8	1.592	1.693	1.735	1.761	1.678	1.673	1.717	1.692
9	1.021	1.053	1.047	1.039	1.052	1.059	1.037	1.053
10	0.937	1.094	1.129	1.113	1.071	1.087	1.076	1.093
11	1.439	1.591	1.578	1.613	1.582	1.582	1.580	1.591
12	0.926	0.946	0.934	0.965	0.957	0.967	0.961	0.945
13	1.376	1.434	1.424	1.417	1.440	1.423	1.420	1.434
14	0.629	0.721	0.722	0.739	0.743	0.733	0.750	0.721
15	1.292	1.326	1.350	1.346	1.335	1.340	1.366	1.325

Table D.14 Comparison of estimates for the item discrimination among algorithms (3PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	1.099	1.366	1.346	1.448	1.303	1.306	1.319	1.366
17	1.128	1.157	1.157	1.163	1.167	1.172	1.161	1.156
18	1.511	1.598	1.615	1.661	1.588	1.598	1.607	1.598
19	1.359	1.443	1.437	1.424	1.470	1.411	1.416	1.442
20	0.677	0.741	0.736	0.742	0.756	0.754	0.730	0.741
21	0.926	0.997	0.984	1.004	1.003	1.009	1.014	0.997
22	1.767	2.101	1.976	2.129	1.976	1.916	1.961	2.101
23	0.689	0.722	0.731	0.713	0.733	0.714	0.722	0.722
24	0.997	1.113	1.109	1.121	1.075	1.091	1.091	1.113
25	0.846	0.880	0.871	0.882	0.890	0.891	0.873	0.879
26	1.534	1.720	1.724	1.732	1.664	1.720	1.709	1.720
27	0.843	0.990	0.968	1.002	1.006	0.961	0.994	0.989
28	1.199	1.260	1.255	1.236	1.256	1.260	1.245	1.260
29	2.123	2.329	2.303	2.290	2.282	2.303	2.291	2.328
30	0.826	0.887	0.901	0.893	0.882	0.885	0.872	0.886

Table D.15 Comparison of SE for the item discrimination among algorithms (3PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.181	0.290	0.282	0.279	0.240	0.225	0.210	0.284
2	0.197	0.274	0.273	0.276	0.235	0.229	0.195	0.274
3	0.072	0.082	0.082	0.079	0.069	0.074	0.069	0.082
4	0.078	0.089	0.090	0.092	0.068	0.079	0.075	0.088
5	0.278	0.410	0.450	0.420	0.255	0.280	0.346	0.409
6	0.090	0.095	0.102	0.110	0.079	0.072	0.076	0.092
7	0.184	0.314	0.289	0.287	0.249	0.240	0.270	0.312
8	0.229	0.296	0.314	0.340	0.258	0.261	0.267	0.294
9	0.111	0.140	0.139	0.139	0.114	0.121	0.099	0.141
10	0.237	0.336	0.319	0.308	0.247	0.275	0.247	0.334
11	0.198	0.284	0.314	0.336	0.241	0.272	0.234	0.285
12	0.106	0.140	0.138	0.121	0.114	0.111	0.112	0.136
13	0.149	0.195	0.222	0.180	0.169	0.175	0.172	0.193
14	0.161	0.195	0.243	0.225	0.173	0.186	0.184	0.196
15	0.143	0.203	0.209	0.193	0.168	0.155	0.191	0.199

Table D.16 Comparison of SE for the item discrimination among algorithms (3PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.317	0.416	0.477	0.544	0.287	0.364	0.348	0.421
17	0.119	0.161	0.139	0.149	0.129	0.135	0.124	0.159
18	0.199	0.275	0.295	0.359	0.223	0.240	0.253	0.275
19	0.174	0.241	0.232	0.237	0.203	0.209	0.199	0.238
20	0.155	0.192	0.191	0.190	0.158	0.181	0.142	0.191
21	0.112	0.174	0.153	0.163	0.139	0.133	0.145	0.174
22	0.216	0.923	0.478	0.627	0.372	0.311	0.351	0.917
23	0.093	0.120	0.105	0.105	0.097	0.085	0.098	0.120
24	0.173	0.267	0.275	0.272	0.221	0.227	0.223	0.271
25	0.114	0.129	0.130	0.140	0.107	0.123	0.120	0.126
26	0.293	0.381	0.451	0.389	0.343	0.362	0.329	0.381
27	0.226	0.297	0.269	0.266	0.262	0.226	0.235	0.294
28	0.151	0.189	0.209	0.193	0.152	0.160	0.165	0.190
29	0.269	0.468	0.407	0.403	0.352	0.330	0.364	0.469
30	0.148	0.172	0.172	0.168	0.135	0.127	0.131	0.170

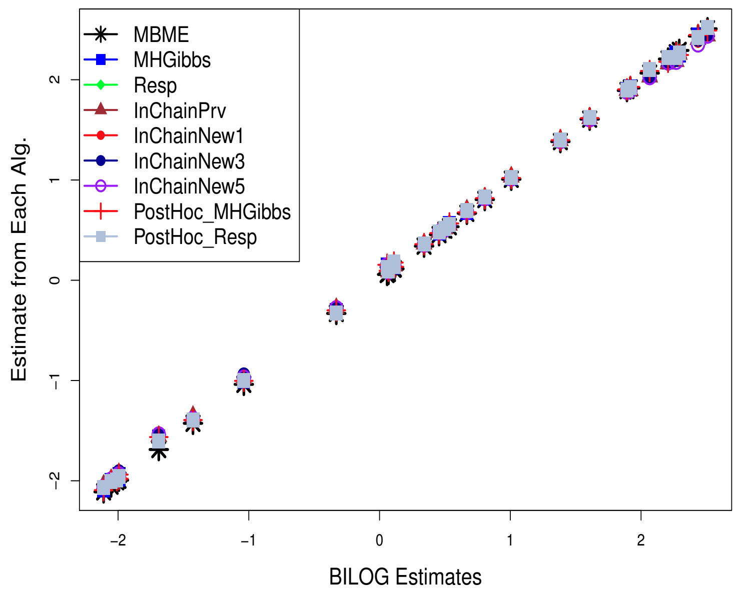

Figure D.9 Comparison of estimates for the item difficulty among algorithms (3PL model with 1000 examinees and 30 items)

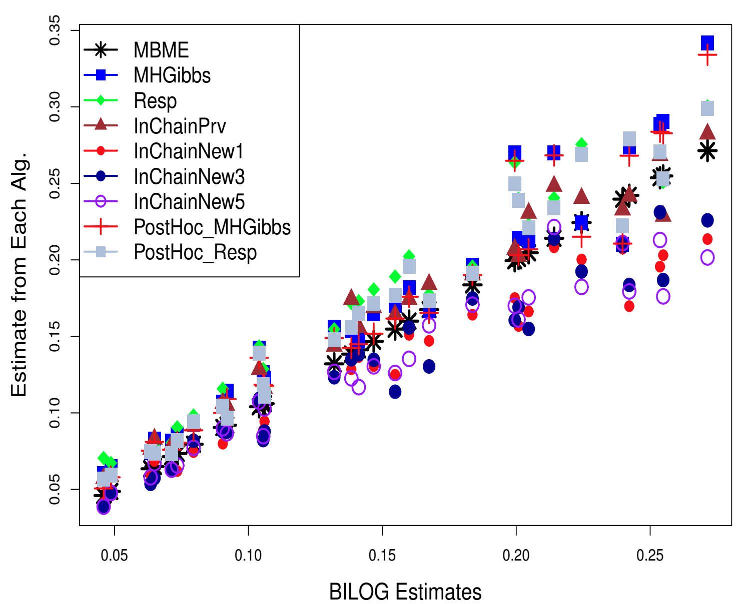

Figure D.10 Comparison of SE for the item difficulty among algorithms (3PL model with 1000 examinees and 30 items)

Table D.17 Comparison of estimates for the item difficulty among algorithms (3PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	2.269	2.266	2.220	2.184	2.198	2.199	2.180	2.266
2	-1.993	-1.980	-1.956	-1.941	-1.956	-1.949	-1.937	-1.980
3	0.058	0.154	0.126	0.134	0.171	0.145	0.148	0.154
4	-1.038	-1.004	-1.001	-0.980	-0.971	-0.942	-0.974	-1.003
5	2.206	2.179	2.223	2.161	2.167	2.164	2.175	2.179
6	-1.994	-1.938	-1.956	-1.927	-1.937	-1.908	-1.938	-1.939
7	0.500	0.517	0.516	0.515	0.516	0.518	0.509	0.517
8	-2.110	-2.094	-2.065	-2.043	-2.082	-2.067	-2.064	-2.094
9	0.068	0.095	0.093	0.086	0.090	0.101	0.086	0.095
10	2.292	2.247	2.266	2.256	2.255	2.243	2.256	2.247
11	-2.053	-1.996	-2.006	-1.987	-1.991	-1.999	-1.989	-1.997
12	-0.331	-0.300	-0.326	-0.281	-0.283	-0.276	-0.281	-0.300
13	0.340	0.359	0.361	0.363	0.362	0.363	0.356	0.359
14	2.066	2.087	2.104	2.029	2.014	2.037	2.028	2.087
15	0.455	0.470	0.480	0.479	0.482	0.476	0.481	0.470

