Approximate Bayesian Inference for Educational Attainment Models

Shuhrah Alghamdi

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Mathematics and Statistics
College of Science and Engineering
University of Glasgow

December 2022

Abstract

The rapidly expanding volume of educational testing data from online assessments has posed a problem for researchers in modern education. Their main goal is to utilise this information in a timely and adaptive manner to infer skills mastery, improve learning facilities and adapt them to individual learners. Over the past few years, several static statistical models have been proposed for extracting knowledge about skills mastery from item response data. However, realistic models typically lead to complex, computationally expensive fitting methods such as Markov chain Monte Carlo (MCMC). In an extensive comparison study, this thesis showed that the MCMC methods are unusable for streaming data, which appear to be very slow even for the efficient and fastest methods such as Hamiltonian Monte Carlo (HMC). On the other hand, the sequential Monte Carlo (SMC) methods have been widely used to reduce the time of dynamic Bayesian analysis. This thesis contributed to the application of two different settings of the SMC algorithms to the item response theory (IRT) model and compared the output to the MCMC results. However, the results showed that these methods were not fast enough to estimate students' ability in real-time and provide immediate feedback even for a small dataset. Moreover, the efficiency of the SMC methods depends on the user settings, which might be difficult for real-time inference or non-professional users.

Therefore, these methods will not scale well for streaming data and large-scale real-time systems. The main objective of this thesis is to develop approximate Bayesian inference based on the Laplace approximation method (LA), which allows faster inference for item response theory (IRT) models.

The LA estimation method's performance for the logistic IRT models has been compared with the MCMC method in simulation studies. Based on the results of several comparison criterion methods such as bias, RMSE, and Kendall's τ, the performance of the LA is very good in small, moderate, and relatively large sample size settings. The LA estimated abilities results are very close to the actual and MCMC values. In addition, LA resulted in between a 120 to 900 times speedup over MCMC, making it a more practical alternative for large educational testing

datasets. Also, this thesis investigated the issue of the high-dimensional covariance matrix for massive datasets, which may slow the LA method. Two solutions: using the LA diagonal and the block matrix techniques, have been proposed to reduce the computation cost. In addition, a novel sequential LA approach was proposed and successfully applied in this thesis to allow using LA in a dynamic inference. The result showed that this method is comparable to the full LA method. Moreover, the use of a real dataset confirmed that the proposed LA inference method provided similar estimates to MCMC estimation with much faster computation.

Declaration of Authorship

I declare that all the work presented in this thesis has been done by myself under the supervision of Dr. Nema Dean and Dr. Ludger Evers, except where otherwise stated. This thesis represents work completed, between 2018 and 2022 in Statistics in the School of Mathematics and Statistics at the University of Glasgow.

Acknowledgements

I would like to express my appreciation to my supervisors, Dr Nema Dean and Dr Ludger Evers for their scientific guidance and their kind support and encouragement through the stages of my PhD. I would not have been able to complete this work without their help.

I would also like to acknowledge the Ministry of Higher Education and the Saudi Arabian Cultural Bureau (SACB), which sponsored my PhD. Thanks also go to my employer, Princess Nourah bint Abdulrahman University, for the scholarship.

I am also grateful to my friends who shared this journey with me: Riham, Shaykhak and Hanadi. Thanks for all your support and the very enjoyable Friday meetings. Thanks also to all my other friends for their love and prayers.

I would also like to thank my parents, without whose encouragement and love I would never have been able to complete this work. Special thanks also to my brother and sisters for their support, love and encouragement.

Many thanks and appreciation go to my husband for his continued support, understanding and patience during my PhD studies. I also thank my lovely children, Ghena and Ahmad (my eyes), for their emotional support, their constant cheerfulness and their enduring love, which inspired me to keep going with my work.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In modern society, exams are widely used in different fields. For example, exams are used more and more for various educational purposes, like evaluating the educational qualification systems, students' learning improvement and measuring individual differences. This has encouraged the development of better exams and improved statistical methods for analysing exam results. Furthermore, the interest in managing common test issues, such as building tests, and investigating and interpreting the results, has increased recently. It all stimulated the development of item response theory (IRT). Fox (2010) stated that "In the second half of the twentieth century, item-based statistical models were used for the measurement of individual states like intelligence, arithmetic ability, customer satisfaction, or neuroticism". Today, some important educational tests, such as the Graduate Record Examination (GRE) and Scholastic Aptitude Test (SAT), are developed by using the IRT (Binh and Duy, 2016).

Educational measurement is an exciting field where many researchers look to construct objective measurements of examinees' knowledge, skills and abilities. Item response theory (Lord, 1952) is one of the most popular methods in education for estimating latent traits of examinees and the test (e.g. difficulty, discrimination, probability of answering correctly without previous knowledge, etc.). Latent traits are a specific type of constructs that refer to unobservable or unmeasurable objects (e.g. ability, attitude, etc.). The focus of the IRT models is on the pattern of the responses, not on the total score. There are two common possible choices of the functions that model the relationship between a latent ability and the probability of a correct response; the normal distribution, which is the cumulative distribution function (CDF) of the standard normal distribution, and the other is the CDF of the standard logistic distribution. Baker (1961) contributed an empirical comparison between logistic and normal ogive functions. This thesis will consider the use of the

logistic function.

Based on the number of latent traits being measured, IRT models can be divided into unidimensional or multidimensional models. Unidimensional IRT (UIRT) models are used when all test items measure one single latent trait (ability). On the contrary, multidimensional IRT (MIRT) models can deal with complex models by providing a different ability for each skill being measured by the test and modelling the relationships between examinees' ability and test items. Using a MIRT model allows separate inferences to be made about each skill or ability measured in the test (Walker and Beretvas, 2000). The focus of this thesis will be on UIRT models. Binary UIRT models can be applied to tests where two response categories are used, such as correct/incorrect or true/false responses. In the test that is designed with more than two opinions, Samejima (1969) proposed the graded response model for polytomous options, which is outside the scope of this thesis. Moreover, various models have been developed in the literature based on the number of items, including the one-parameter, two-parameter, and three-parameter models.

To date, many estimation methods have been developed for implementing item response theory (IRT) models. The utility of the IRT models mostly depends on the accuracy of item and ability parameter estimates. Different factors can influence the frequentist approach (classical marginal maximum likelihood) for estimating IRT model parameters. One of these factors could be sample size, where analyses with small numbers of samples may suffer from worse parameter estimation accuracy, affecting the standard errors for the estimates (more detail about this approach will be described in section 7.2.). To address this issue, using the prior distributions in the Bayesian approach can help increase the accuracy of IRT parameter estimation with small sample sizes, as suggested by Swaminathan et al. (2003). With the help of modern computer techniques, Bayesian estimation methods have been widely used for IRT models via Markov chain Monte Carlo (MCMC). However, the challenge arises when one needs to estimate the parameters of interest in a dynamic system, where the data arrives in real-time continuously, such as in real-life scenarios when students take a test at different times or on other days. In reality, teachers and students are interested in immediate test results, especially the estimation of the students' ability.

Although real-time inference and online methods arise in several areas, including statistics, network and machine learning, the application of online inference in IRT

models is still limited. As far as is known, a few studies have considered the estimation of the IRT model parameters in real-time, such as Weng et al. (2018) and Su et al. (2018). However, both studies are outside of the educational area and focus on online product ratings using IRT models.

In the case of real-time inference, MCMC techniques may be unusable since it is computationally expensive to estimate with streaming data. There is however extensive literature on applying MCMC methods to static IRT models, some of which will be briefly mentioned in Chapter 4. In this thesis, a comparison study will be carried out between two MCMC algorithms; Gibbs Sampler within Metropolis algorithm and Hamiltonian Monte Carlo algorithm, to assess the usability of the MCMC method in real-time inference.

To reduce the computational time of dynamic Bayesian inference, some authors such as Isard and Blake (1996), Berzuini et al. (1997), Liu and Chen (1998) and Gilks and Berzuini (2001) have been developing more efficient methods that combine importance sampling and the Monte Carlo method to explore a sequence of posterior distributions. This method is known as the sequential Monte Carlo (SMC) method, which involves importance sampling and resampling, allowing an efficient inference for real-time whenever new data becomes available. However, as far as is known, currently, there is no application of SMC methods to IRT models. This thesis will contribute applications of two different settings of the SMC methods to the IRT models, including studying their properties, tuning parameters and comparing their performance to MCMC methods. However, most of the SMC techniques become computationally expensive as the dynamic process evolves.

The speed and volume present considerable challenges to apply MCMC and SMC methods to the IRT model when real-time inference is required or for dynamic problems. Approximation methods can be valuable alternatives to MCMC for Bayesian inference. Approximation methods, such as variational Bayes, expectation propagation, Laplace approximation, and so on, are considered simple and computationally cheap. Therefore, they have been widely used in machine learning and neural-network to solve large data issues. Wu et al. (2020) applied variational Bayes method to Multidimensional item response theory (MIRT) models and compared the results to Hamiltonian Monte Carlo and Maximum Likelihood Estimation (MLE). Their results suggested that variational inference is faster than the MCMC method without losing the accuracy of the estimation results. Ulitzsch and Nestler (2022) applied variational inference through Stan (Stan Development Team, 2022) to MIRT and compared the output to MCMC and MLE. Although their focus was on

estimating the item parameters, they concluded that Stan's built-in VB algorithm could not be a useable alternative for estimating MIRT models.

This thesis will contribute to using the Laplace approximation (LA) methods on the IRT models. A comprehensive comparison between the estimation results from LA and MCMC will be carried out, taking into account the accuracy and the speed. This thesis will discuss in large detail the issue of the high dimensional covariance matrix problem, providing two contributed solutions. A novel approach will be provided for Laplace approximation in dynamic IRT models.

1.1 Thesis Goals and Contributions

This thesis aims to find a fast computation method within a fully Bayesian framework to estimate the item response theory models parameters for real-time inference or massive datasets. Although the main goal is to evaluate the students' ability in real-time, the items' parameters estimate are taken into account also. This thesis extensively studies using the Laplace approximation method as an alternative to the Markov chain Monte Carlo (MCMC) method and studies some other possible methods for dynamic IRT models. The main contributions of this thesis can be summarised as follows:

- Study the MCMC methods in detail and discuss the tuning of the two algorithms employed in this work. The Hamiltonian Monte Carlo in this work is implemented in **R** code rather than Stan.
- Apply the sequential Monte Carlo approach in two different algorithm settings to introduce the data in the dynamic 1PL IRT model. Discuss in detail the effect of tuning the parameters, such as the number of particles, the scaling factor of the proposal variance, the number of intermediate distributions and the proposal variance in the MCMC move step.
- Implement an extensive comparison study between the MCMC and Laplace approximation methods for simple 1PL IRT and more complex 2PL models, which involves accuracy, ordering the ability estimates and computational costs.
- Investigate potential high dimensional covariance matrix issues using Laplace approximation and provide some possible solutions.
- Provide a novel approach to the sequential Laplace approximation method in a dynamic IRT model. Discuss the proposed method in greater detail using simulated datasets.

- Apply the proposed Laplace approximation method to the real dataset obtained from the General Aptitude Test (GAT) provided by the Education and Training Evaluation Commission in Saudi Arabia for this thesis.

1.2 Outline of the Thesis

This thesis is divided into 8 chapters. A brief overview of each chapter and a description of the general structure of this thesis is now given.

In Chapter 2: This chapter will present the statistical background to the Bayesian inference methods. The chapter will provide the reader with a literature review of Bayesian inference and the MCMC methods, including three types of algorithms; Gibbs sampler, Metropolis-Hastings and Hamilton Monte Carlo. The idea of sequential Monte Carlo will be discussed in this chapter also. Finally, this chapter will introduce the Laplace approximation method and explain how it works by giving an example.

In Chapter 3: This chapter will present a literature review of the unidimensional item response theory model (UIRT). This will include three IRT parameter logistic models; 1PL, 2PL and 3PL. Identification issues will be discussed and addressed in this chapter. This chapter does not intend to provide a complete overview of previous inference studies but will briefly mention some of them.

In Chapter 4: This chapter will provide an application of Bayesian inference with MCMC on UIRT Models. A general overview of the previous application of MCMC in IRT models and the idea of the prior distribution choices will be considered in this chapter. A comparison study between two MCMC algorithms; Gibbs Sampler within Metropolis and Hamiltonian Monte Carlo, will be presented in this chapter.

In Chapter 5: The application of two different settings for sequential Monte Carlo algorithms will be presented in this chapter. In addition, a comparison study of each method to the MCMC method results will be carried out.

In Chapter 6: A detailed description of the use of the Laplace approximation for the IRT model is given in this chapter. In addition, comparison studies to the MCMC results, considering three possible sample sizes; small, moderate and relatively large, will be implemented and discussed in greater detail. Furthermore,

the issue of the high-dimensional covariance matrix will be discussed, and some possible solutions will be proposed. Finally, this chapter will also provide an application of the Laplace approximation method in dynamic IRT models.

In Chapter 7: A case study which considers an application of the Laplace approximation method explained in Chapter 6 on the General Aptitude Test (GAT) will be provided and compared the output to the MCMC and marginal maximum likelihood method (MML).

In Chapter 8: The results obtained from the experimental work will be reviewed in this chapter, with a brief discussion on possible directions for future work.

Chapter 2

Bayesian Inference Methods

This chapter reviews the statistical concepts and methods in Bayesian inference. The main goals are to review essential concepts of the Bayesian Inference methods that will be used and developed throughout this thesis. The review will consider the basic idea of Bayesian statistics in sections 2.1, 2.2 and 2.3. Bayesian Inference with Markov chain Monte Carlo will be discussed in sections 2.4, 2.5, providing more details on the specific MCMC algorithms: Metropolis-Hastings 2.4.1, Gibbs sampler 2.4.2 and Hamiltonian Monte Carlo 2.4.3. Monte Carlo methods 2.6 and one approximation Method, known as Laplace Approximation 2.8 will also be discussed in detail in this chapter.

2.1 Introduction to Bayesian Statistics

In statistical inference, the interpretations of probability can be divided into two prime categories: classical inference and Bayesian inference. The differences between these views are the essential nature of the probability. In classical inference, it is assumed that the true values of the parameters interest θ are fixed, and the observed data is random. On the other hand, in Bayesian inference, parameters are treated as random variables. Thus, for each parameter, we can assign probability distributions representing our degree of belief. This distribution is the so-called prior distribution, and it can be updated according to further information or new observations to give a posterior distribution.

The idea of updating our beliefs can be done through Bayes theorem. This theorem was developed by Thomas Bayes (Bayes, 1763), which offers a way to combine our confidence using the prior distribution and the data using the likelihood function. According to this theory, the combination of the likelihood function and the prior distribution is expressed in the posterior distribution, which summarises all the information about the parameter of interest θ after observing data.

Originally, Bayes' theorem is applied to probability, and the basic formula simply states:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}, \tag{2.1}$$

where $p(A|B)$ is the conditional probability that tells how likely event A occurs, given event B has happened, and $p(A), p(B)$ is the marginal probability of observing event A and B. This formula can be expressed in terms of random variables as following:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \tag{2.2}$$

$p(\theta)$ is the prior distribution for unknown parameter θ, and $p(D)$ is a normalisation constant (marginal distribution) of the observed data D. The prior belief can then be updated after observing the data D via the likelihood $p(D|\theta)$ to give the posterior distribution $p(\theta|D)$.

The normalisation constant $p(D)$ is the marginal probability of the data D, which does not depend on the parameter θ. Therefore, the posterior distribution can be expressed as proportional to the likelihood multiplied by the prior distribution for the parameters θ.

$$p(\theta|D) \propto p(D|\theta)p(\theta). \tag{2.3}$$

2.2 Prior Distributions

The concept of the prior distribution has already been mentioned in the previous section. This section will provide some details about different types of prior distributions.

The inference for a given parameter θ depends on the data and the choice of prior; therefore, by using different sets of priors, the inference for unknown parameter θ will change. The prior distribution can represent all the information we know about the parameter θ or our ignorance.

Priors can be divided into informative, weakly informative and non-informative. The informative term expresses our knowledge about θ, which might come from available studies on similar data sets, a literature review or experts (O'Hagan et al., 2006). On the other hand, the prior can be chosen simply to represent the ignorance or the lack of prior information, hence being weakly informative priors. In this

case, the information on the posterior distribution will mainly be derived from the data (via the likelihood) since the prior has minimal influence on the posterior, keeping the posterior within reasonable bounds. Finally, the prior distributions are non-informative if they are flat over the entire real number line and thus have no information to influence the posterior distribution.

An example of non-informative priors on the interval $[0, 1]$ would be a uniform distribution; $Uniform(0, 1)$. This can be interpreted as we have no prior information and all possible values are equally likely a priori on the unit interval. A common example of a weakly informative prior is a normal distribution with very large variance (e.g. $N(0, 100)$). Even though weakly informative priors are widely used, extra care should be taken when applying them. A fundamental problem would be produced by a type of priors known as improper priors, such as $Uniform(-\infty, \infty)$, where the prior here is not a finite density and cannot be integrated into one. Improper priors can lead to an improper posterior distribution, where the inference is invalid. To define a posterior distribution as proper, we have to make sure that the integral of the normalising constant $p(D)$ in Bayes' theorem is a positive finite value for all D, where D is the data.

In some cases, it is useful and possible to use a conjugate prior; the basic idea is the posterior distribution, and the prior distribution belongs to the same family. The advantage of that is to obtain a closed-form expression which makes calculation straightforward to evaluate. For example, the beta distribution $Beta(a, b)$ is conjugate to the binomial distribution $binomial(n, \theta)$. Thus, if the likelihood from the binomial distribution with known n and unknown θ and if our prior belief about θ is a beta distribution, the posterior will be simply a beta distribution $Beta(x + a, n - x + b)$ with different parameters where x is the observed success in n trails. For more information about the types of prior distributions, see Gelman (2002).

2.3 Inference

In order to use Bayesian modelling, we should be able to compute the posterior distributions for the model parameters. In some simple models, calculating the posterior distributions is straightforward, such as considering a conjugate prior (as discussed in section 2.2), which can help provide a posterior distribution with standard distributional form. However, in more complex models where a conjugate prior cannot be applied, the computation of the posterior distribution might require more advanced methods such as numerical simulation.
An example of these methods is the Monte Carlo method. These methods are

mathematical approaches that use a large number of samples drawn randomly from the posterior distribution to estimate the distributions of the model parameters. One can then obtain summary statistics by calculating the sample mean, variance and quantiles. The problem with this method appears in high dimensional models where it may not work well. These models might require a large number of different parameters and hence high dimensions of the prior distributions. Consequently, we have to evaluate the posterior distributions numerically in high dimensional space (Van Ravenzwaaij et al., 2018). A greater explanation of this method will be provided in section 2.6.

Next section will introduce the most common method used to draw a sample from a high dimensional space known as Markov chain Monte Carlo (MCMC). Also, this section will explain three different MCMC methods which will be carried out in this thesis; Metropolis-Hastings algorithm, Gibbs sampler and Hamiltonian Monte Carlo.

2.4 Bayesian Inference with Markov chain Monte Carlo (MCMC)

Markov chain Monte Carlo (MCMC) is the most common method for drawing samples from high dimensions and complex posterior distributions (Hastings, 1970). However, these samples will not be independent but will be drawn from a Markov chain. This Markov chain is a sequence of random variables (θ_n) with the property that the new sample only depends on the current sample. This can be seen as:

$$\pi(\theta_{n+1}|\theta_0, \theta_1, ..., \theta_n) = \pi(\theta_{n+1}|\theta_n); n = 0, \ldots, \infty. \tag{2.4}$$

The objective of MCMC simulation is constructing a Markov chain that, after a long run time, will converge to the posterior distribution of interest as its stationary distribution $\pi(\theta) = p(\theta|D)$. For more details about the Markov chain concepts and their properties, see Gamerman and Lopes (2006). The following subsections will introduce three MCMC algorithms, which are the focus of this thesis.

2.4.1 Metropolis-Hastings Algorithm (M-H)

The Metropolis-Hastings algorithm (Hastings, 1970) (M-H) is an MCMC method, which can be used to sample from the posterior distribution by using a proposal distribution. The algorithm is based on simulating a candidate sample θ^* from a proposal distribution q conditional on the current value $\theta^{(t-1)}$, $q(\theta^*|\theta^{(t-1)})$ (Chib and Greenberg, 1995), then make use of a certain acceptance probability $\alpha(\theta^*, \theta^{(t-1)})$ to accept or reject the new value θ^* as following;

$$\alpha(\theta^*, \theta^{(t-1)}) = \min\left\{1, \frac{p(\theta^*)q(\theta^{(t-1)}|\theta^*)}{p(\theta^{(t-1)})q(\theta^*|\theta^{(t-1)})}\right\}; t = 1, \ldots, T. \tag{2.5}$$

$p(\theta^*)$ is the target distribution (posterior distribution in a Bayesian framework) at θ^*. The acceptance probability $\alpha(\theta^*, \theta^{(t-1)})$ is compared to a uniform random variable u on the interval [0,1]; $u \sim Unif(0, 1)$. The new value θ^* is accepted if $\alpha(\theta^*, \theta^{(t-1)})$ is greater than u, otherwise, it will be rejected. If the proposed value θ^* is rejected, the chain will stay at the current value $\theta^{(t-1)}$, and we set $\theta^{(t)} = \theta^{(t-1)}$. The M-H algorithm for generating samples from the posterior (target) distribution is displayed in Algorithm 1.

Algorithm 1 Metropolis-Hastings Algorithm

1: Starting with the initial value $\theta^{(0)}$. Set the maximum number of iterations T
   For iteration $t = 1, \ldots, T$
2: Propose a candidate value θ^* ; $\theta^* \sim q(\theta^*|\theta^{(t-1)})$
3: Compute
$$\alpha(\theta^*, \theta^{(t-1)}) = \min\left\{1, \frac{p(\theta^*)q(\theta^{(t-1)}|\theta^*)}{p(\theta^{(t-1)})q(\theta^*|\theta^{(t-1)})}\right\}$$
4: Set $\theta^{(t)} = \theta^*$ with the probability $\alpha(\theta^*, \theta^{(t-1)})$
   else
5: Reject θ^* and remain at current state, $\theta^{(t)} = \theta^{(t-1)}$

One of the special cases of M-H is known as the Metropolis algorithm; this was first introduced by Metropolis et al. (1953). In this setting, the proposal distribution is chosen to be symmetric e.g. $q(\theta^*|\theta^t) = q(\theta^t|\theta^*)$. When q is symmetric the acceptance probability (2.5) simplifies to:

$$\alpha(\theta^*, \theta^{(t-1)}) = \min\left\{1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right\} \tag{2.6}$$

The M-H algorithm's performance and efficiency rely on the choice of the proposal distribution (Rosenthal et al., 2011). There are two types of proposal distribution, which are used frequently. The first type is normal random walks. In this type of proposal, a random walk is generated using a normal distribution with a mean equal to the current parameter value $\theta^{(t-1)}$;

$$q(\theta^*|\theta^{(t-1)}) = N(\theta^{(t-1)}, \sigma^2). \tag{2.7}$$

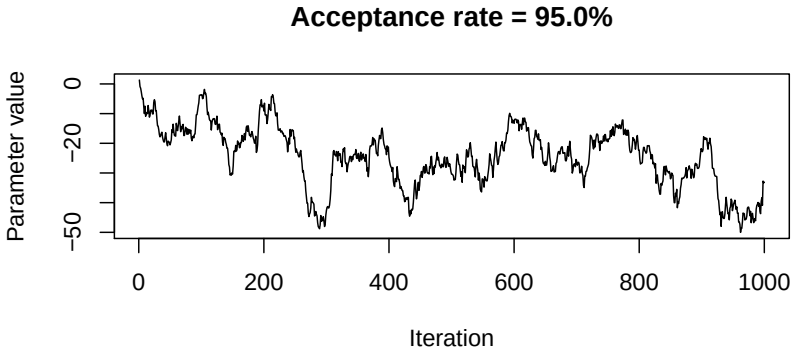
The variance σ^2 should be tuned to control the algorithm's performance and obtain an efficient algorithm. If σ^2 is chosen to be small, the acceptance probability (2.5) will be high, which means that the algorithm will accept a slight movement in the state space. As a result, θ^* will move very incrementally, causing a strong correlation in the resulting Markov chain as displayed in Figure 2.1a. Conversely, if σ^2 is chosen to be large, the θ^* will move very fast. Yet, the acceptance probability will be low, and most of the new proposed values will be rejected, leading to slow convergence towards the target distributions as shown in Figure 2.1b. Both of these situations are referred to as bad mixing, where mixing time for a Markov chain is the number of steps required for the Markov chain to approach the posterior distribution. In the literature, the aim usually is to have an acceptance probability between 0.20 and 0.60. For example, Gelman et al. (2013) suggested that the optimal acceptance probability for one dimension is around 0.44, and in high dimensions, this acceptance probability decreases to 0.23.

The second common choice of the proposal distribution is an independent normal distribution, which is given by;

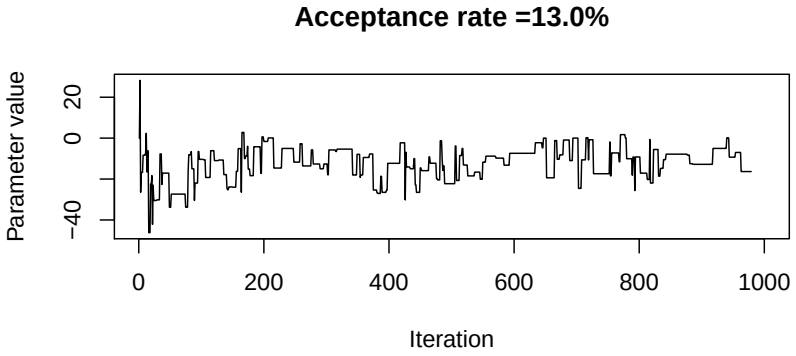
$$q(\theta^*|\theta^{(t-1)}) = q(\theta^*) = N(\mu, \sigma^2). \tag{2.8}$$

In this type of proposal distribution, the proposal value θ^* does not depend on the current value $\theta^{(t-1)}$ which means the previously accepted value cannot be used as guidance to converge to the posterior distribution.

For more explanation of the Metropolis-Hastings algorithm and the way of choosing and tuning the proposal can be found in Roberts et al. (1997), Gelman et al. (2013) and Robert et al. (2010).



(a) Trace plot of small choice of σ^2, and large acceptance probability, a small movement



(b) Trace plot of large choice of σ^2, and small acceptance probability, a large movement

Figure 2.1: Comparison of two Markov chains with different choices of the variance σ^2 for the proposal distribution.

2.4.2 Gibbs Sampler Algorithm

The Gibbs sampler is a special case of the M-H algorithm and was introduced first by Geman and Geman (1984) and then developed by Gelfand and Smith (1990). It is used when the full conditional distribution is available; where the proposal distribution in this method can be split into the conditional distribution of the posterior distributions.

The idea is sampling each variable $\theta_i \in \boldsymbol{\theta}$ from its conditional distribution in turn, depending on the current value of all the other variables, $\boldsymbol{\theta_{-i}} \in \boldsymbol{\theta}$ in the joint distribution. Using a Gibbs sampler requires the conditional distributions of θ_i to follow a standard distributional form. This results in always accepting each move (with probability one) in 2.5. Using the fact that the proposal distributions for the Gibbs sampler are the posterior conditionals, the proposal distribution is given as:

$$q(\theta_i^*, \boldsymbol{\theta}_{-\boldsymbol{i}}^{(\boldsymbol{t})} | \theta_i^{(t)}, \boldsymbol{\theta}_{-\boldsymbol{i}}^{(\boldsymbol{t})}) = p(\theta_i^* | \boldsymbol{\theta}_{-i}^{(t)}) \tag{2.9}$$

Applying this proposal to the Metropolis-Hastings acceptance probability (2.5) yields:

$$\alpha(\theta_i^*, \theta_i^{(t)}) = \min\left\{1, \frac{p(\theta_i^*, \boldsymbol{\theta}_{-i}^{(t)}) q(\theta_i{}^{(t)}, \boldsymbol{\theta}_{-i}^{(t)} | \theta_i^*, \boldsymbol{\theta}_{-i}^{(t)})}{p(\theta_i^{(t)}, \boldsymbol{\theta}_{-i}^{(t)}) q(\theta_i^*, \boldsymbol{\theta}_{-i}^{(t)} | \theta_i^{(t)}, \boldsymbol{\theta}_{-i}^{(t)})}\right\} \tag{2.10}$$

$$= \min\left\{1, \frac{p(\theta_i^*, \boldsymbol{\theta}_{-i}^{(t)}) p(\theta_i^{(t)} | \boldsymbol{\theta}_{-i}^{(t)})}{p(\theta_i^{(t)}, \boldsymbol{\theta}_{-i}^{(t)}) p(\theta_i^* | \boldsymbol{\theta}_{-i}^{(t)})}\right\} \tag{2.11}$$

$$= \min\left\{1, \frac{p(\theta_i^* | \boldsymbol{\theta}_{-i}^{(t)}) p(\boldsymbol{\theta}_{-i}^{(t)}) p(\theta_i^{(t)} | \boldsymbol{\theta}_{-i}^{(t)})}{p(\theta_i^{(t)} | \boldsymbol{\theta}_{-i}^{(t)}) p(\boldsymbol{\theta}_{-i}^{(t)}) p(\theta_i^* | \boldsymbol{\theta}_{-i}^{(t)})}\right\} \tag{2.12}$$

$$= \min(1, 1) = 1 \tag{2.13}$$

Here, we made use of the chain rule, where we wrote the full joint distribution $p(\theta_i^*, \boldsymbol{\theta}_{-i}^{(t)})$ as the product of two terms: $p(\theta_i^* | \boldsymbol{\theta}_{-i}^{(t)}) p(\boldsymbol{\theta}_{-i}^{(t)})$. For more details about the Gibbs sampler, see Gelfand (2000).

The Gibbs sampler algorithm for drawing T samples from the posterior distribution is given in Algorithm 2.

Algorithm 2 Gibbs Sampler Algorithm

1: For each iteration $t = 1, \ldots, T$, and $(\theta_i)_{i=1}^k$; starting with the initial value
$\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)})$
2: Sample $\theta_1^{(t)}$ from its conditional distribution
$\theta_1^{(t)} \sim p(\theta_1^{(*)} | D, \theta_2^{(t-1)}, \ldots, \theta_k^{(t-1)})$
3: Sample $\theta_2^{(t)}$ from its conditional distribution
$\theta_2^{(t)} \sim p(\theta_2^{(*)} | D, \theta_1^{(t)}, \ldots, \theta_k^{(t-1)})$
$\vdots$
4: Sample $\theta_k^{(t)}$ from its conditional distribution
$\theta_k^{(t)} \sim p(\theta_k^{(*)} | D, \theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_k^{(t-1)})$
5: Repeat step 2 and 3 until convergence.

2.4.3 Hamiltonian Monte Carlo (HMC) Algorithm

Many Markov chain Monte Carlo algorithms may suffer from random walk behaviour, which leads to inefficient algorithms. The main problem is that a large number

of sample sizes are needed to obtain reasonable effective samples. Thus, complex models may take a long time to run. Hence, improving mixing and reducing random walk behaviour is important to accelerate MCMC methods. Some strategies such as reparameterisation and adaptive acceptance rate could improve the mixing, but this random walk behaviour may remain (Gelman et al., 2013) for high-dimensional posterior distributions. HMC is an MCMC algorithm that avoids inefficient random walk behaviour and autocorrelated parameters using a physical system known as Hamiltonian dynamics. This method was originally developed in physics by Duane et al. (1987) and then was introduced to statistics by Neal et al. (2011).

This section will introduce a brief introduction to the Hamiltonian Monte Carlo (HMC), and the explanations here will closely follow Neal et al. (2011) and Betancourt (2017).

Hamiltonian dynamics

Before presenting Hamilton Monte Carlo, the basic concept of Hamilton dynamics in one dimension needs to be defined. In physics, Hamiltonian dynamics is a way of describing the movement of a frictionless object. The description of the object's motion depends on its position q and momentum r (equal to the product of the object's mass and its velocity) at some time t. For each position q, there is related energy called potential energy $U(q)$, and for each momentum r, there is a related kinetic energy $K(r)$. The total energy then is the sum of potential energy and kinetic energy, which is known as the Hamiltonian function $H(q, r)$:

$$H(q, r) = U(q) + K(r). \tag{2.14}$$

In Bayesian inference applications, the potential energy $U(q)$ in this system is defined by the negative of the log posterior distribution ($-\log p(q|D)$) where the position q is the variable that we want to estimate (e.g. if we are going to estimate the θ parameter then $q = \theta$). For each variable q, we add an auxiliary momentum variable r to draw a proposal distribution that can use gradient information in the posterior distribution. In most applications of HMC, the kinetic energy $K(r)$ is a standard normal distribution. HMC in Bayesian inference applications will be discussed in detail in the next section.

Hamiltonian dynamics satisfies the following Hamiltonian equations:

$$\frac{dq}{dt} = \frac{\partial H}{\partial r} \tag{2.15}$$

$$\frac{dr}{dt} = -\frac{\partial H}{\partial q}, \tag{2.16}$$

which determine how the system change through time t.

Hamiltonian Monte Carlo Algorithm (HMC)

This subsection will discuss how we can use Hamiltonian dynamics for MCMC. As mentioned earlier, using a proposal distribution in Metropolis-Hastings algorithms will result in a random walk without considering further information about the target distribution. Hence, if the target distribution is differentiable, its shape can be accessed through its gradient. The main idea is to develop the Hamiltonian function $H(q, r)$ and use the results of the Hamiltonian dynamics to help us efficiently explore the posterior distribution using the gradient. This gradient indicates which direction the trajectory goes in to find the high probability state, and we can draw from the proposal distribution in that direction.

In order to relate between the Hamiltonian function $H(q, r)$ and the posterior distribution $p(q|D)$, we can use some basic concepts from statistical mechanics called the canonical distribution. The idea is for some energy functions $E(\theta)$, over a set of variables θ, we can define the canonical distribution as:

$$p(\theta) = \frac{1}{Z} \exp(-E(\theta)), \tag{2.17}$$

where the variable Z is a normalising constant, which is used here to scale the canonical distribution to integrate or sum to one. As we discussed early in section 2.1, MCMC methods can sample from an unnormalised probability distribution, and hence this can be written as:

$$p(\theta) \propto \exp(-E(\theta)) \tag{2.18}$$

Now, as we saw in the previous section in 2.14, the energy function for Hamiltonian dynamics is the sum of potential and kinetic energies:

$$E(\theta) = H(q, r) = U(q) + K(r) \tag{2.19}$$

Therefore, the joint canonical distribution for q and r for the Hamiltonian function is given by

$$\begin{aligned} p(q, r) &= \exp\left(-H(q, r)\right) && (2.20)\\ &= \exp\left(-U(q) - K(r)\right) && (2.21)\\ &= \exp\left(-U(q)\right)\exp\left(-K(r)\right) && (2.22)\\ &= p(q)p(r) && (2.23) \end{aligned}$$

We can clearly see that the two variables are independent, and each variable has canonical distributions as:

$$\begin{aligned} p(q) &= \exp\left(-(U(q)\right) && (2.24)\\ p(r) &= \exp\left(-K(r)\right) && (2.25) \end{aligned}$$

The canonical distribution $p(r)$ is independent of q. Therefore, we can sample the momentum variable r from any distribution. In most applications of HMC, the common choice is a normal distribution with a zero-mean, and for simplicity, the variance is chosen to be one;

$$p(r) = \exp(-r^2/2)) \tag{2.26}$$

With this form of $p(r)$, equation (2.25) can be written as

$$\begin{aligned} \exp(-r^2/2)) &= \exp(-K(r)) && (2.27)\\ K(r) &= r^2/2 && (2.28) \end{aligned}$$

Now we can use equation(2.24) to define a formula for the potential energy as following:

First, take the log of both sides;

$$\log(p(q)) = \log(\exp(-(U(q))) \tag{2.29}$$

The *log* will cancel out the *exp* function then we will get a formula for $U(q)$;

$$U(q) = -\log(p(q)) \tag{2.30}$$

In Bayesian inference, the main interest is in the posterior distribution for the model parameters. Hence, $p(q)$ here represents the posterior distribution which can be written in term of prior distribution $p(q)$ times the likelihood function $p(D|q)$);

$$U(q) = -\log(p(q)p(D|q)), \tag{2.31}$$

where the position variable q can be replaced by the model parameters.

Simulating Hamiltonian dynamics (The Leapfrog Method)

As mentioned early, the differential equations (2.15) and (2.16) determine an object's motion via time t, which is a continuous variable. Hence, in practice, these equations cannot be solved analytically and to simulate Hamiltonian dynamics, numerical methods are required to discretize time. This can be done by dividing the time interval of t into a series of smaller length intervals ϵ. There are several numerical methods, but the interest in this thesis will be in one method called the Leapfrog integrator.

The leapfrog algorithm consecutively updates the momentum variable r and the position variable q. We start by updating the momentum variable r for a small time

interval $(\epsilon/2)$, then update the position variable q for a longer time interval(ϵ), and then end up by completing r update for another small time interval $(\epsilon/2)$.

The leapfrog algorithm can be summarised as follows:

1. Update the momentum variable r a half step in time:

$$r(t+\epsilon/2) = r(t) - (\epsilon/2)\frac{\partial U}{\partial q}(q(t))$$

2. Update the position variable q a full step in time; using the new value of r resulting from step (1).

$$q(t+\epsilon) = q(t) + \epsilon p(t+\epsilon/2)$$

3. Update the momentum variable r another half step in time; using the new value of q resulting from step (2).

$$r(t+\epsilon) = r(t+\epsilon/2) - (\epsilon/2)\frac{\partial U}{\partial q}(q(t+\epsilon))$$

The Leapfrog method can be run for L leapfrog steps to draw a total of $L * \epsilon$ time. At the end of these simulations, the resulting state will be denoted by (r^*, q^*), representing the proposal points.

Hamiltonian Monte Carlo (HMC) Algorithm

Now that we have a better idea about Hamiltonian dynamics and how they can be simulated, the Hamiltonian Monte Carlo algorithm can be introduced. In HMC, to explore the posterior (canonical) distribution, which is defined by $U(q)$, the Hamiltonian dynamics are used as a proposal distribution for a Markov Chain. The first step is drawing a new value for the momentum r that is independent of the current position variable values q;

$$r \sim Normal(0, 1)$$

Next, starting at current state (q, r), we can propose a new state (q^*, r^*) using Hamiltonian dynamics. As explained in the previous section, we can simulate Hamiltonian dynamics using the leapfrog method for L number of steps and ϵ step size. The choice of these two parameters L and ϵ will affect the algorithm's performance (more details will be given in the next section), so extra care should be taken to tune them. Finally, the new proposed values are accepted or rejected using the Metropolis acceptance probability(2.5);

$$\begin{aligned} \alpha &= \min(1, exp(-H(q^*, r^*)/exp(-H(q, r)) && (2.32)\\ &= \min(1, exp(-H(q^*, r^*) + H(q, r)) && (2.33)\\ &= \min(1, exp(-U(q^*) + U(q) - K(r^*) + K(r))) && (2.34) \end{aligned}$$

If the new value is rejected, the next Markov chain state is set the same as the current value (q, r). We can perform an approximate procedure listed as Algorithm 3.

Algorithm 3 Hamilton Monte Carlo

1: Starting with the initial value q^0, set number of iterations T.
   For iteration t;
2: Sample an initial momentum variable $r^0 \sim p(r)$; $r \sim Normal(0, 1)$.
3: Run leapfrog algorithm for L steps and ϵ step size starting at (q^0, r^0) to obtain a new proposed value (q^*, r^*).
4: Compute:
   $\alpha = \min(1, exp(-U(q^*) + U(q^{(t)}) - k(r^*) + k(r^{(t)})))$
5: Draw a random number $u \sim Unif(0, 1)$.
6: If u $\leq \alpha$ accept the proposed state position q^* and set $q^{(t)} = q^*$
   else;
7: Reject q^* and remain at current state; $q^{(t)} = q^{(t-1)}$

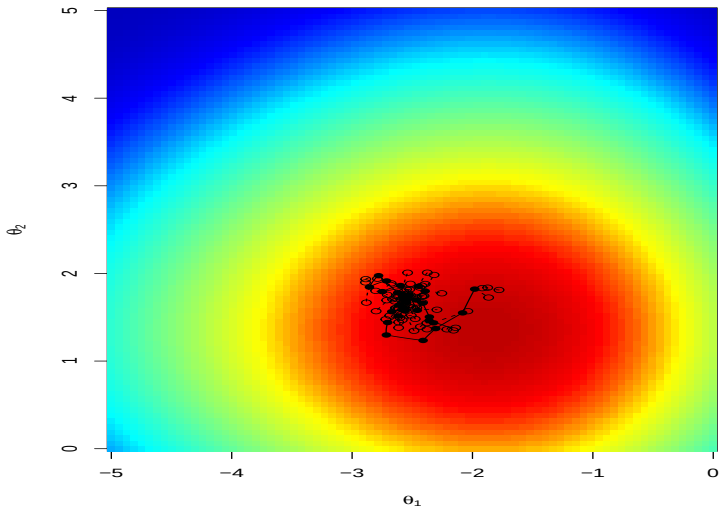
Tuning Hamiltonian Monte Carlo

While HMC is recognised as an efficient method, the algorithm's performance is sensitive to the choice of step size ϵ and the number of steps L. Unfortunately, selecting an optimal step size of ϵ is not easy. When ϵ is chosen to be too small, the Hamiltonian dynamics system does not explore the target distribution rapidly, which causes wasted computation time as shown in figure 2.2a. On the other hand,

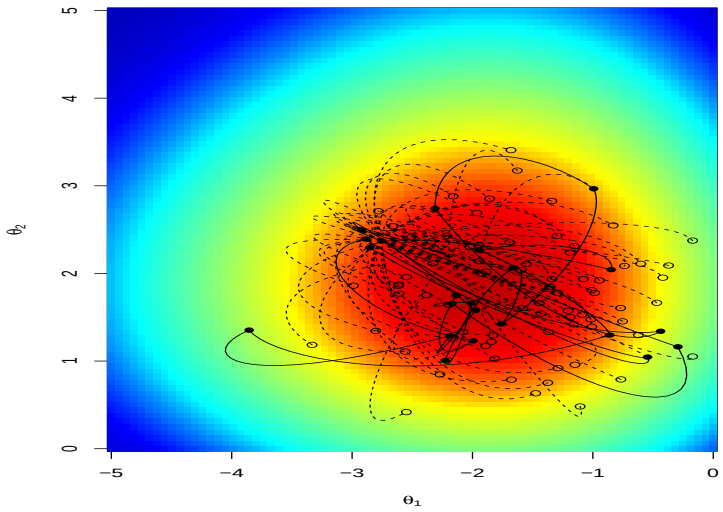
when ϵ is chosen to be too large, it will reject most of the proposal points during the Metropolis step, and hence very low acceptance probability as shown in figure 2.2b.

Also, the number of step size L that represents the trajectory length needs to be selected carefully. The aim of the trajectory length is to avoid random walk behaviour when exploring the target distribution. Even though it is advisable to have a large number of step sizes to reduce random walk behaviour (Neal et al. (2011)), a too large L may take a long time to compute and hence waste time. Alternatively, a too small L may lead to random walk behaviour and thus slow mixing.

In practice, we can fix the number of steps sizes L to 10 per sample and then automatically tune the step size ϵ to reach an optimal acceptance probability. Theoretically, Beskos et al. (2013) and Neal et al. (2011) concluded that the optimal acceptance probability is around 0.65. Figure 2.3 shows the results of tuning ϵ automatically and fixed L to reach 65% acceptance probability.



(a) Small ϵ, hardly move and slowly exploring the target distribution.



(b) large choice of ϵ, high rejected proposal points (empty circles).

Figure 2.2: Plots of HMC sampling with difference step sizes ϵ. The dark circles represent the accepted points, and the empty circles represent the rejected points. The dark red areas represent the high probability of the posterior density.

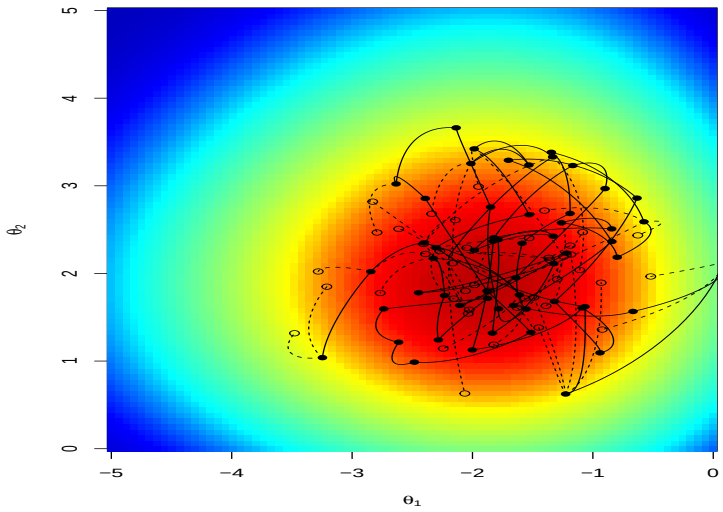


Figure 2.3: Plot of HMC sampling with the step size ϵ automatically to get 65% acceptance probability. The dark circles represent the accepted points, and the empty circles represent the rejected points. The dark red area represents the high probability of the posterior density.

2.5 MCMC Convergence Diagnostics

A major issue when running MCMC algorithms is determining when we should stop sampling and use this result samples to estimate the target distribution. Therefore, we can utilise a set of diagnostics to assess if the simulation process is behaving correctly. There are several methods to check for signs of non-convergence. This section will briefly explain some of these methods.

Trace Plot

Convergence can be evaluated through trace plots for every parameter. Trace plots involve drawing two or more chains of parameter values and plotting the values of each chain against the number of iterations. One possible way is to run two or multiple chains with different initial values. It is assumed that convergence is achieved when all the chains converge to the same distribution. As the early samples of the chain can be a poor representative of the target distribution, it is better to discard these initial samples. This is known as a burn-in. All the samples obtained after the burn-in are assumed to be drawn from the target distribution. However, it is crucial to implement some other convergence diagnostics to ensure the Markov chain converges to the posterior distribution.

Figure 2.4 displays the trace plots for three Markov chains run independently with different starting values. After a period of time (around 200 iterations), all the chains converge to similar distributions even though they started far from each other. Hence, we assume that these chains after 200 iterations represent a reasonable estimate for the target distribution (the true posterior), and we could remove the first 200 iterations (burn-in).

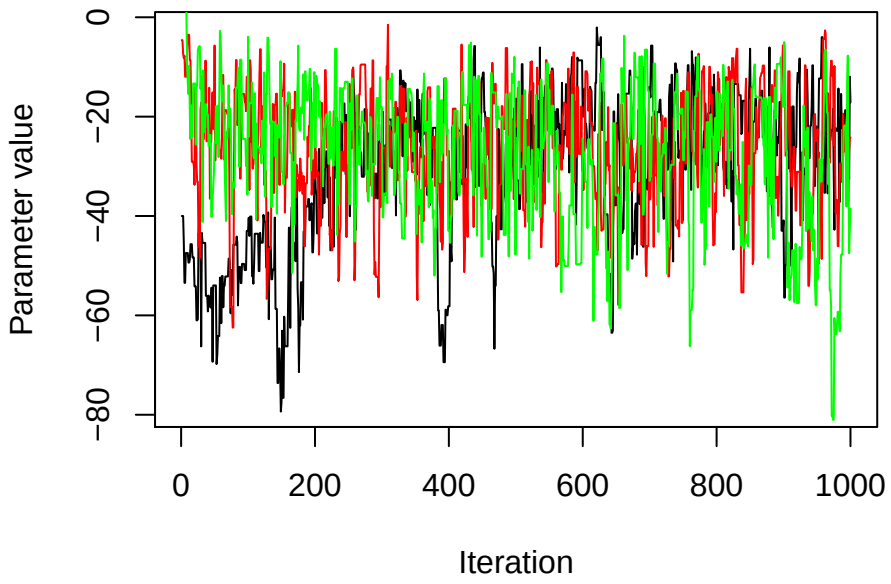


Figure 2.4: Comparison of three MCMC runs for the same parameter with different starting values.

Gelman-Rubin($\widehat{R}$ Statistics) Test

Another strategy for evaluating convergence is a test suggested by Gelman et al. (1992) to assess the convergence of multiple chains initialised from different values based on an analysis of the variance. The idea is we run M parallel chains, each of length T. We can label the individual draws as $\theta_{tm} = (t = 1, \ldots, T, M = 1, \ldots, m)$. Then we calculate the between-chain B and within-chain W variance for each parameter θ:

$$B = \frac{T}{M-1} \sum_{M=1}^{m} (\bar{\theta}_{.m} - \bar{\theta}_{..})^2 \tag{2.35}$$

and

$$W = \frac{1}{M}\sum_{M=1}^{m} s_m^2$$

where

$$\bar{\theta}_{.m} = \frac{1}{T}\sum_{t=1}^{T}\theta_{tM}$$

and

$$\bar{\theta}_{..} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_{.m}$$

$$s_t^2 = \frac{1}{T-1}\sum_{t=1}^{T}(\theta_{tM} - \bar{\theta}_{.M})^2$$

The estimated variance of the marginal posterior can be calculated as:

$$\widehat{var}(\theta|D) = \frac{T-1}{T}W + \frac{1}{T}B \tag{2.36}$$

The estimated potential reduction in the scale of θ, $\widehat{R}$ is defined as:

$$\widehat{R} = \sqrt{\frac{\widehat{var}(\theta|D)}{W}}$$

If all chains converge to the target distribution, the between chains variability should be small, and $\widehat{R}$ should be close to one. Therefore, as $T \to \infty$, the value of $\widehat{R}$ decreases to 1. If the value of $\widehat{R}$ is higher than 1, we could have evidence for non-convergence and increasing the number of simulations may help to reduce the variance $\widehat{var}(\theta|D)$. Gelman et al. (2013) suggested running the chain until achieving $\widehat{R}$ values close to 1.1.

Autocorrelation

Because of the offer dependent nature of the Markov chain, it is common that the autocorrelation exists between the posterior samples from the MCMC chain. This means that the samples are dependent on each other (Fox, 2010). To get an idea of how these samples correlated, we can plot the autocorrelation of our chains using autocorrelation function (ACF). This plot shows how the autocorrelation between samples changes as a function of their lag (k). High autocorrelation can lead to slower convergence, and it can be reduced by storing only every t^{th} iterate (after burn-in) from the sample and discarding all others. This is known as thinning. However, in practice, some researchers such as Link and Eaton (2012) suggested that unthinned Markov chains can produce more precise results, implying that thinning is not necessary. Therefore, in this thesis, all results by MCMC will be based

on summary statistics of posterior distributions resulting from unthinned Markov chains.

Effective Sample Size

In practice, drawing independent random samples from the posteriors distribution is desirable. However, a lot of MCMC chains are strongly autocorrelated, which increases uncertainty relative to an independent sample. The equivalent number of independent samples, which is known as the effective sample size (ESS), can be calculated as following:

$$\text{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho(k)},$$

where T is the number of samples and $\rho(k)$ is the correlation at lag k. If the samples are independent, ESS will be equal to the actual sample size (T). However, If the correlation $\rho(k)$ at lag k is very high and decreases slowly, ESS will be very small.

Thus, the efficiency of the MCMC can be measured by the number of effectively independent samples (ESS) generated per second. More methods are highlighted and described for convergence diagnostics in Cowles and Carlin (1996).

2.6 Monte Carlo Methods

Monte Carlo (MC) methods are a class of algorithms that aim to overcome the numerical issue of intractable target distributions in Bayesian inference. Monte Carlo approaches can be used to obtain numerical estimations of unknown parameters by drawing random samples from the target distributions (Metropolis and Ulam, 1949) (Robert et al., 2010).

The main idea of MC is to approximate a complicated distribution by sampling N independent and identically distributed (i.i.d.) random samples $x_1, \ldots, x_N$ from the target distribution $\pi(x)$, where the probability density $\pi(x)$ corresponds to the posterior density $\pi(\theta|D)$ in Bayesian inference. Monte Carlo method approximates $\pi(x)$ by the empirical measure as follows:

$$\hat{\pi}^N(x_{0:t}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x_{0:t}^i}(x_{0:t}), \tag{2.37}$$

where δ denotes the Dirac delta mass function, with a unit mass at x^i, and $\{x^i\}_{i=1}^N$ is the collection of i.i.d sample (particles) generated from $\pi(x)$. For further information see Robert et al. (2010).

However, with the basic Monte Carlo approach, it is not always possible to sample

efficiently from the target distribution, when $\pi(x)$ is complex or high-dimensional. Therefore, alternative sampling techniques have been developed to draw a sample from the target distribution within the Monte Carlo framework. One of these methods is importance sampling.

2.6.1 Importance Sampling

The main idea of the importance sampling method (IS) is to introduce an importance density $q(x)$, which is also known as the proposal distribution, such that

$$\pi(x) > 0 \Rightarrow q(x) > 0.$$

It then makes use of this density $q(x)$ to generate a sample x^i to approximate the target distribution $\pi(x)$. The difference between the target $\pi(x)$ and the proposal $q(x)$ densities can be measured by the importance weight $w(x)$. The resulting samples are used to approximate $\pi(x)$ as follows:

$$\hat{\pi}^N(x) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\pi(x)}{q(x)} \delta_{x^i}(x), \tag{2.38}$$

where $\frac{\pi(x)}{q(x)}$ is the importance weight $w(x)$, which is used to correct the discrepancy between the target density $\pi(x)$ and the proposal density $q(x)$ since the sample derived from $q(x)$ not from $\pi(x)$. The collection of points $\{x^i\}_{i=1}^N$ is a set of iid samples generated from the target. The Monte Carlo sample methods can be then written as:

$$\hat{\pi}^N(x) \approx \frac{1}{N} \sum_{i=1}^{N} w(x^i) \delta_{x^i}(x) \tag{2.39}$$

This estimator is known as the self-normalised importance sampling (SNIS), and the algorithm is presented in Algorithm 4.

Algorithm 4 SNIS Algorithm

1: Sample $x^i \sim q(x)$ where $i = 0, \cdots, N$
2: Compute the weight $w^{(i)} = \frac{\pi(x^i)}{q(x^i)}$
3: Compute the normalised weight $w(x^i) = \frac{w^{(i)}}{\sum_{j=1}^{N} w^{(j)}}$

In general, importance sampling can give a consistent and effective estimation, comparable to the classical Monte Carlo method (Murphy, 2012). This is because of

the idea that the samples only concentrate in the important region of the probability space. This implies that importance sampling needs fewer samples compared to sampling from the exact distribution. However, in some cases, the method can be ineffective because of the large difference between importance and target densities leading to higher variability of the weights.

Moreover, each time new data arrives, the importance weights have to be recomputed over the entire sequence in real-time data. This means that the computational complexity will increase with time (Smith, 2013).

In the case of a complex high dimensional target distribution $\pi(x)$, utilising importance sampling can be difficult because choosing importance density requires some knowledge about the target distribution that is unavailable. More explanations and details the about importance sampling method can be found in (Glynn and Iglehart, 1989), (Robert et al., 2010) and (Hastings, 1970). The other Monte Carlo methods, such as rejection sampling, are out of this thesis scope. For interest and more details, see (MacKay et al., 2003) and (Hammersley, 2013).

The following section introduces a method that addresses the issue of recomputing the importance weights in real-time data.

2.7 Sequential Monte Carlo Method (SMC)

Importance sampling can be modified by sampling from a sequence of intermediate distributions to compute the estimation of the target distribution without computing previous simulation tracks. This method is known as the Sequential Monte Carlo Method (SMC). The idea is to construct the importance density sequentially as described in (Del Moral et al., 2006) as follows:

$$q_t(x_{0:t}) = q_{t-1}(x_{0:t-1})q_t(x_t|x_{t-1}) = q_0(x_0)\prod_{k=1}^{t} q_k(x_k|x_{0:k-1}) \tag{2.40}$$

This means that firstly at initial time $t = 0$, x_0 can be sampled from the initial proposal, $x_0^i \sim q_0(x_0)$. Then, at time k, where $k = 1, \ldots, t$, the x_k^i can be obtained by sampling from the proposal such that $x_k^i \sim q_k(x_k|x_{0:k-1}^i)$. The associated unnormalised weights can be computed sequentially by:

$$w_t(x_{0:t}) = \frac{\pi_t(x_{0:t})}{q_t(x_{0:t})} = \frac{\pi_{t-1}(x_{0:t-1})}{q_{t-1}(x_{0:t-1})} \quad \frac{\pi_t(x_{0:t})}{\pi_{t-1}(x_{0:t-1})q_t(x_t|x_{0:t-1})} \tag{2.41}$$

This method is known as sequential importance sampling (SIS), which is a particular case of SMC (Doucet and Johansen, 2009). More details and descriptions of this method are given in the following section.

2.7.1 Sequential Importance Sampling (SIS)

Suppose that at time $t-1$, the target distribution π_{t-1} is approximated by using the weighted samples $\{x^i_{0:t-1}, w^i_{0:t-1}\}^N_{i=1}$ as:

$$\hat{\pi}^N_{t-1}(x_{0:t-1}) \approx \sum_{i=1}^{N} \frac{\pi(x^i_{0:t-1})}{q(x^i_{0:t-1})} \delta_{x^i_{0:t-1}}(x_{0:t-1}) \approx \sum_{i=1}^{N} w_t(x^i_{0:t-1}) \delta_{x^i_{0:t-1}}(x_{0:t-1}) \tag{2.42}$$

The next distribution π_t is obtained by propagating these samples (particles) by utilising the proposal distribution q_t to have a set of samples $x^i_{0:t}$. The associated unnormalised weight is evaluated as:

$$w_t(x^i_{0:t}) = w^i_{t-1} \frac{\pi_t(x^i_t)}{q_t(x^i_t)}$$

Hence, the target distribution π_t is estimated as the following:

$$\hat{\pi}^N_t(x_{0:t}) \approx \sum_{i=1}^{N} w_t(x^i_{0:t}) \delta_{x^i_{0:t}}(x_{0:t}) \tag{2.43}$$

The procedure of SIS is described in Algorithm 5.

Particle Degeneracy

The main advantage of SIS is that the particles are placed in important regions of the probability space with a high mass in the target distribution. If the proposal distribution is proportional to the target distribution, the particles will have a similar importance weight. Therefore, the quality of the estimation can be measured by the variance of the weighted particles. However, the major problem encountered by the SIS method is that when t increases, the difference between the importance density q_t and the target density π_t can increase too, which results in particle degeneracy. This means that after a few iterations, only a few particles have high importance weights and others have negligible weights (Doucet et al., 2000), (Cappé et al., 2007). As a result, the variance of the importance weights increases dramatically as time increases. Consequently, the algorithm fails to represent the target distributions

adequately.

Algorithm 5 SIS Algorithm

1: For $i = 1, \cdots, N$, Initialise sample

$$x_0^i \sim q_0(x_{0:t})$$

Assign initial weight:

$$w_0(x_0^i) = \frac{\pi(x_0^i)}{q(x_0^i)}$$

$$w_0^i = \frac{w_0(x_0^i)}{\sum_{j=1}^N w_0(x_0^j)}$$

2: At the next time $t = 1, \cdots, T$ and for $i = 1, \cdots, N$ propagate:

$$x_t^i \sim q(x_t^i | x_{t-1}^i)$$

3: Compute the importance weight:

$$w_t(x_{0:t}^i) = w_{t-1}^i \frac{\pi_t(x_t^i)}{q_t(x_t^i)}$$

4: Compute the normalised weight:

$$w_t^i = \frac{w_t(x_{0:t}^i)}{\sum_{j=1}^N w_t(x_{0:t}^i)}$$

The degeneracy of particles can be quantified by the effective sample size (ESS) through iterations. This measurement is different from 2.5, and can be defined as:

$$EES = \frac{1}{\sum (w_t^i)^2}$$

A small effective sample size indicates the high degeneracy of the algorithm. If all weights are equal, the ESS equals N. On the other hand, if all mass is concentrated in one particle, the ESS equals 1. ESS can be interpreted as minimum number of particles that is needed to present the target distributions.

To avoid this degeneracy, one needs to introduce an additional selection step to minimise the variance between weights. The key idea is to eliminate the particles that have lower weights and multiply the particles with high weight (Gordon et al., 1993). This method is known as sequential importance resampling (SIR).

2.7.2 Sequential Importance Resampling (SIR)

Sequential importance resampling (SIR) was introduced by Gordon et al. (1993) to address the issue of particle degeneracy described in the previous section (2.7.1). The algorithm has three main steps: resampling, propagation and weighting. This algorithm follows the same procedure as the SIS algorithm (2.7.1), but an extra resample step is added. In the resampling step, particles that have higher weights are repeated while particles with small weights are eliminated. Therefore, only high-weight particles will be used. The idea is that the particles with large weights are more expected to represent the target distribution than particles with small wights (Li et al., 2015). As a result, in the next step, new particles will be generated from the region of the large weight. Hence, this will improve the exploration of the parameter space after resampling .

Resampling Method

There are different schemes for resampling particles, such as multinomial sampling, residual resampling and stratified sampling. More details about these strategies are provided by Carpenter et al. (1999), Kitagawa and Sato (2001), Douc and Cappé (2005) and Hol et al. (2006). This thesis will focus on the residual resampling method.

One of the disadvantages of the resampling method, besides being computationally expensive, is that it increases the estimator's variance (Speekenbrink, 2016) by using some method such as multinomial sampling. To address this issue, residual resampling is considered an alternative method with a smaller variance. Residual resampling (Liu and Chen, 1998), which is also called remainder resampling, is based on using a set restriction. The method consists of two steps. In the first step, each particle with a weight greater than $1/N$ is replicated. In the second step, the remaining weights (residuals) will be used for random sampling. The process of this resampling scheme can be summarised as follows:

1. Calculate $R^i = \lfloor Nw_i \rfloor$ copies of particles x_i, For $i = 1, \cdots, N$. Where $\lfloor \rfloor$ denotes the integer part.

2. Allocate R^i to the new distribution.

3. Resample $K = N - \sum R^i$ from x_i, with the probability of selecting the x_i that is proportional to $w_i = Nw_i - R_i$. This step can be done using other resampling schemes; frequently, multinomial sampling is used. In short, the integer value of

the particles is allocated first and copied, leaving the remainder for multinomial resampling. The greater the weights of the particles, the more copies of particles.

Although the resampling step can help to increase the number of active particles, it also can generate serious issues. One of these issues is that performing resampling at each stage can result in decreasing the diversity within the particle set, which is known as sample impoverishment (Carpenter et al., 1999). Because of this, it is not desirable to resample at each time step, which means it should be performed only when necessary. The effective sample (ESS) can be used to determine the necessity of the resampling step. One can define a specific threshold on the ESS, which usually is set to $N/2$ (Doucet and Johansen, 2009), and when the ESS drops below the predefined threshold, resampling should be carried out. The procedure of SIR is described in Algorithm 6.

Algorithm 6 SIR Algorithm

1: for $i = 1, \cdots, N$, Initialise sample

$$x_0^i \sim q_0(x_0)$$

Assign initial weight:

$$w_0^i = \frac{\pi(x_0^i)}{q(x_0^i)}$$

$$w^i = \frac{w_1^i}{\sum_{j=1}^N w_1^j}$$

2: At the next time $t = 1, \cdots, T$ and for $i = 1, \cdots, N$
propagate:

$$x_t^i \sim q(x_t^i | x_{t-1})$$

3: Compute the importance weight:

$$w(x_t, x_{0:t-1}) = \frac{\pi_t(x_{0:t})}{\pi_{t-1}(x_{0:t-1}) q_t(x_t | x_{0:t-1})} w_{t-1}$$

4: Compute the normalised weight:

$$w_t^i = w(x_t^i, x_{0:t-1}^i), \quad w_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}$$

5: Compute the ESS

$$EES = \frac{1}{\sum (w_t^i)^2}$$

6: If ESS $< N/2$ resample

Moreover, replacing high weights with multiple replicas of a unique particle may result in a high correlation between particles (Chopin, 2002). Gilks and Berzuini (2001) developed an algorithm by adding a rejuvenation step which consists of a resample step and a move step. The idea of the rejuvenation step is that the resampled particles are moved from time t to $t+1$ according to a Markov chain transition kernel q with the invariant distribution. The transition kernel q could be, for example, a Gibbs sampler or Metropolis-Hastings. This method could be an alternative method to limit the degeneracy of the particles (see (Doucet et al., 2001) and (Fearnhead, 2002)). The efficiency of the rejuvenation step depends on the choice of this kernel density; for discussion of the choice of the kernel, see Gilks and Berzuini (2001). Elfring et al. (2021) explained in greater detail the challenge of applying SMC methods with several examples. Further details about the applications of the SMC methods will be discussed in Chapter 5.

2.8 Approximation Method

MCMC and SMC have the advantage that they will eventually converge to the target distribution and hence find exact samples from the posterior distributions (Robert et al., 1998). However, these methods may become very slow in a large problem. Therefore, simulation methods are suitable for smaller data sets and when more precise inference samples are required. An alternative way to address the computational issue in inference problems is by using approximation methods such as variational inference (Fox and Roberts, 2012) or Laplace approximation. In order to find the best approximation for the target distributions, we follow an optimisation process. These methods are relatively simple to compute and can derive helpful information about the models' parameters (Gelman et al., 2013). This thesis will consider the use of the Laplace approximation methods.

2.8.1 Laplace Approximation Method

The Laplace approximation (LA) (Tierney and Kadane, 1986) is an analytical approximation method that aims to find a Gaussian approximation to a continuous target distribution. The idea behind the Laplace approximation is using the second-order Taylor expansion of the log-posterior of interest $p(\theta|\mathcal{D})$ around its maximum $\hat{\theta}$, which corresponds to a Gaussian approximation at the mode. Formally, we have:

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\hat{\theta}, \mathbf{H}^{-1}),$$

where

$$\hat{\theta} = \arg\max_{\theta} p(\theta \mid \mathcal{D}) = \arg\max_{\theta} p(\mathcal{D}, \theta) = \arg\max_{\theta} p(\mathcal{D} \mid \theta)p(\theta) = \arg\max_{\theta} [\log p(\mathcal{D} \mid \theta) + \log p(\theta)].$$

$\mathbf{H}$ is the Hessian matrix of the negative log-posterior at the mode ($\hat{\theta}$);

$$\mathbf{H} = -\nabla^2 \log p(\theta \mid \mathcal{D})\big|_{\theta=\hat{\theta}} = -\nabla^2 \log p(\mathcal{D}, \theta)\big|_{\theta=\hat{\theta}} = -\nabla^2 [\log p(\mathcal{D} \mid \theta) + \log p(\theta)]\big|_{\theta=\hat{\theta}}.$$

Now, we can approximate $p(\theta|\mathcal{D})$ using its 2nd order Taylor expansion.

Suppose $\log p(\mathcal{D}, \theta) = f(\theta)$, one can approximate $f(\theta)$ using its 2nd Taylor expansion as following:

$$f(\theta) \approx f(\theta_0) + (\theta - \theta_0)^\top \nabla f(\theta_0) + \frac{1}{2} (\theta - \theta_0)^\top \nabla^2 f(\theta_0) (\theta - \theta_0),$$

where θ_0 is an arbitrary point. If we chose $\theta_0 = \hat{\theta}$, we get

$$\log p(\mathcal{D}, \theta) \approx \log p(\mathcal{D}, \hat{\theta}) + \frac{1}{2} \left(\theta - \hat{\theta}\right)^\top \nabla^2 \log p(\mathcal{D}, \hat{\theta}) \left(\theta - \hat{\theta}\right), \tag{2.44}$$

where we know that $\nabla f(\hat{\theta}) = \nabla \log p(\mathcal{D}, \hat{\theta})) = 0$, since $\hat{\theta}$ is at a maximum.

Now, we can use Bayes' rule to write the unnormalised posterior distribution as following:

$$p(\theta \mid \mathcal{D}) \approx p(\mathcal{D} \mid \theta)p(\theta) = p(\mathcal{D}, \theta) = e^{\log p(\mathcal{D}, \theta)}$$

Plugging in this formula to the 2nd order Taylor approximation for $\log p(\mathcal{D}, \theta)$, using equation (2.44), we get

$$p(\theta \mid \mathcal{D}) \approx e^{\log p(\mathcal{D}, \theta)} \approx e^{\log p(\mathcal{D}, \theta) \approx \log p(\mathcal{D}, \hat{\theta}) + \frac{1}{2} \left(\theta - \hat{\theta}\right)^\top \nabla^2 \log p(\mathcal{D}, \hat{\theta}) \left(\theta - \hat{\theta}\right)}$$

simplify this formula, we get

$$p(\theta \mid \mathcal{D}) \approx e^{-\frac{1}{2} \left(\theta - \hat{\theta}\right)^\top (-\nabla^2 \log p(\mathcal{D}, \hat{\theta})) \left(\theta - \hat{\theta}\right)}$$

Hence, Laplace approximation of the posterior distribution $p(\theta \mid \mathcal{D})$ is a Gaussian;

$$p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\hat{\theta}, \mathbf{H}^{-1}),$$

where $\mathbf{H} = -\nabla^2 \log p(\mathcal{D}, \hat{\theta})$.

The Laplace Approximation is considered to be very fast. The reason is that LA only has to find the posterior mode, and it does not have to explore the whole space of posterior distribution as in MCMC or SMC. Laplace Approximation methods can be more efficient for uni-modal target distributions or when it is possible to apply separately to each mode for multimodal distributions (Gelman et al., 2013). The performance of LA in approximating the posterior distribution is illustrated in the following example.

2.8.2 Example: Binomial Data with a Beta Prior

In this simple example, assume that the likelihood is drawn from a $binomial(n, \theta)$ distribution where n is known, and θ is the (unknown) parameter of interest. The model is, therefore

$$x \mid \theta \sim binomial(n, \theta)$$

$$\theta \sim Beta(\alpha, \beta)$$

Since the beta distribution is a conjugate prior to the binomial likelihood (see section 2.2), the posterior will also be a beta distribution. Therefore, the posterior distribution has the following closed-form:

$$\theta \mid x \sim Beta(\alpha + x, \beta + n - x)$$

Figure 2.5 shows the posterior for the rate parameter θ of a Binomial distribution given $n = 20$ and $x = 10$. The black line represents the true posterior, and the Laplace approximation is the red line. In this case, the Laplace approximation works pretty well around the mode and also when moving farther away from the mode. In Figure 2.6, we can see the posterior of the same model but with less data, $n = 6$ and $x = 4$. In this case, the approximation is reasonable around the neighbourhood of the mode. However, when we move away from the mode, the tail of the true posterior is heavier on the left and slanted to the right, and the symmetric

normal distribution cannot match it. This kind of problem generally occurs when parameters have bounds. This was the case with θ that bounded between $[0, 1]$. There are two ways to address this issue. The first way is that one would expect the approximation to improve by collecting more data. As shown in Figure 2.5, the more data, the better the Laplace approximation, where the posterior becomes asymptotically normally distributed.

However, in practice, collecting more data for better approximations is not always possible. The better way is to re-parameterise the bounded parameters using logarithms so that they extend the real line $[-\infty, \infty]$. Figure 2.7 displays the same posterior, and the only difference is that the approximation is made on the logarithm scale. In this case, the parameter θ from the binomial model re-parameterised using the logit transform $\log\left(\frac{\theta}{1-\theta}\right)$. Thus, θ extends from $[0, 1]$ to $[-\infty, \infty]$. An overview of variable transformation and Laplace approximation can be found in Hobbhahn and Hennig (2021).

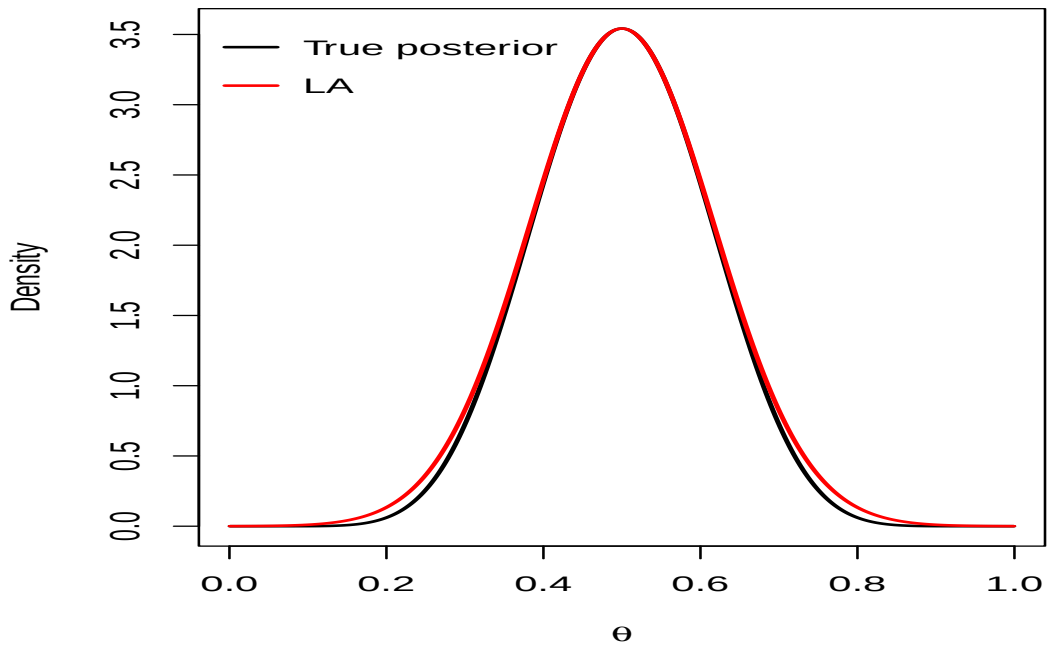


Figure 2.5: Laplace Approximation of Posterior for Binomial Distribution Given $n = 20$, $x = 10$.

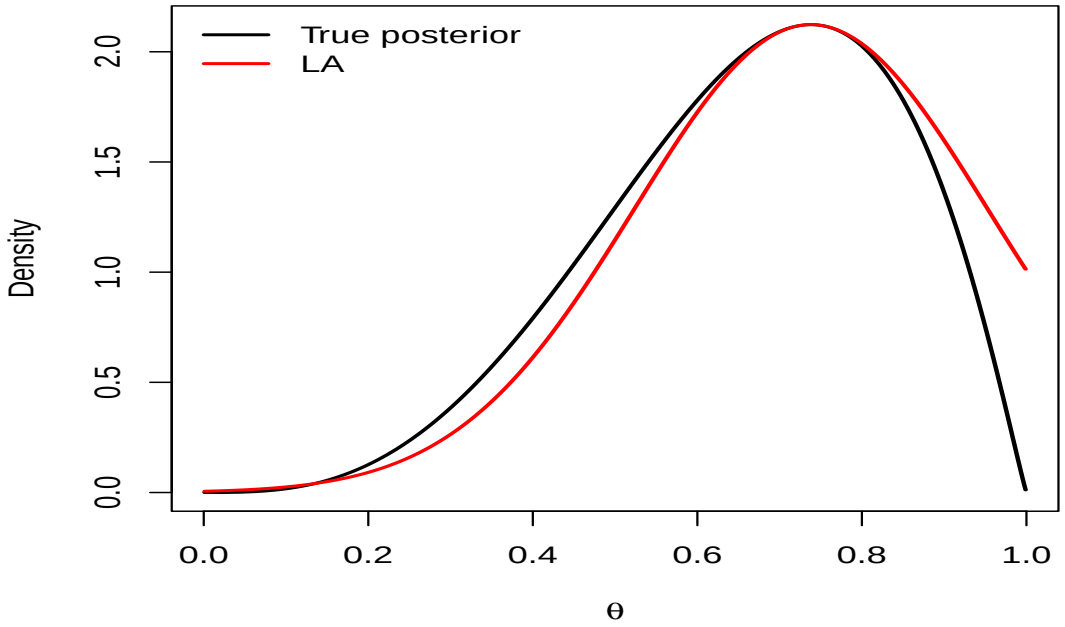


Figure 2.6: Laplace Approximation of Posterior for Binomial Distribution Given $n = 6$, $x = 4$.

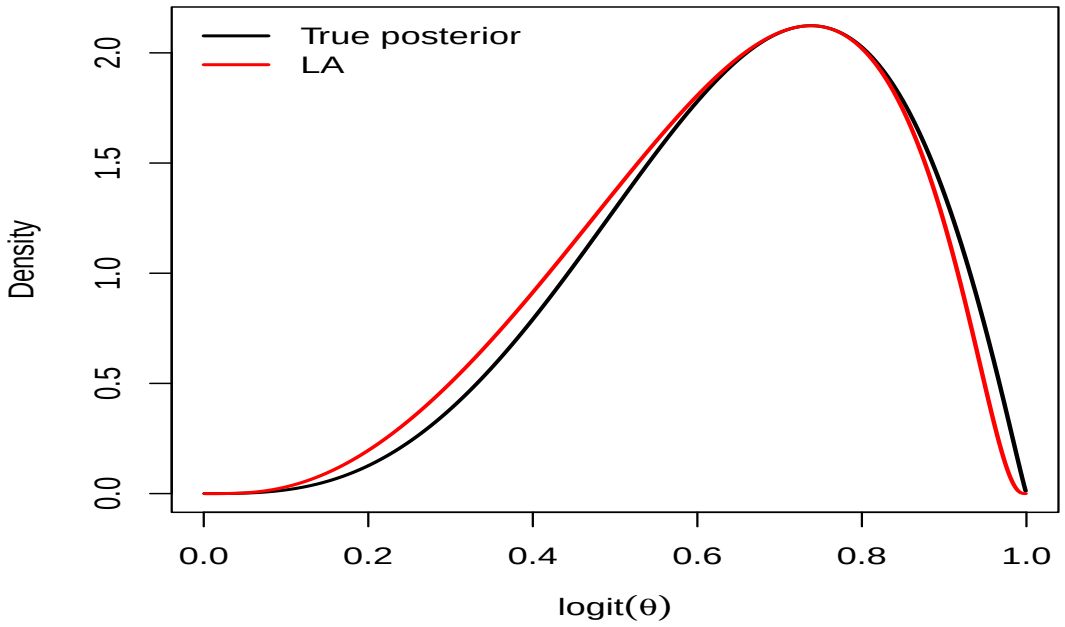


Figure 2.7: Laplace Approximation of Posterior for Binomial Distribution Given $n = 6$, $x = 4$, and using the logit transform.

For further reading about the Laplace method and its applications in Bayesian inference, see Kass et al. (1991). Extensive details about the algorithm setting and the optimisation methods that will be used under the Laplace method will be discussed in Chapter 6.

2.9 Summary of the Chapter

This chapter provided an overview of the statistical methods of interest to this thesis. First, a general introduction to Bayesian statistics was given. The prior distributions greatly affect Bayesian inference; hence, different types of priors are discussed.

In addition, Bayesian inference with Markov chain Monte Carlo was explained in detail. In particular, three different algorithms were presented; Metropolis-Hastings, Gibbs Sampler and Hamiltonian Monte Carlo. Information was provided regarding the efficiency of each method, setting and tuning each technique.

Besides the MCMC methods, the idea of the Monte Carlo method was introduced. The primary MC method, importance sampling, was first explained. We found this method can be inefficient for a dynamic system; hence the idea of sequential Monte Carlo was introduced. Two important methods of SMC were explained in detail; sequential importance sampling and sequential importance resampling. The concept of particle degeneracy and the resampling techniques were also mentioned.

The idea of approximation inference was introduced to address the expensive computational cost that may result from using MCMC or SMC methods. The concept of the Laplace approximation method was first explained, and then the performance of the proposed method in approximating the posterior distributions was clarified using a toy example.

Further explanation about the application of these methods in item response theory models and related works will be provided later. The following chapter will present a literature review of the unidimensional item response theory model (UIRT).

Chapter 3

Item Response Theory

Item response theory modelling improvement has a long history and wide literature. This chapter provides a brief overview of some common IRT models and their assumptions.

Section 3.1 will present an overview of unidimensional item response theory (UIRT). Based on the number of item parameters, this chapter will explain three types of this model; the one-parameter logistic model (1PL), two-parameter logistic model (2PL) and three-parameter logistic model (3PL). Finally, section 3.3 will discuss an identifiability issue for item response theory model and will give some ideas of how it could be addressed.

3.1 Unidimensional IRT Models

Item response theory (IRT) models demonstrate the relationship between the ability or attitude (denoted θ) and an item response (e.g. questions). These models can be categorized based on different factors such as the dimensionality of the ability, type of questions or the number of item parameters. If all items measure one common ability, this will result in a so-called unidimensional item response theory model (UIRT).

The item response might have two categories (dichotomous), like yes or no, right or wrong, agree or disagree. Also, it may have more than two categories such as Likert scale on a survey. In a large-scale quizzes, multiple choice format is most commonly used for IRT, where the answer is either correct (1) or incorrect (0). Often also a single ability variable θ(unidimensional) is assumed. For dichotomous (UIRT), there are three common types of models which are named according to the number of item parameters, and they will be described in the following sections.

3.1.1 One-Parameter Logistic (1PL) Model

The first and most straightforward item response theory model is the one-parameter logistic (1PL) model. This model is also known as the Rasch model and was introduced first by Rasch (1960). The model contains one item parameter which is the difficulty parameter.

Each IRT model has a unique item characteristic curve (ICC). This ICC shows how changing the ability variable (θ) results in changing the probability of a correct item response. Different item response models can be obtained by writing the (ICC) in various mathematical forms. The ICC of 1PL can be written as:

$$p(X_{ij} = 1) = \frac{exp(\theta_i - b_j)}{(1 + exp(\theta_i - b_j))},\ \theta_i\ and\ b_j \in \mathbb{R}, \tag{3.1}$$

where X_{ij} is the result for the j^{th} item by examinee i, and 1 indicates a correct response. The ability parameter (latent trait) in this formula is represented by θ_i, and the difficulty parameter is b_j. The subscript $j = (1, 2, ..., m)$ represents the items where m is the total number of items. The subscript $i = (1, 2, ..., n)$ represents the examinees where n is the total number of examinees.

In this formula, the difficulty parameter measures the difficulty of getting the correct answers. High (low) values of b_j means hard (easy) questions. In other words, given the same ability level, for an item to be easier than another, the probability of correct answer should be higher. Also, high (low) values of θ_i mean high (low) levels of examinee skill. The theoretical range of a person ability θ_i as well as item difficulty b_j is from $-\infty$ to ∞ . However, in general, $b's$ tend to range between -2 to to 2, so as for questions not to be too easy or too hard (DeMars, 2010). In practice, the b_j values are estimated from data.