Table D.18 Comparison of estimates for the item difficulty among algorithms (3PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	2.508	2.491	2.521	2.439	2.451	2.455	2.441	2.491
17	0.114	0.134	0.137	0.144	0.140	0.143	0.140	0.134
18	-2.010	-1.994	-1.991	-1.951	-1.974	-1.978	-1.965	-1.994
19	0.534	0.567	0.559	0.551	0.562	0.553	0.554	0.567
20	1.918	1.929	1.921	1.894	1.892	1.894	1.894	1.929
21	-1.689	-1.564	-1.604	-1.556	-1.549	-1.551	-1.540	-1.564
22	-1.427	-1.394	-1.393	-1.357	-1.383	-1.381	-1.385	-1.394
23	0.110	0.177	0.181	0.154	0.173	0.158	0.153	0.177
24	1.383	1.397	1.399	1.395	1.384	1.389	1.390	1.397
25	0.668	0.677	0.697	0.702	0.700	0.699	0.684	0.677
26	1.607	1.617	1.622	1.615	1.599	1.604	1.606	1.617
27	2.436	2.447	2.419	2.404	2.377	2.400	2.354	2.447
28	0.805	0.819	0.830	0.817	0.820	0.824	0.820	0.819
29	1.007	1.020	1.025	1.031	1.012	1.021	1.016	1.020
30	1.892	1.904	1.897	1.896	1.878	1.884	1.882	1.904

Table D.19 Comparison of SE for the item difficulty among algorithms (3PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.255	0.290	0.251	0.229	0.203	0.187	0.176	0.283
2	0.141	0.147	0.173	0.156	0.137	0.138	0.117	0.145
3	0.240	0.210	0.222	0.233	0.207	0.209	0.211	0.211
4	0.184	0.197	0.196	0.190	0.164	0.175	0.171	0.190
5	0.199	0.270	0.264	0.207	0.175	0.161	0.170	0.265
6	0.205	0.211	0.223	0.231	0.166	0.155	0.175	0.207
7	0.049	0.065	0.067	0.059	0.047	0.048	0.048	0.058
8	0.147	0.165	0.181	0.169	0.131	0.135	0.130	0.152
9	0.092	0.114	0.102	0.105	0.093	0.088	0.087	0.109
10	0.224	0.224	0.276	0.240	0.200	0.192	0.182	0.215
11	0.160	0.182	0.202	0.174	0.151	0.156	0.135	0.176
12	0.132	0.156	0.155	0.144	0.123	0.123	0.127	0.149
13	0.063	0.074	0.075	0.076	0.059	0.054	0.058	0.075
14	0.242	0.274	0.279	0.242	0.170	0.184	0.179	0.268
15	0.065	0.083	0.079	0.083	0.069	0.057	0.067	0.081

Table D.20 Comparison of SE for the item difficulty among algorithms (3PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.254	0.288	0.271	0.268	0.196	0.231	0.213	0.284
17	0.080	0.093	0.098	0.092	0.074	0.082	0.077	0.089
18	0.138	0.154	0.172	0.174	0.129	0.135	0.123	0.142
19	0.073	0.086	0.091	0.081	0.062	0.083	0.066	0.081
20	0.201	0.214	0.240	0.202	0.157	0.169	0.161	0.203
21	0.214	0.270	0.240	0.248	0.208	0.214	0.221	0.268
22	0.104	0.142	0.143	0.128	0.110	0.109	0.107	0.136
23	0.167	0.167	0.178	0.184	0.147	0.130	0.157	0.165
24	0.106	0.126	0.129	0.112	0.089	0.082	0.085	0.118
25	0.106	0.122	0.117	0.116	0.094	0.088	0.103	0.118
26	0.090	0.107	0.116	0.106	0.080	0.088	0.090	0.100
27	0.271	0.342	0.300	0.282	0.214	0.226	0.202	0.334
28	0.071	0.082	0.080	0.079	0.073	0.062	0.063	0.076
29	0.046	0.060	0.070	0.057	0.041	0.039	0.039	0.051
30	0.155	0.170	0.189	0.164	0.125	0.114	0.126	0.162

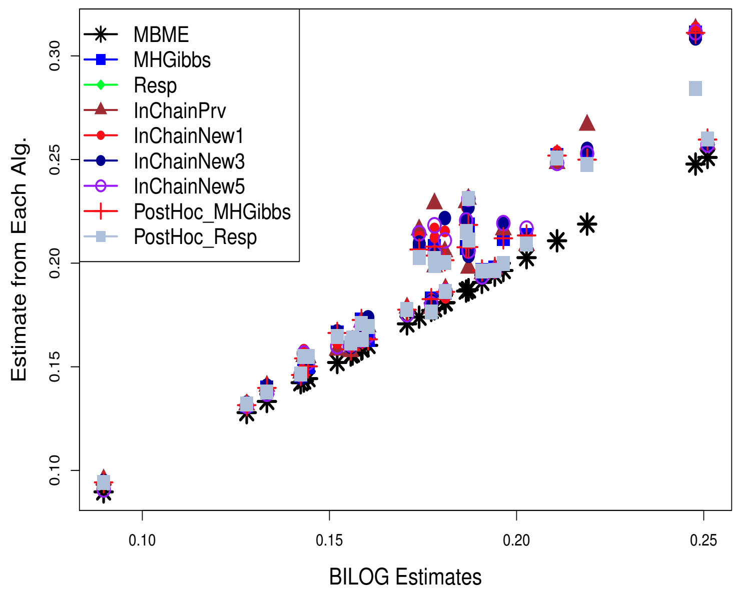

Figure D.11 Comparison of estimates for the item guessing among algorithms (3PL model with 1000 examinees and 30 items)

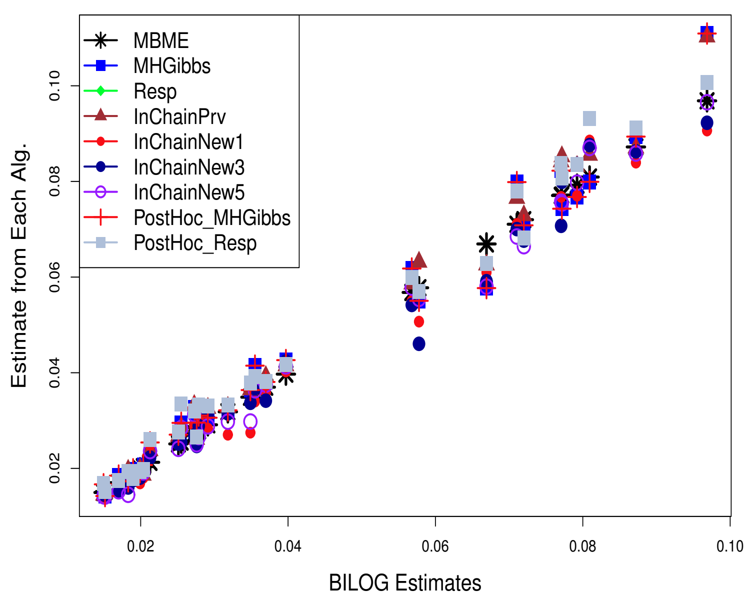

Figure D.12 Comparison of SE for the item guessing among algorithms (3PL model with 1000 examinees and 30 items)

Table D.21 Comparison of estimates for the item guessing among algorithms (3PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.159	0.173	0.171	0.172	0.169	0.166	0.170	0.173
2	0.187	0.208	0.215	0.229	0.227	0.221	0.221	0.208
3	0.178	0.204	0.199	0.198	0.212	0.200	0.205	0.204
4	0.181	0.201	0.200	0.206	0.215	0.222	0.211	0.201
5	0.191	0.196	0.196	0.195	0.193	0.193	0.194	0.196
6	0.174	0.207	0.203	0.216	0.213	0.210	0.214	0.207
7	0.133	0.140	0.138	0.141	0.138	0.141	0.137	0.140
8	0.187	0.218	0.231	0.231	0.230	0.227	0.231	0.218
9	0.152	0.166	0.165	0.158	0.161	0.167	0.160	0.166
10	0.156	0.158	0.161	0.157	0.160	0.160	0.161	0.158
11	0.211	0.252	0.251	0.248	0.254	0.251	0.248	0.252
12	0.196	0.212	0.200	0.216	0.220	0.220	0.219	0.212
13	0.171	0.178	0.178	0.178	0.179	0.178	0.175	0.178
14	0.203	0.213	0.209	0.209	0.212	0.212	0.217	0.213
15	0.144	0.150	0.155	0.155	0.154	0.153	0.152	0.150

Table D.22 Comparison of estimates for the item guessing among algorithms (3PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.142	0.146	0.146	0.147	0.146	0.145	0.146	0.146
17	0.143	0.154	0.155	0.155	0.158	0.155	0.156	0.154
18	0.178	0.208	0.202	0.229	0.217	0.205	0.218	0.208
19	0.251	0.260	0.260	0.257	0.261	0.258	0.257	0.260
20	0.177	0.183	0.176	0.178	0.183	0.182	0.180	0.183
21	0.248	0.311	0.284	0.313	0.314	0.308	0.311	0.311
22	0.219	0.250	0.248	0.267	0.255	0.255	0.253	0.250
23	0.187	0.208	0.211	0.198	0.209	0.204	0.206	0.208
24	0.181	0.186	0.186	0.188	0.183	0.186	0.186	0.186
25	0.160	0.163	0.169	0.169	0.172	0.174	0.165	0.163
26	0.194	0.198	0.197	0.197	0.195	0.196	0.197	0.198
27	0.159	0.164	0.163	0.163	0.165	0.163	0.163	0.164
28	0.156	0.162	0.164	0.160	0.162	0.164	0.162	0.162
29	0.128	0.131	0.132	0.132	0.131	0.133	0.131	0.131
30	0.090	0.094	0.094	0.096	0.095	0.095	0.091	0.094