As an example of 1PL, according to the model in equation (3.1), the probability that randomly selected person with ability equal to 0.5 gets item with difficulty equal to 1 right would be ;

$$p(X_{ij} = 1) = \frac{exp(0.5 - 1)}{(1 + exp(0.5 - 1))} = 0.3775407$$

This means this person has a probability of about 0.38 to answer this question correctly.

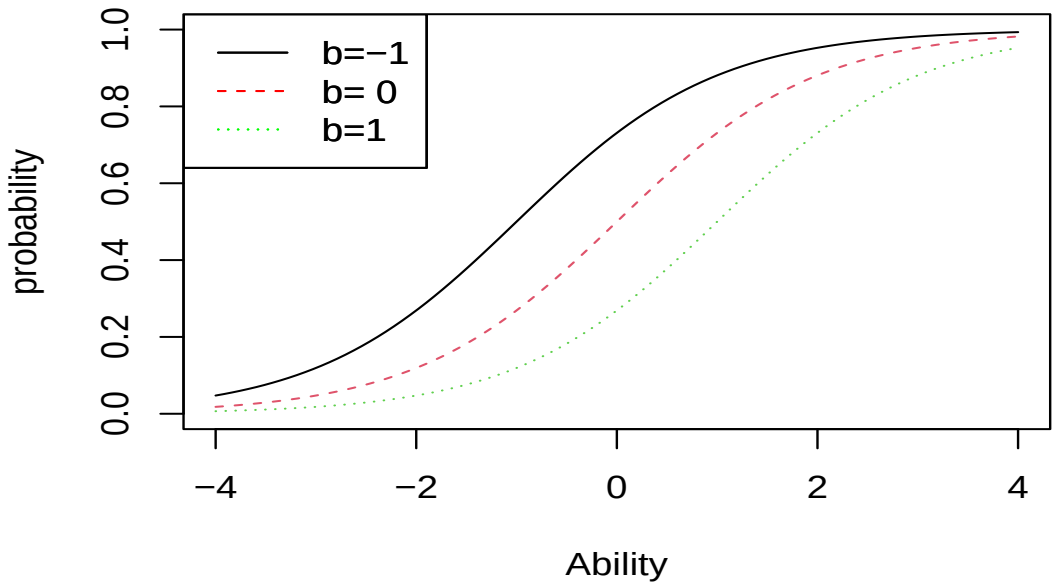


Figure 3.1: ICCs for the one-parameter model corssponding to three item difficulty levels.

Figure 3.1 shows three ICCs with different values of item difficulty. It can be seen that the ICC increases from the left to the right when the ability increases. The difficulty parameters are -1, 0 and 1, respectively. Item 3 with $b = 1$ is more difficult than item 1 ($b = -1$) and item 2 ($b = 0$); for any given value of θ, the probability of answering item 1 correctly is lower than answering item 2 or item 3 correctly.

3.1.2 Two-Parameter Logistic (2PL) Model

The main limitation of 1PL model (Rasch model) is that items only vary in terms of difficulty where the discrimination between the items is assumed to be fixed as equal. As a result, all item discriminate between examinees with different levels of ability in the same way. Thus, the model can be extended to set a different discrimination parameter a_j for each item; this model is called the two-parameter model (2PL).

The discrimination parameter a_j measures the slope of the curve of ICC, which tells how steeply the probability of correct answer changes at the steepest point; where the probability of correct answer changes rapidly when ability increases.

In other words, the discrimination parameter can tell us how well this question discriminates between students with high and low ability levels.

The probability of correct answer for the 2PL model or ICC is written as:

$$p(X_{ij} = 1) = \frac{exp[a_j(\theta_i - b_j)]}{(1 + exp[a_j(\theta_i - b_j)])};\ \theta_i\ ,\ b_j\ and\ a_j \in \mathbb{R} \tag{3.2}$$

Theoretically, the discrimination parameters a_j can range between $-\infty$ to ∞. Items with positive discrimination values suggest that lower ability students have a low probability of answering an item correctly, and higher ability students have a high chance of getting the item right. Items with negative discrimination parameters suggest that examinees with high abilities are less likely to answer the items correctly. Hence, these items should be removed or edited. Hence, this parameter can measure the differential capability of items. A higher value means that the item better discriminates between examinees with different ability levels. In practice, the discrimination values of a good test item can take range between 0.5 to 2 (DeMars, 2010).

Figure 3.2 shows three ICCs of the two-parameter model with the same difficulty parameter ($b_j = 0.5$) and different discrimination parameters a_j. The discrimination parameter values are 0.5, 1 and 2, respectively. It is noticeable that as the a_j values increase the slope becomes steeper around 0. The ICC with $a_j = 2$ (highest value) has the steepest slope. The higher (lower) values of a_j means better (less) discrimination between high and low ability levels. For the 1PL model, $a_j = 1$ for all items.

3.1.3 Three-Parameter Logistic (3PL) Model

In reality, it is reasonable to assume that examinees with low ability skills might choose the correct answer in a difficult item by luck or guess. Therefore, the two-parameter can be extended to measure the probability of answering questions correctly in spite of low ability by adding a third item parameter. This parameter is called the item guessing parameter c_j. In the 1PL and 2PL models, we assume that the guessing parameters are zero for all items.

The probability of the correct answer for the 3PL model (ICC) is described as following:

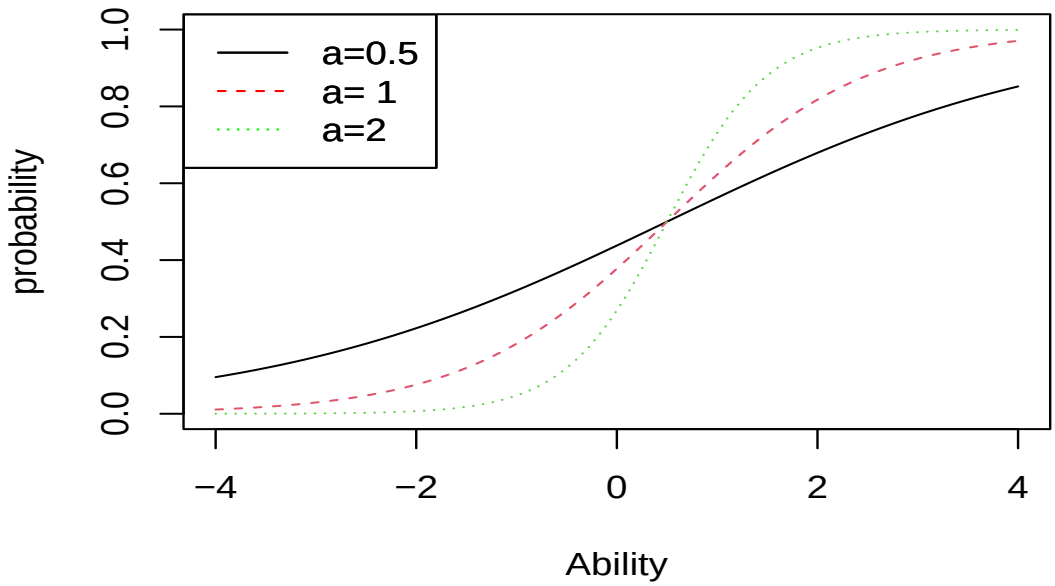


Figure 3.2: ICCs for the two-parameter model corresponding to three discrimination level (with an equal difficulty level).

$$p(X_{ij}=1) = c_j + (1-c_j)\,\frac{exp[a_j(\theta_i - b_j)]}{(1+exp[a_j(\theta_i - b_j)])};\ \theta_i\ ,\ b_j\ and\ a_j\ \in \mathbb{R}\ and\ c_j > 0. \quad (3.3)$$

In the case of the multiple-choices test, the probability of answering multiple choice question with k choices is $\frac{1}{k}$ even for examinees who has a low ability. For example, If we have four multiple choices item ($k = 4$), c_j would be approximately 0.25, which is the chance that an examinee with an extremely low ability could randomly answer this item correctly. Theoretically, c_j can range between 0 and 1.

Figure 3.3 displays three ICCs of 3PL plotted with the same difficulty ($b = 0$) and discrimination level ($a = 1$) , but with three different guessing parameters, low ($c = 0$), medium ($c = 0.2$) and high(0.5). The guessing parameters in the ICCs are the height of the lower asymptote, so is referred to as the lower asymptote parameter. It can be seen that for the low ability such as -2 the probability of getting the correct answer is 0.5 for item 3 ($c = 0.5$). However, as the ability level increases , the effect of guessing parameter on the probability of getting the correct answer is very small

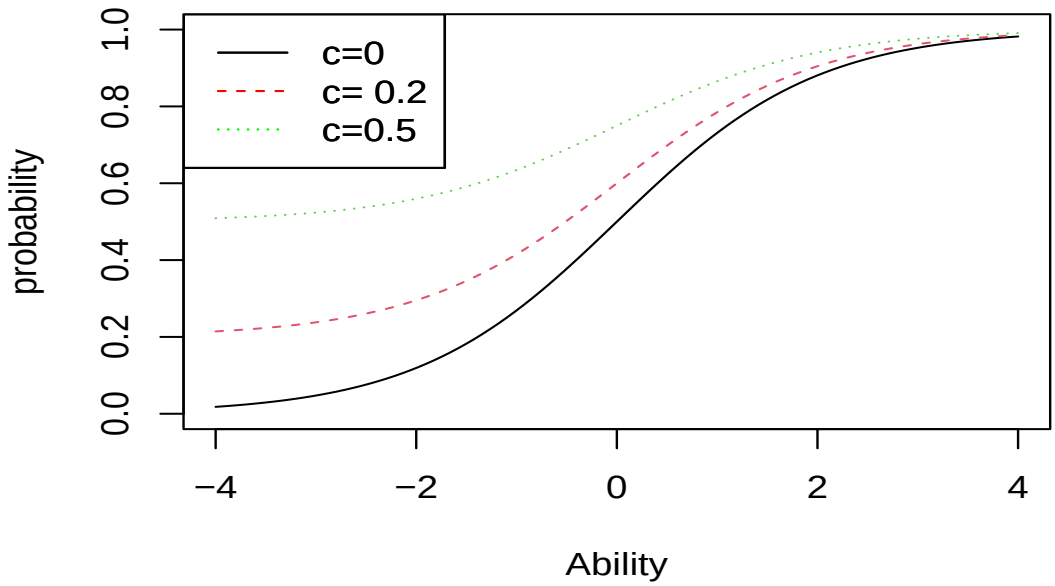


Figure 3.3: ICCs for the two-parameter model corssponding to three guessing levels and an equal difficulty and discrimination level.

since the ICCs are almost identical at the end of the ability scale.

3.1.4 Model Assumptions

The unidimensional item response theory models consider that any change in the ability variable θ will lead to a change in the probability of selecting the correct response $p(\theta)$. In other words, as the ability level increases the probability $p(\theta)$ will increase too which is usually called the monotonicity assumption. This assumption can be clearly described by the item characteristic curve (ICC). This ICC shows how the changing in the latent or ability variable result in changing the probability of an item response. Different item response models can be obtained by writing the (ICC)s in various mathematical forms, as introduced before.

Besides this assumption, there are two main assumptions that are (1) *unidimensionality* of the latent variable and (2) *local independence*. A unidimensional IRT model is shown in Figure 3.4 as Model A. In this model, each item has a single common ability θ_1.

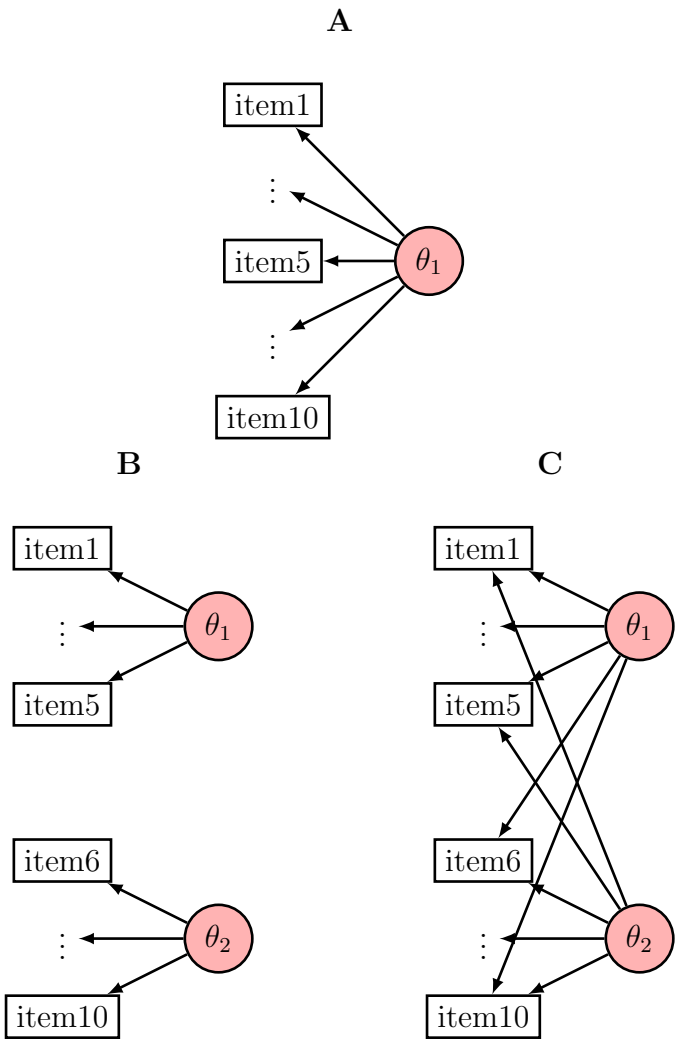


Figure 3.4: Alternative models: A)- unidimensional model, B) Between-Item (dimensionality), C)- Within-Item (dimensionality) structure

In the other hand, in C (within-item dimensionality), each item can be associated with two or more abilities measured by the test. In between-item (model B), item 1 to item 5 are associated with the first ability, θ_1, while item from 6 to 10 are associated with the second ability, θ_2. Which means θ_1 and θ_2 do not share any common items. Therefore, the test becomes a multi-unidimensional structure where each ability is defined by a set of unidimensional items while the overall structure becomes multidimensional. In the within-item model C, the items are associated with both θ_1 and θ_2, and thus the test has a complex structure. This thesis focuses on model A.

The second assumption, *local independence*, indicates that if the first assumption is held, then examinees responses to one item will be independent of their responses to other items conditional on the latent ability. In other words, locally independent

items are assumed to be uncorrelated after conditioning on θ_i (DeMars, 2010). Several diagnostic methods have been discussed in the literature to test the violations of local independence assumption in different simulation studies (e.g. Yen (1984), Chen and Thissen (1997), Kim et al. (2011) and Edwards et al. (2018) for testing the local independence assumption in the 2PL and 3PL models). Liu and Maydeu-Olivares (2013) described 6 different statistics and assessed the performance of these statistics in detecting local independence in 2PL IRT models for binary data under various simulated conditions. Debelak and Koller (2020) proposed and evaluated two new quasi-exact non-parametric methods for testing the local independence assumption for the Rasch model.

3.2 Parameter Estimation in IRT Models

In the literature, there have been extensive estimation techniques in IRT models in both Bayesian and frequentist approaches. For example, Joint Maximum Likelihood Estimation (JML), Marginal Maximum Likelihood Estimation (MML) have wide applications and different techniques for estimating both item and ability parameters. For more details, see for example, Embretson et al. (2000), Baker and Kim (2004) and De Ayala (2013).

Many researchers such as Swaminathan and Gifford (1982), Baker (1998) and Baker and Kim (2004) suggested that Bayesian estimations methods can be useful for complex IRT models and for small data sets. Using Bayes' theorem allows us to combine the previous knowledge belief (prior distribution) with data (the likelihood function) to obtain the probability distribution of possible parameter values (the posterior distribution). With the help of modern computer techniques, Bayesian estimation methods have been widely used for IRT models via Markov chain Monte Carlo (MCMC). For example, Patz and Junker (1999b) developed a Metropolis-Hastings algorithm and demonstrated the performance of this method in the two-parameter logistic (2PL) model: Martin and Quinn (2002) and Wang et al. (2013) introduced and discussed the idea of the MCMC methods for dynamic IRT models. See Albert (2015) for a brief overview of the developments of Bayesian inference in the IRT model. Weng et al. (2018) discussed the idea of estimating parameters in real-time and developing an efficient online algorithm for online product ratings and IRT models.

Focusing on IRT models in the context of educational testing, the challenge arises when the data arrives in real-time continuously and the parameters need to be estimated online. For example, if we want to update the estimates in a model quiz

where students can answer the questions at different stages. In this case, when there is new data arriving from individuals or questions through time, dynamic structures of student abilities and questions difficulties need to be included in the model, to accommodate changes in ability and difficulty. In this case, we cannot see the results of all individuals at the same time. Therefore, in order to use frequentist approaches, the person who actually took the test has to wait for other colleagues to take the test to get feedback on their own ability. Contrary to the frequentist statistics, using the Bayesian methods do not require many students to take the test, because using a prior distribution can direct the result. Assuming prior knowledge is available that allows formulation of the Bayesian model. According to this setting, the inference of unknown parameters is based on the posterior distribution obtained from Bayes' theorem. Hence, if the data arrive sequentially in real-time, and one is interested in making inference about an unknown parameter online, it is important to update the posterior distribution as new data arrive. This thesis aims to investigate some possible methods that can be used for real-time inference, taking into account the accuracy of the estimation results and the velocity of the methods.

3.3 IRT Model Identifiability

IRT models often suffer from model identification problems. This section will illustrate the issue of identifiability in IRT. To illustrate non-identifiability in IRT, the two-parameter logistic model will be given as an example. Also, some ideas for addressing this problem will be given in this section.

Non-identifiability of a model means that more than one set of parameter values can lead to the same likelihood. For the 2PL model, it is enough to show that different sets of parameters can lead to the same likelihood function (San Martín et al., 2013).

The likelihood function for IRT can be written as:

$$L(\boldsymbol{x}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{j=1}^{m} \pi_{ij}^{x_{ij}} (1 - \pi_{ij})^{1 - x_{ij}},$$

where $logit(\pi_{ij}) = a_j(\theta_i - b_j)$.

Applying the following linear transformation for any constant value δ

$$\theta_i' = \theta_i * \delta$$

and

$$b'_j = b_j * \delta$$

and

$$a'_j = \frac{a_j}{\delta}$$

where $b \neq b'$, $a \neq a'$ and $\theta \neq \theta'$.

With some simple algebra, one can see that;

$$L(\boldsymbol{x}, \boldsymbol{\pi}) = L(\boldsymbol{x}, \boldsymbol{\pi}')$$

$$\pi'_{ij} = a'_j(\theta'_i - b'_j) = \frac{a_j}{\delta}(\theta_i\delta - b_j\delta) = a_j(\theta_i - b_j) = \pi_{ij}$$

This means different parameter values induce the same probability distribution; $\pi'_{ij} = \pi_{ij}$. Similarly, the parameters for both 1PL and 3PL are not identifiable, because one can add the same constant to all θ_i and all b_j and obtain the same probability distribution.

In 1PL (Rasch model), the only concern is the difference between the ability parameter θ and the difficulty parameter b. If θ =2 and b=1, then the difference $(\theta - b)$ is 1.0 and the probability of answering the item correctly $p(X_{ij} = 1)$ is 0.7311. However, if we add the same constant (δ) to θ and b, the difference $(\theta + \delta - b - \delta)$ will still equal 1.0, and hence, the Rasch model would result in exactly the same probability of answering item correctly. For example, if δ =5, then $\theta' = \theta + \delta = 7$ and $b' = b + \delta = 6$, so the the difference $(\theta - b)$ is one and $p(X_{ij} = 1)$ is 0.7311, which exactly the same probability as before. For more information of how to show non-identifiability in 3PL see Maris and Bechger (2009).

From a Bayesian perspective, several authors in the literature review (e.g., Chaloner and Verdinelli (1995), Bernardo and Smith (2009)) have pointed out that there is no identifiability issue when proper prior (2.2) distributions are assumed for all parameters as this case will lead to proper posterior distributions. Therefore, every parameter can be well estimated; see Shariati et al. (2009) for more details about this point of view. However, taking the view that non-identifiability may cause concerns for the Bayesian inference, these concerns can appear clearly in practice. Practically, there might be strong correlations between estimated parameters in the posterior distribution. Hence, the appearance of strong correlations results in poor mixing of the Markov chain. In this way, the chain may not be able to converge to

the target distribution within a reasonable time.

In order to resolve the non-identifiability issue in IRT, we can impose various constraints upon the estimated parameters. For instance, setting θ_1=1 (that means using first person as a baseline), or we can set b_1=0 (using first item as a comparison point). We can also constrain the all θ_i to sum to 0, or constrain all b_j to sum to 0.

In Bayesian inference, a common way is assuming that the ability parameters are standard normally distributed by setting μ_θ= 0 and σ^2_θ=1. Another possible solution is rescaling the sample difficulty values in each MCMC iteration. For instance, restricting their sum to be zero: Define $\widehat{\mu_{b^t}} = \sum_{j=1} b_j^{(t)}/m$ and transform the sample as $b_j^{*(t)}$= $b_j^{(t)}$-$\widehat{\mu_{b^t}}$. Then, one can sample θ parameters using the values of rescaled sample of item difficulty (Fox, 2010).

The alternative approach is to constrain the $a's$ to have positive signs, as explained by Gelman and Hill (2006). This thesis will use this approach since it is desirable in educational testing for items j to have high values of a_j to discriminate better between high and low abilities. . If a_j is not fixed, changing the sign of a_j can be compensated by changing the sign of θ_i with no change in the probability of getting the correct answer.

In summary, this chapter reviewed the basic concept of the unidimensional item response theory models. Three standard models; 1PL, 2PL and 3PL, have been discussed in this chapter. However, this thesis will consider the application of the 1PL and 2PL models. The next chapter will consider the application of Bayesian inference with MCMC methods on the UIRT models.

Chapter 4

Bayesian Inference with MCMC on the UIRT Models

4.1 Introduction to Parameter Estimation with MCMC methods

The popularity of the MCMC method has increased due to the flexibility of its application, especially for complex models. To date, many researchers have investigated the application of the MCMC methods to the IRT models. Patz and Junker (1999b) introduced a general MCMC methodological guidance in complex IRT models for Bayesian inference. Their results suggested that MCMC, based on Metropolis Hastings sampling, can accurately fit the two-parameter logistic (2PL) model. Patz and Junker (1999a) extended their methodology to address some issues such as missing data and non-response and studied the behaviour of a guessing parameter in the 3PL model. They succeeded in applying MCMC based on Metropolis Hastings within Gibbs, but they pointed out that the 3PL model required a longer time to run the Markov chain. Hence, they have found some difficulty in the computational efficiency, which was not a concern for their study.

Many studies with the help of software development have been carried out to improve the application of the MCMC in the IRT model for educational uses. For examples, see Kim and Bolt (2007), Levy et al. (2011) and Junker et al. (2016) for more explanation of the applications of MCMC in the IRT models. Moreover, several studies have shown an advantage feature of Bayesian estimation in the IRT model for small sample sizes by using MCMC methods, such as Finch and French (2019).

Within MCMC methods, several algorithms have been widely examined in the IRT models, such as Gibbs sampling (Jiang and Templin 2019, Do 2021 and Fu et al.

2021), Metropolis Hastings (Patz and Junker 1999b), Metropolis Hastings within Gibbs (Patz and Junker 1999a), blocked Metropolis and Metropolis Hastings Robins Monroe (Cai 2010). However, a few researchers have studied the application of the Hamiltonian Monte Carlo (HMC) algorithm in the IRT models, which is considered the new MCMC method, such as Luo and Jiao (2018), Ames and Au (2018) and Do (2021). These studies have been implemented through a software program called Stan (Stan Development Team, 2022).

This chapter aims to compare the application of two MCMC methods extensively: Metropolis Hastings within Gibbs and Hamiltonian Monte Carlo to the IRT model, which explores the accuracy, complexity of the implementation, and computational costs.

4.2 Prior Distributions for UIRT Models

In Bayesian inference, the accuracy of parameter estimations is expected to improve by taking into account the prior information for the unknown parameter in the estimation procedures, while misspecification of the prior distribution could lead to incorrect inference (Evans and Moshonov, 2006). As mentioned in 2.2, the prior distribution is typically specified by the user, based on personal belief from previous experience with the parameter or from the statistical properties of the parameter that need to be estimated. Therefore, one of the critical issues in Bayesian analysis is the specification of prior distributions.

Despite the increased popularity of using Bayesian inference via MCMC methods in the IRT models, a few studies have considered the impact of the choices of the prior distributions on the accuracy of the estimates. For example, Ghosh et al. (2000) discussed the choice of priors and its effect on posterior propriety for the one-parameter normal model, $p(X_{ij} = 1) = \boldsymbol{\pi}(\theta_i - b_j)$, where $\boldsymbol{\pi}$ is the standard normal cumulative distribution function. Also, Sheng (2010) investigated the impact of prior specification on the accuracy of the three parameters IRT normal model. However, this model is outside of the scope of this thesis.

Marcoulides (2018) investigated the effectiveness of using different specified priors in estimating item parameters in the two-parameter logistic (2PL) using simulated data. The study focused on a comparison of using three different types of prior; non-informative, bad informative and good informative, using some statistical measurements, such as relative bias (Flora and Curran, 2004), to quantify the accuracy of parameter estimates. The results were all different and presented varying

levels of estimation bias, suggesting that Bayesian estimation of a 2PL model appears quite sensitive to the prior choice. However, the researcher recommended that more investigation was needed to include different types of priors for other models and sample sizes. Also, one of the recent studies investigated how the choice of prior distribution in the IRT model may influence the sensitivity of the posterior predictive checks; see (Ames, 2018) for further details.

The choice of prior distributions becomes more critical for smaller sample sizes. One way of specifying prior distribution, as recommended in Kim and Bolt (2007), is by eliciting experts' information. In this method, experts face a carefully created list of questions, which they answer according to their knowledge. However, the applications of this method have been limited in the IRT models. For more information about the prior elicitation and its application in the IRT models, see the Ames and Smith (2018) and Andrade and Gosling (2018).

Although it is expected that suitable informative priors would give the best estimation result, it is more common in practice to specify a weakly informative prior for different IRT models, such as Sinharay (2006), Kim and Bolt (2007) and Junker et al. (2016). This thesis will also consider the use of weakly informative prior distributions. In practice, the prior distributions will independently be determined on the person (ability) and item parameters. For more details about prior distributions for Bayesian IRT models, see Fox (2010) and Bürkner (2019).

Prior Setting for this Application

With respect to the ability parameters $\boldsymbol{\theta}$, it is common to apply the same prior for all examinees. The most common choice is a normal distribution (e.g. Patz and Junker (1999a) and Bürkner (2020)) with a mean equal to zero and a variance set in advance by the user; $\theta_i \sim \text{Normal}\,(0, \sigma_\theta^2)$. The interest in this chapter is in the 2PL IRT model (3.1.2). Therefore, there will be two item parameters; the difficulty ($\boldsymbol{b}$) and the discrimination ($\boldsymbol{a}$). In practice, it is also common to choose the normal distribution with the mean being zero and a reasonable value for the variance for $\boldsymbol{b}$; $b_j \sim \text{Normal}\,(0, \sigma_b^2)$. By using this prior distribution for $\boldsymbol{b}$, it is assumed that the item parameters of the same item are uncorrelated (Fox, 2010). The variance of both priors of $\boldsymbol{\theta}$ and $\boldsymbol{b}$ will be relatively large for the weakly informative prior goal. In terms of the item discrimination parameters ($\boldsymbol{a}$), in most testing settings, a_j is typically greater than 0 (DeMars, 2010), suggesting that the prior distribution of $\boldsymbol{a}$ can be modelled by positively skewed distribution. For that purpose, a gamma distribution will be used in this application; $\boldsymbol{a} \sim \text{gamma}\,(\alpha, \beta)$. An important

advantage of the MCMC methodology is that there is much freedom in choosing the prior distributions, where less or more informative priors or using different types of distribution forms could be chosen as well. The prior specifications are summarised in Table 4.1.

Table 4.1: Prior Specification for the 2PL Model.

Parameter	Prior
Ability	$\boldsymbol{\theta} \sim N(0, \sigma_\theta^2)$
Difficulty	$\boldsymbol{b} \sim N(0, \sigma_b^2)$
Discrimination	$\boldsymbol{a} \sim gamma(\alpha, \beta)$

4.3 MCMC Algorithms Settings

The most common Bayesian algorithms for estimating IRT model parameters are Metropolis Hastings (explained in 2.4.1) and Gibbs sampling (explained in 2.4.2). The Gibbs sampling algorithm is frequently used for the normal model, where full conditional posterior distributions of parameters can be derived in closed-form expressions. However, this is not the case for the logistic models. Jiang and Templin (2019) provided a new method for deriving conditional distributions of IRT and using Gibbs sampling for the logistic model. Still, this approach is outside of the focus of this thesis since it will not lead to an improvement in computation time.

The Metropolis Hastings algorithm requires a decision to reject/accept the new samples from the proposal distribution for each parameter in every step of the Markov chain. However, controlling the acceptance rate for IRT models with large numbers of examinees and items for large datasets becomes hard. As suggested by Patz and Junker (1999a) and Patz and Junker (1999b), combining strategies from the Gibbs and MH algorithms could simplify the process of generating the Markov chain for logistic IRT models. Therefore, this thesis will consider the combination of the Gibbs sampler and MH algorithm called Metropolis-Hasting within Gibbs (MH/Gibbs).

In addition to Gibbs sampling and the MH algorithm, Hamiltonian Monte Carlo has recently gained researchers' attention in the IRT models. HMC can provide preciser proposal values than MH by using the Hamiltonian dynamic as explained in 2.4.3. Therefore, this method will also be considered in this thesis because of the computational efficiency of the algorithm (Brooks et al. 2011 and Hoffman et al. 2014).

4.3.1 Metropolis Algorithm within Gibbs Sampler

The main structure of the MH/Gibbs algorithm is similar to the Gibbs sampler 2.4.2, in which we can sample one or a couple of parameters simultaneously based on the full conditional distribution. However, the MH algorithm 2.4.1 is added to overcome the difficulty of computing the full conditional distribution. The MH/Gibbs algorithm for the two-parameter 3.1.2 model can be composed of 4 steps as follows;

Repeat for $t = 1, 2, ..., T$, where T is the number of iterations:

1. sample $\theta_i^{(t+1)}$ from $p(\theta_i|X, \boldsymbol{\theta}_{-\boldsymbol{i}}^{(t)})$ for $1 \leq i \leq n$

   (a) Generate a candidate value, θ_i^* from a proposal density; $\theta_i^* \sim N(\theta_i^{(t)}, \sigma_\theta^2)$.

   (b) Update $\theta_i^{(t+1)} = \theta_i^*$ with acceptance probability, $\alpha(\theta_i^{(t)}, \theta_i^*)$.

      i. Compute $\alpha(\theta_i^{(t)}, \theta_i^*)$;

$$\alpha(\theta_i^{(t)}, \theta_i^*) = \min\left\{1, \frac{p(\theta_i^*)p(X|\theta_i^*, \boldsymbol{\theta}_{-\boldsymbol{i}}^{(t)})}{p(\theta_i^{(t)})p(X|\theta_i^{(t)}, \boldsymbol{\theta}_{-\boldsymbol{i}}^{(t)})}\right\}$$

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

      iii. If $u < \alpha$ then accept the proposal $\theta_i^{(t+1)} = \theta_i^*$, otherwise reject the proposal and set $\theta_i^{(t+1)} = \theta_i^{(t)}$.

2. sample $b_j^{(t+1)}$ from $p(b_j|X, \boldsymbol{b}_{-\boldsymbol{j}}^{(t)})$ step for $1 \leq j \leq m$.

   (a) Generate a candidate value, b_j^* from a proposal density; $b_j^* \sim N(b_j^{(t)}, \sigma_b^2)$.

   (b) Update $b_j^{(t+1)} = b_j^*$ with acceptance probability; $\alpha(b_j^{(t)}, b_j^*)$.

      i. Compute $\alpha(b_j^{(t)}, b_j^*)$;

$$\alpha(b_j^{(t)}, b_j^*) = \min\left\{1, \frac{p(b_j^*)p(X|b_j^*, \boldsymbol{b}_{-\boldsymbol{j}}^{(t)})}{p(b_j^{(t)})p(X|b_j^{(t)}, \boldsymbol{b}_{-\boldsymbol{j}}^{(t)})}\right\}$$

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

      iii. If $u < \alpha$ then accept the proposal $b_j^{(t+1)} = b_j^*$, otherwise reject the proposal and set $b_j^{(t+1)} = b_j^{(t)}$.

3. sample $a_j^{(t+1)}$ from $p(a_j|X, \boldsymbol{a}_{-\boldsymbol{j}}^{(t)})$ step for $1 \leq j \leq m$.

   (a) Generate a candidate value, a_j^* from proposal density; $a_j^* \sim N(a_j^{(t)}, \sigma_a^2)$

   (b) Update $a_j^{(t+1)} = a_j^*$ with acceptance probability, $\alpha(a_j^{(t)}, a_j^*)$.

i. Compute $\alpha(b_j^{(t)}, b_j^*)$;

$$\alpha(a_j^{(t)}, a_j^*) = \min\left\{1,\ \frac{p(a_j^*)p(X|a_j^*, \boldsymbol{a}_{-\boldsymbol{j}}^{(t)})}{p(a_j^{(t)})p(X|a_j^{(t)}, \boldsymbol{a}_{-\boldsymbol{j}}^{(t)})}\right\}$$

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

iii. If $u < \alpha$ then accept the proposal $a_j^{(t+1)} = a_j^*$, otherwise reject the proposal and set $a_j^{(t+1)} = a_j^{(t)}$.

4. Repeat steps 1, 2 and 3 until t reaches the total number of iterations T.

The Gibbs sampler is summarised in steps 1, 2 and 3, while the sub-steps (a) and (b), in each step, contain the same structure as the single iteration MH algorithm. Those sub-steps generate the full conditional distribution by proposal distributions of $\boldsymbol{\theta}$, $\boldsymbol{b}$ and $\boldsymbol{a}$. Due to their simplicity, symmetric distributions are a popular choice for the proposal distribution when the MH/Gibbs algorithm is used for IRT models. Therefore, a normal distribution with mean equal to a current iteration and standard deviation, which is fixed in advance, is used in this setting. This will simplify the algorithm to the Metropolis algorithm 2.4.1. Therefore, from now on, this method will be referred to as M/Gibbs. The choice of the proposal variance, as mentioned in section 2.4.1 affects the algorithm's performance. In this study, the variance is tuned automatically to obtain an acceptance probability between 25% to 50%, as recommended by Patz and Junker (1999a).

4.3.2 Hamiltonian Monte Carlo Algorithm in the UIRT Models

The main structure for the HMC algorithm has been explained in greater detail in section 2.4.3. In order to use the HMC algorithm in the IRT models, the first derivatives of the log-posterior distribution with respect to each parameter are required, which can be implemented in equation 2.31, and hence the following processes of the HMC algorithm. The log-posterior is the sum of the log-likelihood and the log-prior distribution for each parameter. Thus, we can start by finding the first derivative of the log-likelihood concerning each parameter. The likelihood function for the 2PL model can be written as:

$$L(\boldsymbol{x}, \boldsymbol{\pi}) = \prod_{i=1}^{n}\prod_{j=1}^{m} \pi_{ij}^{x_{ij}}(1 - \pi_{ij})^{1-x_{ij}},$$

where $\pi_{ij} = e^{\eta_{ij}}$ and $\eta_{ij} = a_j(\theta_i - b_j)$.

Hence, The log-likelihood function for the 2PL model can be written as:

$$l_{ij} = \log(\pi_{ij})^{x_{ij}} + \log(1-\pi_{ij})^{1-x_{ij}}.$$

This can be simplify as:

$$l_{ij} = x_{ij}\left(\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) + \log(1-\pi_{ij})\right).$$

$$l_{ij} = x_{ij}\left(\log\left(1 - \frac{\pi_{ij}}{1+\pi_{ij}}\right)\right)$$

$$l_{ij} = x_{ij}\left(\log\left(\frac{1}{1+\pi_{ij}}\right)\right)$$

$$l_{ij} = x_{ij}\left(\log\left(\frac{1}{1+e^{\eta_{ij}}}\right)\right)$$

$$l_{ij} = x_{ij}\eta_{ij} - \log(1+e^{\eta_{ij}})$$

The first derivative derivative with respect to η_{ij}:

$\frac{dl_{ij}}{d\eta_{ij}} = (x_{ij} - \pi_{ij})$

The first derivative derivative with respect to θ_i:

$\frac{\partial l_{ij}}{\partial \theta_i} = \frac{\partial l_{ij}}{\partial \eta_{ij}}\frac{\partial \eta_{ij}}{\partial \theta_i} = \sum_{i=1}^{i=n}(x_{ij} - \pi_{ij}) * a_j.$

The first derivative derivative with respect to b_j:

$\frac{\partial l_{ij}}{\partial b_j} = \frac{\partial l_{ij}}{\partial \eta_{ij}}\frac{\partial \eta_{ij}}{\partial b_j} = -\sum_{j=1}^{j=m}(x_{ij} - \pi_{ij}) * a_j.$

The first derivative derivative with respect to a_j:

$\frac{\partial l_{ij}}{\partial a_j} = \frac{\partial l_{ij}}{\partial \eta_{ij}}\frac{\partial \eta_{ij}}{\partial a_j} = -\sum_{j=1}^{j=m}(x_{ij} - \pi_{ij}) * (\theta_i - b_j).$

The specifications of the prior distribution for each parameter, that will be used in this application, can be found in Table 4.1. Regarding the first derivative of the prior distributions, the first derivative of the normal distribution with respect to θ_i:

$$p(\theta_i) = \left(2\pi\sigma_\theta^2\right)^{-1/2}\exp\left(-\frac{1}{2}\frac{(\theta_i - \mu_\theta)^2}{\sigma_\theta^2}\right)$$

$$\frac{dp_i}{d\theta_i} = \frac{-(\theta_i - \mu_\theta)}{\sigma_\theta^2}$$

The first derivative of the normal distribution with respect to b_j :

$$p(b_j) = \left(2\pi\sigma_b^2\right)^{-1/2} \exp\left(-\frac{1}{2}\frac{(b_j - \mu_b)^2}{\sigma_b^2}\right)$$

$$\frac{dp_j}{db_j} = \frac{-(b_j - \mu_b)}{\sigma_b^2}$$

The first derivative of the gamma distribution with respect to a_j :

$$p(a_j) = \frac{a_j^{\alpha-1} e^{-\beta a} \beta^\alpha}{\Gamma(\alpha)} \quad \text{for } a_j > 0 \quad \alpha, \beta > 0$$

$$\frac{dp_j}{da_j} = \left(\frac{\alpha - 1}{a_j}\right) - 1$$

Finally, the resulting derivative from the log-likelihood concerning each parameter is added to its first derivative of the prior distribution to find the log-posterior distribution. This result will be coded in the R programming code in order to be able to apply the HMC algorithm. Implementing the HMC algorithm also requires setting and tuning two parameters carefully; trajectory length (L) and step size (ϵ) 2.4.3. These will be set automatically to achieve the recommended acceptance probability (2.32) between 60% to 70%.

4.4 Comparison Study

This section will compare the performance of the Hamiltonian Monte Carlo (HMC) and Metropolis within Gibbs samplers (M/Gibbs) for UIRT models. The comparison between the two algorithms will be carried out in terms of accuracy, efficiency and computational time.

4.4.1 Simulated Data

The data in this setting will be represented as a matrix $\boldsymbol{X}$; where

$$X_{ij} = \begin{cases} 1 & \text{if examinee i answer item j correctly} \\ 0 & \text{if examinee i answer item j incorrectly,} \end{cases}$$

and $i = 1, 2, ..., n$ (number of rows (examinees)) and $j = 1, 2, ..., m$ (number of columns (items)). In this setting, the questions or items are measure the same skill (unidimensional ability), and the examinees are assumed to answer all questions. Therefore, $X_{ij} \sim Bernoulli(\pi_{ij})$ where $logit(\pi_{ij}) = a_j(\theta_i - b_j)$ which give the

likelihood of this model as:

$$L(\boldsymbol{x}|\theta_i, b_j, a_j) = \prod_{i=1}^{n}\prod_{j=1}^{m} \pi_{ij}^{x_{ij}}(1-\pi_{ij})^{1-x_{ij}}$$

The $\theta_i's$ range uniformly from -4 to 4, $b_j's$ range uniformly from -2 to 2, as these ranges suggested by DeMars (2010). Both parameters $\boldsymbol{\theta}$ and $\boldsymbol{b}$ are centred around zero. The $a_i's$ is set to be positive (as explained in 3.1.2) and range uniformly between 0.5 to 1.

This experiment will be carried out for the two-parameter logistic model (2PL) with binary responses (correct answer=1, incorrect answer=0) and unidimensional ability. The comparison study will consider the case of moderate sample size and test length; $n = 200$ and $m = 20$.

4.4.2 Simulation Framework

The main aim of performing this experiment is to compare the computational expensiveness, efficiency and the accuracy of M/Gibbs and HMC.

For a fair comparison, both algorithms will be applied to the same dataset. Also, the same prior distributions and same initial values will be used. Finally, the comparison will be carried out by checking for accuracy, mixing, efficiency and convergence to the desired target distribution using the following:

- To determine the quality of the approximation obtained from the two methods, the resulting posterior distributions will be compared to the true values.
- The trace plots (2.5) will be used to visually assess the mixing of the Markov chains for both methods.
- The autocorrelation (2.5)and the effective sample size (2.5) per second will be used to compare the efficiency between these two algorithms.
- To assess the convergence of the chains, four chains with different initial values will be run and use the plot of Gelman-Rubin statistics ($\widehat{R}$) test (2.5). For convergence, $\widehat{R}$ should be approximately 1 ± 0.1.
- The computational time for both algorithms will also be recorded.

Some previously studied, such as Kim and Bolt (2007),Patz and Junker (1999b) and Patz and Junker (1999a), have used 10,000, 25,0000 and 50,000 iterations,

respectively for estimating 2PL models. In this thesis, 100,000 iterations will be used to ensure both algorithms will converge to the desired posterior distributions, and obtain stable parameter estimates. The convergence diagnostics that will be carried out in this section, will be implemented through coda package in R (Plummer et al., 2006).

4.4.3 Comparison

The comparison of results between the two proposed algorithms will be discussed in this section. The priors' parameters are set as follows;

$$\boldsymbol{\theta} \sim N(0, \sigma_\theta^2 = 10)$$

$$\boldsymbol{b} \sim N(0, \sigma_b^2 = 10)$$

$$\boldsymbol{a} \sim gamma(\alpha = 1, \beta = 2)$$

The tuning of other parameters, such as the proposals' variances, trajectory length and step size, which are set for this result, has been discussed early in 4.3.1 and 4.3.2.

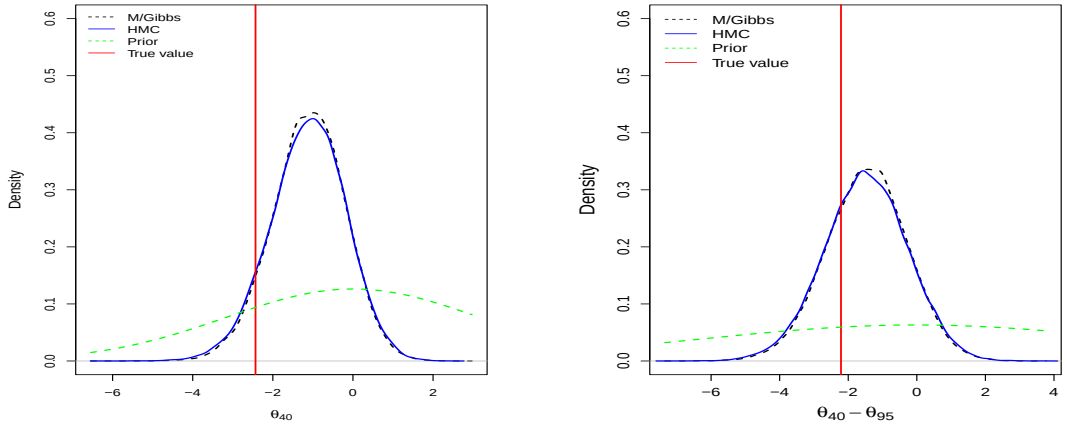
Figure 4.1 displays the resulting posterior distributions from both algorithms for three levels of examinees' abilities. These abilities were randomly selected according to the number of correct answers to represent low, moderate and high abilities. Also, the figure shows the posterior distributions of the difference between the two abilities. The result shows that the two posterior densities (black dashed line for M/Gibbs and blue line for HMC) are very close to being identical, which indicates that both methods track the same target distributions.
Because weakly informative priors were used (dashed green line), as detailed in 4.2, most of the information in the posteriors is obtained from the data. Taking the prior distribution into account, we notice that both methods result in biased estimations. However, by taking the difference between these two parameters, the estimations of these differences become almost unbiased, where the actual values (red line) become close to the mean of the posterior distributions.

Figure 4.2 shows the trace plots after the burn-in period, which represent sampling histories of the chain obtained from M/Gibbs (left) and HMC (right). The first 1000 iterations were discarded based on an initial visualisation. The results do not show a lack of convergence using either algorithm. The chains appear to mix well in both algorithms. The parameters move quickly to the target distribution, and the

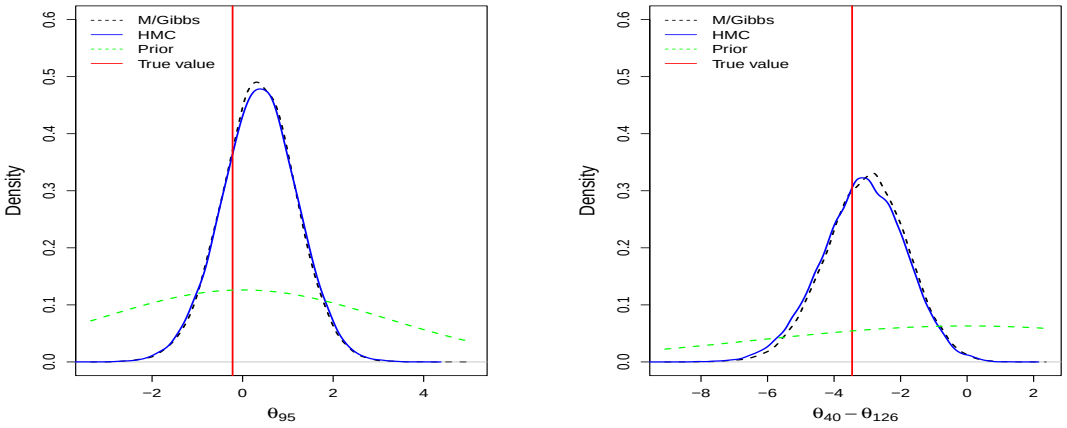
algorithms explore the space well by moving rapidly through the range of the target distribution.

Figure 4.3 displays the autocorrelation function plots (ACF) for three different parameters; θ_{40}, θ_{95} and θ_{126} . These plots show how the autocorrelation between samples decreases as a function of their lag. The ACF plots resulting from both algorithms do not provide evidence for any problem where we can see that the autocorrelation at lag 1 is already less than 0.8 and then dropped to zero quickly. However, it is noticeable that HMC dropped faster than M/Gibbs.

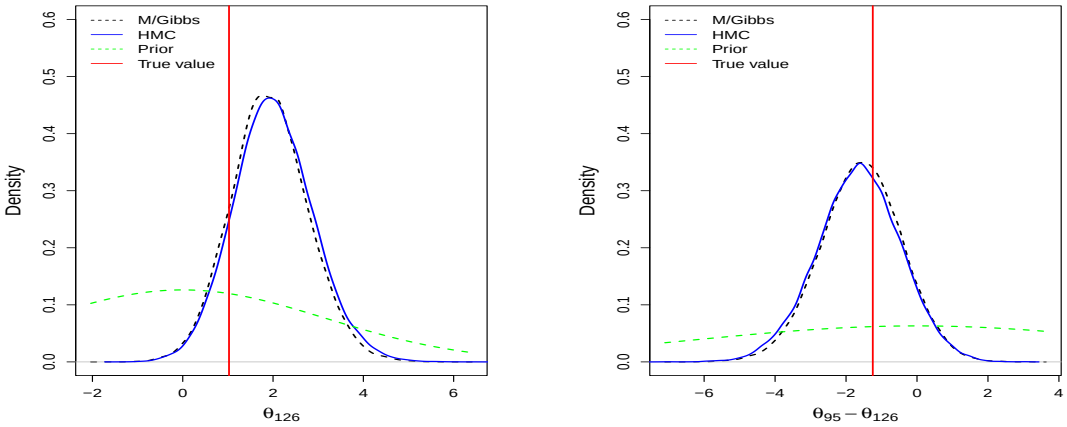
To measure the efficiency of the proposed algorithms, the effective sample size per second (ESS/Sec) was recorded for both methods and plotted in Figure 4.4 against the ability estimates. The ESS/Sec for M/Gibbs ranged from 1 to 20, and HMC ranged from 10 to 50. The results indicate that HMC produces samples from the posterior distribution with much lower autocorrelations compared to M/Gibbs. Therefore, in order to get the same information, we have to run M/Gibbs for a longer time. Thus, HMC is sampling more efficiently from the target distribution than M/Gibbs.



Number of Correct Answers= 6



Number of Correct Answers= 10



Number of Correct Answers= 15

Figure 4.1: Posterior density plots for M/Gibbs and HMC methods of selected examinees' abilities with different numbers of correct answers.

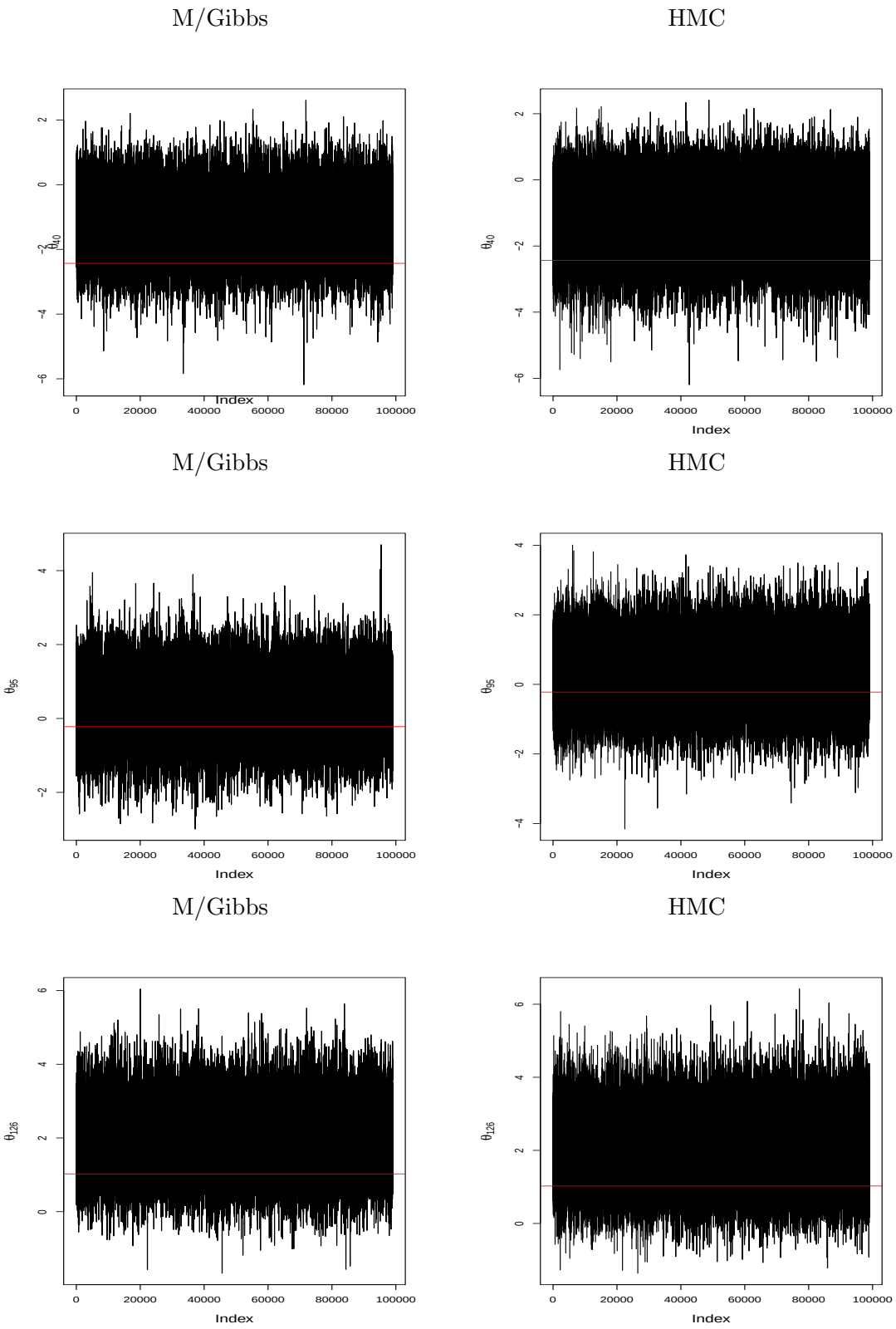


Figure 4.2: Trace plots of three levels of randomly selected examinees' abilities obtained from M/Gibbs (left) and HMC (right). The red line indicates the true parameter value.

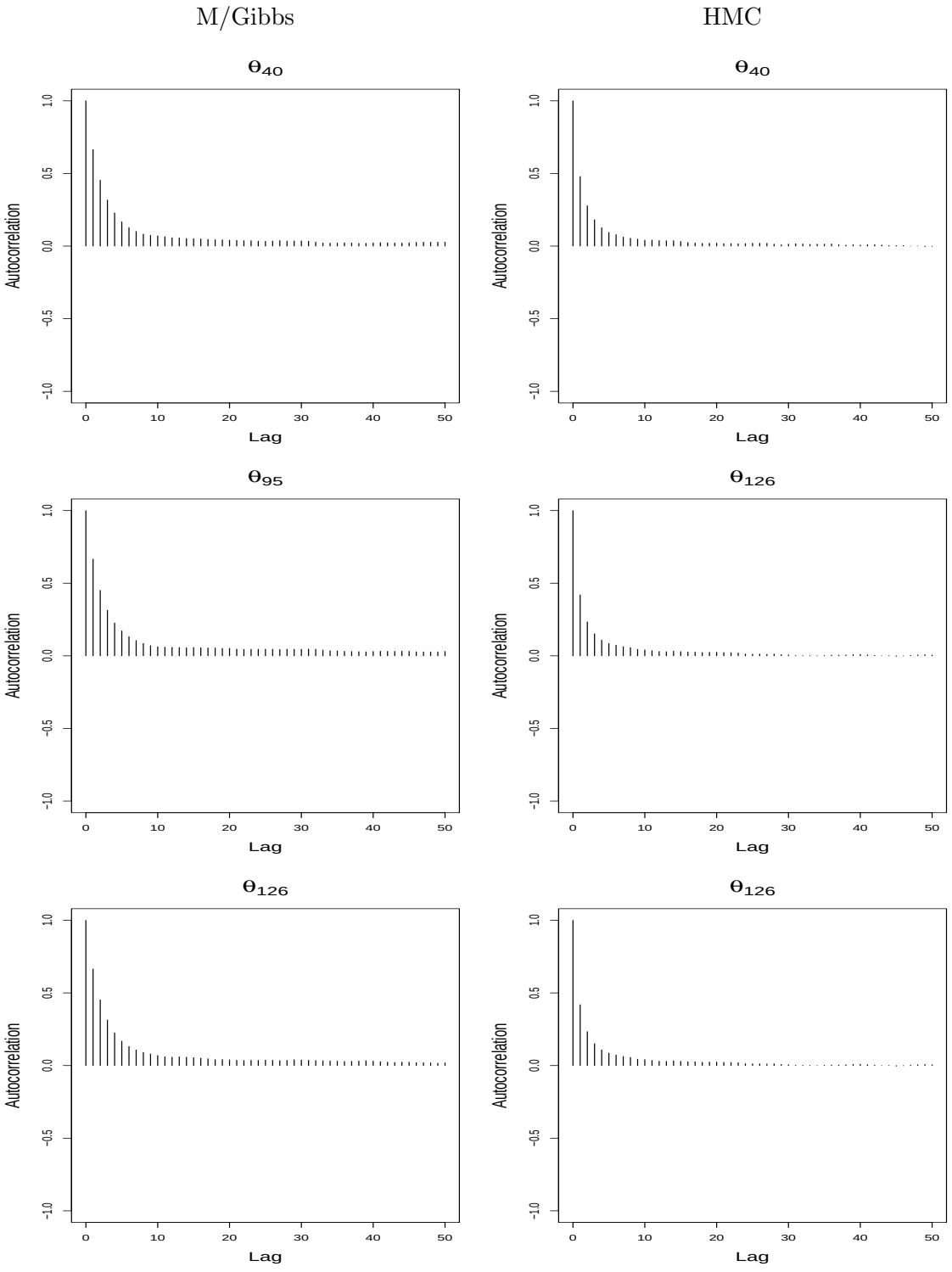


Figure 4.3: Autocorrelations between the samples returned by M/Gibbs (left) and HMC (right) for three levels of randomly selected examinees' abilities.

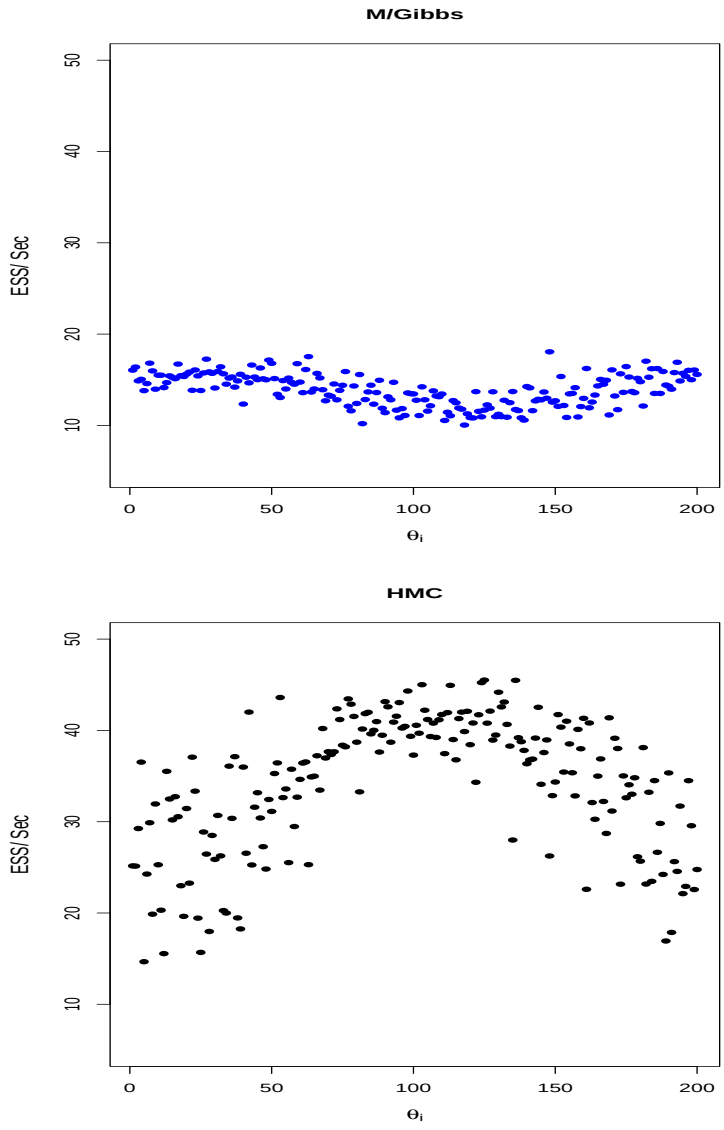


Figure 4.4: ESS per second from the performance of M/Gibbs (blue) and HMC (black).

The impact of using different initial values on convergence has been investigated using the Gelman-Rubin statistic test ($\widehat{R}$), also known as the potential scale reduction. Figure 4.5 shows the plots of the Gelman-Rubin statistic test . These plots result from running four chains initialised from different values. Two chains are initialised far from the true values, one chain is initialised from zero, and the final chain is initialised from values close to the true values. The plots show that after 100,000 iterations, the samples generated by M/Gibbs show a lack of convergence where $\widehat{R}$ has not reached the suggested value for convergence 1 ± 0.1. For example, the $\widehat{R}$ for θ_{40} ranged between 1.33 to 3.13. However, samples generated by HMC reached $\widehat{R} = 1$ fast, after approximately, 1000 iterations. The lack of convergence that appears in this experiment suggests that M/Gibbs are more sensitive to the starting values from HMC. Moreover, a large number of iterations is required if the starting

values are not chosen carefully. Figure 4.6 shows the plot of $\widehat{R}$ for two different chains initialised from values not so far from the actual values by using the logit function to estimate those values. It is clear the convergences have improved for M/Gibbs, where $\hat{R} \leq 1.1$. However, the potential scale reduction values become stable after 40,000 iterations.

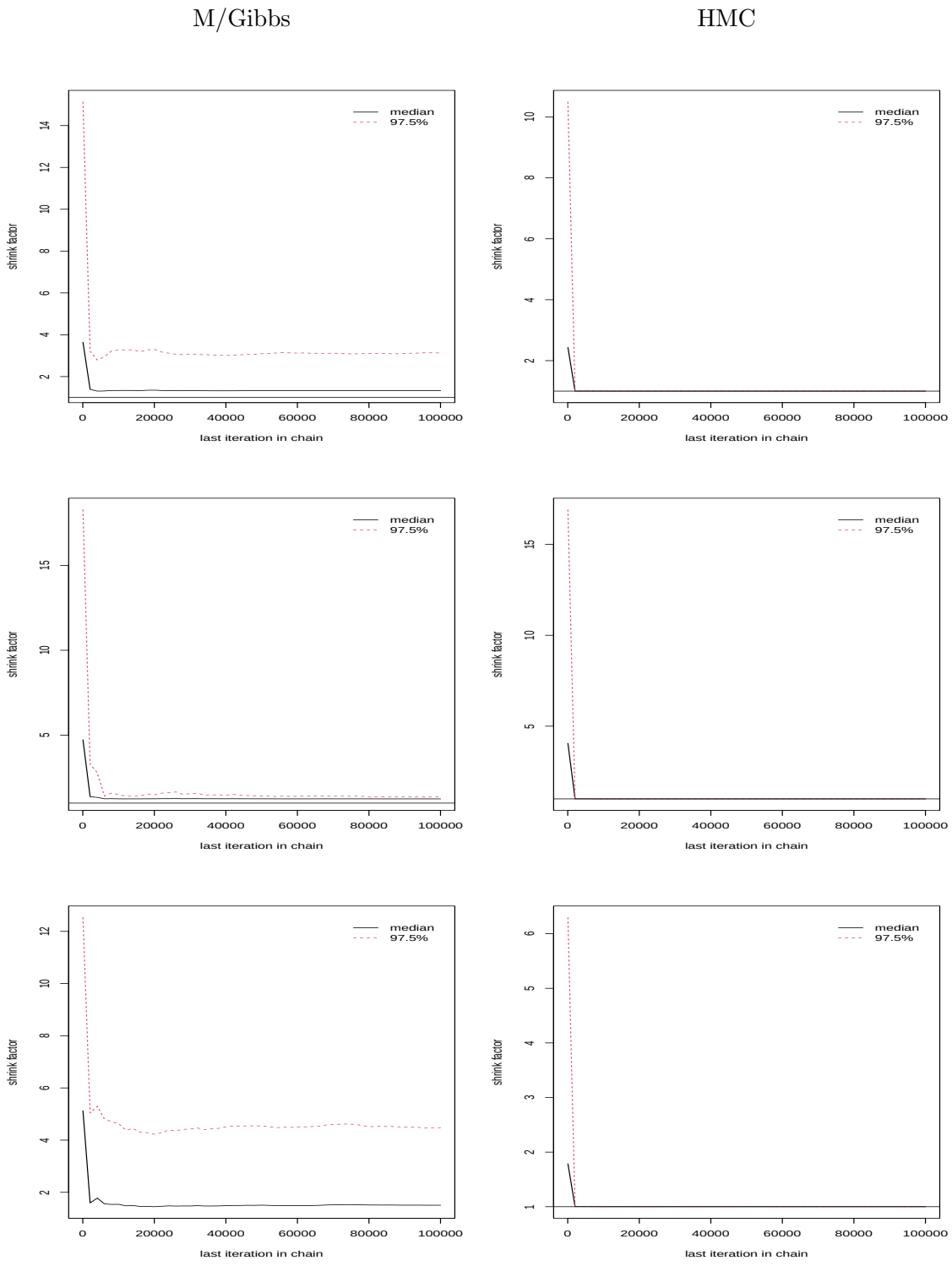


Figure 4.5: Potential scale reduction (shrink factor $\widehat{R}$) resulting from M/Gibbs (left) and HMC (right).The first row represents the result for θ_{40}, the second row θ_{95}, and the third row is the result of θ_{126}.

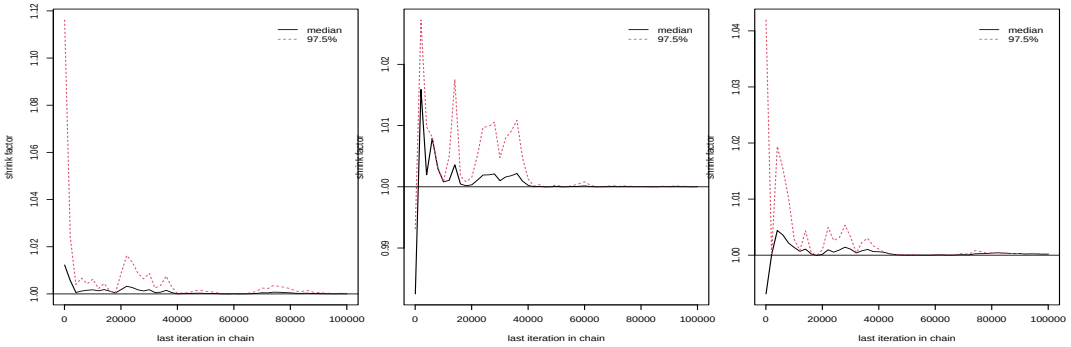
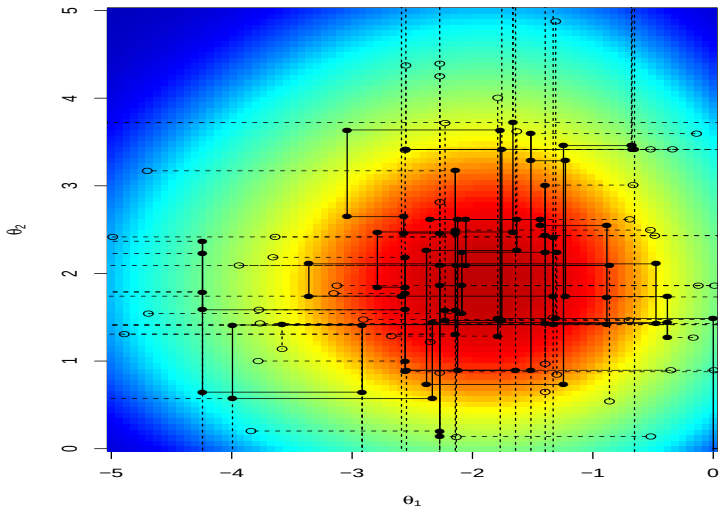


Figure 4.6: Potential scale reduction (shrink factor $\widehat{R}$) resulting from M/Gibbs for θ_{40}, θ_{95}, and θ_{126}.

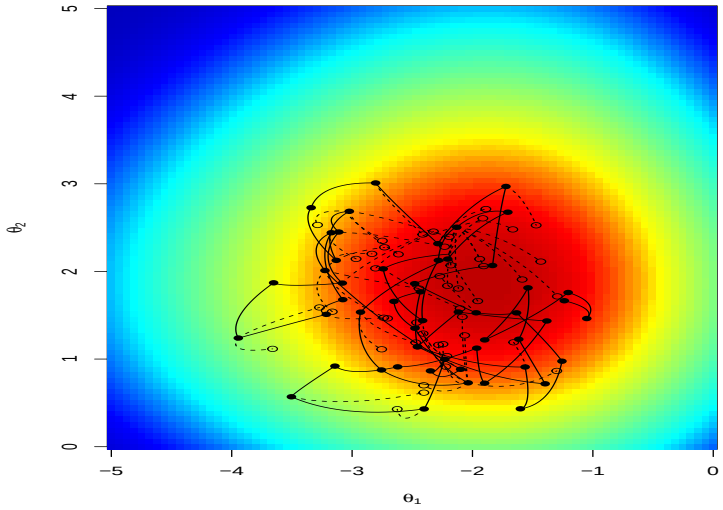
Regarding computational time, the average running times of repeating both algorithms 20 times are 960 and 540 seconds for M/Gibbs and HMC, respectively. This result suggests that HMC is almost twice as fast as M/Gibbs. That can be seen by tracing the path (trajectory) of the movements of both methods. The trajectory of 100 HMC, and M/Gibbs iterations are shown in Figure 4.7. It is clearly seen that the HMC converges to the high probability density region of the posterior distribution (red area) faster than M/Gibbs. The computational times will remain high, even using fewer iterations, such as 10,000. For example, Table 4.2 summarises the average run time of repeating both algorithms 20 times for different scenarios of simulated data and 10,000 iterations. Although HMC produces results faster than M/Gibbs, it still provides some high computational costs, especially in the case that real-time inference is required.

Table 4.2: Average running time of M/Gibbs and HMC for 10,000 iterations and different amounts of datasets.

Time in seconds		
Amount of Data	HMC	M/Gibbs
n=200, m=20	49	92
n=500, m=20	131	201
n=1000, m=20	254	386
n=1500, m=20	400	599



M/Gibbs



HMC

Figure 4.7: Trajectory of 100 iterations of HMC method, and M/Gibbs for two dimensions (2D) posterior distribution. The dark circles represent the accepted points, and the empty circles represent the rejected points.

4.5 Summary of the Chapter

Over the past years, many researchers have been using MCMC methods to estimate IRT model parameters. This chapter provided a brief summary of the application of MCMC methods to the IRT models. Moreover, Metropolis-Hastings within Gibbs (M/Gibbs), which is one common MCMC method for the logistic IRT model, was implemented for the 2PL IRT model. The result of this method was compared to the Hamiltonian Monte Carlo (HMC) algorithm, which is a new MCMC method

and is considered to be fast. The comparison was carried out in terms of accuracy, efficiency and computational time. The two-parameter logistic model (2PL) was first used to simulate the dataset. Each of the two MCMC algorithms was then separately implemented to estimate the posterior distributions of the ability parameters ($\boldsymbol{\theta}$) and the item parameters ($\boldsymbol{b}$ and $\boldsymbol{a}$). Both methods were applied successfully.

In this experiment, we found that HMC appeared less sensitive to the initial values, where the result of $\hat{R} \leq 1.1$ was fast, even where we started the chain far from the actual values. As a result, HMC was able to achieve a reasonable convergence to the target distribution more quickly with fewer iterations. Moreover, HMC appeared to be more efficient with high effective sample sizes per second and negligible autocorrelation. On the other side, using reasonable initial values for M/Gibbs showed no signs of non-convergence. However, M/Gibbs appeared to have sensitivity to the initial values that require running the algorithm for a longer time. Moreover, for the same number of iterations, HMC has less computational time compared to M/Gibbs. This comparison study suggests using the HMC method would be preferable to the M/Gibbs. Although this chapter focuses on presenting the result of the ability parameter $\boldsymbol{\theta}$, the same conclusion is valid for the item parameters. See Appendix A for some results of estimation of the difficulty parameter.

A bigger challenge arises when the data arrives in real-time continuously, and the parameters need to be estimated online. When new data comes from individuals or questions through time, dynamic structures of student abilities and questions difficulties need to be included in the model to accommodate changes in ability and difficulty. Focusing on real-time response data, as we have seen from this chapter, the methods such as MCMC appear to be computationally expensive. Therefore, these methods will not tend to scale well for streaming data and large-scale real-time systems. To reduce the computational time of dynamic Bayesian inference, sequential Monte Carlo methods (SMC) 2.7 have been widely used to explore a posterior sequence. The application of two settings for SMC algorithms will be presented in the next chapter, and the results will be compared to the MCMC method.

Chapter 5

Sequential Monte Carlo Methods on the Dynamic IRT Model

The main goal of this thesis is to provide an efficient Bayesian inference for both massive data and online inference, taking into account the accuracy and speed. We have seen from the previous chapter that Markov chain Monte Carlo (MCMC) techniques are infeasible for massive data due to computational cost. Standard MCMC methods generally require re-computing the posterior distributions every time new data becomes available. Even though for the fastest MCMC method, such as Hamiltonian Monte Carlo and moderate sample size as provided in Chapter 4, MCMC remains computationally expensive. Therefore, the velocity and volume present considerable challenges to apply MCMC methods when real-time inference is required or for dynamic problems where the posterior distribution develops over time.

Sequential Monte Carlo (SMC) methods (2.7) have been widely used in the literature to reduce the computational cost of dynamic Bayesian analysis (e.g Liu and Chen (1998)). In this method, the posterior distribution is constructed in such a way to avoid re-computing the likelihood of old data when new data arrive, allowing use of the information without great computational cost. However, although SMC methods have become very common over the last few years to solve a variety of sequential Bayesian inference problems, the application of SMC remains limited in IRT models, and as far as is known, currently, there is no application of SMC methods to IRT models. Therefore, the main contribution of this chapter is applying the SMC method to the IRT model and a comprehensive investigation into the performance of the proposed method.

5.1 Classic Sequential Monte Carlo Methods

In this section, the SMC algorithm described in 2.7 will be developed using the classical SMC method; moving from prior to posterior (Del Moral et al., 2006). This method is the most commonly used in the literature for different types of models for both online and offline inference. However, most of the methodological results occurred outside the scope of education. For example, see Schäfer and Chopin (2013) and McLean et al. (2017). This method will be denoted by SMC1. The goal of using this method is that we want to estimate the ability of students every time a new student answers the test. Therefore, we start from the prior distribution, and the likelihood is updated gradually until we reach the desired posterior distribution. The intermediate distributions in this setting help us approximate the final target distribution. In educational scenarios, these intermittent distributions can be developed in different ways. For example, we could assume these intermediate distributions to be a sequence of students' ability distributions. Every time a new student answers the test, the likelihood is updated until the last student finishes the test. Hence, there is no need to re-evaluate the whole process every time students take the test in this setting. Therefore, it could be helpful for dynamic systems.

The performance of the classical SMC methods will be demonstrated in this section. This section aims to explain the basic SMC1 method and cover some of the comparison results to MCMC that justify this method in practice.

5.1.1 Algorithm Setting

The algorithm explained in section 2.7 is very general. There is a wide range of possible options to consider when setting an SMC algorithm, such as the appropriate sequence of intermediate distributions π and the choice of importance density.

There are different ways of choosing a sequence of π. One way is given by Neal (2001) as the following:

$$\pi_i(\boldsymbol{\theta} \mid \mathcal{D}) = p(\mathcal{D} \mid \boldsymbol{\theta})^{\tau_i} p(\boldsymbol{\theta}), \qquad i = 0, \ldots, s$$

where s is the number of stages, and τ_i is non-decreasing such that:

$$0 = \tau_0 \leq \cdots \leq \tau_i \leq \cdots \leq \tau_s = 1$$

When $\tau_0 = 0$, the samples are coming from the prior $p(\boldsymbol{\theta})$, and when $\tau_s = 1$, the samples are coming from the posterior distribution. Therefore, the effect of the likelihood is included gradually in order to obtain at the end($i = s$) an

approximation of the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathcal{D})$. The intermediate distributions, i.e. $\tau_i(\boldsymbol{\theta}_{1:i})$ for $i < s$, are useful in helping us to approximate the final target posterior $\tau_s(\boldsymbol{\theta}_{1:s})$. In practice, it shows that it is essential to have τ_i closer to each other near the prior; therefore, many researchers have been using the following sequence of τ_i:

$$\tau_i = \left(\frac{i}{s}\right)^c,$$

where c is a small natural number. Frequently $c = 3$ or $c = 4$. In this thesis $c = 4$.

To investigate the performance of SMC1, where the sequence moves from prior to posterior, the basic SIR algorithm (6) described in subsection (2.7.2) will be run to make inference about the posterior distributions of the ability parameter $\boldsymbol{\theta}$, and the difficulty $\boldsymbol{b}$. This experiment will look at a small dataset simulated from the one-parameter logistic model (1PL) with binary responses (correct answer=1, incorrect answer=0) and unidimensional ability.

The proposal (kernel) densities (2.7) for the ability parameter $q_i(\boldsymbol{\theta})$ and difficulty $q_i(\boldsymbol{b})$ are chosen to be an adaptive normal distribution with variance estimated with a population variance of the previous SIR stage. This variance will be multiplied by a scaling factor so that every single intermediate density is slightly smaller than the previous one. More information about choosing and adapting the proposal density can be found in Fearnhead and Taylor (2013).

The ESS is monitored during the run at each stage to ensure particle diversity (2.7.1). If the ESS falls below $\frac{N}{2}$ samples, the particles are re-sampled according to their weights. Therefore, at every stage of the sequential sampler, N particles are used.

5.1.2 Comparison Study

Simulated Data

The probability of getting the correct answer for 1PL can be written as:

$$p(X_{ij} = 1) = \frac{exp(\theta_i - b_j)}{(1 + exp(\theta_i - b_j))},\ \theta_i \ and\ b_j \in \mathbb{R}$$

Therefore, the data in this setting will be represented as a matrix $\boldsymbol{X}$; where

$$X_{ij} = \begin{cases} 1 & \text{if examinee i answers item j correctly} \\ 0 & \text{if examinee i answers item j incorrectly,} \end{cases}$$

and $i = 1, 2, ..., n$ (number of rows) and $j = 1, 2, ..., m$ (number of columns). In this setting, the questions or items are measure the same skill (unidimensional ability), and the examinees are assumed to answer all questions.

Hence, $X_{ij} \sim Bernoulli(\pi_{ij})$ where $logit(\pi_{ij}) = (\theta_i - b_j)$ which give the likelihood of this model as;

$$L(\boldsymbol{x}|\theta_i, b_j) = \prod_{i=1}^{n} \prod_{j=1}^{m} \pi_{ij}^{x_{ij}} (1 - \pi_{ij})^{1-x_{ij}}$$

The $\theta_i's$ range uniformly from -4 to 4, $b_j's$ from -2 to 2, as these ranges suggested by DeMars (2010). Both parameters $\boldsymbol{\theta}$ and $\boldsymbol{b}$ are centred around zero.

The comparison study will be carried out in small sample size, with $n = 10$ students and a test of length $m = 5$.

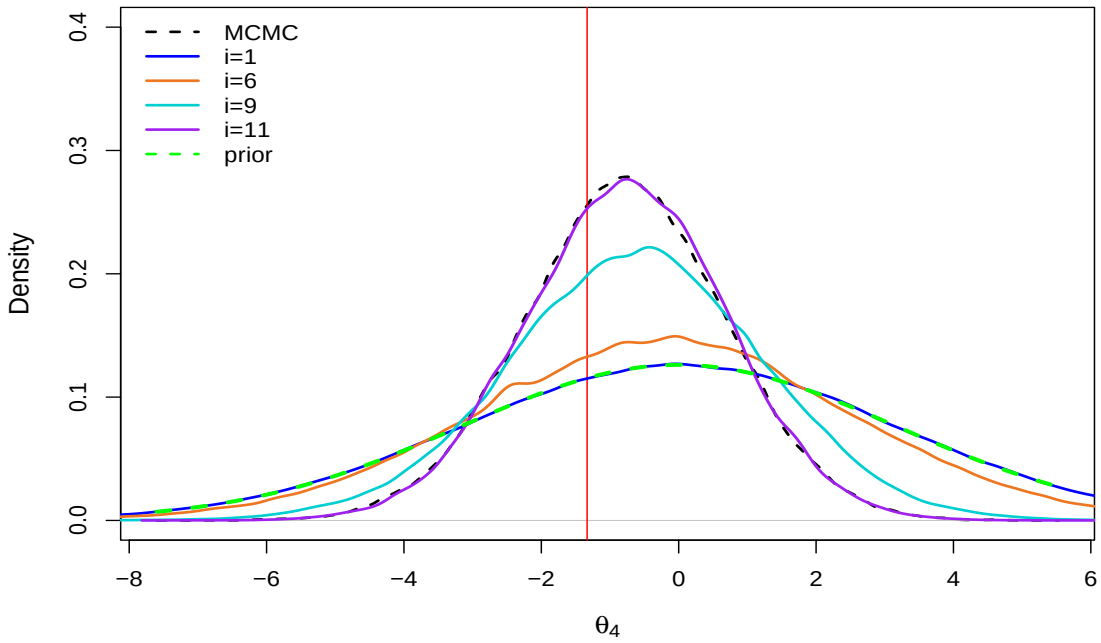


Figure 5.1: Distributions of the ability parameter (θ_4) at the first stage ($i = 1$), the intermediate stages ($i = 6$), ($i = 9$) and the final stage ($i = 11$) of SMC1 sampler comparing to MCMC method (M/Gibbs). The vertical red line represents the actual value.

Comparison Studies Results

This section will present the results of the comparison study between the SMC1 method (5.1.1) and one of the MCMC methods, M/Gibbs (4.3.1). The comparison will be carried out to compare the point estimates, the shape of the posterior (density distributions) and the computational cost resulting from each method. The M/Gibbs was applied first using a $100,000$ iterations. The mixing and convergence of the samples were then assessed through the suggested methods presented in section 2.5. Therefore, according to the visual result of the trace plots, the initial 500 iterations are discarded. For a fair comparison, the same setup in terms of priors and initial conditions is used.

Comparison of Distributions:

The posterior distributions generated from M/Gibbs are used to assess the accuracy of its corresponding approximation posterior distributions obtained from the SMC1 algorithm.