Table D.23 Comparison of SE for the item guessing among algorithms (3PL model with 1000 examinees and 30 items, item 1-15)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
1	0.028	0.030	0.027	0.026	0.025	0.025	0.025	0.030
2	0.079	0.077	0.084	0.083	0.077	0.084	0.080	0.077
3	0.067	0.058	0.063	0.063	0.061	0.059	0.058	0.058
4	0.072	0.071	0.068	0.073	0.068	0.068	0.066	0.071
5	0.018	0.017	0.019	0.020	0.016	0.016	0.014	0.017
6	0.077	0.082	0.084	0.084	0.077	0.071	0.076	0.082
7	0.021	0.025	0.026	0.024	0.025	0.022	0.024	0.025
8	0.081	0.080	0.093	0.086	0.089	0.088	0.087	0.080
9	0.040	0.043	0.042	0.041	0.040	0.042	0.041	0.043
10	0.019	0.020	0.018	0.019	0.018	0.018	0.018	0.020
11	0.087	0.089	0.091	0.087	0.084	0.086	0.086	0.089
12	0.057	0.062	0.060	0.059	0.055	0.054	0.058	0.062
13	0.028	0.031	0.033	0.032	0.030	0.029	0.027	0.031
14	0.035	0.036	0.038	0.036	0.027	0.034	0.030	0.036
15	0.027	0.033	0.032	0.033	0.029	0.027	0.031	0.033

Table D.24 Comparison of SE for the item guessing among algorithms (3PL model with 1000 examinees and 30 items, item 16-30)

Item	Algorithm							
	MBME	MH/Gibbs	MH/Gibbs with same response treatment	In-chain centered MCMC (previous ability)	In-chain centered MCMC (one extra ability set)	In-chain centered MCMC (three extra ability sets)	In-chain centered MCMC (five extra ability sets)	Post-Hoc
16	0.015	0.014	0.015	0.015	0.014	0.014	0.014	0.014
17	0.036	0.041	0.039	0.037	0.034	0.037	0.036	0.041
18	0.077	0.074	0.081	0.085	0.084	0.080	0.075	0.074
19	0.029	0.031	0.033	0.033	0.028	0.032	0.029	0.031
20	0.032	0.032	0.033	0.033	0.027	0.031	0.030	0.032
21	0.097	0.111	0.101	0.110	0.091	0.092	0.097	0.111
22	0.071	0.080	0.078	0.077	0.071	0.070	0.068	0.080
23	0.058	0.055	0.057	0.063	0.051	0.046	0.055	0.055
24	0.026	0.030	0.033	0.029	0.027	0.025	0.028	0.030
25	0.037	0.038	0.038	0.039	0.036	0.034	0.038	0.038
26	0.017	0.018	0.017	0.016	0.017	0.015	0.015	0.018
27	0.020	0.021	0.020	0.019	0.018	0.020	0.019	0.021
28	0.025	0.027	0.028	0.027	0.026	0.025	0.024	0.027
29	0.015	0.017	0.017	0.014	0.016	0.015	0.014	0.017
30	0.020	0.019	0.019	0.020	0.017	0.018	0.019	0.019