Figure 5.1 shows the posterior distribution resulting from the MCMC method (M/Gibbs) for one randomly selected ability parameter (θ_4) and three corresponding intermediate distributions resulting from SMC1. The figure illustrates the samples at the first stage ($i = 1$), where all the particles are coming from the prior distribution, two of the intermediate distributions ($i = 6$) and ($i = 9$), and the samples at the final stage which represent the density of the posterior distribution ($i = 11$). It is noticeable that the density distribution starts quite wide at the first stage and matches the prior very well. Then as we move away from the prior and include more stages (more data), the density becomes tighter and defines a reasonable posterior at the final stage, which matches the target estimate posterior generated by MCMC.

One of the main objectives is to investigate how much the quality of the approximation has been affected by various choices of the number of particles N. Thus, the SMC1 is performed with different particle numbers $N = \{10,000, 50,000, 100,000, 500,000\}$.

The results are shown in Figure 5.2 for two different examinees θ_3 and θ_6, where the number of correct answers is 2 and 3 respectively. The results indicate that increasing the number of particles used in the SMC1 algorithm has a powerful effect on improving the approximation. As we can see that using a small number of particles ($N = 10,000$) provides a poor approximation to the target estimate posterior (black dashed line). It is clear from Figure 5.2 (green line) that the samples have not fully discovered the parameter space compared to M/Gibbs. The approximation is improved by increasing the number of particles to $N = 50,000$ and $N = 100,000$. Moreover, by increasing the number of particles to $N = 500,000$, the posterior distributions resulting from SMC1 become almost identical to the posterior distributions generated from M/Gibbs. Therefore, according to this result in this particular case, at least $500,000$ particles are needed for the posterior distributions of the IPL model parameters generated from the SMC1 algorithm to cover the whole target posterior distributions, even the tails of the distributions.

The ESS of performing SMC1 with different numbers of particles is recorded and presented in Figure 5.3. The ESS for all parameters remain high at most stages. When ESS falls below the defined threshold $\frac{N}{2}$, the resampling step is performed. In this experiment, the ESS is dropped more frequently. This means that the resampling step in the SMC1 algorithm is carried out more and will add additional computation time. However, the ESS values for all runs have not dropped to very small value near to zero. The ESS can be improved by changing the proposal's scaling factor for the estimate variance, which has an important impact on the

performance of the SMC1 algorithm. Figure 5.4 shows the effect of using different scaling factors on improving the quality of ESS and hence the performance of the algorithm. For a small scaling factor of 0.2, the ESS is very small, ranging from 1.7 to 5. In this experiment, a scaling factor of 0.7 seems reasonable to get a larger ESS. Moreover, the ESS can also be improved by adding more stages between the prior and the posterior distribution (more intermediate distributions). As shown in Figure 5.5, by increasing the number of stages, the minimum ESS increases too. For example, in this particular case, ESS at the last stage was 2.8, 171.81 and 346.81 for the number of stages 5, 10 and 15, respectively. However, the computational time is increased as well, such as the total run time for this experience, where $n = 10$, $m = 5$ and $N = 10,000$, was 318, 682 and $1,148$ seconds for the number of stages 5, 10 and 15, respectively.

Comparison of the Point Estimates:

Table 5.1 provides the numerical results for the samples' mean resulting from M/Gibbs and SMC1 for different numbers of particles. Each method was repeated 20 times for the same dataset, and the average of the posteriors' mean was recorded in this table. In terms of the point estimates, The numerical results indicate that using a small number of particles $N = 10,000$ can provide a reasonable approximation compared to the M/Gibbs for point estimates values. However, by increasing the number of particles, the point estimates resulting from each method become almost identical and only vary in the second or third decimal places.

Summary:

Although the classical SMC method (SMC1) has been successfully applied to the 1PL model, from this experiment, we can see that the efficiency of the SMC1 algorithm depends on the user settings. First, the number of particles can affect the ESS and hence the accuracy of the approximation. Many particles will require more computational time to achieve the target distributions. Second, the variance of the proposal distribution plays a substantial effect. Therefore, it needs to be chosen carefully to result in accurate estimations. Finally, the number of intermediate distributions is also essential. For example, if the prior distribution is very far from the posterior, a large number of intermediate distributions may be required. As a result, the computational time will be expensive.

Therefore, the time cost is not only caused by the size of the data but also by the

previous setting, where these settings can affect the ESS. Hence, the re-sample step is required if the ESS is small, so extra running time costs are needed. Therefore, for educational use to estimate the students' ability in real-time, using this algorithm requires more effort and sometimes more computational time. The following section will introduce a more efficient SMC method to improve the quality of the estimation results with less computational cost.

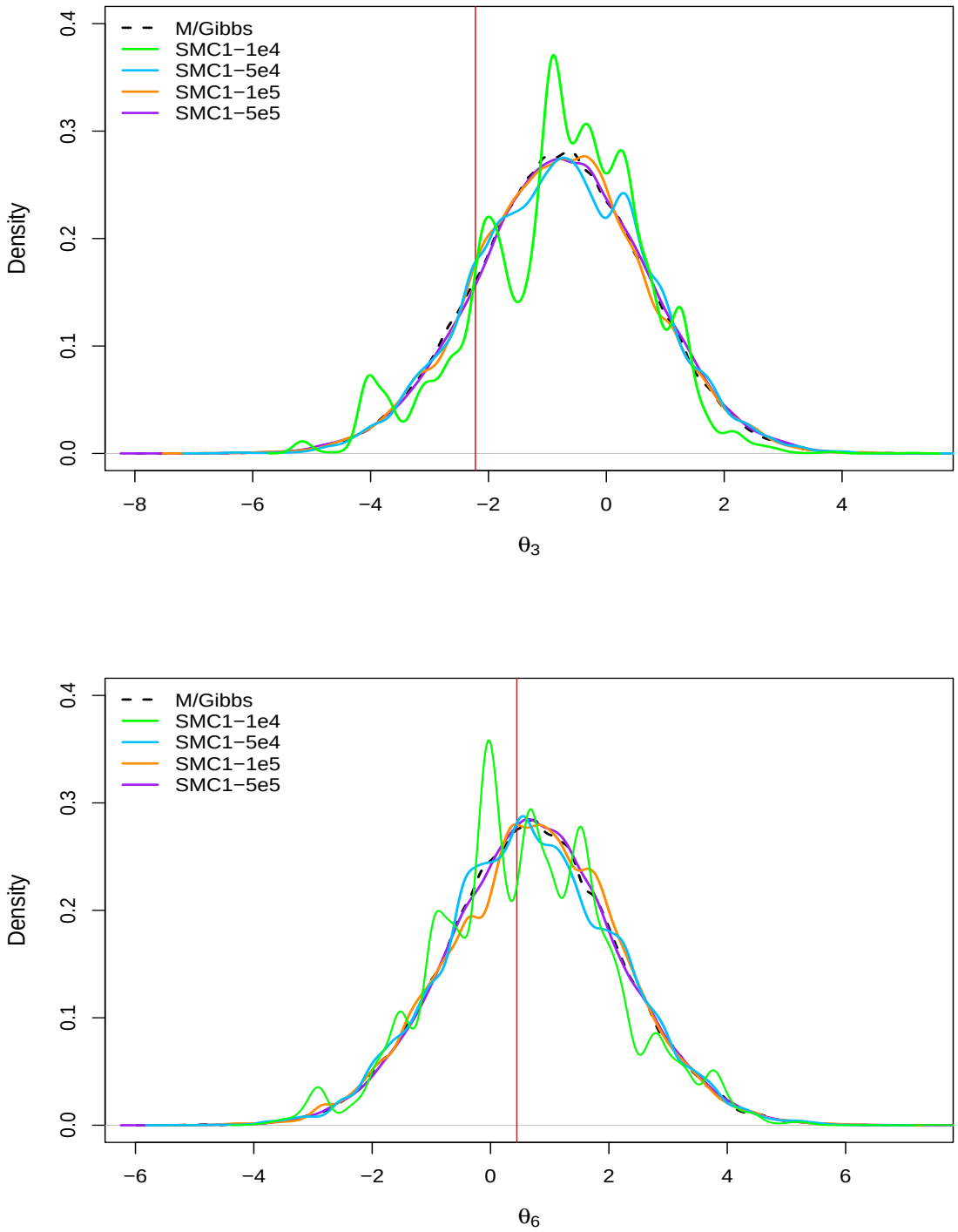


Figure 5.2: Posterior density of θ_3 and θ_6 obtained from M/Gibbs algorithm (black dashed line) compared with the approximated posteriors obtained from the SMC1 with different number of particles N. The vertical red line represents the actual value.

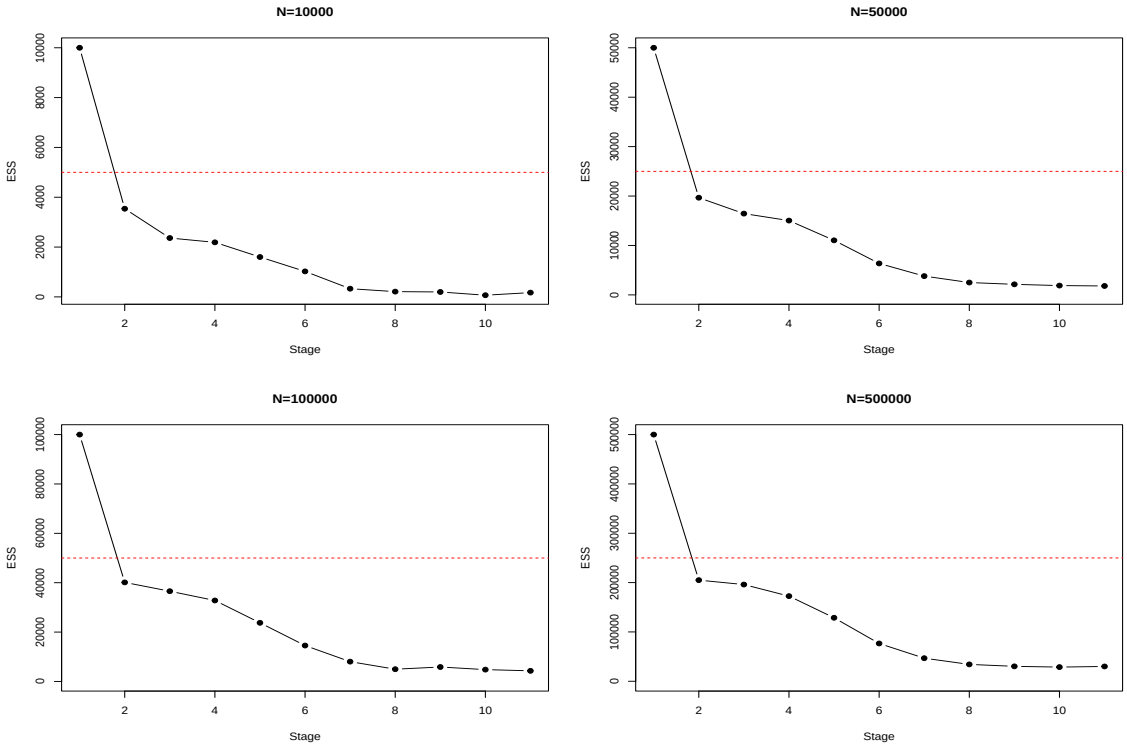


Figure 5.3: ESS values from performing the SMC1 algorithm with different number of particles N. X-axis is represented the number of SMC1 stages. The horizontal dashed red line represents the threshold ($N/2$).

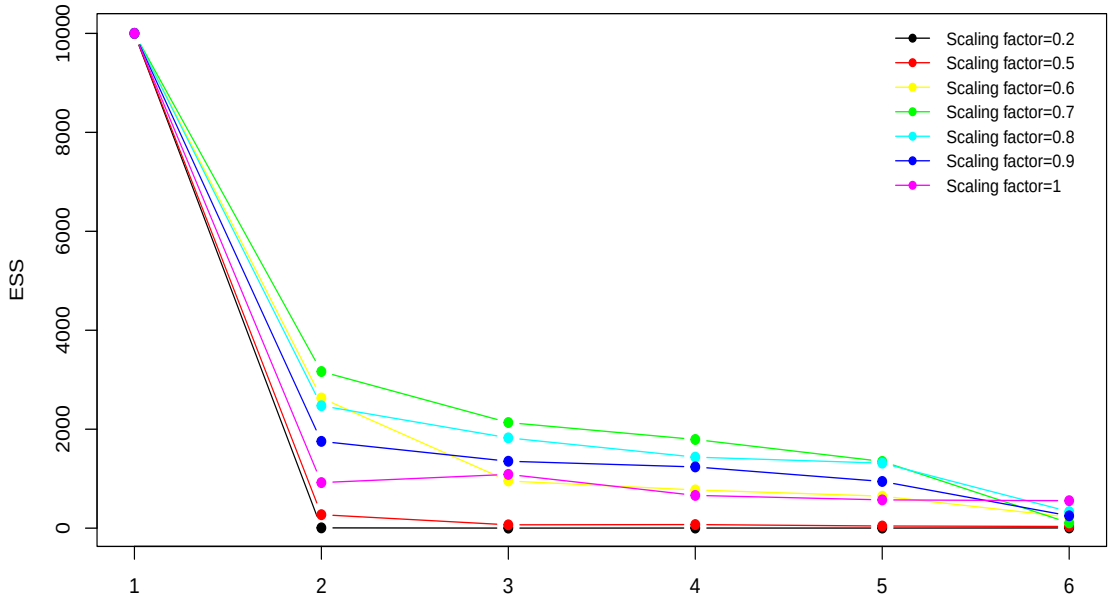


Figure 5.4: Impact of using different proposal variance scaling factors in the ESS for five intermediate (s=5) distributions and 10,000 particles.

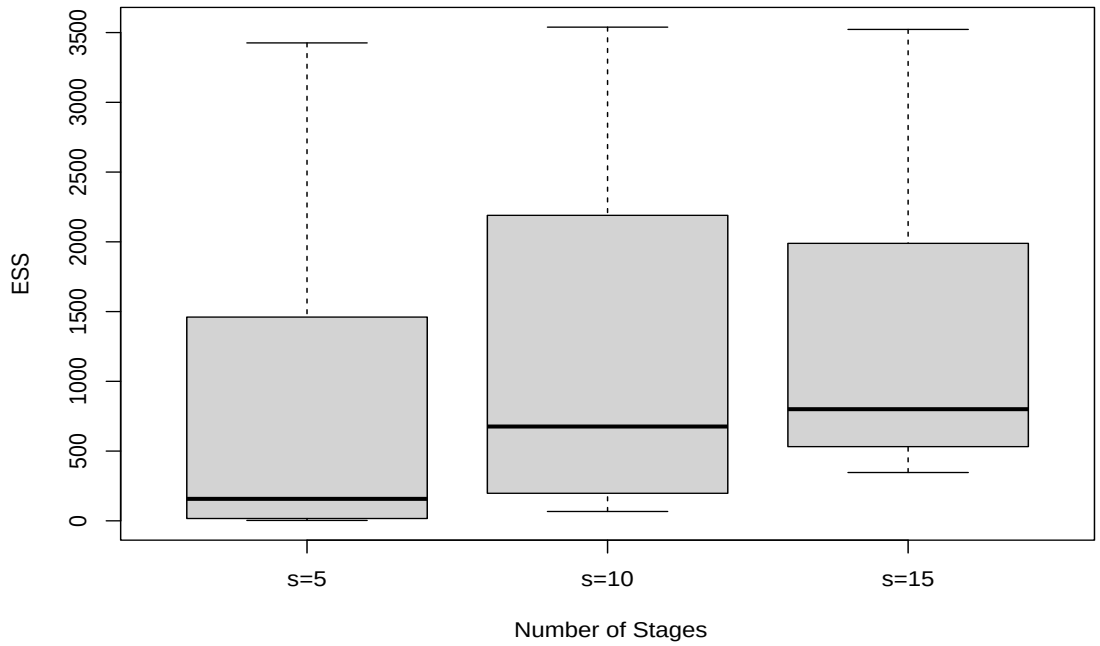


Figure 5.5: Impact of using different Number of intermediate distributions in the range of ESS.

Table 5.1: Comparison of point estimates for the ability parameter ($\boldsymbol{\theta}$) among different numbers of particles for SMC1.

			Numbers of particles			
Parameter	True value	M/Gibbs	N=1e4	N=5e4	N=1e5	N=5e5
θ_1	-4.000	-4.6029	-4.5973	-4.5743	-4.6351	-4.5982
θ_2	-4.620	-4.6117	-4.7081	-4.6684	-4.6448	-4.6072
θ_3	-0.770	-0.7598	-0.7853	-0.7453	-0.7886	-0.7537
θ_4	-0.783	-0.7642	-0.8510	-0.7430	-0.7909	-0.7681
θ_5	-0.757	-0.7607	-0.8773	-0.7691	-0.7933	-0.7670
θ_6	0.742	0.7397	0.5938	0.7081	0.7430	0.7402
θ_7	2.350	2.3678	2.1781	2.3110	2.3464	2.3552
θ_8	-0.769	-0.7535	-0.7158	-0.8081	-0.8005	-0.7677
θ_9	4.674	4.6137	4.5774	4.5772	4.5843	4.6180
θ_{10}	4.626	4.6401	4.3612	4.5655	4.5931	4.6238

5.2 Sequential Monte Carlo Samplers with Markov Chain Monte Carlo Proposals

The previous setting of classic SMC (SMC1) is less time consuming than the MCMC method. This is because we can use the most updated information in SMC1 without

having to re-run the entire procedure like MCMC methods. However, we have seen that the resampling step is frequently required, adding extra time. Moreover, in this setting, all particles need to be stored, even those not useable, and using a lot of particles will slow the running time. Also, in the classic setting, the data is added incrementally, and if the prior is very far from the posterior, many intermediate distributions are needed. Therefore, in this section, two techniques, one for the resampling step and one for introducing the data, will be added to SMC to improve the estimation's quality with a faster time. For short, this method will be denoted by SMC2.

5.2.1 Data Update and Algorithm Setting

This section will discuss the two techniques that could be used to improve the efficiency of the SMC1 method. First, an improved way of introducing the data gradually as new data arrive is presented. We will then consider one way of improving the resampling step given by 2.7.2.

Data Update

In this setting, the focus is on adding data sequentially in the likelihood instead of sampling from the sequence of intermediate distributions until getting the posterior distribution, such as $\pi_i(\boldsymbol{\theta} \mid \mathcal{D}) = p(\mathcal{D} \mid \boldsymbol{\theta})^{\tau_i} p(\boldsymbol{\theta})$, where $0 = \tau_0 \leq \cdots \leq \tau_i \leq \cdots \leq \tau_s = 1$.

The sequence of π can be introduced as the following:

$$\pi_i(\theta_i \mid \mathcal{D}) = p(\mathcal{D} \mid \theta_{1:i}) p(\theta), \qquad i = 1, \ldots, n \qquad ,$$

where $\pi_i(\theta_i \mid \mathcal{D})$ represents the posterior distribution for a current student i, and $p(\theta)$ is the prior distribution. The likelihood; $p(\mathcal{D} \mid \theta_{1:i})$ represents the information from the previous students (from 1 to $i-1$ students) until the current student i. This means that the data will be updated sequentially every time a new student answers the test. For example, in the IRT model, the data can be stored as a matrix $\boldsymbol{X}$; where

$$X_{ij} = \begin{cases} 1 & \text{if examinee i answers item j correctly} \\ 0 & \text{if examinee i answers item j incorrectly,} \end{cases}$$

and $i = 1, 2, ..., n$ (number of rows) and $j = 1, 2, ..., m$ (number of columns).

The posterior distribution for a current student will then be the product of the updated likelihood and the prior. Therefore, in the context of the SMC2, it is unsuitable to have a posterior distribution for a single student; it has to be a joint posterior distribution, updated every time a new student answers the test.

Algorithm Setting

The goal of using this method in an educational setting is to estimate the ability of students in real-time inference or a dynamic system such as students answering the test at different times. Therefore, the algorithm setting presented in this section is considered the 1PL model 3.1.1. Although the focus is on the 1PL model, the same setting can be straightforwardly applied to 2PL 3.1.2 or 3PL 3.1.3 models.

Given the 1PL model, as mentioned earlier, the likelihood can be written as:

$$L(\boldsymbol{x}|\theta_i, b_j) = \prod_{i=1}^{n}\prod_{j=1}^{m} \pi_{ij}^{x_{ij}}(1-\pi_{ij})^{1-x_{ij}},$$

The SMC2 algorithm begins by sampling $k = 1, \ldots, N$ particles from the prior distribution of the ability parameter $p(\theta)$, which is assumed to be the same for all students. Similarly, we sample N particles from the prior distributions of the questions' difficulties $p(b)$. In this setting, it is assumed that we have no prior belief about the difficulties of the questions, and hence the prior distribution is assumed to be the same for all questions. However, in most educational testing, we expect to have some knowledge about the difficulty of the questions, and then different prior distributions for each question may be required. This can also be applicable in the SMC2 algorithm.

For each particle, we assign an initial weight; $w_0 = 1$ so that $\{\theta_0^k, w_0^k\}$ is a weighted sample from the prior $p(\theta)$, and $\{b_0^k, w_0^k\}$ is a weighted sample from the prior $p(b)$.

In most applications of this method, the weighted particles from distribution π_{s-1} are used to produce particles from the distribution π_s, where s represents the stage. However, the important technique of using SMC2 is that method will use a mixture of weighted particles from all previous $\pi_{1:(i-1)}$ to produce particles from π_i. For π_1, the initial particles from the prior distributions will be used to produce the sample from the posterior distribution for a first student. This technique can be done through three different possible steps at each stage as follows:

Re-weight step: For given weighted samples $\{\theta_{i-1}^k, w_{i-1}^k\}$ and $\{b_{i-1}^k, w_{i-1}^k\}$, set the weights from π_i:

$$w_i^k = w_{i-1}^k \frac{\pi_i(\boldsymbol{\theta}^k, \boldsymbol{b}^k)}{\pi_{i-1}(\boldsymbol{\theta}^k, \boldsymbol{b}^k)},$$

where $k = 1, \ldots, N$ is the number of particles and $i = 1, \ldots, n$ is the students sequence. Then the weight need to be normalised by setting $w^k \leftarrow w^k / \sum_{k=1}^{N} w^k$.

Re-sample step: If ESS $< \frac{N}{2}$, residual resampling is carried out; see (Douc and Cappé, 2005). In this step, particles with low weights will be discarded and multiplied particles with high weights. Finally, the weights of the resampled particles are reset to 1.

Move step: In order to increase particle diversity and overcome problems such as sample impoverishment (2.7.2), some selected parameters (from θ_1 to θ_i) samples are replaced according to MCMC transition kernel density q (proposal density) such that:

$$\boldsymbol{\theta}_i^{k,*} \sim q_i(\boldsymbol{\theta}_i'^k, .),$$

where $\{\boldsymbol{\theta}_i', w_i^k\}$ is a sample from the current posterior π_i after re-weighting and (possibly) resampling. Similarly, for $\{\boldsymbol{b}_i', w_i^k\}$,

$$\boldsymbol{b}_i^{k,*} \sim q_i(\boldsymbol{b}_i'^k, .).$$

The selected parameter in the MCMC move step can be one or more. Moreover, the parameter $\boldsymbol{\theta}$ and the $\boldsymbol{b}$ are updated in two different stages. Therefore, the kernel density can be chosen differently. However, in this application, the kernel density is chosen as a normal random walk density for both parameters. The mean is the current particles for the selected parameters, and the user sets the variances. Similarly to the standard Metropolis-Hastings, for each $k = 1, \ldots, N$, we set $\boldsymbol{\theta}_i^{k,*} = \boldsymbol{\theta}_i^k$ with probability

$$\alpha\left(\boldsymbol{\theta}_i^k, \boldsymbol{\theta}_i^{k,*}\right) = \min\left(1, \frac{\pi_i(\boldsymbol{\theta}^{k,*}, \boldsymbol{b}^k)}{\pi_i(\boldsymbol{\theta}^k, \boldsymbol{b}^k)}\right)$$

The exact process is repeated for the difficulty parameter $\boldsymbol{b}$. Hence, for each $k = 1, \ldots, N$, we set $\boldsymbol{\theta}_i^{k,*} = \boldsymbol{\theta}_i^k$, we set $\boldsymbol{b}_i^{k,*} = \boldsymbol{b}_i^k$ with probability

$$\alpha\left(\boldsymbol{b}_i^k, \boldsymbol{b}_i^{k,*}\right) = \min\left(1, \frac{\pi_i(\boldsymbol{\theta}^k, \boldsymbol{b}^{k,*})}{\pi_i(\boldsymbol{\theta}^k, \boldsymbol{b}^k)}\right)$$

This MCMC movement step can be repeated several times. The resulting particles are then used in the denominator of the re-weighting step (5.2.1).

This method has wide applications outside the scope of the educational models, such as the IRT model. There are also more advanced strategies in the MCMC move step. See for example, Fan et al. (2008), Creal (2012) and Everitt et al. (2020).

The performance of the SMC2 algorithm will be investigated in the next section through a comparison study. The comparison study will be carried out between the proposed method (SMC2) and one of the MCMC methods (M/Gibbs).

5.2.2 Comparison Study

To investigate the performance of SMC2, the same dataset presented in 5.1.2 will be used. Therefore, the investigation will be conducted into a small dataset with a sample size of 10 students and 5 questions. The SMC2 algorithm will be run for different numbers of particles to check the effect of increasing the number of particles on the approximation result; $N = \{10,000, 50,000, 100,000, 500,000\}$. The comparison study will be focused on the posterior distributions and the point estimates. For the M/Gibbs method, the same results provided in section 5.1.2, for the SMC1, will be used in this comparison study as well.

Comparison Studies Result

Comparison of Distributions:

Figure 5.6 shows two different ability distributions, where the number of correct answers is 2 and 3 respectively. The figures indicate that starting with a small number of particles, $10,000$ can still provide a reasonable approximation to the parameters space compared to the posterior distribution resulting from M/Gibbs. However, the two resulting distributions from SMC2 (green line) and M/Gibbs (black dashed line) are not quite the same. As we can notice, the posterior distributions of SMC2 have larger variances and do not fully cover the peaks and the tails of the posterior distributions of M/Gibbs. That indicates higher numbers of particles are required.

Increasing the number of particles to $50,000$ has a noticeable effect on improving the shape of the distributions (blue line), where the variances of the posterior distributions generated from SMC2 become smaller. However, the peaks of the SMC2 distributions (blue line) are slightly shifted toward the right, indicating that the two posteriors' means are not quite the same yet.

Therefore, using $100,000$ numbers of particles seems reasonable in this experiment, where the two resulting distributions (yellow line and black dashed line) become identical. Also, we can see that increasing the number of particles from $100,000$ to $500,000$ has an invisible effect since the approximation posteriors generated by the SMC2 algorithm with $N = 100,000$ can cover the entire parameters space.

In terms of ESS, Figure 5.7 shows the ESS resulting from the four different number of particles. The X-axis represents the sequence of the students. The ESS follows the same scenarios for all numbers of particles, such as the lowest ESS is for the third student, and the largest ESS is for the fifth student. However, the ESS for the current student depends on the results of the previous students since we are using a mixture of the particles on the proposal density to produce samples for a current student. Therefore, if there are big differences between the previous students' abilities and a current student's ability, some of these particles will have negligible weights. Hence, the ESS will be smaller, such as in the ability's estimate of the third student is -0.78 and the ability's estimates for the first and second students is -4.62.

In this experiment, the ESS never dropped below 500. However, the ESS drooped below the defined threshold $\frac{N}{2}$ every time. Therefore, the resampling step is performed in each step.

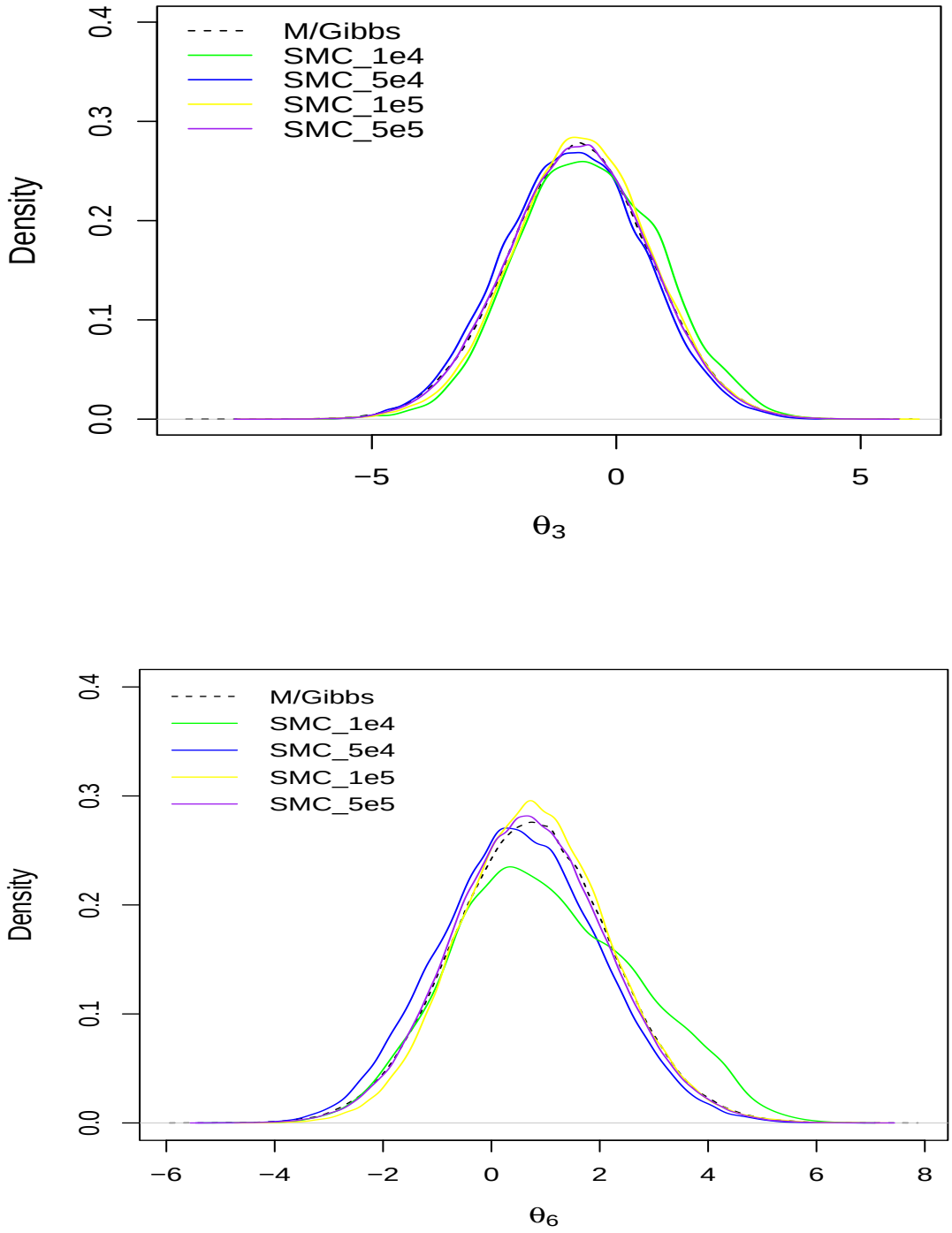


Figure 5.6: Posterior density of θ_3 and θ_6 obtained from M/Gibbs algorithm (black dashed line) compared with the approximated posteriors obtained from the SMC2 with different number of particles N.

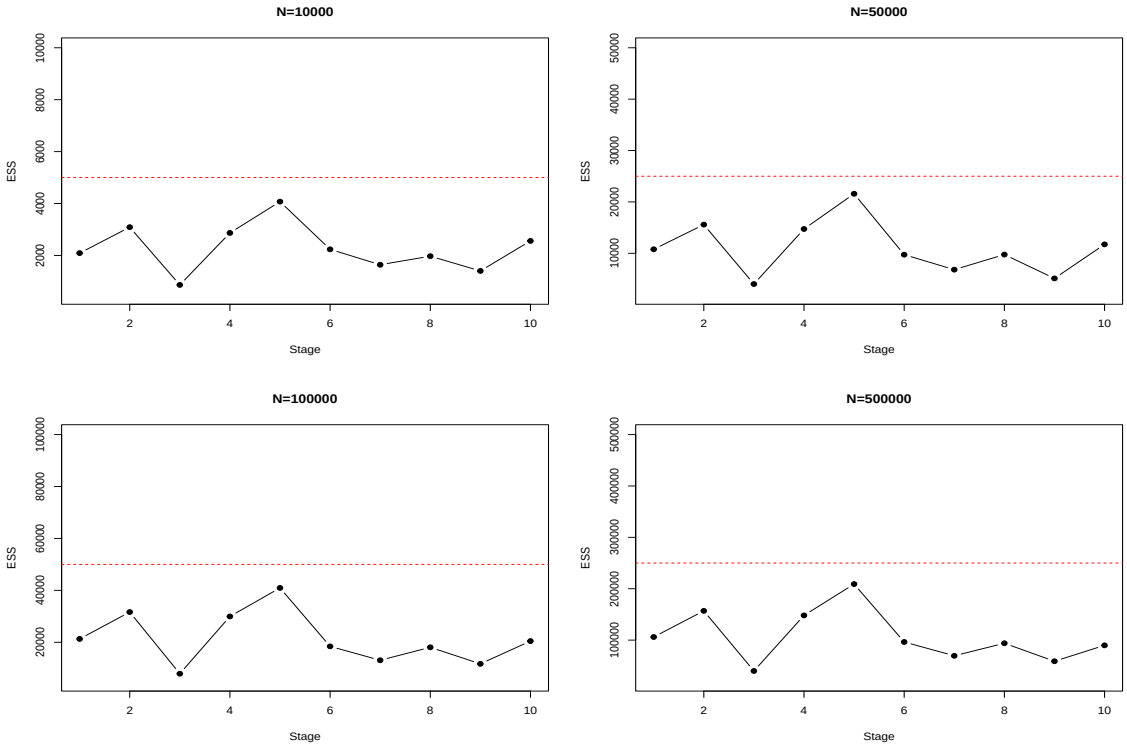


Figure 5.7: ESS values from performing the SMC2 algorithm with different numbers of particles N. The x-axis represents the sequence of students. The horizontal dashed red line represents the threshold $(N/2)$.

For further analysis of the ESS, Figures 5.8 shows ESS values from performing the SMC2 algorithm for six different datasets with $n = 10$ and $m = 5$. In each experiment, the order of the students' ability estimates is different. Hence, the ESS is affected by the previous particle results. However, in all experiments, the ESS did not drop below 500, but it fell below the defined threshold $(\frac{N}{2})$, indicating the resampling step was needed at every stage.

The ESS is also affected by the size of the dataset. Figure 5.9 presents the ESS values for four different scenarios of the simulated dataset and $10,000$ particles. For example, there are 20 students and 5 questions in the sub-figure 5.9a. The ESS values range from 932 to 4739, with a mean equal to 2607. By increasing the number of questions to 10 in sub-figure 5.9b, the minimum ESS drooped to 73. The average ESS also dropped to 1517, but the maximum ESS increased to 8113, with one outlier point. A similar scenario occurs in sub-figures 5.9c and 5.9d . For example, for a dataset of n= 30 and m= 5, the ESS range from 658 to 5486, with a mean equal to 2563. However, these values are drooped sharply by increasing the number of questions to 10 in 5.9c . Hence, the ESS values range from 29 to 5043, with a mean equal to 1555.

This result is expected since an increase in the number of questions will increase the variety of the students' abilities. Therefore, this could increase the difference between the current posterior distribution and the proposal distribution. Hence, the difference affects the weight, which will lead to small weights if the difference is large. Consequently, the ESS will be small, where $ESS = \frac{\left(\sum_{k=1}^{N} w_k\right)^2}{\sum_{K=1}^{N} w_k^2}$, and then more particles are required.

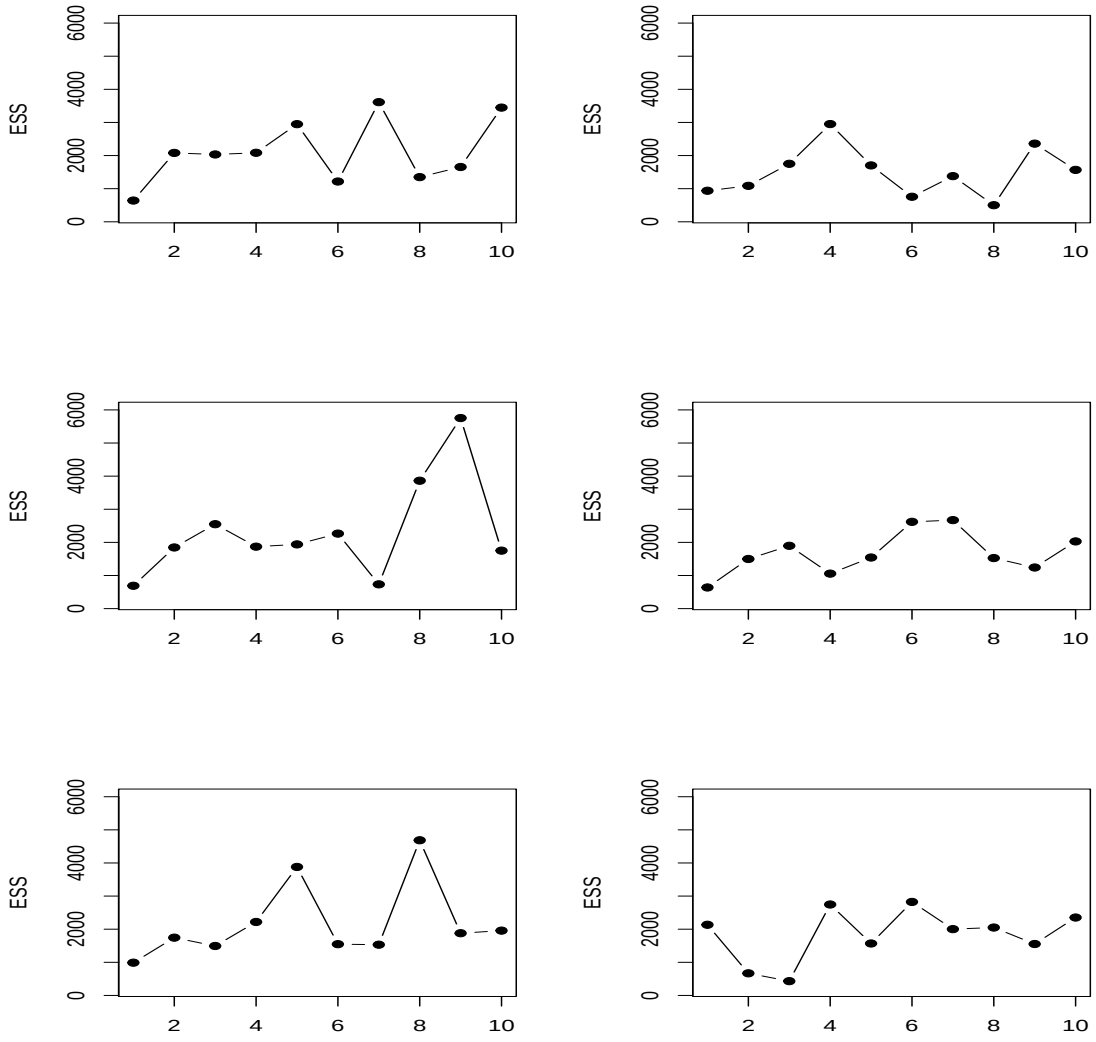
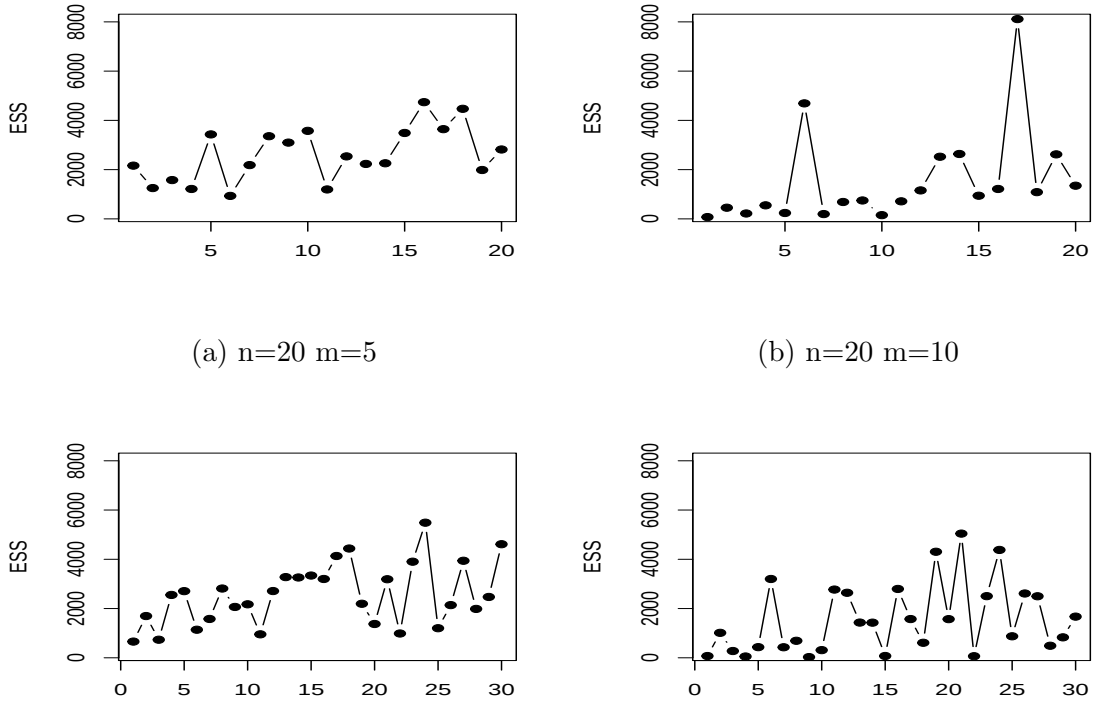


Figure 5.8: ESS values from performing the SMC2 algorithm for six different datasets with $n = 10$ and $m = 5$. The x-axis represents the sequence of students.

Adding the MCMC move step in the SMC2 re-samples procedure has an important effect on the performance of the algorithm. As explained earlier in 2.4.1, the variance of normal random walk density should be tuned carefully to control the algorithm performance and thus obtain an efficient algorithm. The objective of this study is

to obtain between 30% to 50% acceptance rate in the MCMC step .

Figure 5.10 displays a comparison of the density plot for θ_6 with different choices of the proposal distribution variance σ^2 in the MCMC step and different numbers of particles. The top panel of Figure 5.10 shows the result of the density estimate with a large choice of the variance in random walk kernel density (MCMC move step). We can see that by using a larger proposal variance, the density estimate generated by SMC2 can converge faster to the objective density estimate obtained M/Gibbs by a smaller number of particles. However, this choice will result in a smaller acceptance rate and therefore more computational time.



(a) n=20 m=5

(b) n=20 m=10

(c) n=30 m=5

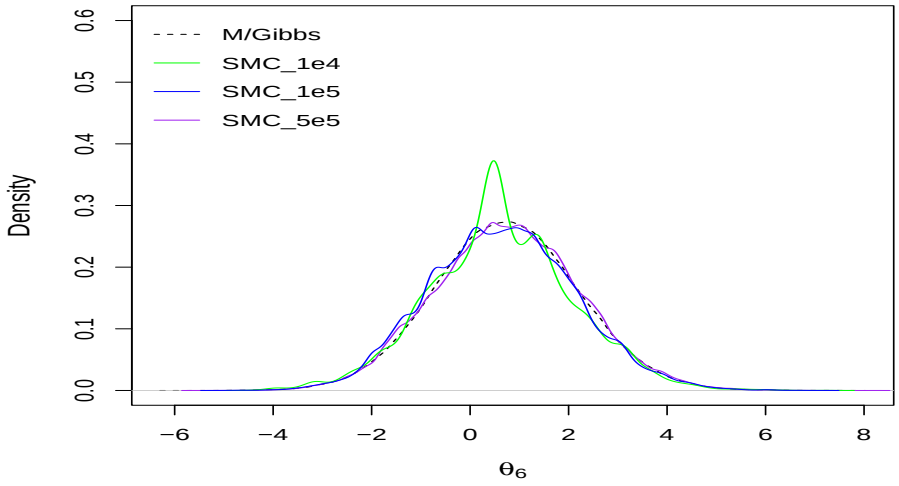
(d) n=30 m=10

Figure 5.9: ESS values from performing the SMC2 algorithm with 10000 numbers of particles N and different datasets, where n is the number of students and m is the number of questions. The X-axis represents the sequence of students.

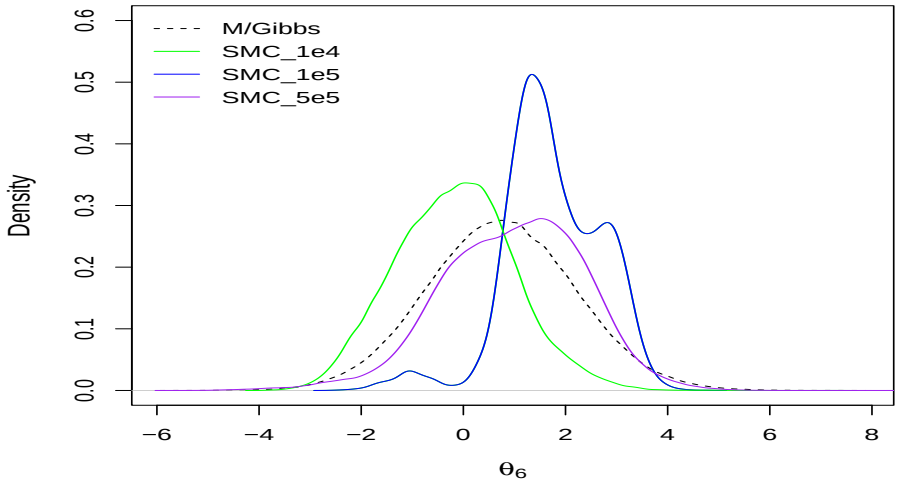
On the other hand, the bottom panel of Figure 5.10 shows the result of the density estimate with a smaller choice of the variance in the random walk kernel density. We can see clearly that using a smaller variance in the kernel density will take longer to explore the entire parameter space. As a result, a larger number of particles is required to achieve an acceptable density estimate. It is noticeable in

this plot that even with a high number of particles $N = 500,0000$, the posterior estimate has not converged to the objective density estimate obtained by M/Gibbs.

Figure 5.11 emphasises that using a smaller variance will require a higher number of particles to be able to converge to the target posterior mean obtained by M/Gibbs. However, using a larger variance will require a smaller number of particles, but in both cases will be computationally expensive. In the terms of ESS, Figure 5.12 shows that the choice of the proposal variance can also affect the value of the ESS. The figure shows that if the variance is small, the ESS will be smaller as well.

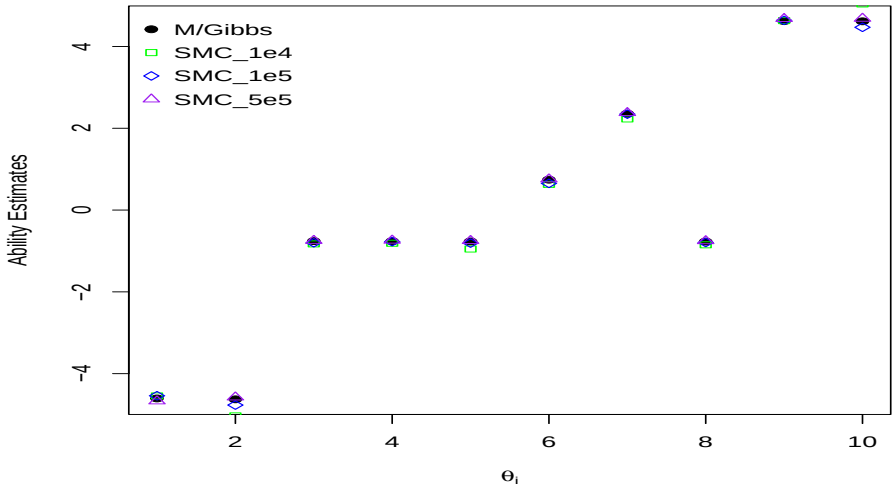


(a) The density plot of large choice of σ^2, and a small acceptance rate.

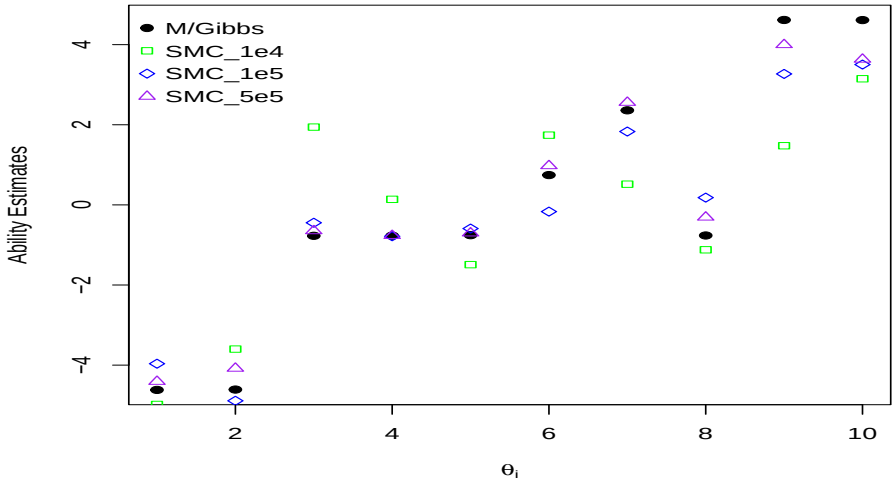


(b) The density plot of small choice of σ^2, and large acceptance rate.

Figure 5.10: Density plot of the ability parameter θ_6 with different choices of the proposal distribution variance σ^2 in the MCMC step for the SMC2 algorithm with varying numbers of particles.



Ability estimates of large choice of σ^2, and a small acceptance rate.



Ability estimates of small choice of σ^2, and large acceptance rate.

Figure 5.11: Ability point estimates with different choices of the proposal distribution variance σ^2 in MCMC step for SMC2 algorithm and different numbers of particles.

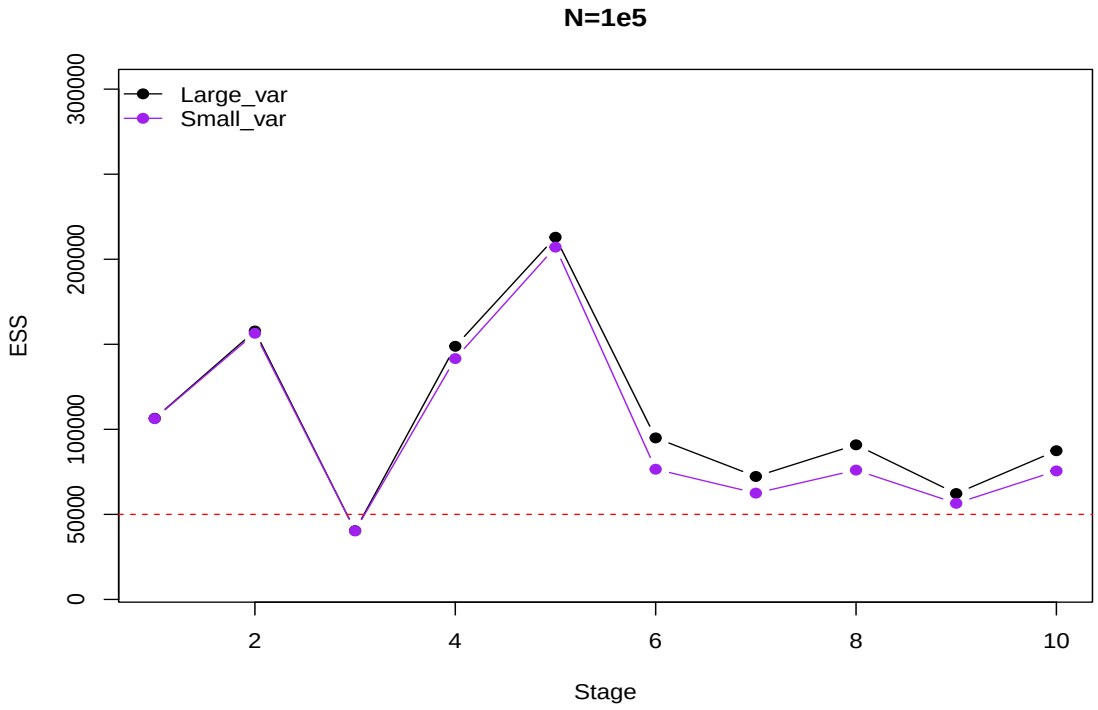


Figure 5.12: ESS values from performing the SMC2 algorithm with large and small proposal distribution variance σ^2 in MCMC step. The horizontal dashed red line represents the threshold $(N/2)$.

Comparison of the Point Estimates:

From comparing the posterior distribution resulting from SMC2 and M/Gibbs, we have seen that many factors can affect the resulting distributions, such as the variance in the proposal distribution, the number of selected parameters in the MCMC move steps and the number of particles. In the same way, the numerical results, such as the posterior mean, can be affected by the previous factors. Table 5.2 shows the numerical results for the samples' mean resulting from M/Gibbs and SMC2 for different numbers of particles. Each method was repeated 20 times for the same dataset, and the average of the posteriors' mean was recorded in this table. The proposal variance in this experiment is chosen based on several initial experiments to reach an acceptance rate between 30% to 50%. The average acceptance rate from repeating the SMC2 20 times is 45%. We can see that when $N = 500,0000$, the point estimates resulting from SMC2 become close enough to those resulting from M/Gibbs.

Table 5.2: Comparison of point estimates for the ability parameter ($\boldsymbol{\theta}$) among different numbers of particles for the SMC2.

			Numbers of particles			
Parameter	True value	M/Gibbs	N=1e4	N=5e4	N=1e5	N=5e5
θ_1	-4.000	-4.6029	-4.5063	-4.6129	-4.5873	-4.6239
θ_2	-4.620	-4.6117	-4.2773	-4.5227	-4.5972	-4.6756
θ_3	-0.770	-0.7598	-0.5458	-0.8814	-0.6906	-0.7729
θ_4	-0.783	-0.7642	-0.6802	-1.0033	-0.7619	-0.7901
θ_5	0.757	-0.7607	-0.7987	-0.9065	-0.7566	-0.7932
θ_6	0.742	0.7397	1.0148	0.5133	0.8236	0.7132
θ_7	2.350	2.3678	2.4100	1.9546	2.3862	2.2747
θ_8	-0.769	-0.7535	-0.6992	-1.0128	-0.6641	-0.7697
θ_9	4.674	4.6137	4.7833	4.3537	4.8363	4.5808
θ_{10}	4.626	4.6401	4.5133	5.0738	4.5159	4.4341

5.3 Comparison of the Computational Time

This section compares the computational time between the two SMC proposed methods; SMC1 and SMC2. The runtime for each algorithm will be recorded for a single student since both methods can be sequentially used when a new student answer the test.

The average runtime for M/Gibbs for $n = 10$, $m = 5$ and $100,000$ number of iterations is 44.0 seconds, where the algorithm was repeated 20 times. Table 5.3 represents the runtime between SMC1 and SMC2 in seconds for the different number of particles. It is clear that the classic SMC method (SMC1) is very slow compared to both MCMC and SMC2. For example, even for a small number of particles (N=10,000), it took about 6 minutes (360 seconds). The reason is that the number of intermediate stages is an essential algorithmic parameter that the user should choose, and for each stage, we need the same number of particles (10,000). Therefore, for the experiments presented in this chapter, the algorithm required $10 \times N$ particles, where 10 is the number of stages. However, if the data is increased, the number of stages should also increase to minimise the difference between the prior and the posterior distributions. Hence, this requires expensive computational time.

On the other hand, we can see that the SMC2 is less expensive than SMC1. However, we have seen that the approximation results can be improved by increasing the number of particles. Therefore, for this particular experiment, many particles were needed ($N = 500,000$) to provide a sample that could represent the posterior

distributions well. Thus, the cost time for this experiment is 248 seconds.

Table 5.3: Comparison of the computation time between SMC1 and SMC2 method for sample size of $n = 10$ and numbers of items $m = 5$, and different number of particles (N). MCMC took 44 seconds.

Time (in seconds)		
N	SMC1	SMC2
$1e4$	360	7
$5e4$	662	27
$1e5$	8713	54
$5e5$	190898	248

5.4 Summary of the Chapter

This chapter contributes to applying the sequential Monte Carlo methods to the IRT model. The SMC method has been applied using two different settings of algorithms; the classical SMC method and the SMC method with MCMC update. Both methods have been successfully applied to the 1PL model for a small dataset. The performance of each method has been compared to one of the MCMC methods; M/Gibbs. After several experiments in each algorithm, a comparative estimation to the MCMC method in terms of the shape of the posterior distributions and the point estimates has been achieved.

However, this thesis aims to find a fast and not very complex method for real-time online inference for an educational model. As we have seen from the experiments presented in this chapter, the efficiency of the SMC methods depends on the user settings, which might be difficult for real-time inference or non-professional users. Moreover, even for a small dataset, the SMC methods were not fast enough to estimate students' ability in real-time and provide immediate feedback.

However, further investigation could be made if one is interested in other uses of the IRT model, such as online rating. For example, one could investigate the performance of SMC methods in a large dataset. Moreover, the performance of the SMC can be primarily affected by the choice of the proposal density. Hence, the effect of different proposals can also be investigated. Daviet (2018) proposed a new Monte Carlo method combining the advantages of sequential Monte Carlo simulators and the Hamiltonian Monte Carlo method. Also, South et al. (2019) developed a new and efficient SMC method using independent MCMC proposals. The authors compared the normal random walk kernel, which was chosen and applied in section 6, and an independent MCMC proposal kernel. The result suggested that there are

some improvements in the efficiency of the estimation of the parameter. This method could be applied to the IRT model and compared the results with the random walk kernel.

The following chapter will consider applying the Bayesian approximation method based on the Laplace approximation, which is expected to be fast and straightforward for educational uses.

Chapter 6

Laplace Approximation Method

6.1 Laplace Approximation on the UIRT Models

The comparison study in Chapter 4 showed that MCMC techniques may be unsuitable for dynamic IRT models as they need to generate a different chain run for each posterior as new data arrives, and most of the time do not take into account the previous generations posterior. Hence, it is computationally expensive for streaming data. On the other hand, in several previous studies, the sequential Monte Carlo method (SMC) was shown to be an effective method to explore a sequence of posterior distributions. However, as we saw in Chapter 5 most of the SMC techniques become computationally expensive as the dynamic process evolves. Moreover, the efficiency of the SMC algorithms depends on the user setting, which appears to be more complex for education petitioners design.

The Laplace approximation (LA) is mathematically simple and computationally cheap for Bayesian inference. This chapter aims to explore the performance of the LA method in IRT models. This will include a comprehensive discussion of the techniques and challenges in implementing the LA method in the IRT setting and performing an extensive comparison of this approach with MCMC method on simulated data.

Algorithm settings

The general idea of Laplace approximation (LA) in Bayesian inference as explained in (2.8.1) is to take a differentiable uni-mode posterior distribution and approximate it with a normal distribution. The implementation of the LA algorithm requires the estimation of the maximum posterior through optimization and construction of a normal distribution around it, where the covariance matrix $\hat{\mathbf{\Sigma}}$ is the inverse curvature around the mode.

At the first stage, we need to find the maximum points of the log posterior distributions for each parameter. For example, for the 1PL IRT model, we need to get the maximum point for each individual's ability θ_i and the maximum point for each question's difficulty b_j. There are many different optimisation methods in the literature for finding the maximum points of the log posteriors. Some of these methods are described to be derivative-free and straightforward methods, such as the Nelder-Mead algorithm (Nelder and Mead, 1965). However, this method can be slow (Wang et al., 2019). It is known that Newton's method is very efficient and fast. Since the posteriors distribution in IRT models are differentiable, a generalisation of the classical Newton's method known as the Quasi-Newton family will be used to find the maximum points of the model parameters. The Broden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is one of the most popular quasi-Newton methods. It is a type of second-order optimisation algorithm, which means it uses a second-order derivative to find the minimum (or maximum) of an objective function. The method was developed and published simultaneously in 1970 by Broyden, Fletcher, Goldfarb and Shanno. The BFGS algorithm will be used in this thesis through the **optim** function in R.

The resulting maximum points vectors (e.g. $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{b}}$) are used to calculate the Hessian matrix ($\mathbf{H}$), which is the second derivative at these estimated points. Hence, the approximate covariance matrix $\hat{\boldsymbol{\Sigma}}$ is obtained by inverting the ($\mathbf{H}$) matrix. This calculation can be time-consuming for high-dimensional models. The problem of the high-dimensional covariance matrix will be discussed in detail in Section 6.3.

Figure 6.1 illustrates the performance of LA (contours in red lines) in approximating the log posterior of the 1PL model (blue circles) for two parameters. To explore the performance of the proposed approximation method, two simulation studies will be conducted in the next sections.

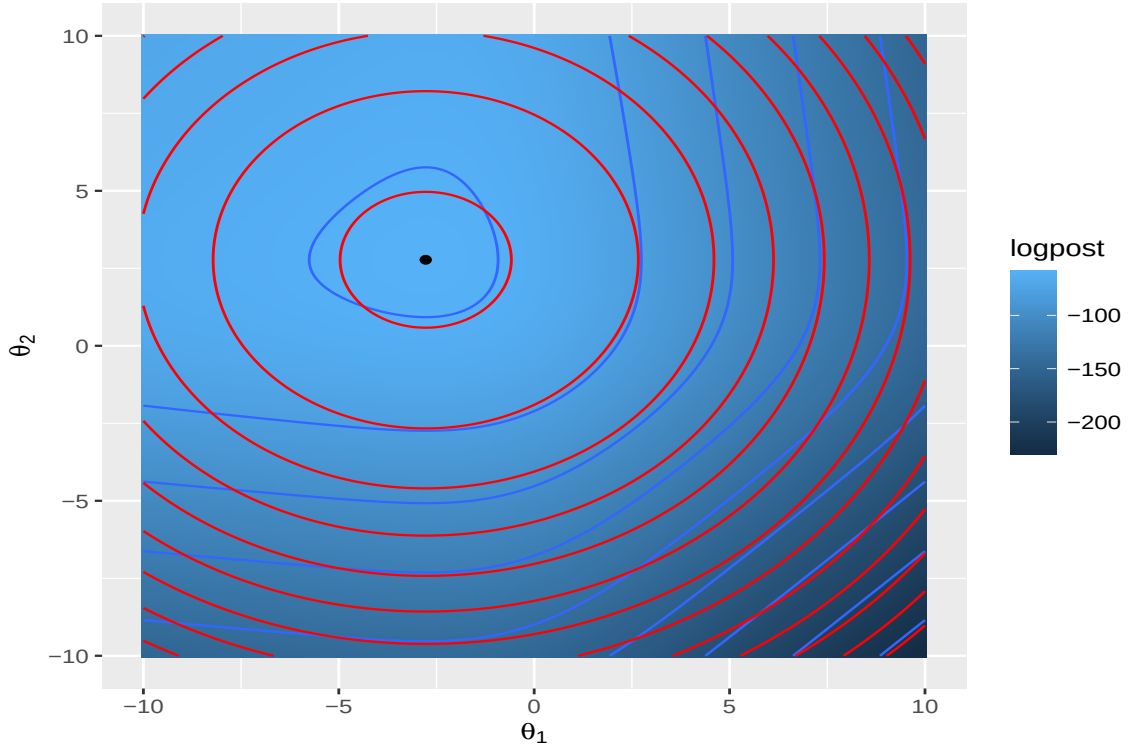


Figure 6.1: The contour plot of the log posterior (blue lines) and the approximate posterior (red lines) resulting from the LA method for θ_1 and θ_2.

6.2 Comparison Studies

The shape of the posterior distribution in IRT scenarios has been investigated in Chapter 4 using some MCMC methods such as Hamiltonian Monte Carlo (HMC) and Metropolis within Gibbs samplers (M/Gibbs). From these previous experiences, it is found that the posterior distributions for model parameters are uni-modal, and hence LA is appropriate for the IRT model.

In this section, simulation studies are designed and carried out to compare the performance of the Laplace approximation method presented above to the estimation method M/Gibbs. The main objective of the comparison study is to investigate the performance of the LA in different scenarios; small, moderate and large data sets for the 1PL IRT model.

When it comes to determining the prior distributions on the model parameters in these experiments, it is assumed that there is a lack of information about the difficulty of the questions and the ability level of the students. Therefore, all students are given the same prior distributions, and the same prior distribution is given to all questions.

Comparison Criterion

The main objective of conducting this comparison study is to evaluate the accuracy of using the Laplace approximation method. For this purpose, the posterior distributions generated from M/Gibbs will be used to compare the approximation posterior distributions resulting from the LA method. Various criteria methods will be used to assess the differences between the posteriors (resulting from the M/Gibbs) and approximated posteriors (resulting from the LA). Some of the suggested methods are mentioned below:

Comparison of Distributions:

- The density plot of the posterior distributions generated from M/Gibbs will be compared visually to the density plot of the approximate posterior distributions obtained from the LA methods.
- The **Jensen-Shannon divergence (JSD)** (Menéndez et al., 1997) is one method that is used to measure the difference between two probability distributions over the same variable. It will be used here to quantify the discrepancy between the resulting target posterior distributions (M/Gibbs) and approximate (LA) posterior distributions that are obtained from each algorithm, which is defined as:

$$\text{JSD}\left(\pi_i \mid \pi_i^*\right) = \frac{1}{2}\sum_i \left(\pi_i \log \frac{\pi_i}{\frac{1}{2}\left(\pi_i + \pi_i^*\right)}\right) + \frac{1}{2}\sum_i \left(\pi_i^* \log \frac{\pi_i^*}{\frac{1}{2}\left(\pi_i + \pi_i^*\right)}\right),$$

  where the term π_i^* indicates the approximate density obtained from LA and π_i represents the density obtained from the MCMC. The index i represents the parameter (e.g. for $\boldsymbol{\theta}$, given n is the total number of students, $i = 1, ..., n$). The value of the JSD describe the amount of divergence between two distributions. The higher the JSD value, the greater the discrepancy between the two distributions. The original Jensen-Shannon divergence (JSD) ranges between [0, 1], where 0 means the two distributions are identical, and 1 is strongly different. See Fuglede and Topsoe (2004) and Nielsen (2019) for more explanation of this method.

Comparison of the Point Estimates:

- The idea of a rank distance measure can be used to determine whether there is a correspondence between two measurements. For example, one can look at

the order of the students' point estimated ability resulting from both methods, and hence look at the distance measure between the two orders.

Kendall's $\boldsymbol{\tau}$ (Noether, 1967) is one method that can be used to compare evaluation measures. For two different rankings of the same size n, this method counts the number of pairs agreed in the same order in each of the two orders and which are not agreed in reverse order. If C is the number of agreements, and D is the number of disagreements,

$$\tau = \frac{C - D}{C + D}.$$

The value of τ ranges from -1 to 1, where 1 means the two rankings are identical, and -1 means one is opposite of the other. Each τ value can be mapped directly to a corresponding percentage. If the value of τ is 0, this means that 50% of the pairs are identical (concordant) and 50% discordant. The value of τ will be equal to 1 if the two lists are identical, and $n(n-1)/2$ if one list is the reverse of the other. Kendall's τ distance is the number of discordant pairs D. The greater the distance, the more different the two lists are. The idea of the rank order has been used several times in the literature to measure the difference between the abilities estimate resulting from item response theory (IRT) and classical test theory (CTT); see for examples Zaman et al. (2008) and Binh and Duy (2016). For more details about Kendall's τ method, see Abdi (2007).

- There are two common methods that can be used to measure the difference between true parameters and their estimated values: **Bias**, which explains for systematic error, and root mean square error **(RMSE)**, which indicates the overall variability in estimation error for point estimates. Therefore, the quality of the point estimates for the ability parameters can be evaluated in terms of the closeness between the estimated and true values using the average estimated bias and RMSE. These values can be calculated using the following equations:

$$\text{Bias}(\hat{\boldsymbol{\theta}}) = \frac{1}{R}\frac{1}{N}\sum_{n=1}^{N}\left(\hat{\theta}_{nr} - \theta_{nr}\right),$$

and

$$\text{RMSE}(\hat{\boldsymbol{\theta}}) = \frac{1}{R}\sum_{r=1}^{R}\sqrt{\frac{\sum_{n=1}^{N}\left(\hat{\theta}_{nr} - \theta_{nr}\right)^2}{N}},$$

where R is the number of replications (number of repeated simulated datasets), θ_{nr} is the true ability parameter for student n, $\hat{\theta}_{nr}$ is its estimated value in the rth replication for student n, where N is the total number of parameters. For an unbiased estimator, the mean bias is expected to be close to zero. The reason is that we can get both positive and negative deviations from the true parameter values and thus cancel each other out. The value of the RMSE indicates how widely the estimates are spread around the true parameter. The smaller the RMSE, the closer the point estimates to the true parameter values and hence better estimates.

- **Credible interval** (CI) is used to summarise the uncertainty about the estimates parameter. As the Bayesian inference returns a posterior distribution, the credible interval is just a range that contains a certain percentage of potential values. For example, the 95% credible interval is simply the central portion of the posterior distribution that contains 95% of the values. For an in-depth explanation and interpretation of the credible interval, see Hespanhol et al. (2019).

The simulation study data and framework will be explained in detail in the following sections, and the results will be illustrated.

6.2.1 Comparison Study for 1PL

In the literature, there are still concerns about the accuracy of the parameter estimation for small samples. Finch and French (2019) discussed a comparison of estimation techniques for IRT models with small samples. Their result suggested that under many small sample sizes (n = 25, 50, 100, 250, 500 and 1000), MCMC estimation methods can provide greater accuracy compared to other methods. However, their comparison study only focused on item parameters estimation.

Current recommendations in the literature are generally for samples of at least 200 to 300 for the Rasch model (Chen et al., 2014), and this increase to 500 in 2PL model De Ayala (2009). A comprehensive study of the effects of test length and sample size to estimate item parameters accurately in the unidimensional binary IRT models can be found in Sahin and Anil (2017). The comparison studies in this section aim to investigate the performance of the LA method on three levels of the datasets; small, moderate and relatively large. Also, the investigation will consider a variety of test lengths from short $m = 10$ to large test $m = 100$.

Simulated Data

The data in this setting will be represented as a matrix $\boldsymbol{X}$; where

$$X_{ij} = \begin{cases} 1 & \text{if examinee i answer item j correctly} \\ 0 & \text{if examinee i answer item j incorrectly,} \end{cases}$$

and $i = 1, 2, ..., n$ (number of rows) and $j = 1, 2, ..., m$ (number of columns). In this setting, the questions or items measure the same skill (unidimensional ability), and the examinees are assumed to answer all questions.

Therefore, $X_{ij} \sim Bernoulli(\pi_{ij})$ where $logit(\pi_{ij}) = (\theta_i - b_j)$ which give the likelihood of this model as;

$$L(\boldsymbol{x}|\theta_i, b_j) = \prod_{i=1}^{n}\prod_{j=1}^{m} \pi_{ij}^{x_{ij}}(1-\pi_{ij})^{1-x_{ij}}$$

The $\theta_i's$ range uniformity from -4 to 4, $b_j's$ from -2 to 2, as these ranges suggested by DeMars (2010). Both parameters $\boldsymbol{\theta}$ and $\boldsymbol{b}$ are centred around zero.

This experiment will be run for one parameter logistic model (1PL) with binary responses (correct answer=1, incorrect answer=0) and unidimensional ability (explained in 3.1.1). The first comparison study for 1PL will investigate the performance of the LA in a small sample size $n = 30$ and test of length $m = 10$. The second study will consider the case of moderate data set, where n=200 and the test length $m = 10$. The final comparison study for the 1PL model will assume a larger data set with a sample size $n = 600$ and the test length $m = 10$. In further analysis, the effect of increasing the test length to 30, 50, 70 and 100 will be investigated and compared to M/Gibbs results. The summary of these simulated datasets is described in Table 6.1.

Table 6.1: Simulated Data for 1PL Model

Variable	Setting
Study	1
Number of Examinees	n=30
Number of Items	m=10
Study	2
Number of Examinees	n=300
Number of Items	m=10
Study	3
Number of Examinees	n=600
Number of Items	m=10

Comparison Studies Result

This section presents the results of the comparison study between the MCMC estimation method (M/Gibbs) described in 4.3.1 and the approximate Laplace method. MCMC is used to produce a target inference, and the performance of the approximate approach (LA) on the 1PL model is illustrated and then compared with the MCMC inference.

The proposed inference methods will be applied to the simulated datasets listed in Table 6.1. The MCMC method will be run for a $100,000$ iterations, and the initial 2,000 iterations will be discarded (burn-in). After that, the average of the posterior distribution for each parameter is used as the point estimate. The comparison study will be focused on the accuracy of the posterior distribution approximation resulting from LA. Hence, different numerical and graphical diagnostics will be discussed in this section to assess the quality of the estimates provided by the two inference methods.

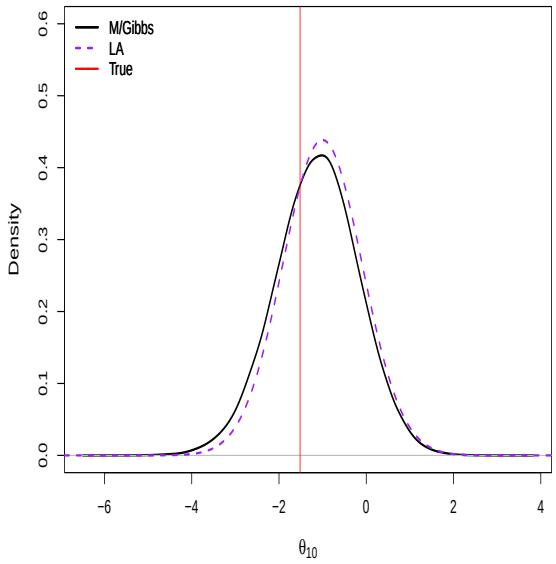
Results of Study 1

Comparison of Distributions:

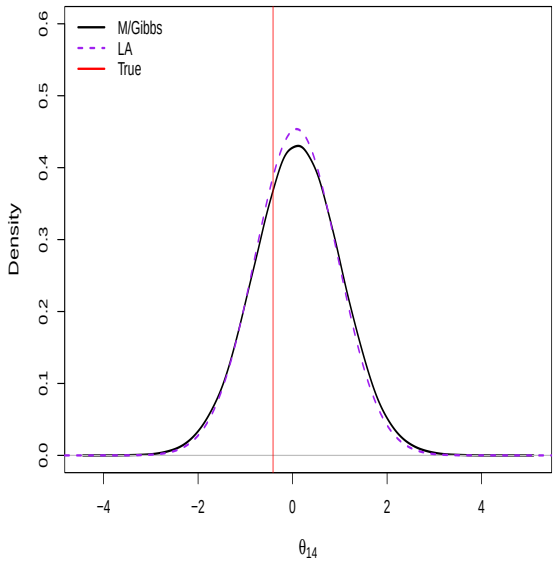
Figure 6.2 shows the approximated posterior resulting from the LA and the posterior resulting from M/Gibbs. The figure presents three randomly selected abilities of different levels. We can see that LA provide similar estimates of the abilities parameter, with the posterior mode closely matching M/Gibbs for all three ability level. However, due to the fact that the posterior densities resulting from M/Gibbs are not completely symmetric, the mode of the two densities are not precisely in the same place. When the density of the posterior resulting from M/Gibbs is almost symmetric, such as θ_{14}, the two posteriors' modes become almost identical. Moreover, we can see that the shape and the highest of the densities are almost the same, suggesting that the approximate posterior distributions generated from LA explore the proper parameters space well and in a similar way to MCMC.

The results of the JSD divergences for each student's ability parameter are recorded and visualised in Figure 6.3 to measure the dissimilarity between the resulting target (M/Gibbs) and approximate (LA) posterior distributions that are obtained from each method. The average JSD values in these experiments range between 0.005 to 0.24. The most largest difference (JSD values) between the two posterior distributions resulting from each method appears for very high/low ability students, such as student 28 with a very high ability and student 2 with a very low

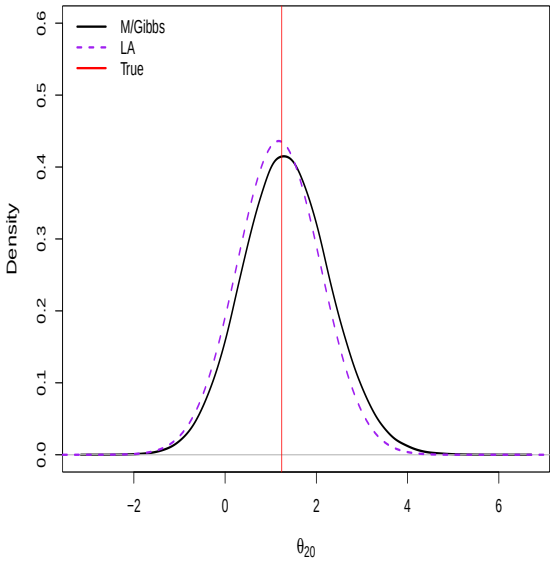
ability. However, this maximum value of 0.24 is relatively small since the original JSD ranges between 0 and 1, where 0 means the two distributions are identical, and 1 means strongly different.



(a) Number of Correct Answers= 3



(b) Number of Correct Answers= 5



(c) Number of Correct Answers = 7

Figure 6.2: Posterior density plots for M/Gibbs and LA methods of selected examinees' abilities with different numbers of correct answers for sample size $n = 30$ and $m = 10$.

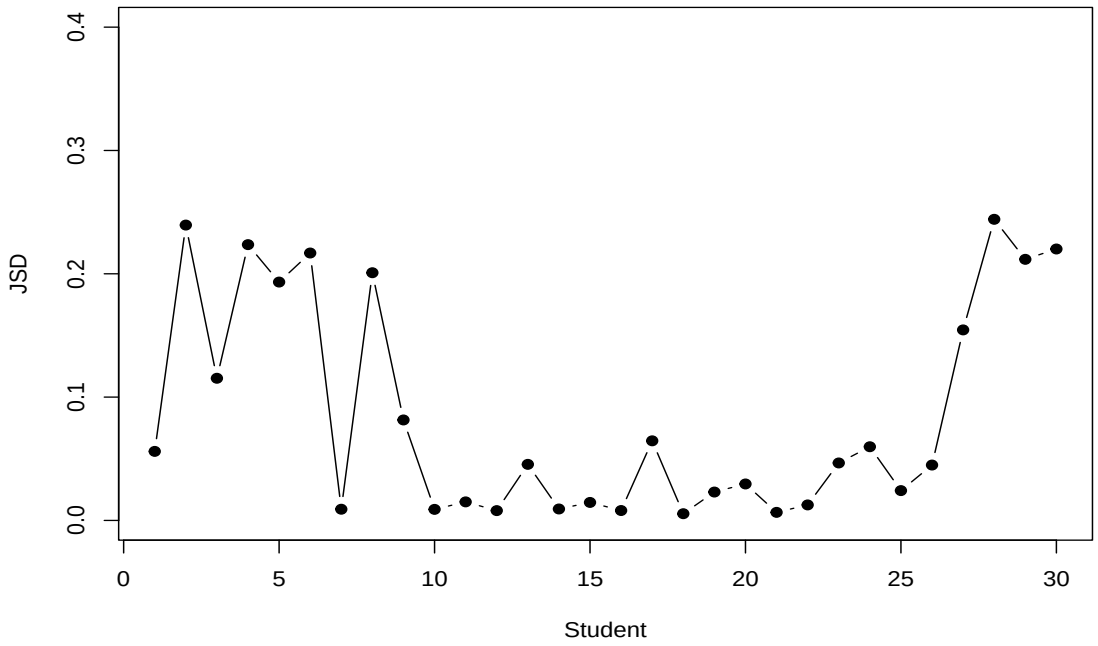


Figure 6.3: Jensen-Shannon divergence (JSD) for each student's ability parameter obtained from M/Gibbs and LA for sample size $n = 30$ and $m = 10$.

Comparison of the Point Estimates:

The 95% credible intervals for all students' abilities estimates are calculated and visualised in Figure 6.4 to compare the uncertainty (interval width) resulting from the two methods. Due to the fact that the posterior distributions resulting from the LA method (black line) are symmetric, distance from the centre to the lower and upper credible interval are equal. However, this is not the case of the posterior distributions resulting from M/Gibbs. Therefore, we can notice some differences between the two intervals range results from each method. Moreover, as we have seen before that the most biggest difference between the two posteriors resulting from each method appears for high/low abilities; the credible intervals are also quite wide for high/low abilities. Although the credible intervals resulting from the LA are wider for these students, the posteriors' means (black circles) are lie inside the M/Gibbs credible intervals and even very close to the M/Gibbs posteriors' means (red circles). For the moderate students' abilities, such as students from 10 to 25, the two credible intervals seem to be very close to each other. From a practical perspective, the accuracy of the point estimates is more critical than the interval width. In other words, teachers are usually more curious about the ability estimates of the students. Hence, for the usual practical view, ability estimates are used to rank students. However, the interval width can give an idea of how much overlap may

be in these estimates. So, for example, if the intervals are very wide, we may have a lot of uncertainty about these estimates, and the ranking may not be meaningful. Section 6.4 will discuss using the interval width to estimate the students' abilities in a dynamic IRT model.

Figure 6.5 displays the comparison of the point estimates of the examinees' abilities resulting from the M/Gibbs and LA (y-axis) versus actual values (x-axis). As we can see in this figure, the point estimates align very closely for abilities ranging between -2 to 2. However, the point estimates varied slightly outside this range for very high/low abilities. The correlation between the abilities estimates and the actual values is 0.98 for both methods and 0.99 between the point estimates resulting from the two methods, where we can see in Figure 6.6 the strong relationship between the two sets of ability estimates. However, it is noted that M/Gibbs overestimated high abilities such as the ability estimate of students 27, 28, 29 and 30, and it underestimated low abilities such as the ability estimate of students 2, 4, 5, 6 and 8. The average absolute difference between the points estimates resulting from each method is 0.37, and the absolute maximum is 0.85, which occurs between very high/low abilities.