Appendix E

Fortran Code

```
!-- Functions supporting MCMC
!***********************************************
!-- Normal Distribution
subroutine dnorm(x,mean,std,px)
implicit none
double precision, intent(in) :: x,mean,std
double precision, intent(out) :: px
double precision, parameter :: pi=3.14159265359
!-----------------------------------------------------------
px=1/(sqrt(2*pi*std**2))*exp(-(x-mean)**2/(2*std**2))
end subroutine dnorm

!-- Log Normal Distribution
subroutine dlognorm(x,mean,std,px)
implicit none
double precision, intent(in) :: x,mean,std
double precision, intent(out) :: px
double precision, parameter :: pi=3.14159265359
!-----------------------------------------------------------
if (x<0) then
        px=0
else
        px=1/(sqrt(2*pi)*x*std)*exp(-(log(x)-mean)**2/(2*std**2))
endif
end subroutine dlognorm

!-- Beta Distribution
subroutine dbeta(x,a,b,px) ! alp=5, beta=17, pxior for c
implicit none
double precision, intent(in) :: x
```

```
integer, intent(in) :: a,b
double precision, intent(out) :: px
!------------------------------------------------------------------
if(x>1) then
        px=0
else
        if(x<0) then
                px=0
        else
                px=101745*(x**(a-1))*((1-x)**(b-1))
        end if
end if
end subroutine dbeta

!-- Random Number Generator from Normal Dist.
subroutine rnorm(mean,sd,rnum) ! Use Box-Muller transformation
implicit none
double precision, intent(in) :: mean, sd
double precision, intent(out) :: rnum
double precision :: u,v,r,c
!------------------------------------------------------------------
r=2
do while (r>1.or.r==0)
        u=2*rand()-1
        v=2*rand()-1
        r=u**2+v**2
end do
c=sqrt(-2*log(r)/r)
rnum=mean+sd*(u*c)
end subroutine rnorm

!-- ICC Curve
subroutine icc(x,XIi,px)
implicit none
double precision, intent(in) :: x
double precision, dimension(3), intent(in) ::XIi
double precision, intent(out) :: px
double precision :: temp
!------------------------------------------------------------------
```

```
temp=XIi(3)+(1-XIi(3))/(1+exp(-1.7*XIi(1)*(x-XIi(2))))
px=min(max(temp,0.000001),0.999999)
end subroutine icc

!-- Likelihood Function
subroutine lik(t,XIi,Uij,pr)
implicit none
!-----------------------------------------------------------------
!Input variables
double precision, intent(in) :: t
double precision, dimension(3), intent(in) ::XIi
integer, intent(in) :: Uij
!-----------------------------------------------------------------
!Output variables
double precision, intent(out) :: pr
!-----------------------------------------------------------------
!Internal variables
double precision :: temp
!-----------------------------------------------------------------
call icc(t,XIi,temp)
pr=(temp**Uij)*((1-temp)**(1-Uij))
end subroutine lik

!-- Log Likelihood Function
subroutine loglik(t,XIi,Uij,pr)
implicit none
!-----------------------------------------------------------------
!Input variables
double precision, intent(in) :: t
double precision, dimension(3), intent(in) ::XIi
integer, intent(in) :: Uij
!-----------------------------------------------------------------
!Output variables
double precision, intent(out) :: pr
!-----------------------------------------------------------------
!Internal variables
double precision :: temp
!-----------------------------------------------------------------
call icc(t,XIi,temp)
```

```
pr=Uij*log(temp)+(1-Uij)*log(1-temp)
end subroutine loglik

!-- Sum of Log Likelihood for All Item
subroutine probLogSumItem(I,t,XI,Uj,pr)
implicit none
!----------------------------------------------------------------
!Input variables
integer, intent(in) :: I
double precision, intent(in) :: t
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(I), intent(in) :: Uj
!----------------------------------------------------------------
!Output variables
double precision, intent(out) :: pr
!----------------------------------------------------------------
!Internal variables
integer :: ii
double precision, dimension(I) :: temp
!----------------------------------------------------------------
do ii=1,I
        call loglik(t,XI(ii,:),Uj(ii),temp(ii))
end do
pr=sum(temp)
end subroutine probLogSumItem

!-- Log Likelihood Function for All Examinees
subroutine probLogSumExam(N,t,XIi,Ui,pr)
implicit none
!----------------------------------------------------------------
!Input variables
integer, intent(in) :: N
double precision, dimension(N), intent(in) :: t
double precision, dimension(3), intent(in) :: XIi
integer, dimension(N), intent(in) :: Ui
!----------------------------------------------------------------
!Output variables
double precision, intent(out) :: pr
!----------------------------------------------------------------
```

```
!Internal variables
integer :: jj
double precision, dimension(N) :: temp
!-----------------------------------------------------------------
do jj=1,N
        call loglik(t(jj),XIi,Ui(jj),temp(jj))
end do
pr=sum(temp)
end subroutine probLogSumExam

!-- Acceptance Function for Ability
subroutine acpt_t_pr(I,th_s,th_p,XI,Uj,sd0,alp_t)
implicit none
!-----------------------------------------------------------------
!Input variables
integer, intent(in) :: I
double precision, intent(in) :: th_s,th_p
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(I), intent(in) :: Uj
double precision, intent(in) :: sd0
!-----------------------------------------------------------------
!Output variables
double precision, intent(out) :: alp_t
!-----------------------------------------------------------------
!Internal variables
double precision, parameter :: aimmean=0, maxAlpha=1
double precision :: pjsp,pjpp,pr1,pr2,tempAlpha
!-----------------------------------------------------------------
call probLogSumItem(I,th_s,XI,Uj,pjsp)
call probLogSumItem(I,th_p,XI,Uj,pjpp)
call dnorm(th_s,aimmean,sd0,pr1)
call dnorm(th_p,aimmean,sd0,pr2)
tempAlpha=pjsp+log(pr1)-pjpp-log(pr2)
alp_t=min(exp(tempAlpha),maxAlpha)
end subroutine acpt_t_pr

!-- Update Candidate Function for Ability
subroutine up_t(I,th_s,th_p,XI,Uj,sd0,new_t,ind)
implicit none
```

```
!-----------------------------------------------------------------------
!Input variables
integer, intent(in) :: I
double precision, intent(in) :: th_s,th_p
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(I), intent(in) :: Uj
double precision, intent(in) :: sd0
!-----------------------------------------------------------------------
!Output variables
double precision, intent(out) :: new_t
integer, intent(out) :: ind
!-----------------------------------------------------------------------
!Internal variables
double precision :: unif,alp_t
!-----------------------------------------------------------------------
call acpt_t_pr(I,th_s,th_p,XI,Uj,sd0,alp_t)
call random_number(unif)
if (unif<alp_t) then
        new_t=th_s
        ind=1
else
        new_t=th_p
        ind=0
end if
end subroutine up_t

!-- Acceptance Fucntion Function
!--  for Discrimination Parameter for Rasch
subroutine acpt_aRCH_pr(N,I,th_s,XI_str,XI_prv,U,sd0,alp_a)
implicit none
!-----------------------------------------------------------------------
!Input variables
integer, intent(in) :: N,I
double precision, dimension(N), intent(in) :: th_s
double precision, dimension(I,3), intent(in) :: XI_str,XI_prv
integer, dimension(N,I), intent(in) :: U
double precision, intent(in) :: sd0
!-----------------------------------------------------------------------
!Output variables
```

```
double precision, intent(out) :: alp_a
!--------------------------------------------------------------------------
!Internal variables
integer :: ii
integer, dimension(N) :: Ui
double precision, dimension(3) :: XIi_str,XIi_prv
double precision, parameter :: aimmean=0, maxAlpha=1
double precision, dimension(I) :: temppjsp, temppjpp
double precision :: pjsp,pjpp,pr1,pr2,tempAlpha
!--------------------------------------------------------------------------
do ii=1,I
        Ui=U(:,ii)
        XIi_str=XI_str(ii,:)
        XIi_prv=XI_prv(ii,:)
        call probLogSumExam(N,th_s,XIi_str,Ui,temppjsp(ii))
        call probLogSumExam(N,th_s,XIi_prv,Ui,temppjpp(ii))
end do

pjsp=sum(temppjsp)
pjpp=sum(temppjpp)

call dlognorm(XIi_str(1),aimmean,sd0,pr1) !lognormal for a
call dlognorm(XIi_prv(1),aimmean,sd0,pr2) !lognormal for a
tempAlpha=pjsp+log(pr1)-pjpp-log(pr2)
alp_a=min(exp(tempAlpha),maxAlpha)
end subroutine acpt_aRCH_pr

!-- Update Fucntion Function
!--  for Discrimination Parameter for Rasch
subroutine up_aRCH(N,I,th_s,XI_str,XI_prv,U,sd0,XI_new,ac)
implicit none
!--------------------------------------------------------------------------
!Input variables
integer, intent(in) :: N,I
double precision, dimension(N), intent(in) :: th_s
double precision, dimension(I,3), intent(in) :: XI_str,XI_prv
integer, dimension(N,I), intent(in) :: U
double precision, intent(in) :: sd0
!--------------------------------------------------------------------------
```

```
!Output variables
double precision, dimension(I,3), intent(out) :: XI_new
integer :: ac
!-----------------------------------------------------------------
!Internal variables
double precision :: unif,alp_a
!-----------------------------------------------------------------
call acpt_aRCH_pr(N,I,th_s,XI_str,XI_prv,U,sd0,alp_a)
call random_number(unif)
if (unif<alp_a) then
        XI_new=XI_str
        ac=1
else
        XI_new=XI_prv
        ac=0
end if
end subroutine up_aRCH

!-- Acceptance Fucntion Function for Guessing Parameter
subroutine acpt_c_pr(N,th_s,XIi_str,XIi_prv,Ui,alp_c)
implicit none
!-----------------------------------------------------------------
!Input variables
integer, intent(in) :: N
double precision, dimension(N), intent(in) :: th_s
double precision, dimension(3), intent(in) :: XIi_str,XIi_prv
integer, dimension(N), intent(in) :: Ui
!-----------------------------------------------------------------
!Output variables
double precision, intent(out) :: alp_c
!-----------------------------------------------------------------
!Internal variables
integer, parameter :: aima=5, aimb=17
double precision, parameter :: maxAlpha=1