To measure the accuracy of the estimated methods and ensure the findings in the current comparison study, the simulated data in Table 6.1 for study 1 is repeated 20 times with the same conditions and sample size $n = 30$ and test of length $m = 10$. Repeating the simulated data will result in a different pattern of students' answers and hence a different order for the true abilities. The average results of the bias and the root mean square error (RMSE) are calculated and presented in Table 6.2. The result showed that the average bias by both methods was considerably small. However, bias under the LA estimation is slightly smaller (-0.004) than bias under M/Gibbs (-0.005). In terms of average RSME, M/Gibbs produced a greater value (1.04) than LA (0.84). In terms of bias concerning students' abilities, results demonstrated that the larger values are for students with more extreme parameter values (e.g. around -3 and 3) for each estimation technique.

The average value of Kendall's τ rank correlation between the point estimates resulting from both methods and the actual values are also presented in Table 6.2. This method evaluates the degree of similarity between the ranking of the true values set and the ability estimates set resulting from each method. The value of the Kendall rank (τ) between the order given by both methods is large 0.80 and 0.86 for the M/Gibbs and the LA, respectively. These large values of (τ) indicate that the two methods the rank order resulting from abilities estimates are very close to the

true abilities order. However, we can see that the LA method has slightly stronger order abilities than the M/Gibbs. Moreover, The Kendall's (τ) value between the rank order of the abilities given by these two methods is 0.98, indicating that the two methods strongly agree on evaluating the order of the students' abilities.

In comparing the point estimates for the ability parameters, the measurement values (bias, RMSE and Kendall's τ) indicate that the LA method provides a slightly more accurate estimation than M/Gibbs.

The comparison between the point estimates of the difficulty parameters resulting from each method and the true values are presented in Table 6.3. The result shows that both methods are very similar in estimating the difficulty parameters. The small positive bias, 0.02 and 0.01 for M/Gibbs and LA, respectively, indicates that both methods slightly overestimated the true parameter values. We can see that RMSE are pretty small for both methods. Moreover, the difference between RMSE resulting from both methods for the point estimates of the difficulty parameters is smaller than RMSE between the two methods for the point estimates of the ability parameters. The average Kendall's τ value between point estimates resulting from the methods and the true values are 0.84, indicating that the order of the difficulty of the question estimates are strongly similar to the true order of the question's difficulties. The average Kendall's τ value between the rank order of the abilities given by these two methods is 0.95, indicating that the two methods strongly agree on evaluating the questions' difficulties.

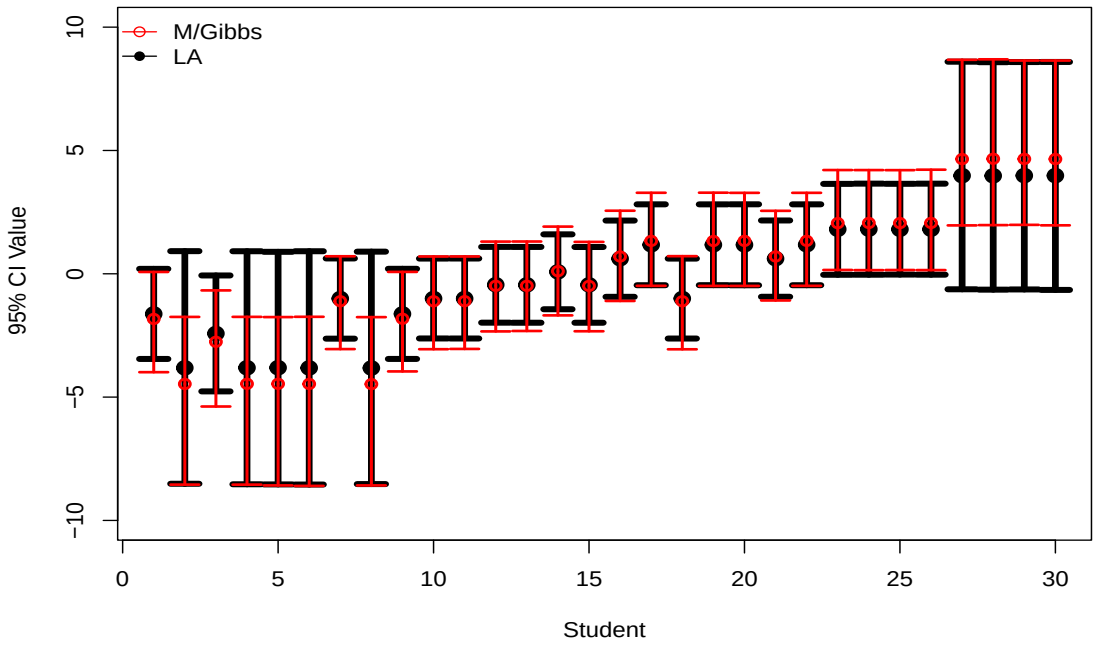


Figure 6.4: Posterior means and 95% credible intervals (CI) of the point estimates resulting from M/Gibbs and approximation method LA for sample size $n = 30$ and $m = 10$.

The number of items can influence the difficulty and ability parameters estimation result. Researchers have investigated the effect of sample size and test length on parameter estimations in the literature. For example, Uyigue and Orheruata (2019) discussed the impact of the test length and sample size for difficulty parameter estimation in IRT and compared the result to some published works. Their result recommended that the sample size, n, should be at least 1000 for a test of length 10 to have high accuracy of difficulty parameters estimation. The following section will consider the impact of increasing the number of questions (test length) in the accuracy of estimating the ability and difficulty parameters.

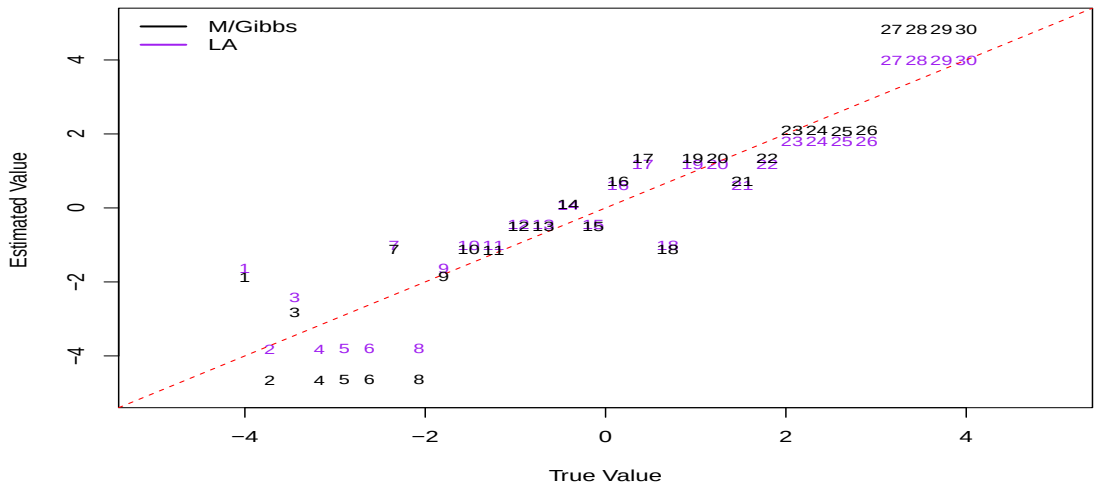


Figure 6.5: Point estimates of the examinees' abilities resulting from the M/Gibbs, and LA versus true values for sample size $n = 30$ and $m = 10$. The red line illustrates the quality line.

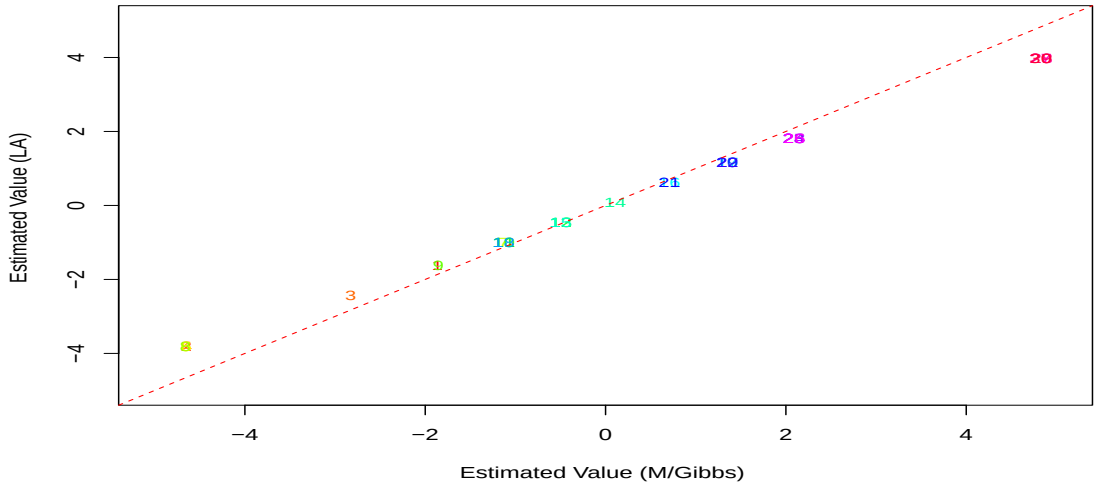


Figure 6.6: Point estimates of the examinees' abilities resulting from the M/Gibbs versus LA for sample size $n = 30$ and $m = 10$. The red line illustrates the quality line.

Table 6.2: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$, averaged across 20 different simulated data sets with sample size $n = 30$ and $m = 10$.

Method	Bias	RMSE	Kendall's τ
MCMC	-0.005	1.04	0.80
LA	-0.004	0.85	0.86

Table 6.3: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$, averaged across 20 different simulated data sets with sample size $n = 30$ and $m = 10$.

Method	Bias	RMSE	Kendall's τ
MCMC	0.02	0.57	0.84
LA	0.01	0.50	0.84

Further Analysis

This section considers the effect of increasing the number of questions from 10 to 30, 50, 70 and 100 in the accuracy of the estimations for a small sample size $n = 30$. These values are identified to represent a variety of test lengths, from short (10 items) to long (100 items). The results of the Laplace approximation method will be compared to the M/Gibbs method. The comparison will be implemented with respect to points estimates, and the accuracy of the results will be examined using the previous measurements; bias, RMSE and Kendall's τ.

There are four different experiments, where every test length is considered one experiment. The simulated data is repeated 20 times for each experiment with the same conditions and sample size $n = 30$. The average bias, RMSE and Kendall's τ for the abilities point estimates resulting from M/Gibbs and LA for each experiment is shown in Table 6.4.

Table 6.4: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$ with sample size $n = 30$ and a different number of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
30	MCMC	-0.006	0.75	0.88
	LA	-0.007	0.58	0.90
50	MCMC	0.015	0.63	0.92
	LA	0.013	0.50	0.93
70	MCMC	-0.03	0.57	0.93
	LA	-0.02	0.44	0.94
100	MCMC	0.017	0.48	0.95
	LA	0.015	0.37	0.95

Table 6.5: Average Kendall's τ values between the point estimates of the students abilities resulting from LA and M/Gibbs for a sample size $n = 30$ and different numbers of items.

Number of Items	Kendall's τ
10	0.98
30	0.98
50	0.99
70	0.99
100	100

Table 6.6: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$ for sample size $n = 30$ and different numbers of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
30	MCMC	0.009	0.58	0.76
	LA	0.006	0.53	0.77
50	MCMC	-0.009	0.56	0.75
	LA	-0.008	0.52	0.77
70	MCMC	0.014	0.58	0.74
	LA	0.013	0.54	0.77
100	MCMC	-0.005	0.57	0.74
	LA	-0.004	0.54	0.77

Table 6.7: Average Kendall's τ values between the point estimates of the difficulty of the questions $\boldsymbol{b}$ resulting from LA and M/Gibbs for a sample size $n = 30$ and different numbers of items.

Number of Items	Kendall's τ
10	0.95
30	0.95
50	0.96
70	0.97
100	0.97

From the results in Table6.4, we can see that increasing the number of questions leads to a slight increase in the average bias values. Furthermore, the increase in the bias values is more noticeable when the number of questions is larger than the number of sample size ($n = 30$). However, the average bias values are still quite small for all four experiments.

The average RMSE values dropped from 1.04 and 0.85 in the shorter test of length 10 to 0.75 and 0.58 in the longer test of length 30 for M/Gibbs and LA, respectively. The average RMSE values are reduced noticeably by adding more

questions for both methods. We can also notice that the RMSE resulting from LA are smaller than RMSE resulting from M/Gibbs. The average RMSE resulting from LA is approximately 0.14 smaller than RMSE resulting from M/Gibbs. This may imply that LA is more accurate in estimating ability parameters than M/Gibbs.

The average Kendall's τ values are also increased by adding more questions for both methods. For a short test length ($m = 30$), LA appears to order the ability point estimates more similar to the true values than M/Gibbs, where the average Kendall's τ for LA is 0.90 and 0.88 for M/Gibbs. From Table 6.5, we can see that the difference between the two methods in ordering the ability point estimates dropped gradually as we added more questions. The two sets of orders become identical for the test of length 100.

Regarding difficulty parameter $\boldsymbol{b}$, Table 6.6 shows that the bias values resulting from M/Gibbs and LA dropped from 0.01 and 0.02 for a test of length 10 to 0.009 and 0.006 for a test of length 30. However, the average bias values have no considerable changes by increasing the questions from 30 to 100. Also, RMSE values are not noticeably affected by increasing the test length. The RMSE resulting from LA is approximately smaller than M/Gibss by 0.04.

The Kendall's τ values between the methods and the true values dropped from 0.84 for a test of length 10 to 0.76 (M/Gibbs) and 0.77 (LA) for a test of length 30. These values decreased slightly by increasing the test length. However, from Table 6.6, we can see that Kendall's τ values between the two methods increase by adding more questions. Kendall's τ values indicate that increasing the test length for a small sample size $n = 30$ results in the less accurate ordering of the difficulty of the point estimates compared to the true values for both methods. However, the LA is approximately better in ordering the difficulty estimates of the questions than M/Gibbs by 0.03.

Table 6.8: Comparison of the computation time between M/Gibbs method and LA method for sample size $n = 30$ and different numbers of items.

	Time (in seconds)	
Number of Items	Gibbs/M	LA
10	120	0.01
30	195	0.02
50	269	0.03
70	339	0.04
100	442	0.07

In terms of the computational cost, the average computational time for each experiment is recorded in seconds and presented in Table 6.8. The M/Gibbs took approximately 120 to 442 seconds (2 to 7 minutes) for a small sample size $n = 30$ and different tests length, while the LA took between 0.01 to 0.07 seconds.

We have seen from this comparison study that the performance of Laplace is good in a small sample size setting, where we have only 30 students. Based on the comparison criterion presented in this study, the approximation abilities results are very close and sometimes even better than the estimation abilities result from M/Gibbs. Moreover, the approximations of the difficulty of the questions are also very similar to M/Gibbs. The following section will consider the result of increasing the sample size to 300.

Results of Study 2

This section presents and discusses the results of a moderate sample size; $n = 300$. The main comparison result will be focused on the test of length 10. The framework of the results will follow the same order of study 1.

Comparison of Distributions:

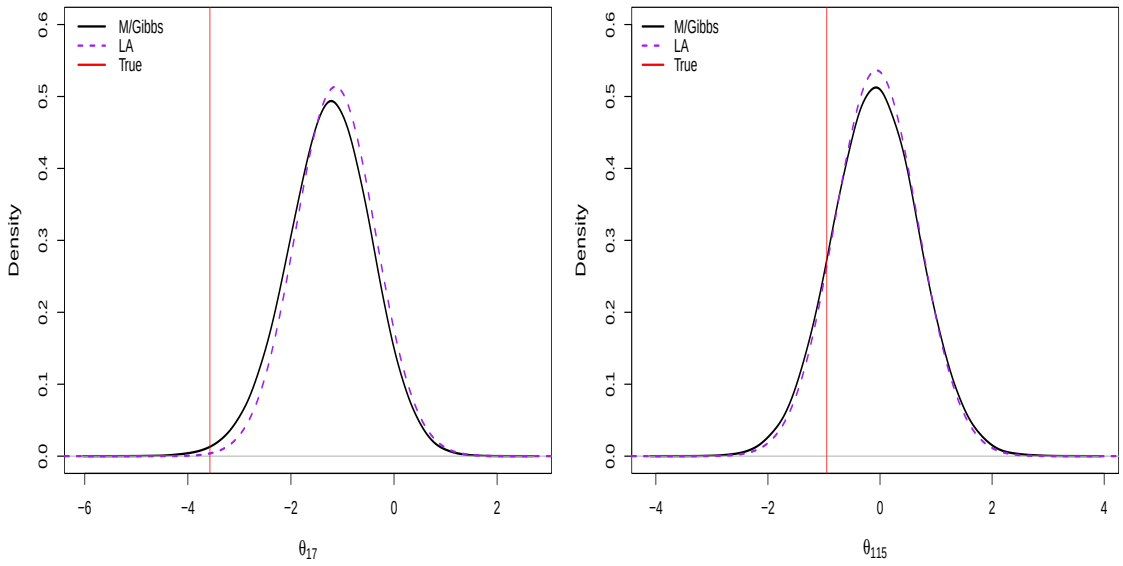
Figure 6.7 demonstrates the approximated posterior resulting from the LA and the posterior resulting from M/Gibbs for sample size n = 300 and a test of length 10. The figure presents three different level randomly selected abilities, which are the same level shown in the first study. We can see that increasing the sample size from 30 to 300 did not clearly affect or improve the approximation results of the posterior densities. The relation between the two posterior densities resulting from each method is the same as study 1 for all three different levels (explained in detail in 6.2.1).

The results of the JSD divergences for each students ability parameter are recorded and visualised in Figure 6.8. The JSD values that measured the differences between the two posterior densities in these experiments range between 0.0 to 0.4. The largest difference (JSD values) between the two posterior distributions appears for very high/low ability students. The maximum differences between the two resulting densities for the 300 sample size are higher than the maximum differences between the two resulting densities for the 30 sample size. However, there are only ten students with JSD values greater than 0.3. These differences dropped below 0.1

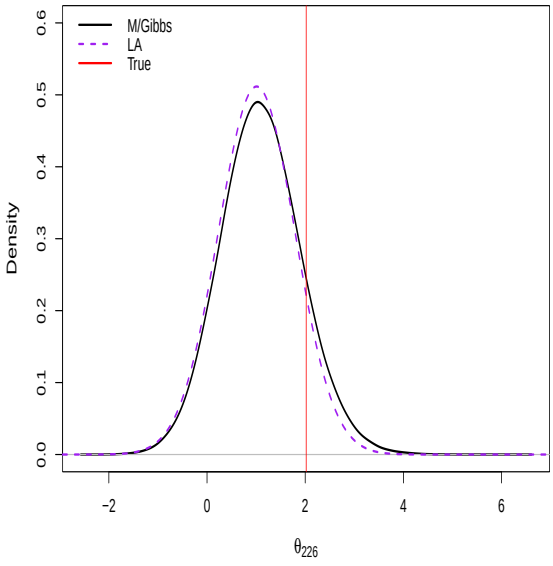
for moderate students abilities, such as students from 50 to 250.

Comparison of the Point Estimates:

The comparison of the point estimates of the 300 students' abilities resulting from the M/Gibbs and LA versus actual values are presented in Figure 6.9. Similarly to the finding of the first study (6.5), the point estimates resulting from each method are very close for moderate abilities ranging between -2 to 2. However, the point estimates varied slightly outside this range for very high/low abilities. Also, as we noticed for the small sample size ($n = 30$), M/Gibbs overestimated high abilities and underestimated low abilities. The average absolute difference between the points estimates resulting from each method is 0.30, and the absolute maximum is 0.76, which occurs between very high/low abilities. The correlation between the abilities estimates and the actual values are 0.92 and 0.93 for M/Gibbs and LA, respectively, which is less than the correlation for a sample size of 30. However, the correlation between the points estimates resulting from the two methods is 0.99, which indicates a strong relationship between the two sets of ability estimates methods, as shown in Figure 6.10.



(a) Number of Correct Answers= 3 (b) Number of Correct Answers= 5



(c) Number of Correct Answers = 7

Figure 6.7: Posterior density plots for M/Gibbs and LA methods of selected examinees' abilities with different numbers of correct answers for sample size $n = 300$ and $m = 10$.

The outcomes of the measurement distances (bias, RMSE and Kendall's τ) between the point estimates and the true values resulting from both methods, M/Gibbs and LA, is presented in Table 6.9. The results are the average values of repeating the simulations data 20 times. The result showed that the average

bias by both methods was very small and almost close to zero. Bias under the LA estimation is affected by increasing the sample size, which dropped from (-0.004) to nearly zero (-0.000003). However, bias under M/Gibbs is slightly dropped from (-0.005) to (-0.0004). The average result of RSME also decreases by increasing the sample size for both methods. However, RMSE under LA is smaller than RMSE under M/Gibbs by 0.23 differences. In terms of ordering the point estimates of the abilities, Kendall's τ values show that the LA method is much closer to the order set of the true abilities (0.83) than M/Gibbs (0.77).

Regarding the comparison between the point estimates of the difficulty parameters generating from each method and the true values, the results of the average bias, RMSE and Kendall's τ are presented in Table 6.10. The outcomes show that increasing the sample size from 30 to 300 improve the accuracy of the difficulty parameter estimations. As we can see, the average bias and RMSE dropped sharply for both methods. The average biases are almost zero; 0.0004 and -0.0005 for M/Gibbs and LA. Moreover, the resulting RMSE from both methods are very small; 0.18 and 0.159 for M/Gibbs and LA. Kendall's τ values increase mainly from 0.84 to 0.99 for both methods by increasing sample size. The large values of Kendall's τ (0.99) indicates that both methods are ordering the difficulties of the questions almost in the same way as the actual order. Additionally, the order of abilities resulting from each method is identical, where Kendall's τ value between the two methods is 1.

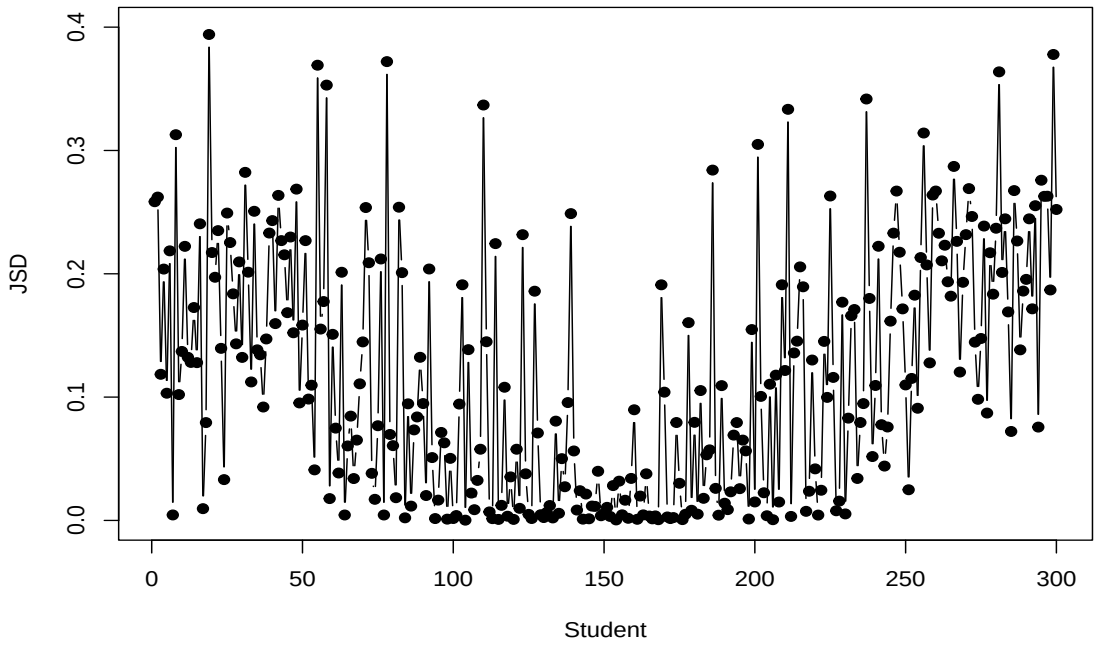


Figure 6.8: Jensen-Shannon divergence (JSD) method for each student's ability parameter obtained from M/Gibbs and LA for sample size $n = 300$ and $m = 10$.

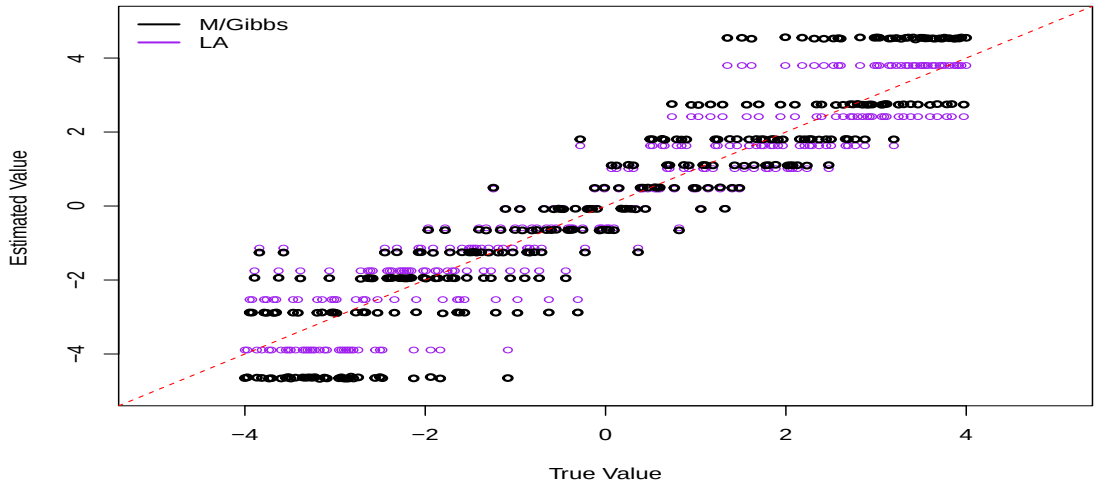


Figure 6.9: Point estimates of the examinees' abilities resulting from the; M/Gibbs, and LA versus true values. The red line illustrates the quality line.

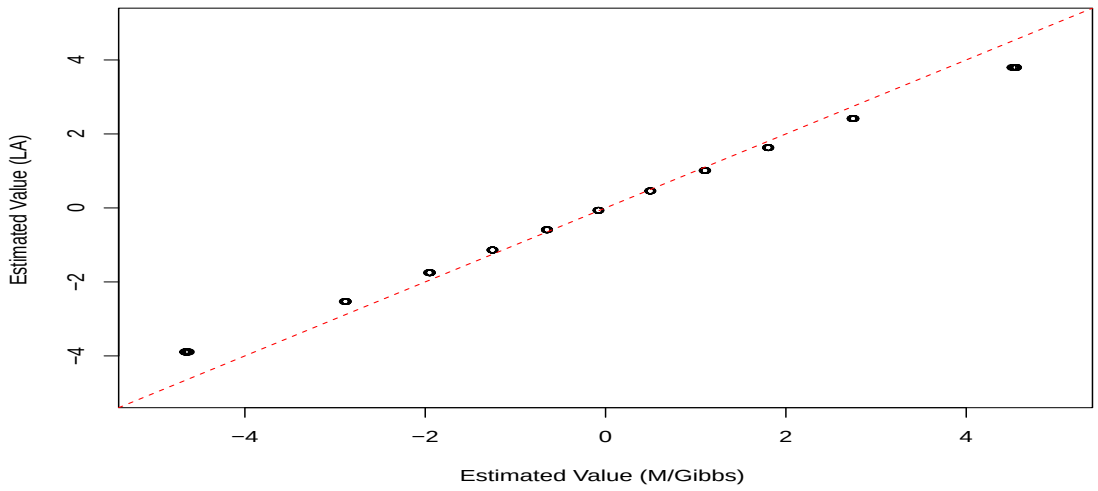


Figure 6.10: Point estimates of the examinees' abilities resulting from the M/Gibbs versus LA for sample size $n = 300$ and test length $m = 10$. The red line illustrates the quality line.

Table 6.9: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$, averaged across 20 different simulated datasets with sample size $n = 300$ and $m = 10$.

Method	Bias	RMSE	Kendall's τ
MCMC	-0.0004	1.014	0.77
LA	-0.000003	0.84	0.83

Table 6.10: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$, averaged across 20 different simulated data sets with sample size $n = 300$ and $m = 10$.

Method	Bias	RMSE	Kendall's τ
MCMC	-0.0004	0.185	0.99
LA	0.0005	0.159	0.99

Further Analysis

This section discusses the effect of increasing the test length to 30, 50,70 and 100 for a sample size of 300. Table 6.11 presents a comparison of average bias, average RMSE, and Kendall's τ values between the estimated points resulting from M/Gibbs and LA and the actual values for the ability parameter $\boldsymbol{\theta}$ and a different number of items. The result shows that increasing the test length did not affect the average bias, where both methods remain with a very small bias close to zero. However, the RMSE values dropped gradually by increasing the test length. It is clear that

RMSE values under LA are smaller than RMSE under M/Gibbs, where the values range between 0.71 to 0.40 for M/Gibbs and between 0.59 to 0.36 for LA. The differences between the RMSE resulting from each method decreased as the test length increased.

Kendall's τ values, which measure the difference between the actual abilities order and the order of the ability estimates resulting from each method, also increased by increasing the test length. Moreover, Kendall's τ values are higher for a sample size of 300 than 30. This implies that the accuracy in ordering the students' abilities is increased by increasing the sample size or the test length. Also, we can see that the order abilities resulting from LA are much closer to the true abilities order, where the value of Kendall's τ is higher. Table 6.12 shows Kendall's τ that measure the differences between the two methods. As we can see from the result that the two methods become almost identical by increasing the test length.

Table 6.11: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$ with sample size $n = 300$ and different number of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
30	MCMC	-0.0004	0.71	0.86
	LA	-0.0002	0.59	0.88
50	MCMC	-.000002	0.56	0.89
	LA	-0.0005	0.48	0.90
70	MCMC	-0.002	0.48	0.90
	LA	-0.003	0.42	0.91
100	MCMC	-0.0004	0.40	0.90
	LA	0	0.36	0.92

Table 6.12: Average Kendall's τ values between the point estimates of the students abilities resulting from LA and M/Gibbs for a sample size $n = 300$ and different numbers of items.

Number of Items	Kendall's τ
10	0.97
30	0.98
50	0.99
70	0.99
100	0.99

Table 6.13: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$ for sample size $n = 300$ and different numbers of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
30	MCMC	0.003	0.17	0.94
	LA	0.004	0.16	0.94
50	MCMC	0.0017	0.17	0.92
	LA	0.0018	0.16	0.93
70	MCMC	0.009	0.17	0.92
	LA	0.007	0.16	0.92
100	MCMC	-0.000003	0.17	0.92
	LA	-0.0004	0.17	0.92

Table 6.14: Average Kendall's τ values between the point estimates of the difficulty of the questions $\boldsymbol{b}$ resulting from LA and M/Gibbs for a sample size $n = 300$ and different numbers of items.

Number of Items	Kendall's τ
10	1
30	0.99
50	1
70	0.99
100	0.99

Regarding difficulty parameter $\boldsymbol{b}$ estimations, the outcomes of measurements result between the estimations methods; M/Gibbs and LA and the true values are presented in Table 6.13. The accuracy of the difficulty point estimates is improved by increasing the sample size to 300, where both bias and RMSE values are much smaller than the study of sample size 30. Also, the differences between the order of the true questions' difficulty and the order of difficulty of the point estimates become very small, where Kendall's τ increased from 0.70 to 0.90. All the three measurements, bias, RMSE and Kendall's τ, show that both methods perform in the same way in estimating the difficulty parameter $\boldsymbol{b}$ and result in the same accuracy level.

In terms of the computational cost, Table 6.15 summarises the average computational time of each experiment in seconds for M/Gibbs and LA. We can see that LA is computationally very cheap. For example, even for a large test of length 100, the LA only took approximately one second, while the M/Gibbs took about 31 minutes (1874 seconds).

This comparison study also emphasises the first study's results of small sample size ($n = 30$). The performance and the accuracy of the Laplace approximation are also very good in a moderate sample size setting ($n = 300$). Based on the comparison criterion; bias, RMSE and Kendall's τ, the approximation abilities results are close to the true values and sometimes even better than the estimation abilities from M/Gibbs. Moreover, the approximations of the difficulty of the questions are almost identical to M/Gibbs. The following section will consider the result of increasing the sample size to 600 and test of length 10. The effect of varying the test length for a large sample size $n = 600$ will also be discussed.

Table 6.15: Comparison of the computation time between M/Gibbs method and LA method for sample size $n = 300$ and different numbers of items.

	Time (in seconds)	
Number of Items	Gibbs/M	LA
10	871	0.13
30	1130	0.33
50	1375	0.54
70	1617	0.63
100	1874	0.98

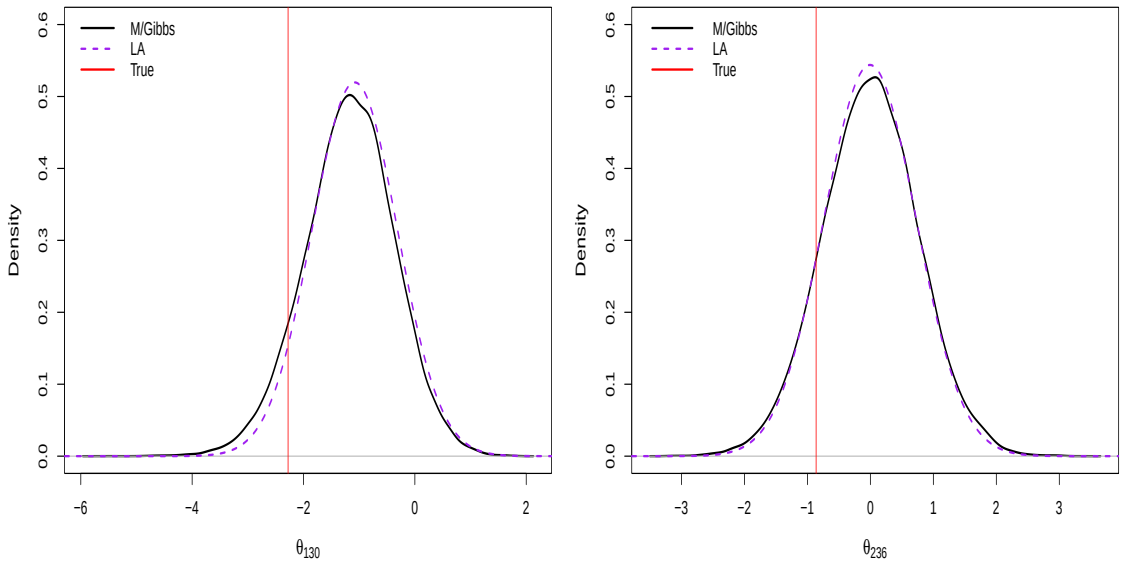
Results of Study 3

This section summaries the result of large sample size; $n = 600$. The main comparison result will be focused on the test of length 10. The effect of varying the test length for this samples size will be considered in the further analysis section.

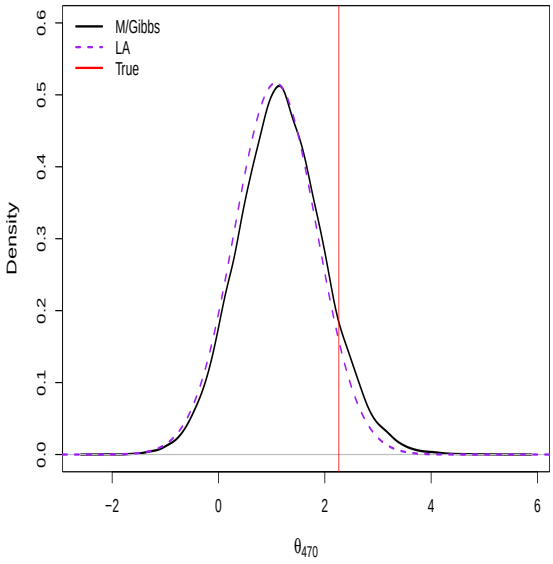
Comparison of Distributions:

Figure 6.11 displays the approximated posterior resulting from the LA and the posterior resulting from M/Gibbs for sample size $n = 600$ and a test of length 10. The figure presents three different levels of randomly selected abilities, which are the same level indicated in the first and second studies. The difference between the two densities resulting from each method remains almost the same by increasing the sample size to 600, where we cannot see, for example, evidence of changing the modes of the densities' places. The relation between the two posterior densities resulting from each method is the same, as explained in studies 1 and 2 for all three different ability levels.

In order to measure the distance between the two posterior densities resulting from M/Gibbs and LA, JSD divergences for each students ability parameter are computed and visualised in Figure 6.12. Similarly to study 2, The JSD values range from 0.0 to 0.4. However, only a few students with JSD values are greater than 0.3. The largest difference (JSD values) between the two posterior distributions also appears for very high/low ability students. These differences decreased by less than 0.1 for moderate students abilities, such as students from 100 to 500.



(a) Number of Correct Answers= 3 (b) Number of Correct Answers= 5



(c) Number of Correct Answers = 7

Figure 6.11: Posterior density plots for M/Gibbs and LA methods of selected examinees' abilities with different numbers of correct answers for sample size $n = 600$ and $m = 10$.

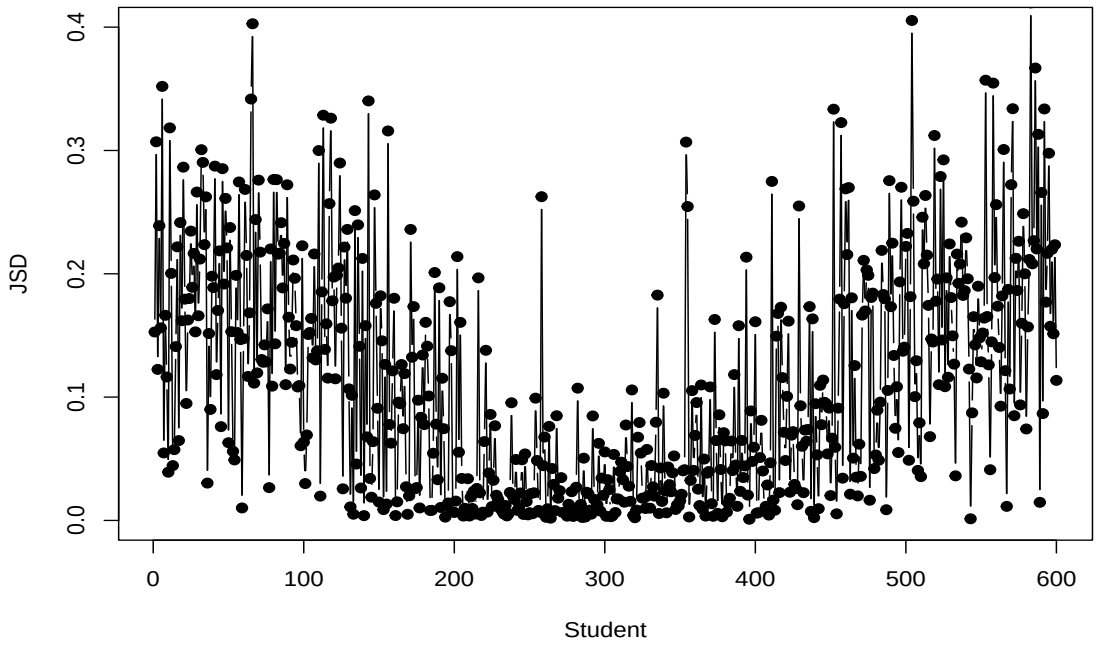


Figure 6.12: Jensen-Shannon divergence (JSD) for each student's ability parameter obtained from M/Gibbs and LA for sample size $n = 600$ and $m = 10$.

Comparison of the Point Estimates:

In terms of point estimates, Figure 6.13 shows the plot of the point estimates of the 600 students' abilities resulting from the M/Gibbs and LA versus actual values. In the same way, to study 1 and 2, the point estimates resulting from each method are very close when students' abilities range between -2 to 2 and vary slightly outside this range for very high/low abilities. Also, for this large sample size, M/Gibbs appears to overestimate high abilities and underestimate low abilities. The average absolute difference between the points estimates resulting from each method is 0.30, and the absolute maximum is 0.78, which occurs between very high/low abilities.

The correlation between the abilities estimates and the actual values remain the same as in study 2 ($n = 300$) 0.92 and 0.93 for M/Gibbs and LA. The correlation between the points estimates resulting from the two methods is 0.99, indicating a strong relationship between the two sets of ability estimates approaches, as shown in Figure 6.14.

To measure and compare the accuracy of the point estimates resulting from M/Gibbs and LA methods to the actual values, Table 6.16 presents the average bias, RMSE and Kendall's τ. The results show only small noticeable changes by

increasing the sample size from 300 to 600. For example, RMSE under M/Gibbs increased slightly from 1.01 to 1.02 and 0.84 to 0.86 for LA. On the other hand, Kendall's τ values drooped from 0.77 and 0.83 to 0.76 and 0.82 for M/Gibbs and LA, respectively. These results may indicate that the test length of 10 is insufficient for a sample size of 600.