double precision :: pjsp,pjpp,pr1,pr2,tempAlpha
!-----------------------------------------------------------------
call probLogSumExam(N,th_s,XIi_str,Ui,pjsp)
call probLogSumExam(N,th_s,XIi_prv,Ui,pjpp)
call dbeta(XIi_str(3),aima,aimb,pr1) ! beta for c
```

```
call dbeta(XIi_prv(3),aima,aimb,pr2) ! beta for c
tempAlpha=pjsp+log(pr1)-pjpp-log(pr2)
alp_c=min(exp(tempAlpha),maxAlpha)
end subroutine acpt_c_pr

!-- Update Fucntion Function for Guessing Parameter
subroutine up_c(N,th_s,XIi_str,XIi_prv,Ui,XIi_new,Ind)
implicit none
!------------------------------------------------------------
!Input variables
integer, intent(in) :: N
double precision, dimension(N), intent(in) :: th_s
double precision, dimension(3), intent(in) :: XIi_str,XIi_prv
integer, dimension(N), intent(in) :: Ui
!------------------------------------------------------------
!Output variables
double precision, dimension(3), intent(out) :: XIi_new
integer :: Ind
!------------------------------------------------------------
!Internal variables
double precision :: unif,alp_c
!------------------------------------------------------------
call acpt_c_pr(N,th_s,XIi_str,XIi_prv,Ui,alp_c)
call random_number(unif)
if (unif<alp_c) then
        XIi_new=XIi_str
        Ind=1
else
        XIi_new=XIi_prv
        Ind=0
end if
end subroutine up_c

!-- Acceptance Fucntion Function for a and b
subroutine acpt_ab_pr(N,th_s,XIi_str,XIi_prv,Ui,sdA,sdB,alp_ab)
implicit none
!------------------------------------------------------------
!Input variables
integer, intent(in) :: N
```

```
double precision, dimension(N), intent(in) :: th_s
double precision, dimension(3), intent(in) :: XIi_str,XIi_prv
integer, dimension(N), intent(in) :: Ui
double precision, intent(in) :: sdA,sdB
!-----------------------------------------------------------------
!Output variables
double precision, intent(out) :: alp_ab
!-----------------------------------------------------------------
!Internal variables
double precision, parameter :: aimmean=0, maxAlpha=1
double precision :: pjsp,pjpp,prA1,prA2,prB1,prB2,tempAlpha
!-----------------------------------------------------------------
call probLogSumExam(N,th_s,XIi_str,Ui,pjsp)
call probLogSumExam(N,th_s,XIi_prv,Ui,pjpp)
call dlognorm(XIi_str(1),aimmean,sdA,prA1) !lognormal for a
call dlognorm(XIi_prv(1),aimmean,sdA,prA2) !lognormal for a
call dnorm(XIi_str(2),aimmean,sdB,prB1) ! normal for b
call dnorm(XIi_prv(2),aimmean,sdB,prB2) ! normal for b
tempAlpha=pjsp-pjpp+log(prA1)+log(prB1)-log(prA2)-log(prB2)
alp_ab=min(exp(tempAlpha),maxAlpha)
end subroutine acpt_ab_pr

!-- Update Fucntion Function for a and b
subroutine up_ab(N,th_s,XIi_str,XIi_prv,Ui,sdA,sdB,XIi_new,Ind)
implicit none
!-----------------------------------------------------------------
!Input variables
integer, intent(in) :: N
double precision, dimension(N), intent(in) :: th_s
double precision, dimension(3), intent(in) :: XIi_str,XIi_prv
integer, dimension(N), intent(in) :: Ui
double precision, intent(in) :: sdA,sdB
!-----------------------------------------------------------------
!Output variables
double precision, dimension(3), intent(out) :: XIi_new
integer, intent(out) :: Ind
!-----------------------------------------------------------------
!Internal variables
double precision :: unif,alp_ab
```

```
!------------------------------------------------------------------------
call acpt_ab_pr(N,th_s,XIi_str,XIi_prv,Ui,sdA,sdB,alp_ab)
call random_number(unif)
if (unif<alp_ab) then
        XIi_new=XIi_str
        Ind=1
else
        XIi_new=XIi_prv
        Ind=0
end if
end subroutine up_ab
```

```
!-- Core Code for MCMC (2PL)
!-- 1PL uses "up_aRCH" instead of "up_a"
!-- 3PL adds "up_c"
!**************************************************
!Classic MH within Gibbs
subroutine mc2_1(seed,K,N,I,R,th,XI,U,IU,qsd,tOut,a,b,c,tAc,iAc)
!N=nexam, I=nitem, K=niter
!R=number of same responses
implicit none
!-----------------------------------------------------------------
!Input variables
integer, intent(in) :: seed,K,N,I,R
double precision, dimension(N), intent(in) :: th
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(N,I), intent(in) :: U
integer, dimension(R,2), intent(in) :: IU
double precision,dimension(4),intent(in) :: qsd
!-----------------------------------------------------------------
!Output variables
double precision, dimension(K,N), intent(out) :: tOut
double precision, dimension(K,I), intent(out) :: a,b,c
integer, dimension(K-1,N), intent(out) :: tAc
integer, dimension(K-1,I,3), intent(out) :: iAc
!-----------------------------------------------------------------
!Internal variables for overall
integer :: iter,jj,ii,usi,ind !iteration iter
double precision, dimension(K,N) :: Thiter
double precision, dimension(K,I,3) :: XIiter
!Internal variables for updata theta
double precision :: th_p, th_s, th_n ! t candidate
double precision, dimension(I,3) :: XIth
integer, dimension(I) :: Uj
!Internal variables for updata updata a,b
double precision, dimension(N) :: thXI
integer, dimension(N) :: Ui
double precision :: a_str,b_str,c_str ! a,b,c candidate
double precision, dimension(3) :: XI_p, XI_s, XI_n ! XI candidate
double precision, dimension(I,3) :: XI_pRC, XI_sRC, XI_nRC
!Internal variables for purposed standard deviations
```

```
double precision, parameter :: pTsd=1.0,pAsd=0.5,pBsd=2.0,pCsd=0.2
!Internal variables for target standard deviations
double precision :: qTsd,qAsd, qBsd,qCsd !q sd for a,b
!------------------------------------------------------------------
!Internal variables for centered MCMC and extra sets of ability
integer :: nb,dino
integer, parameter :: Nboots=1
double precision, dimension(N,Nboots) :: thTemp
double precision, dimension(I,3) :: xiTemp
double precision :: muBT,sdBT
!------------------------------------------------------------------
call srand(seed)
XIiter(1,:,:)=XI(:,:)
Thiter(1,:)=th
qAsd=qsd(1);qBsd=qsd(2);qCsd=qsd(3);qTsd=qsd(4)
dino=N*Nboots
!------------------------------------------------------------------
do iter=1,(K-1)
        !update theta
        do jj=1,N
                th_p=Thiter(iter,jj)
                XIth=XIiter(iter,:,:)
                Uj=U(jj,:)
                call rnorm(th_p,qTsd,th_s)
                call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                tAc(iter,jj)=ind
                Thiter(iter+1,jj)=th_n
        end do
        !update chain
        thXI=Thiter(iter+1,:)
        !update chain a,b
        do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                call rnorm(XI_p(1),qAsd,a_str)
                call rnorm(XI_p(2),qBsd,b_str)
                XI_s=XI_p
                XI_s(1)=a_str
                XI_s(2)=b_str
```

```
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                iAc(iter,ii,1)=ind
                iAc(iter,ii,2)=ind
                XIiter(iter+1,ii,:)=XI_n
        end do
end do
tOut=Thiter
a=XIiter(:,:,1)
b=XIiter(:,:,2)
c=XIiter(:,:,3)
end subroutine mc2_1

!-------------------------------------------------
!Same response treat
subroutine mc2_2(seed,K,N,I,R,th,XI,U,IU,qsd,tOut,a,b,c,tAc,iAc)
!N=nexam, I=nitem, K=niter
!R=number of same responses
implicit none
!-----------------------------------------------------------
!Input variables
integer, intent(in) :: seed,K,N,I,R
double precision, dimension(N), intent(in) :: th
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(N,I), intent(in) :: U
integer, dimension(R,2), intent(in) :: IU
double precision,dimension(4),intent(in) :: qsd
!-----------------------------------------------------------
!Output variables
double precision, dimension(K,N), intent(out) :: tOut
double precision, dimension(K,I), intent(out) :: a,b,c
integer, dimension(K-1,N), intent(out) :: tAc
integer, dimension(K-1,I,3), intent(out) :: iAc
!-----------------------------------------------------------
!Internal variables for overall
integer :: iter,jj,ii,usi,ind !iteration iter
double precision, dimension(K,N) :: Thiter
double precision, dimension(K,I,3) :: XIiter
!Internal variables for updata theta
double precision :: th_p, th_s, th_n ! t candidate
```

```
double precision, dimension(I,3) :: XIth
integer, dimension(I) :: Uj
!Internal variables for updata updata a,b
double precision, dimension(N) :: thXI
integer, dimension(N) :: Ui
double precision :: a_str,b_str,c_str ! a,b,c candidate
double precision, dimension(3) :: XI_p, XI_s, XI_n ! XI candidate
double precision, dimension(I,3) :: XI_pRC, XI_sRC, XI_nRC
!Internal variables for purposed standard deviations
double precision, parameter :: pTsd=1.0,pAsd=0.5,pBsd=2.0,pCsd=0.2
!Internal variables for target standard deviations
double precision :: qTsd,qAsd, qBsd,qCsd !q sd for a,b
!-----------------------------------------------------------------
!Internal variables for centered MCMC and extra sets of ability
integer :: nb,dino
integer, parameter :: Nboots=1
double precision, dimension(N,Nboots) :: thTemp
double precision, dimension(I,3) :: xiTemp
double precision :: muBT,sdBT
!-----------------------------------------------------------------
call srand(seed)
XIiter(1,:,:)=XI(:,:)
Thiter(1,:)=th
qAsd=qsd(1);qBsd=qsd(2);qCsd=qsd(3);qTsd=qsd(4)
dino=N*Nboots
!-----------------------------------------------------------------
do iter=1,(K-1)
        !update theta
        usi=1
        if (R==0) then
                do jj=1,N
                        th_p=Thiter(iter,jj)
                        XIth=XIiter(iter,:,:)
                        Uj=U(jj,:)
                        call rnorm(th_p,qTsd,th_s)
                        call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                        tAc(iter,jj)=ind
                        Thiter(iter+1,jj)=th_n
                end do
```

```
        else
                do jj=1,N
                        if (jj==IU(usi,2)) then
                                Thiter(iter+1,jj)=Thiter(iter+1,IU(usi,1))
                                usi=usi+1
                        else
                                th_p=Thiter(iter,jj)
                                XIth=XIiter(iter,:,:)
                                Uj=U(jj,:)
                                call rnorm(th_p,qTsd,th_s)
                                call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                                tAc(iter,jj)=ind
                                Thiter(iter+1,jj)=th_n
                        endif
                end do
        endif
        !update chain
        thXI=Thiter(iter+1,:)

        !update chain a,b
        do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                call rnorm(XI_p(1),qAsd,a_str)
                call rnorm(XI_p(2),qBsd,b_str)
                XI_s=XI_p
                XI_s(1)=a_str
                XI_s(2)=b_str
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                iAc(iter,ii,1)=ind
                iAc(iter,ii,2)=ind
                XIiter(iter+1,ii,:)=XI_n
        end do
end do
tOut=Thiter
a=XIiter(:,:,1)
b=XIiter(:,:,2)
c=XIiter(:,:,3)
end subroutine mc2_2
```

```
!-----------------------------------------------
!Same response treat & in-Chain using previous
subroutine mc2_3(seed,K,N,I,R,th,XI,U,IU,qsd,tOut,a,b,c,tAc,iAc)
!N=nexam, I=nitem, K=niter
!R=number of same responses
implicit none
!------------------------------------------------------------
!Input variables
integer, intent(in) :: seed,K,N,I,R
double precision, dimension(N), intent(in) :: th
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(N,I), intent(in) :: U
integer, dimension(R,2), intent(in) :: IU
double precision,dimension(4),intent(in) :: qsd
!------------------------------------------------------------
!Output variables
double precision, dimension(K,N), intent(out) :: tOut
double precision, dimension(K,I), intent(out) :: a,b,c
integer, dimension(K-1,N), intent(out) :: tAc
integer, dimension(K-1,I,3), intent(out) :: iAc
!------------------------------------------------------------
!Internal variables for overall
integer :: iter,jj,ii,usi,ind !iteration iter
double precision, dimension(K,N) :: Thiter
double precision, dimension(K,I,3) :: XIiter
!Internal variables for updata theta
double precision :: th_p, th_s, th_n ! t candidate
double precision, dimension(I,3) :: XIth
integer, dimension(I) :: Uj
!Internal variables for updata updata a,b
double precision, dimension(N) :: thXI
integer, dimension(N) :: Ui
double precision :: a_str,b_str,c_str ! a,b,c candidate
double precision, dimension(3) :: XI_p, XI_s, XI_n ! XI candidate
double precision, dimension(I,3) :: XI_pRC, XI_sRC, XI_nRC
!Internal variables for purposed standard deviations
double precision, parameter :: pTsd=1.0,pAsd=0.5,pBsd=2.0,pCsd=0.2
!Internal variables for target standard deviations
```

```
double precision :: qTsd,qAsd, qBsd,qCsd !q sd for a,b
!------------------------------------------------------------
!Internal variables for centered MCMC and extra sets of ability
integer :: nb,dino
integer, parameter :: Nboots=1
double precision, dimension(N,Nboots) :: thTemp
double precision, dimension(I,3) :: xiTemp
double precision :: muBT,sdBT
!------------------------------------------------------------
call srand(seed)
XIiter(1,:,:)=XI(:,:)
Thiter(1,:)=th
qAsd=qsd(1);qBsd=qsd(2);qCsd=qsd(3);qTsd=qsd(4)
dino=N*Nboots
!------------------------------------------------------------
do iter=1,(K-1)
        !update theta
        usi=1
        if (R==0) then
                do jj=1,N
                        th_p=Thiter(iter,jj)
                        XIth=XIiter(iter,:,:)
                        Uj=U(jj,:)
                        call rnorm(th_p,qTsd,th_s)
                        call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                        tAc(iter,jj)=ind
                        Thiter(iter+1,jj)=th_n
                end do
        else
                do jj=1,N
                        if (jj==IU(usi,2)) then
                                Thiter(iter+1,jj)=Thiter(iter+1,IU(usi,1))
                                usi=usi+1
                        else
                                th_p=Thiter(iter,jj)
                                XIth=XIiter(iter,:,:)
                                Uj=U(jj,:)
                                call rnorm(th_p,qTsd,th_s)
                                call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
```

```
                        tAc(iter,jj)=ind
                        Thiter(iter+1,jj)=th_n
                    endif
                end do
            endif
            !update chain
            thXI=Thiter(iter+1,:)
            muBT=sum(thXI)/N
            sdBT=sqrt((sum(thXI**2)-sum(thXI)**2/N)/(N-1))

            !update chain a,b
            do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                call rnorm(XI_p(1),qAsd,a_str)
                call rnorm(XI_p(2),qBsd,b_str)
                XI_s=XI_p
                XI_s(1)=a_str*sdBT
                XI_s(2)=(b_str-muBT)/sdBT
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                iAc(iter,ii,1)=ind
                iAc(iter,ii,2)=ind
                XIiter(iter+1,ii,:)=XI_n
            end do
end do
tOut=Thiter
a=XIiter(:,:,1)
b=XIiter(:,:,2)
c=XIiter(:,:,3)
end subroutine mc2_3

!------------------------------------------------------------
!Same response treat & in-Chain link, NB=1
subroutine mc2_4(seed,K,N,I,R,th,XI,U,IU,qsd,tOut,a,b,c,tAc,iAc)
!N=nexam, I=nitem, K=niter
!R=number of same responses
implicit none
!-----------------------------------------------------------------------
!Input variables
```

```
integer, intent(in) :: seed,K,N,I,R
double precision, dimension(N), intent(in) :: th
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(N,I), intent(in) :: U
integer, dimension(R,2), intent(in) :: IU
double precision,dimension(4),intent(in) :: qsd
!-----------------------------------------------------------------
!Output variables
double precision, dimension(K,N), intent(out) :: tOut
double precision, dimension(K,I), intent(out) :: a,b,c
integer, dimension(K-1,N), intent(out) :: tAc
integer, dimension(K-1,I,3), intent(out) :: iAc
!-----------------------------------------------------------------
!Internal variables for overall
integer :: iter,jj,ii,usi,ind !iteration iter
double precision, dimension(K,N) :: Thiter
double precision, dimension(K,I,3) :: XIiter
!Internal variables for updata theta
double precision :: th_p, th_s, th_n ! t candidate
double precision, dimension(I,3) :: XIth
integer, dimension(I) :: Uj
!Internal variables for updata updata a,b
double precision, dimension(N) :: thXI
integer, dimension(N) :: Ui
double precision :: a_str,b_str,c_str ! a,b,c candidate
double precision, dimension(3) :: XI_p, XI_s, XI_n ! XI candidate
double precision, dimension(I,3) :: XI_pRC, XI_sRC, XI_nRC
!Internal variables for purposed standard deviations
double precision, parameter :: pTsd=1.0,pAsd=0.5,pBsd=2.0,pCsd=0.2
!Internal variables for target standard deviations
double precision :: qTsd,qAsd, qBsd,qCsd !q sd for a,b
!-----------------------------------------------------------------
!Internal variables for centered MCMC and extra sets of ability
integer :: nb,dino
integer, parameter :: Nboots=1
double precision, dimension(N,Nboots) :: thTemp
double precision, dimension(I,3) :: xiTemp
double precision :: muBT,sdBT
!-----------------------------------------------------------------
```

```
call srand(seed)
XIiter(1,:,:)=XI(:,:)
Thiter(1,:)=th
qAsd=qsd(1);qBsd=qsd(2);qCsd=qsd(3);qTsd=qsd(4)
dino=N*Nboots
!-------------------------------------------------------------
do iter=1,(K-1)
        !update theta
        usi=1
        if (R==0) then
                do jj=1,N
                        th_p=Thiter(iter,jj)
                        XIth=XIiter(iter,:,:)
                        Uj=U(jj,:)
                        call rnorm(th_p,qTsd,th_s)
                        call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                        tAc(iter,jj)=ind
                        Thiter(iter+1,jj)=th_n
                end do
        else
                do jj=1,N
                        if (jj==IU(usi,2)) then
                                Thiter(iter+1,jj)=Thiter(iter+1,IU(usi,1))
                                tAc(iter,jj)=tAc(iter,IU(usi,1))
                                usi=usi+1
                        else
                                th_p=Thiter(iter,jj)
                                XIth=XIiter(iter,:,:)
                                Uj=U(jj,:)
                                call rnorm(th_p,qTsd,th_s)
                                call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                                tAc(iter,jj)=ind
                                Thiter(iter+1,jj)=th_n
                        endif
                end do
        endif
        !update chain
        thXI=Thiter(iter+1,:)
!--------------------------------------------------
```

```
        !Initial chain a,b
        do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                call rnorm(XI_p(1),qAsd,a_str)
                call rnorm(XI_p(2),qBsd,b_str)
                XI_s=XI_p
                XI_s(1)=a_str
                XI_s(2)=b_str
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                xiTemp(ii,:)=XI_n
        end do
!--------------------------------------------------------------------
        !extra theta sample to adjust item parameters
        do jj=1,N
                th_p=Thiter(iter+1,jj)
                Uj=U(jj,:)
                do nb=1,Nboots
                        call rnorm(th_p,qTsd,th_s)
                        call up_t(I,th_s,th_p,xiTemp,Uj,pTsd,thTemp(jj,nb),ind)
                end do
        end do
        muBT=sum(thTemp)/(N*Nboots)
        sdBT=sqrt((sum(thTemp**2)-sum(thTemp)**2/(dino))/(dino-1))
!--------------------------------------------------------------------
        !update chain a,b
        do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                XI_s=XI_p
                XI_s(1)=xiTemp(ii,1)*sdBT
                XI_s(2)=(xiTemp(ii,2)-muBT)/sdBT
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                iAc(iter,ii,1)=ind
                iAc(iter,ii,2)=ind
                XIiter(iter+1,ii,:)=XI_n
        end do
!--------------------------------------------------------------------
end do
```

```
tOut=Thiter
a=XIiter(:,:,1)
b=XIiter(:,:,2)
c=XIiter(:,:,3)
end subroutine mc2_4

!-------------------------------------------------------
!Same response treat & in-Chain link, NB=3
subroutine mc2_5(seed,K,N,I,R,th,XI,U,IU,qsd,tOut,a,b,c,tAc,iAc)
!N=nexam, I=nitem, K=niter
!R=number of same responses
implicit none
!-------------------------------------------------------------------
!Input variables
integer, intent(in) :: seed,K,N,I,R
double precision, dimension(N), intent(in) :: th
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(N,I), intent(in) :: U
integer, dimension(R,2), intent(in) :: IU
double precision,dimension(4),intent(in) :: qsd
!-------------------------------------------------------------------
!Output variables
double precision, dimension(K,N), intent(out) :: tOut
double precision, dimension(K,I), intent(out) :: a,b,c
integer, dimension(K-1,N), intent(out) :: tAc
integer, dimension(K-1,I,3), intent(out) :: iAc
!-------------------------------------------------------------------
!Internal variables for overall
integer :: iter,jj,ii,usi,ind !iteration iter
double precision, dimension(K,N) :: Thiter
double precision, dimension(K,I,3) :: XIiter
!Internal variables for updata theta
double precision :: th_p, th_s, th_n ! t candidate
double precision, dimension(I,3) :: XIth
integer, dimension(I) :: Uj
!Internal variables for updata updata a,b
double precision, dimension(N) :: thXI
integer, dimension(N) :: Ui
double precision :: a_str,b_str,c_str ! a,b,c candidate
```

```
double precision, dimension(3) :: XI_p, XI_s, XI_n ! XI candidate
double precision, dimension(I,3) :: XI_pRC, XI_sRC, XI_nRC
!Internal variables for purposed standard deviations
double precision, parameter :: pTsd=1.0,pAsd=0.5,pBsd=2.0,pCsd=0.2
!Internal variables for target standard deviations
double precision :: qTsd,qAsd, qBsd,qCsd !q sd for a,b
!------------------------------------------------------------------
!Internal variables for centered MCMC and extra sets of ability
integer :: nb,dino
integer, parameter :: Nboots=3
double precision, dimension(N,Nboots) :: thTemp
double precision, dimension(I,3) :: xiTemp
double precision :: muBT,sdBT
!------------------------------------------------------------------
call srand(seed)
XIiter(1,:,:)=XI(:,:)
Thiter(1,:)=th
qAsd=qsd(1);qBsd=qsd(2);qCsd=qsd(3);qTsd=qsd(4)
dino=N*Nboots
!------------------------------------------------------------------
do iter=1,(K-1)
        !update theta
        usi=1
        if (R==0) then
                do jj=1,N
                        th_p=Thiter(iter,jj)
                        XIth=XIiter(iter,:,:)
                        Uj=U(jj,:)
                        call rnorm(th_p,qTsd,th_s)
                        call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                        tAc(iter,jj)=ind
                        Thiter(iter+1,jj)=th_n
                end do
        else
                do jj=1,N
                        if (jj==IU(usi,2)) then
                                Thiter(iter+1,jj)=Thiter(iter+1,IU(usi,1))
                                tAc(iter,jj)=tAc(iter,IU(usi,1))
                                usi=usi+1
```

```
            else
                th_p=Thiter(iter,jj)
                XIth=XIiter(iter,:,:)
                Uj=U(jj,:)
                call rnorm(th_p,qTsd,th_s)
                call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                tAc(iter,jj)=ind
                Thiter(iter+1,jj)=th_n
            endif
        end do
    endif
    !update chain
    thXI=Thiter(iter+1,:)
!------------------------------------------------------------
    !Initial chain a,b
    do ii=1,I
        XI_p=XIiter(iter,ii,:)
        Ui=U(:,ii)
        call rnorm(XI_p(1),qAsd,a_str)
        call rnorm(XI_p(2),qBsd,b_str)
        XI_s=XI_p
        XI_s(1)=a_str
        XI_s(2)=b_str
        call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
        xiTemp(ii,:)=XI_n
    end do
!------------------------------------------------------------
    !extra theta sample to adjust item parameters
    do jj=1,N
        th_p=Thiter(iter+1,jj)
        Uj=U(jj,:)
        do nb=1,Nboots
            call rnorm(th_p,qTsd,th_s)
            call up_t(I,th_s,th_p,xiTemp,Uj,pTsd,thTemp(jj,nb),ind)
        end do
    end do
    muBT=sum(thTemp)/(N*Nboots)
    sdBT=sqrt((sum(thTemp**2)-sum(thTemp)**2/(dino))/(dino-1))
!------------------------------------------------------------
```

```
        !update chain a,b
        do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                XI_s=XI_p
                XI_s(1)=xiTemp(ii,1)*sdBT
                XI_s(2)=(xiTemp(ii,2)-muBT)/sdBT
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                iAc(iter,ii,1)=ind
                iAc(iter,ii,2)=ind
                XIiter(iter+1,ii,:)=XI_n
        end do
!---------------------------------------------------
end do
tOut=Thiter
a=XIiter(:,:,1)
b=XIiter(:,:,2)
c=XIiter(:,:,3)
end subroutine mc2_5
!---------------------------------------------------
!Same response treat & in-Chain link, NB=5
subroutine mc2_6(seed,K,N,I,R,th,XI,U,IU,qsd,tOut,a,b,c,tAc,iAc)
!N=nexam, I=nitem, K=niter
!R=number of same responses
implicit none
!------------------------------------------------------------
!Input variables
integer, intent(in) :: seed,K,N,I,R
double precision, dimension(N), intent(in) :: th
double precision, dimension(I,3), intent(in) :: XI
integer, dimension(N,I), intent(in) :: U
integer, dimension(R,2), intent(in) :: IU
double precision,dimension(4),intent(in) :: qsd
!------------------------------------------------------------
!Output variables
double precision, dimension(K,N), intent(out) :: tOut
double precision, dimension(K,I), intent(out) :: a,b,c
integer, dimension(K-1,N), intent(out) :: tAc
integer, dimension(K-1,I,3), intent(out) :: iAc
```

```
!——————————————————————————————————————————————————————————————
!Internal variables for overall
integer :: iter,jj,ii,usi,ind !iteration iter
double precision, dimension(K,N) :: Thiter
double precision, dimension(K,I,3) :: XIiter
!Internal variables for updata theta
double precision :: th_p, th_s, th_n ! t candidate
double precision, dimension(I,3) :: XIth
integer, dimension(I) :: Uj
!Internal variables for updata updata a,b
double precision, dimension(N) :: thXI
integer, dimension(N) :: Ui
double precision :: a_str,b_str,c_str ! a,b,c candidate
double precision, dimension(3) :: XI_p, XI_s, XI_n ! XI candidate
double precision, dimension(I,3) :: XI_pRC, XI_sRC, XI_nRC
!Internal variables for purposed standard deviations
double precision, parameter :: pTsd=1.0,pAsd=0.5,pBsd=2.0,pCsd=0.2
!Internal variables for target standard deviations
double precision :: qTsd,qAsd, qBsd,qCsd !q sd for a,b
!——————————————————————————————————————————————————————————————
!Internal variables for centered MCMC and extra sets of ability
integer :: nb,dino
integer, parameter :: Nboots=5
double precision, dimension(N,Nboots) :: thTemp
double precision, dimension(I,3) :: xiTemp
double precision :: muBT,sdBT
!——————————————————————————————————————————————————————————————
call srand(seed)
XIiter(1,:,:)=XI(:,:)
Thiter(1,:)=th
qAsd=qsd(1);qBsd=qsd(2);qCsd=qsd(3);qTsd=qsd(4)
dino=N*Nboots
!——————————————————————————————————————————————————————————————
do iter=1,(K-1)
        !update theta
        usi=1
        if (R==0) then
                do jj=1,N
                        th_p=Thiter(iter,jj)
```

```
                XIth=XIiter(iter,:,:)
                Uj=U(jj,:)
                call rnorm(th_p,qTsd,th_s)
                call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                tAc(iter,jj)=ind
                Thiter(iter+1,jj)=th_n
            end do
        else
            do jj=1,N
                if (jj==IU(usi,2)) then
                    Thiter(iter+1,jj)=Thiter(iter+1,IU(usi,1))
                    tAc(iter,jj)=tAc(iter,IU(usi,1))
                    usi=usi+1
                else
                    th_p=Thiter(iter,jj)
                    XIth=XIiter(iter,:,:)
                    Uj=U(jj,:)
                    call rnorm(th_p,qTsd,th_s)
                    call up_t(I,th_s,th_p,XIth,Uj,pTsd,th_n,ind)
                    tAc(iter,jj)=ind
                    Thiter(iter+1,jj)=th_n
                endif
            end do
        endif
        !update chain
        thXI=Thiter(iter+1,:)
!-----------------------------------------------------------
        !Initial chain a,b
        do ii=1,I
            XI_p=XIiter(iter,ii,:)
            Ui=U(:,ii)
            call rnorm(XI_p(1),qAsd,a_str)
            call rnorm(XI_p(2),qBsd,b_str)
            XI_s=XI_p
            XI_s(1)=a_str
            XI_s(2)=b_str
            call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
            xiTemp(ii,:)=XI_n
        end do
```

```
!------------------------------------------------------------
        !extra theta sample to adjust item parameters
        do jj=1,N
                th_p=Thiter(iter+1,jj)
                Uj=U(jj,:)
                do nb=1,Nboots
                        call rnorm(th_p,qTsd,th_s)
                        call up_t(I,th_s,th_p,xiTemp,Uj,pTsd,thTemp(jj,nb),ind)
                end do
        end do
        muBT=sum(thTemp)/(N*Nboots)
        sdBT=sqrt((sum(thTemp**2)-sum(thTemp)**2/(dino))/(dino-1))
!------------------------------------------------------------
        !update chain a,b
        do ii=1,I
                XI_p=XIiter(iter,ii,:)
                Ui=U(:,ii)
                XI_s=XI_p
                XI_s(1)=xiTemp(ii,1)*sdBT
                XI_s(2)=(xiTemp(ii,2)-muBT)/sdBT
                call up_ab(N,thXI,XI_s,XI_p,Ui,pAsd,pBsd,XI_n,ind)
                iAc(iter,ii,1)=ind
                iAc(iter,ii,2)=ind
                XIiter(iter+1,ii,:)=XI_n
        end do
!------------------------------------------------------------
end do
tOut=Thiter
a=XIiter(:,:,1)
b=XIiter(:,:,2)
c=XIiter(:,:,3)
end subroutine mc2_6
```

Appendix F

MCMC R Code

```
# ICC Function for Logistic model
irf.logistic<-function(ability=0,items=data.frame(A=1,B=0,C=0)){
irf.aux<-function(x){items$C+(1-items$C)/(1+exp(-1.7*items$A*(x-items$B)))}
t(sapply(ability,irf.aux))
}

# Generate response U
irtgen<-function(ability=0,A=1,B=0,C=0,type=c("logistic","normal")){
items<-data.frame(A,B,C)
n<-dim(items)[1]
nexmn<-length(ability)
data<-matrix(0,nrow=nexmn,ncol=n)
type<-match.arg(type)
if (type=="logistic"){
p<-irf.logistic(ability,items)}
if (type=="normal"){
p<-irf.normal(ability,items)}
```

```
try<-runif(nexmn*n,0,1)
data[try < p]<-1
return(data)
}

# Limit range of parameter
limitItemPar<-function(par,min,max){
par[par>max]<-max
par[par<min]<-min
return(par)
}

# Generate whole respose matrix
GenU<-function(seed=2016,nexam=1000,nitem=40,model="2PL",
                              thRange=c(-3,3),
                              aRange=c(0.5,2),bRange=c(-2,2),cRange=c(0,0.2)){
set.seed(seed)
        thTrue<-limitItemPar(round(rnorm(nexam,0,1),2),
                                    min=thRange[1],max=thRange[2])
                                    # Normal prior for ability
        cTrue<-rep(0,nitem)
        bTrue<-limitItemPar(round(rnorm(nitem,0,2),2),
                                    min=bRange[1],max=bRange[2])
                                    # Standard normal prior for B
if(model=="1PL"){
        aTrue<-rep(limitItemPar(round(rlnorm(1,0,0.5),2),
```

```
                                        min=aRange[1],max=aRange[2]),nitem)
                                        # Lognormal prior for A
} else if(model=="2PL"){
        aTrue<-limitItemPar(round(rlnorm(nitem,0,0.5),2),
                                        min=aRange[1],max=aRange[2])
                                        # Lognormal prior for A
} else if(model=="3PL"){
        aTrue<-limitItemPar(round(rlnorm(nitem,0,0.5),2),
                                        min=aRange[1],max=aRange[2])
                                        # Lognormal prior for A
        cTrue<-limitItemPar(round(rbeta(nitem,5,17),2),
                                        min=cRange[1],max=cRange[2])
                                        # Beta prior for C
}
U<-irtgen(thTrue,aTrue,bTrue,cTrue,type="logistic")
return(list(thTrue=thTrue,aTrue=aTrue,bTrue=bTrue,cTrue=cTrue,U=U))
}

# Estimate ability and item parameter using BILOG
RunBilog<-function(U,model){
library(irtoys)
setwd("C:/D_Drive/Research/01_Dissertation/97_BILOGMG/")

BilogOut<-est(U,model=model, engine="bilog",bilog.defaults=F,rasch=F, nqp=40)
Xi<-BilogOut$est
estT<-eap(resp=U, ip=Xi, qu=normal.qu())
```

```
Xise<-BilogOut$se
Xi[,1]=Xi[,1]/1.7
Xise[,1]<-Xise[,1]/1.7

detach("package:irtoys")
return(list(xiEst=Xi, xiSE=Xise, thEst=estT[,1], thSE=estT[,2]))
}

# Run MCMC using Fortran
mcmc<-function(outID,
               vol,
               model="2PL",
               ver=1,
               date=TRUE,
               simSize=c(25000,1000,20),
               qsd=c(0.3,0.2,0.1,1.4),
               seed=NA,
               sameResp=TRUE, rasch=TRUE){
### Setup Directory and initial time
baseDir="C:/D_Drive/Research/01_Dissertation/"
fortranDir=paste(baseDir,"41_Coding/",sep="")
saveDir=paste(baseDir,"45_Workspace/",sep="")
BilogDir<-paste(baseDir,"45_Workspace/bilogEst/",sep="")
niter=simSize[1];nexam=simSize[2];nitem=simSize[3];
t0<-Sys.time();print(paste("Begin ",t0,sep=""));flush.console()
```

```
### Call bilog result and U
loadBilog(BilogDir,nexam,nitem)
U<-get(paste("U",nexam,nitem,model,sep="_"))
BL<-get(paste("BL",nexam,nitem,model,sep="_"))
iniT=BL$thEst;iniXI=BL$xiEst

## Set the seed & qsd setting
if(is.numeric(seed)!=TRUE){seed=sample(1:1000,1)}

## Identical Response or same score
if(sameResp==TRUE&&rasch==TRUE&&model=="1PL"){IU<-sameScore(U);nIUrow=nrow(IU)
} else if (sameResp==TRUE){IU<-IRTsameAns(U);nIUrow=nrow(IU)
} else {IU=NULL}

## Single identical response handing
if(is.null(IU)){IU=c(1,1);nIUrow=0
} else if(is.null(nIUrow)){nIUrow=1}

dyn.load(paste(fortranDir,"MCvol",vol,".dll",sep="")) # Calling fortran code
mcResult<-.Fortran(paste("mc",substr(model,1,1),"_",ver,sep=""),
                                as.integer(seed),
                                as.integer(niter),as.integer(nexam),
                                as.integer(nitem),as.integer(nIUrow),
                                as.double(iniT),as.double(iniXI),
                                as.integer(U),as.integer(IU),
                                as.double(qsd),
```

```
                                          th=double(niter*nexam),
                                          a=double(niter*nitem),
                                          b=double(niter*nitem),
                                          c=double(niter*nitem),
                                          tAc=integer((niter-1)*nexam),
                                          iAc=integer((niter-1)*nitem*3))
dyn.unload(paste(fortranDir,"MCvol",vol,".dll",sep=""))
## Readable format
thetaDimName=list(paste("exam",1:nexam,sep=""),
                      paste("iter",1:niter,sep=""))
thetaDimNameAc=list(paste("exam",1:nexam,sep=""),
                      paste("iter",1:(niter-1),sep=""))
itemDimName=list(paste("item",1:nitem,sep=""),
                      paste("iter",1:niter,sep=""),c("a","b","c"))
itemDimNameAc=list(paste("item",1:nitem,sep=""),
                      paste("iter",1:(niter-1),sep=""),c("a","b","c"))
th<-matrix(mcResult$th,ncol=niter,byrow=TRUE,dimnames=thetaDimName)
thAc<-matrix(mcResult$tAc,ncol=(niter-1),byrow=TRUE,dimnames=thetaDimNameAc)
xi<-array(c(mcResult$a,mcResult$b,mcResult$c),dim=c(niter,nitem,3))
iAc<-array(mcResult$iAc,dim=c((niter-1),nitem,3))
xi<-array(c(t(xi[,,1]),t(xi[,,2]),t(xi[,,3])),
                      dim=c(nitem,niter,3),dimnames=itemDimName)
iAc<-array(c(t(iAc[,,1]),t(iAc[,,2]),t(iAc[,,3])),
                      dim=c(nitem,niter-1,3),dimnames=itemDimNameAc)
## Record Running time and save all objects
runTime=round(difftime(Sys.time(),t0,units="mins"),4)
```

```
assign(paste("mcmc",outID,"result",sep=""),
       list(outID=outID,model=model,
            ver=ver,niter=niter,
            nexam=nexam,nitem=nitem,
            runTime=runTime,seed=seed,
            rasch=rasch,idt=sameResp,
            qsd=qsd,
            preTh=th,preXi=xi,
            thAc=thAc,xiAc=iAc),
       envir = .GlobalEnv)
print(c(paste(niter," iterations",sep=""),paste(runTime," mins",sep="")));flush.console()
saveMCMCout(saveDir,outID,date)
}

# mcmc thinning
mcThin<-function(outID,nburn,nthin){
mcResult<-get(paste("mcmc",outID,"result",sep=""))
thinSeq1=seq(nburn+1,mcResult$niter,by=nthin)
thinSeq2=seq(nburn+1,mcResult$niter-1,by=nthin)
## MCMC Iteration
postTh<-mcResult$preTh[,thinSeq1]
postXi<-mcResult$preXi[,thinSeq1,]
## Acceptance Rate
postThAc<-mcResult$thAc[,thinSeq2]
postXiAc<-mcResult$xiAc[,thinSeq2,]
## Returning
```

```
return(list(outID=mcResult$outID,ver=mcResult$ver,
            model=mcResult$model,
            niter=mcResult$niter,nexam=mcResult$nexam,
            nitem=mcResult$nitem,
            runTime=mcResult$runTime,seed=mcResult$seed,
            rasch=mcResult$rasch,idt=mcResult$identicalResp,
            qsd=mcResult$qsd,
            preTh=mcResult$preTh,preXi=mcResult$preXi,
            thAc=mcResult$thAc,xiAc=mcResult$xiAc,
            th=postTh,xi=postXi,
            thinXiAc=postThAc,thinXiAc=postXiAc))
}

# linking function
linking<-function(outID,nburn,nthin,linkType=c("NOR","BLG")){
baseDir="C:/D_Drive/Research/01_Dissertation/"
fortranDir=paste(baseDir,"41_Coding/",sep="")
saveDir=paste(baseDir,"45_Workspace/",sep="")
BilogDir<-paste(baseDir,"45_Workspace/bilogEst/",sep="")

postThin<-mcThin(outID,nburn,nthin)
model=postThin$model
nexam=postThin$nexam
nitem=postThin$nitem
loadBilog(BilogDir,nexam,nitem)
```

```
BILOG<-get(paste("BL",nexam,nitem,model,sep="_"))
MCMCth<-apply(postThin$th,1,mean)
BILOGth<-BILOG$thEst
if(linkType=="NOR"){
        muBILOG=0;sdBILOG=1
        muMCMC=mean(MCMCth);sdMCMC=sd(MCMCth)
        A=sdBILOG/sdMCMC
        B=muBILOG-A*muMCMC
} else if(linkType=="BLG"){
        muBILOG=mean(BILOGth);sdBILOG=sd(BILOGth)
        muMCMC=mean(MCMCth);sdMCMC=sd(MCMCth)
        A=sdBILOG/sdMCMC
        B=muBILOG-A*muMCMC
} else {print("Wrong Link Type");break}
linkXi<-array(,dim=c(dim(postThin$xi)[1],dim(postThin$xi)[2],3))
linkTh<-A*postThin$th+B
linkXi[,,1]<-postThin$xi[,,1]/A
linkXi[,,2]<-A*postThin$xi[,,2]+B
linkXi[,,3]<-postThin$xi[,,3]
return(list(outID=postThin$outID,ver=postThin$ver,
                              model=postThin$model,
                              niter=postThin$niter,nexam=postThin$nexam,
                              nitem=postThin$nitem,
                              runTime=postThin$runTime,seed=postThin$seed,
                              rasch=postThin$rasch,idt=postThin$identicalResp,
                              qsd=postThin$qsd,
```

```
                        preTh=postThin$preTh,preXi=postThin$preXi,
                        thAc=postThin$thAc,xiAc=postThin$xiAc,
                        th=linkTh,xi=linkXi,
                        thinXiAc=postThin$thinXiAc,thinXiAc=postThin$thinXiAc))
}

# Post Hoc Linking
postHocLinking<-function(outID,linkType=c("NOR","BLG")){
baseDir="C:/D_Drive/Research/01_Dissertation/"
fortranDir=paste(baseDir,"41_Coding/",sep="")
saveDir=paste(baseDir,"45_Workspace/",sep="")
BilogDir<-paste(baseDir,"45_Workspace/bilogEst/",sep="")

mcResult<-get(paste("mcmc",outID,"result",sep=""))

model=mcResult$model
nexam=mcResult$nexam
nitem=mcResult$nitem
loadBilog(BilogDir,nexam,nitem)

BILOG<-get(paste("BL",mcResult$nexam,mcResult$nitem,mcResult$model,sep="_"))

#MCMCth<-apply(mcResult$preTh,1,mean)
BILOGth<-BILOG$thEst

muMCMC<-apply(mcResult$preTh,2,mean)
```

```
sdMCMC<-apply(mcResult$preTh,2,sd)

if(linkType=="NOR"){
        muBILOG=0;sdBILOG=1
        A=sdBILOG/sdMCMC
        B=muBILOG-A*muMCMC
} else if(linkType=="BLG"){
        muBILOG=mean(BILOGth);sdBILOG=sd(BILOGth)
        A=sdBILOG/sdMCMC
        B=muBILOG-A*muMCMC
} else {print("Wrong Link Type");break}

postHocXi<-array(,dim=c(mcResult$nitem,mcResult$niter,3))
postHocTh<-t(t(mcResult$preTh)*A+B)
postHocXi[,,1]<-t(t(mcResult$preXi[,,1])/A)
postHocXi[,,2]<-t(t(mcResult$preXi[,,2])*A+B)
postHocXi[,,3]<-mcResult$preXi[,,3]
return(list(outID=mcResult$outID,ver=mcResult$ver,
                                  model=mcResult$model,
                                  niter=mcResult$niter,nexam=mcResult$nexam,
                                  nitem=mcResult$nitem,
                                  runTime=mcResult$runTime,seed=mcResult$seed,
                                  rasch=mcResult$rasch,idt=mcResult$idt,
                                  qsd=mcResult$qsd,
                                  thAc=mcResult$preThAc,xiAc=mcResult$xiAc,
                                  th=postHocTh,xi=postHocXi))
```

```
}

# Post Hoc method
mcmcPostHoc<-function(dataID,saveID,date,linkType=c("NOR","BLG")){
baseDir="C:/D_Drive/Research/01_Dissertation/"
fortranDir=paste(baseDir,"41_Coding/",sep="")
saveDir=paste(baseDir,"45_Workspace/",sep="")

mcResult<-postHocLinking(dataID,linkType)
assign(paste("mcmc",saveID,"result",sep=""),
        list(outID=saveID,ver=mcResult$ver,
                model=mcResult$model,
                niter=mcResult$niter,nexam=mcResult$nexam,
                nitem=mcResult$nitem,
                runTime=mcResult$runTime,seed=mcResult$seed,
                rasch=mcResult$rasch,idt=mcResult$idt,
                qsd=mcResult$qsd,
                thAc=mcResult$thAc,xiAc=mcResult$xiAc,
                preTh=mcResult$th,preXi=mcResult$xi),
        envir = .GlobalEnv)
saveMCMCout(saveDir,saveID,date)
}
```