The results of the comparison of the difficulty parameter $\boldsymbol{b}$ point estimates resulting from M/Gibbs and LA for $n = 600$ and test of length 10 are shown in Table 6.17. From the results values, we can see that bias and RMSE values under LA are slightly smaller than those under M/Gibbs. However, both methods are identical in ordering the difficulties of the questions, where Kendall's τ values are 1 for both methods.
Further analysis will be done in the next section to investigate the effect of increasing the test length to 30, 50, 70 and 100 for a sample size of 600.

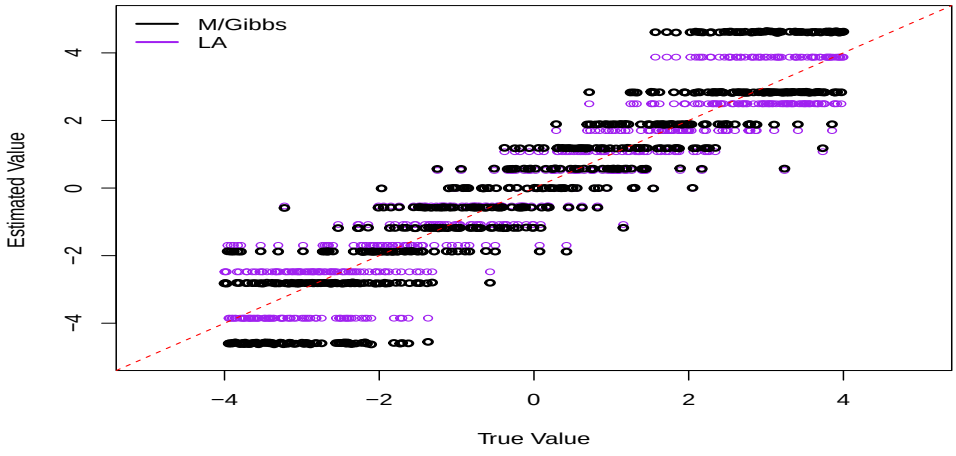


Figure 6.13: Point estimates of the examinees' abilities resulting from the; M/Gibbs, and LA versus true values for sample size $n = 600$ and $m = 10$. The red line illustrates the quality line.

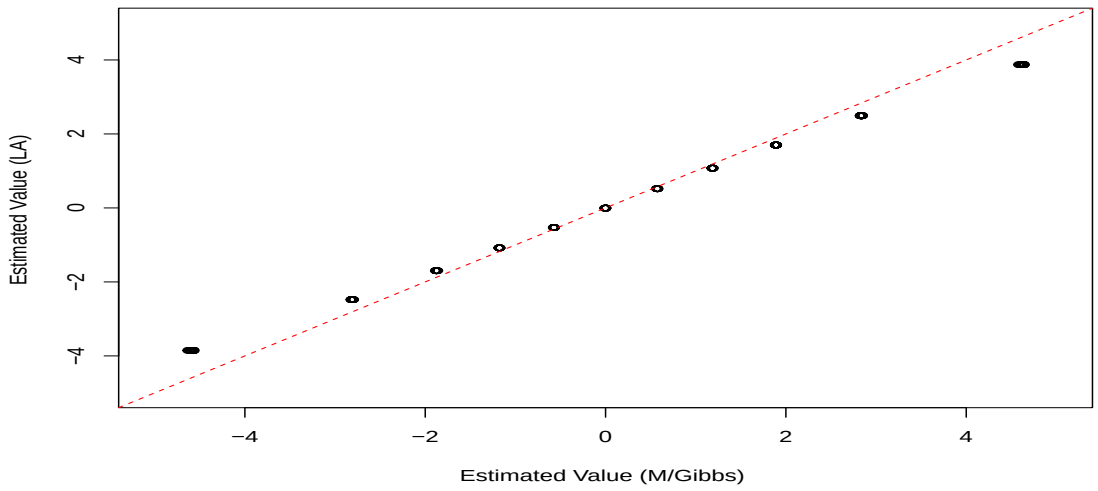


Figure 6.14: Point estimates of the examinees' abilities resulting from the M/Gibbs versus LA for sample size $n = 600$ and $m = 10$.

Table 6.16: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$, averaged across 20 different simulated datasets with sample size $n = 600$ and $m = 10$.

Method	Bias	RMSE	Kendall's τ
MCMC	-0.0005	1.02	0.76
LA	-0.0002	0.86	0.82

Table 6.17: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$, averaged across 20 different simulated data sets with sample size $n = 600$ and $m = 10$.

Method	Bias	RMSE	Kendall's τ
MCMC	0.014	0.13	1
LA	0.011	0.12	1

Further Analysis

The summary results of increasing the test length to 30, 50, 70 and 100 for a sample size of 600 will be presented in this section.

Table 6.18 shows the outcomes of average bias, RMSE and Kendall's τ resulting from measuring the accuracy between the actual values and point estimates resulting from M/Gibbs and LA. Regarding bias values, there are no noticeable changes by increasing the test length or the sample sizes, where both methods produce very

similar small biases. However, regarding RMSE dropped largely by expanding the test length from 10 to 30 and then started dropping slightly by increasing the test length. In terms of increasing the sample size from 300 to 600, RMSE under both methods only decreased by 0.01 points. For example, when $m = 30$, RMSE were 0.71 and 0.59 for M/Gibbs and LA and became 0.70 and 0.58. Moreover, RMSE values under LA are smaller than values under M/Gibbs.

Increasing the test length to 30 and more yields more accuracy in ordering the point estimates of the students' abilities resulting from both methods. For example, as we can see, Kendall's τ values that measure the differences between the order of the point estimates of the abilities and the true abilities increased from 0.76 and 0.82 when $m = 10$ to 0.86 and 0.88 when $m = 30$ M/Gibbs and LA. Moreover, Kendall's τ values for LA are slightly higher than M/Gibbs. This difference between the two methods in ordering the abilities become smaller by adding more questions, as shown in Table 6.19.

The comparison results between the actual values and the point estimates of the difficulty parameter $\boldsymbol{b}$ are presented in Table 6.20. Similarly to the results of studies 1 and 2, according to the outcomes of the criteria measurements, the point estimates resulting from each method are very close to being identical. However, bias and RMSE under LA are slightly smaller. Kendall's τ in Table 6.21 shows that the two methods are almost identical in ordering the questions' difficulties.

The average computational time for 20 different simulated datasets of sample size 600 and different tests length are calculated and presented in Table 6.22. As we can see, the LA method is high-speed, and even for a large dataset and test of length of 100, the method only took 4 seconds to produce the results. In contrast, M/Gibbs spent about one hour (3729 seconds) on the same dataset.

Table 6.18: Comparison of average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$ with sample size $n = 600$ and different number of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
30	MCMC	0.0008	0.70	0.86
	LA	0.0012	0.58	0.88
50	MCMC	0.0004	0.54	0.89
	LA	0.002	0.47	0.90
70	MCMC	-0.0003	0.46	0.90
	LA	-0.001	0.40	0.91
100	MCMC	-0.000002	0.39	0.91
	LA	-0.0012	0.35	0.92

Table 6.19: Average Kendall's τ values between the point estimates of the students abilities resulting from LA and M/Gibbs for a sample size $n = 600$ and different numbers of items.

Number of Items	Kendall's τ
10	0.96
30	0.99
50	0.99
70	0.99
100	0.99

Table 6.20: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$ for sample size $n = 600$ and different numbers of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
30	MCMC	-0.005	0.124	0.97
	LA	-0.003	0.114	0.97
50	MCMC	-0.008	0.120	0.95
	LA	-0.006	0.116	0.95
70	MCMC	0.003	0.118	0.94
	LA	0.002	0.115	0.95
100	MCMC	0.003	0.119	0.95
	LA	0.0008	0.116	0.95

Summary of the Comparison Results for 1PL

Regarding the point estimates of ability parameter $\boldsymbol{\theta}$, the biases for the LA were generally smaller than those from M/Gibbs. The only exceptions to this result occurred for 50 questions and a sample size of 300 and for 100 questions and 600

Table 6.21: Average Kendall's τ values between the point estimates of the difficulty of the questions $\boldsymbol{b}$ resulting from LA and M/Gibbs for a sample size $n = 600$ and different numbers of items.

Number of Items	Kendall's τ
10	1
30	1
50	0.99
70	0.99
100	0.99

Table 6.22: Computation time comparison between M/Gibbs method and LA method for sample size $n = 600$ and different numbers of items.

	Time (in seconds)	
Number of Items	Gibbs/M	LA
10	1691	0.53
30	2122	1.33
50	3060	1.80
70	3360	2.40
100	3729	4.00

sample size, in which cases the M/Gibbs approach yielded the lowest biases. In addition, the RMSE for the LA estimates were lower than those of the M/Gibbs estimator across all sample sizes and the number of questions, with the most noticeable differences in smaller sample sizes and shorter test lengths. Kendall's τ values were generally larger for the LA method, with the most marked differences occurring with smaller sample sizes and shorter test lengths. However, Kendall's τ values became almost identical for longer tests under both methods.

In terms of the point estimates of difficulty parameter $\boldsymbol{b}$, mostly the biases were slightly smaller for the LA than those for M/Gibbs. In addition, the RMSE were also smaller for the LA method, with the most apparent differences occurring in sample sizes of 30 and 300 for all test lengths. However, this difference between the two methods of RMSE was very small for the sample size of 600. Kendall's τ values were smaller under the LA method for the sample size of 30, and the two resulting values become almost identical for the sample sizes of 300 and 600.

From the results of these comparison studies, we can see that the LA method provides very accurate approximations in very cheap computational time. Therefore, the LA method seems to be a useful tool for researchers interested in obtaining estimates of students' abilities in real-time. Further analysis for larger sample sizes; $n = 1000$ and $n = 2000$ can be found in the Appendix B, where the results confirm

the findings of the comparison studies presented in this section.

6.2.2 Comparison Study for 2PL Model

In the literature, more complex models tend to be more complicated to fit, and can take longer when using MCMC estimation methods with more convergence issues (e.g. Gelman et al. (2013), Bürkner (2019)). Therefore, this section will investigate the performance of the Laplace approximation method in the 2PL model. The performance of two MCMC methods; Hamiltonian Monte Carlo (HMC) and Metropolis within Gibbs samplers (M/Gibbs), in estimating the 2PL model's parameters, have been investigated in detail in Chapter 4. The procedure of the comparison study will be carried out in the same way as the comparison study for the 1PL model. However, this study will be focused on one sample size and test length.

Simulated Data

The data in this setting will be generated using the 2PL model described in 4.4.1. In the previous comparison studies, there is only one item parameter: the difficulty parameter b_j, which measures the difficulty of the questions. In the 2PL model, we can also estimate how strongly question j distinguishes the student's ability θ_i by adding the discrimination parameter a_j. However, extra care should be taken when using the LA method to approximate the discrimination parameter. The following is a brief description of the issue one may face when estimating this parameter and a proposed solution.

Discrimination Parameter and Transformation

As mentioned in 2.8.2, a general issue that may occur when using LA for bounded parameters is that the approximations of the posterior distributions may become less accurate when moving away from the posteriors' mode. Since the discrimination parameter is bounded between $[0.5, 1]$ and to avoid this issue, this parameter will be re-parametrised using the logarithm transformation. If a transformation is applied to the discrimination parameter; $g(a) : \mathbb{R} \to \mathbb{R}$, then its probability density function (pdf) will change too. Hence, if g(a) is a monotonic and differentiable function, the pdf $p_{\boldsymbol{a'}}(a)$ of the transformed random variable $\boldsymbol{a'}$ can be computed as:

$$p_{\boldsymbol{a'}}(a') = p_{\boldsymbol{a}}\left(g^{-1}(a')\right)\left|\frac{d}{da'}\left(g^{-1}(a')\right)\right|,$$

where g^{-1} represent the inverse function of g. The Jacobian part, $|\frac{d}{da'}\left(g^{-1}(a')\right)|$, makes sure that the new pdf $p_{\boldsymbol{a'}}(a')$ is a valid pdf, which is still integrated to 1. For more explanation about parameter transformation, see Casella and Berger (2021).

We can apply a non-linear transformation to the discrimination parameter, such as $a' = log(a)$. In this case of the logarithm, small values ($0.5 < a < 1$) are mapped to larger negative numbers. Since the gamma distribution is used as prior, the log transformation can be found as follows:

$$\begin{aligned} p_{\boldsymbol{a'}}(a') &= f_{\boldsymbol{a}}\left(g^{-1}(a')\right)\left|\frac{d}{da'}\left(g^{-1}(a')\right)\right| \\ &= \frac{\beta^{\alpha}}{\Gamma(\alpha)} g^{-1}(a')^{\alpha-1}\exp\left(-\beta g^{-1}(a')\right)\left|\frac{d}{da'}\left(g^{-1}(a')\right)\right| \\ &= \frac{\beta^{\alpha}}{\Gamma(\alpha)}\exp(a')^{\alpha-1}\exp(-\beta\exp(a'))\cdot\exp(a') \\ &= \frac{\beta^{\alpha}}{\Gamma(\alpha)}\exp(a')^{\alpha}\exp(-\beta\exp(a')) \end{aligned}$$

Hence, the transformation prior can be used now for the posterior distribution.

Regarding sample sizes, we have seen from the previous studies that the two methods, M/Gibbs and LA, become comparable for a sample size of 300 students. However, as suggested in the literature, as the complexity of the model increases, more data is required (e.g. Sahin and Anil (2017)). Therefore, this study will aim to use a relatively large sample size of 600 students and a moderate test length of 50 questions to ensure the accuracy of the M/Gibbs method, which is aimed to use as a baseline to compare the LA method results to its results.

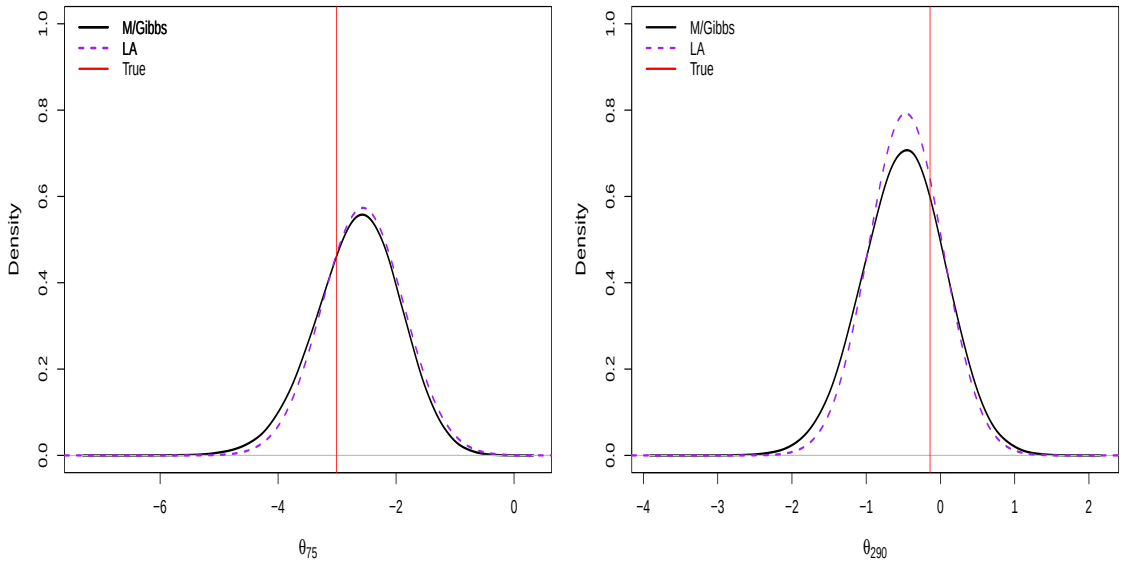
Comparison Results

This section presents the results of the comparison study between the M/Gibbs and the LA methods for estimating the 2PL model's parameters. The MCMC method will be run for a $500,000$ number of iterations, and the initial 2,000 part of iterations will be discarded (burn-in) based on initial visualisation. The same comparison criterion 6.2 used for previous comparison studies will also be used here.

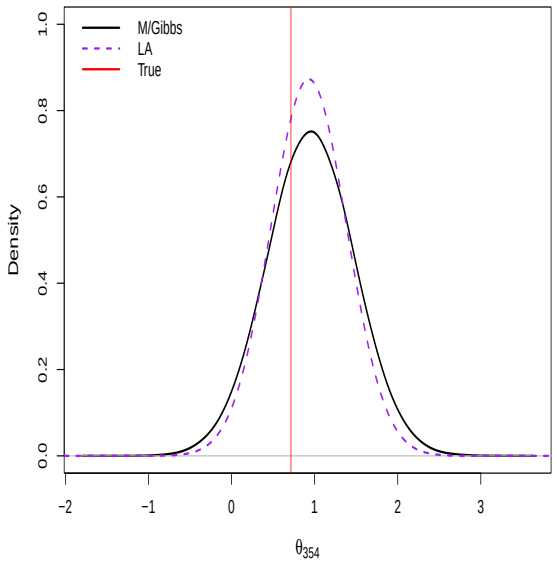
Comparison of Distributions:

Figure 6.15 shows the approximated posterior resulting from the LA and the posterior resulting from M/Gibbs for sample size n = 600 and a test of length 50. The posterior distribution generated from LA seems slightly higher than M/Gibbs's posteriors, indicating that LA may underestimate the variance. To measure the divergence between the two posteriors, JSD divergences for each student's ability parameter are computed and visualised in Figure 6.16. The JSD values range from 0.001 to 0.32. However, only a few students with JSD values are greater than 0.25, about 1%.

The largest difference (JSD values) between the two posterior distributions appears for very high ability students. In most cases, these differences become less than 0.1 for moderate students' abilities, such as students from 200 to 400.



(a) Number of Correct Answers= 10 (b) Number of Correct Answers= 20



(c) Number of Correct Answers= 30

Figure 6.15: Posterior density plots for M/Gibbs and LA methods of selected examinees' abilities with different numbers of correct answers for the 2PL mode.

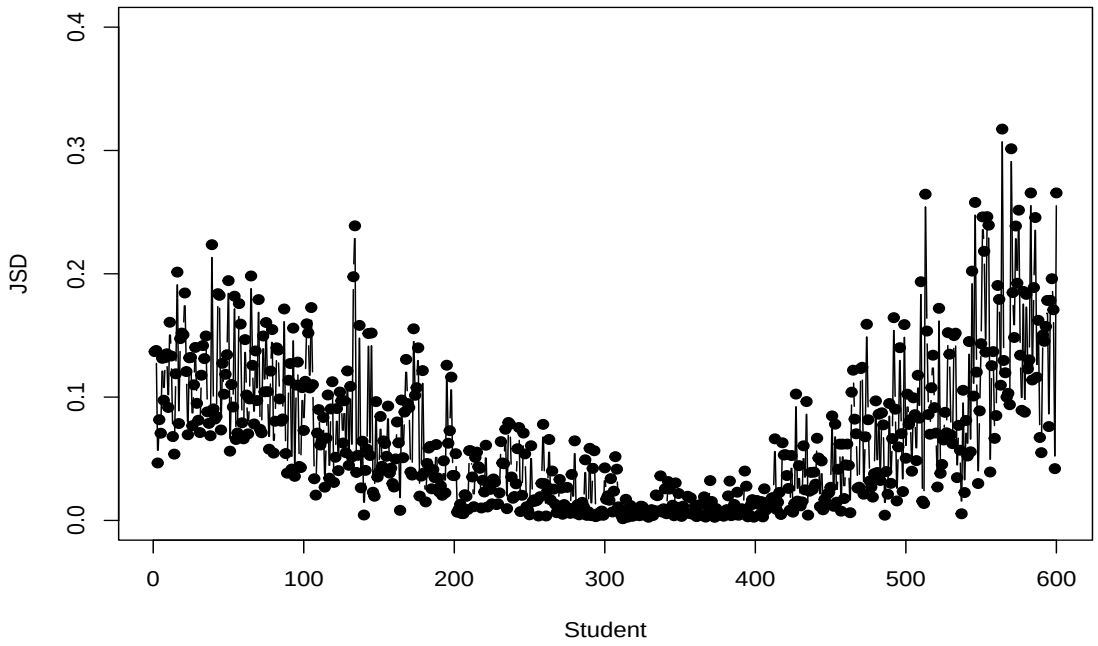


Figure 6.16: Jensen-Shannon divergence (JSD) for each student's ability parameter obtained from M/Gibbs and LA for the 2PL model.

Comparison of the Point Estimates:

In terms of point estimates, Figure 6.17 displays the comparison of the point estimates of the students' abilities resulting from the M/Gibbs and LA (y-axis) versus actual values (x-axis). Similarly to the 1PL, we can see that the point estimates are very close for abilities ranging between -2 and 2. However, the point estimates varied slightly outside this range for very high/low abilities, whereas M/Gibbs slightly underestimated low abilities and overestimated high abilities. Figure 6.18 also confirms that a strong relationship between the two sets of ability estimate methods and a slight variation appears for low/high abilities.

The simulated data is repeated 30 times under the same conditions and the average results of the bias, RMSE and Kendall's τ are calculated and presented in Table 6.23 for the ability, difficulty and discrimination parameters. The biases of both estimation methods are small enough to be regarded as practically negligible in the ability and difficulty parameter, although the LA estimates have a slightly larger bias. The bias of the discrimination parameter is larger than other parameters' biases; however, it is slightly smaller for the LA. In addition, the RMSEs of LA estimates are generally smaller than the M/Gibbs method for all three parameters.

The average value of Kendall's τ (Table 6.23), which evaluates the degree of similarity between the true values set and the point estimates set, indicates that the two approaches are ordering the ability of the students and the difficulty/ discrimination of the questions almost in the same way. Furthermore, the average values of the Kendall rank (τ) between the order given by both methods in Table 6.24 are large, confirming the high similarity between the point estimates sets.

Regarding time cost, M/Gibbs took 17,521 seconds on average (almost 5 hours) to produce the results, while LA only took 6 seconds. Therefore, in general, applying the LA method to the 2PL model does not appear to have any issues, as the results are very comparable to the MCMC method and much faster.

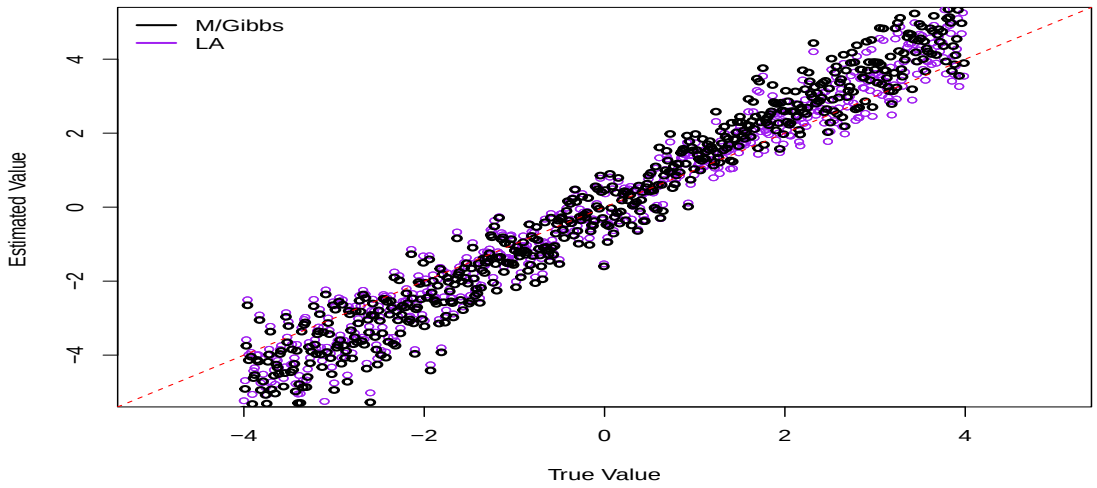


Figure 6.17: Point estimates of the examinees' abilities resulting from the; M/Gibbs, and LA versus true values for sample size for 2PL model. The red line illustrates the equality line.

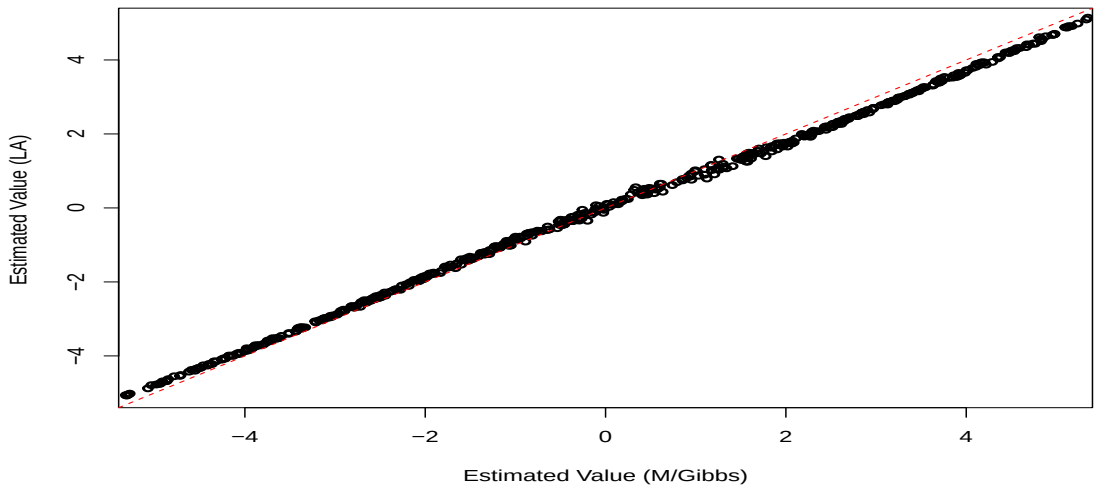


Figure 6.18: Point estimates of the examinees' abilities resulting from the M/Gibbs versus LA for 2PL model. The red line illustrates the equality line.

Table 6.23: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability ($\boldsymbol{\theta}$), difficulty ($\boldsymbol{b}$) and discrimination ($\boldsymbol{a}$) parameters for the 2PL model.

Parameter	Method	Bias	RMSE	Kendall's τ
$\boldsymbol{\theta}$	MCMC	0.0008	0.91	0.87
	LA	-0.0068	0.58	0.87
$\boldsymbol{b}$	MCMC	-0.005	0.38	0.94
	LA	-0.009	0.17	0.94
$\boldsymbol{a}$	MCMC	-0.15	0.17	0.77
	LA	0.02	0.09	0.76

Table 6.24: Average Kendall's τ values between the point estimates of the 2PL model's parameters resulting from LA and M/Gibbs.

Parameter	Kendall's τ
$\boldsymbol{\theta}$	0.99
$\boldsymbol{b}$	0.99
$\boldsymbol{a}$	0.95

6.3 High-Dimensional Covariance Matrix Problems

Computational time is an essential criterion for assessing the efficiency of the Laplace approximation method for the purpose of online inference or massive data sets. The performance of the LA method depends on obtaining the covariance matrix

$\hat{\boldsymbol{\Sigma}}$ by inverting the Hessian matrix ($\mathbf{H}$). However, computing the inverse of $\mathbf{H}$ is computationally expensive when a posterior distribution is fitted on high-dimensional data when there is a large sample size of students (n).

This section will discuss two different methods to reduce the computational cost. The first method is to use the idea of the block matrix. The second method is to approximate the posterior distribution with the diagonal of the H matrix. The speed efficiency of these methods will be assessed by comparing the running time to the standard Laplace approximation 6.1. The following sections will explain the uses and benefits of the two proposed methods.

6.3.1 Covariance Matrix Structure with Laplace Approximation

Before making any assumptions about the covariance matrix, a toy example will be used to explain the structure of the Hessian and covariance matrices. A small sample size of students and a short test length will be considered to make it easy to look at the matrices' structure. Therefore, this example will assume 3 students are answering 5 questions. The same procedure of applying the LA method on the 1PL model will be used here.

Toy Example

As mentioned early, the $\mathbf{H}$ matrix of the full posterior distribution $P(\boldsymbol{\theta}, \boldsymbol{b}|\boldsymbol{X})$ is the matrix of the second derivative of the log posterior at its maximum point estimates; $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{b}}$. Which can be formally written as:

$$\mathbf{H}_p = \begin{pmatrix} \frac{\partial^2 p}{\partial \theta_1^2} & \frac{\partial^2 p}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 p}{\partial \theta_1 \partial \theta_n} & \frac{\partial^2 p}{\partial \theta_1 \partial b_1} & \cdots & \frac{\partial^2 p}{\partial \theta_1 \partial b_m} & \frac{\partial^2 p}{\partial b_1^2} & \cdots & \frac{\partial^2 p}{\partial b_1 \partial b_m} \\ \frac{\partial^2 p}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 p}{\partial \theta_2^2} & \cdots & \frac{\partial^2 p}{\partial \theta_2 \partial \theta_n} & \frac{\partial^2 p}{\partial \theta_2 \partial b_1} & \cdots & \frac{\partial^2 p}{\partial \theta_2 \partial b_m} & \frac{\partial^2 p}{\partial b_2 \partial b_1} & \cdots & \frac{\partial^2 p}{\partial b_2 \partial b_m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 p}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 p}{\partial \theta_n \partial \theta_2} & \cdots & \frac{\partial^2 p}{\partial \theta_n^2} & \frac{\partial^2 p}{\partial \theta_n \partial b_1} & \cdots & \frac{\partial^2 p}{\partial \theta_n \partial b_m} & \frac{\partial^2 p}{\partial b_m \partial b_1} & \cdots & \frac{\partial^2 p}{\partial b_m^2} \end{pmatrix}$$

Hence, the resulting matrix is a square $(n + m) \times (n + m)$ matrix, where n is the number of students and m is the number of questions. Therefore, the $\mathbf{H}$ matrix is a way to combine all the information from the second derivative of the posterior distribution. The following is the $\mathbf{H}$ matrix for the toy example, where $n = 3$ and $m = 5$.

$$\boldsymbol{H} = \begin{pmatrix} 0.95 & 0.00 & 0.00 & -0.20 & -0.20 & -0.20 & -0.05 & -0.20 \\ 0.00 & 1.15 & 0.00 & -0.22 & -0.22 & -0.22 & -0.18 & -0.22 \\ 0.00 & 0.00 & 0.35 & -0.03 & -0.03 & -0.03 & -0.14 & -0.03 \\ -0.20 & -0.22 & -0.03 & 0.54 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.20 & -0.22 & -0.03 & 0.00 & 0.54 & 0.00 & 0.00 & 0.00 \\ -0.20 & -0.22 & -0.03 & 0.00 & 0.00 & 0.54 & 0.00 & 0.00 \\ -0.05 & -0.18 & -0.14 & 0.00 & 0.00 & 0.00 & 0.47 & 0.00 \\ -0.20 & -0.22 & -0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.54 \end{pmatrix}$$

In statistics, the Hessian matrix of the log posterior distribution with respect to the parameters is the negative inverse of the covariance of the parameter estimates; $\boldsymbol{H} = -\boldsymbol{\Sigma}^{-1}$, which is known as the precision matrix (Yuen, 2010). The elements in this matrix tell us about the conditional information because they are obtained by fixing all other parameters. Hence, the diagonal elements are the negative of the conditional variance of the parameters. We see that in 6.3.1 the sub-matrix of $\theta's$ (red rectangle) and the sub-matrix of the $b's$ (blue rectangle) have zero for off-diagonal elements. This implies that $\theta's$ and $b's$ are conditionally independent of each other. In other words, a student's response to a question is independent of another students conditional on the student's ability. Therefore, there should not be any correlation between questions after conditioning on $\boldsymbol{\theta}$ but only correlation in the full problem. This result confirms a critical assumption for the IRT model, known as local independence or conditional independence (explained in Chapter 3).

Figure 6.19 displays a graphical representation of conditional independence for two students and two questions. As we see, questions (b_1 and b_2) are not linked to each other and are only associated via the ability (θ_1 and θ_2) . Therefore, a correct/wrong response to one question should not lead to a correct/wrong response to another question.

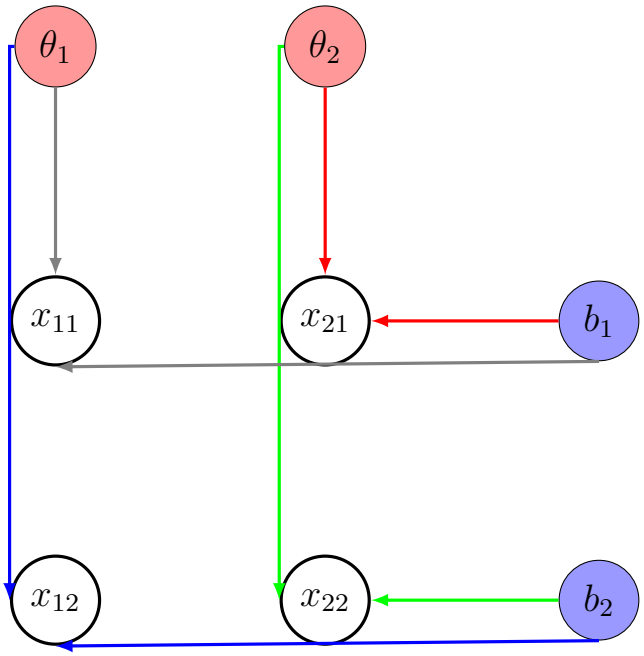


Figure 6.19: Graphical representation of conditional independence.

To apply the LA method, we need to invert the $\boldsymbol{H} = -\boldsymbol{\Sigma}^{-1}$ matrix. The following is the result of inverting $\boldsymbol{H}$ matrix and multiplying by -1, which is the covariance matrix $\boldsymbol{\Sigma}$.

$$\boldsymbol{\Sigma} = \begin{pmatrix} 2.17 & 1.10 & 0.76 & 1.27 & 1.27 & 1.27 & 0.88 & 1.27 \\ 1.10 & 1.98 & 0.85 & 1.23 & 1.23 & 1.23 & 1.13 & 1.23 \\ 0.76 & 0.85 & 3.68 & 0.80 & 0.80 & 0.80 & 1.50 & 0.80 \\ 1.27 & 1.23 & 0.80 & 2.84 & 1.00 & 1.00 & 0.85 & 1.00 \\ 1.27 & 1.23 & 0.80 & 1.00 & 2.84 & 1.00 & 0.85 & 1.00 \\ 1.27 & 1.23 & 0.80 & 1.00 & 1.00 & 2.84 & 0.85 & 1.00 \\ 0.88 & 1.13 & 1.50 & 0.85 & 0.85 & 0.85 & 3.08 & 0.85 \\ 1.27 & 1.23 & 0.80 & 1.00 & 1.00 & 1.00 & 0.85 & 2.84 \end{pmatrix}$$

Unlike the $\boldsymbol{H}$ matrix, which presents the conditional correlations among the parameters, the covariance matrix $\boldsymbol{\Sigma}$ shows the marginal correlations. Therefore, the diagonal elements in the covariance matrix are the marginal variances of the parameters. Also, as we can see, the covariance matrix $\boldsymbol{\Sigma}$ is dense, implying that every pair of the parameter is marginally correlated.

Figure 6.20 shows the correlation matrix between 1PL model's parameters for $n = 3$ and $m = 5$. The result shows moderately strong correlations between students' ability and questions' difficulty. These correlations occur because of the model identifiability issue, explained in 3.3. The 1PL model suffers from additive

identifiability issues, i.e adding a constant number to each θ_i and b_j will result in the same likelihood. There are also moderately high correlations between $b's$, and less correlation occurs between $\theta's$. However, it might not be evident, for this is a small example. Hence in Appendix B, Figures B.2, B.3, and B.4 provide more examples of the correlation between the model's parameters for larger datasets.

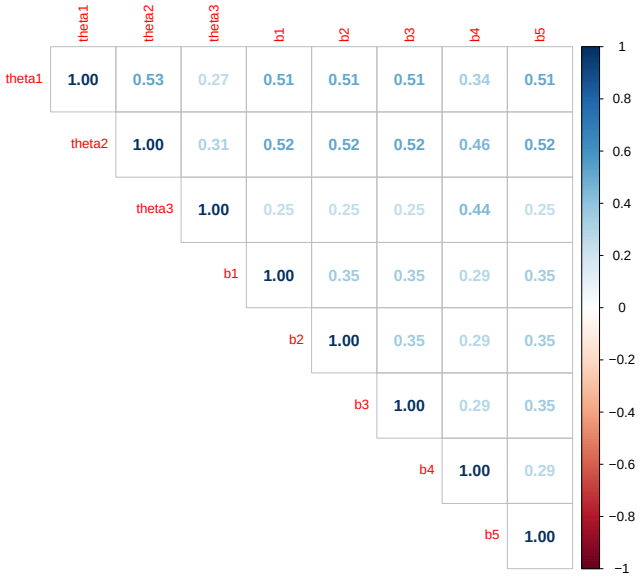


Figure 6.20: Correlation matrix between 1PL model parameters for $n = 3$ and $m = 5$.

Summary

In summary, there are some important results that can help make assumptions about Hessian; $\boldsymbol{H}$ and covariance; $\boldsymbol{\Sigma}$ matrices to reduce the computational time of inverting the $\boldsymbol{H}$ matrix for massive data sets. First, from the analysis of the $\boldsymbol{H}$ matrix, we found that $\theta's$ and $b's$ are conditionally independent, and hence these two sub-matrices are sparse matrices. This will help in storing matrices in special block structures. The idea of this method will be discussed in detail in section 6.3.2. The second assumption that could be made is using only the diagonal of the $\boldsymbol{H}$ matrix. From the correlation between the model's parameters, we found less correlation between $\theta's$ than between $b's$ or $\theta's$ and $b's$. This knowledge will help us understand and track the differences between using the full Hessian matrix and using only the diagonal of the $\boldsymbol{H}$ matrix. This method will be discussed in detail in section 6.3.3.

6.3.2 Block Matrix Strategy for Laplace Approximation

This section provides a strategy to invert the Hessian matrix to reduce the computational time by using a block matrix strategy. The idea is that we can divide the $\boldsymbol{H}$ matrix into sub-matrices or block matrices and use some linear algebra strategies to simplify calculations of inverting the $\boldsymbol{H}$ matrix.

Block Matrix Strategy

We have seen from the structure of the $\boldsymbol{H}$ matrix 6.3.1 that this matrix can be broken into four sub-matrices or blocks. Therefore, the $\boldsymbol{H}$ matrix can be interpreted as a flowing block matrix:

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H_{\theta\theta}} & \boldsymbol{H_{\theta b}} \\ \boldsymbol{H_{b\theta}} & \boldsymbol{H_{bb}} \end{pmatrix},$$

where $\boldsymbol{H_{\theta\theta}}$ represents Hessian matrix for students and $\boldsymbol{H_{bb}}$ is the Hessian matrix for questions. The two sub-matrices $\boldsymbol{H_{\theta b}}$ and $\boldsymbol{H_{b\theta}}$ are for both students and questions. The $\boldsymbol{H}$ matrix has been convert from $(n+m)\times(n+m)$ to 2×2 matrix. Therefore, working with the 2×2 matrix is mathematically more straightforward. This will allow us to use a common inverse formula for 2×2. There are several formulas, but in this thesis, the focus will be on the procedure related to the Hessian matrix, which is the square diagonal partition, where $\boldsymbol{H_{\theta\theta}}$ and $\boldsymbol{H_{bb}}$ are square matrices. This method can be explained shortly as follows;

A nonsingular or invertible square matrix $\boldsymbol{M}$ (its determinant is non-zero), that can be divided into 2×2 blocks is given by:

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix},$$

and its inverse is given by:

$$\boldsymbol{M^{-1}} = \begin{pmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{G} & \boldsymbol{H} \end{pmatrix}.$$

In the case that A, D, E and H are square matrices, where A and E have the same size, as well as D and H, the matrix M is invertible if and only if A is nonsingular, and the Schur complement $(D - CA^{-1}B)$ of A is invertible. Thus, the following formula can be used to invert $\boldsymbol{M}$;

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1}CA^{-1} & -A^{-1}B\left(D - CA^{-1}B\right)^{-1} \\ -\left(D - CA^{-1}B\right)^{-1}CA^{-1} & \left(D - CA^{-1}B\right)^{-1} \end{bmatrix}.$$

From comparing $\boldsymbol{M}$ matrix to the $\boldsymbol{H}$ matrix, we can see that $\boldsymbol{A} = \boldsymbol{H_{\theta\theta}}$, $\boldsymbol{B} = \boldsymbol{H_{\theta b}}$, $\boldsymbol{C} = \boldsymbol{H_{b\theta}}$ and $\boldsymbol{D} = \boldsymbol{H_{bb}}$. More details about this method and other formulas can be found in Lu and Shiou (2002).

While this formula still looks complex, using some linear algebra strategies with some advanced computer programming in R can make it mathematically cheap. For example, $\boldsymbol{A}(\boldsymbol{H_{\theta\theta}})$ is a diagonal matrix and only needs to take the inverse of the diagonal. However, if we use the full $\boldsymbol{H}$ matrix, the programme does not take this point into account and calculates for the whole matrix, which is mathematically expensive. Another essential strategy is that $\boldsymbol{B}(\boldsymbol{H_{\theta b}})$ and $\boldsymbol{C}(\boldsymbol{H_{b\theta}})$, in this case, are symmetric, which means $\boldsymbol{C} = \boldsymbol{B}^T$. Hence, we can only calculate the top right of the $\boldsymbol{M^{-1}}$; $\boldsymbol{E}$ and then transpose the result to get $\boldsymbol{G}$.

Comparison of Computation Time

Converting the Hessian matrix into a 2×2 block matrix and using the previously mentioned formula to invert the new matrix will not change the point estimates or the approximate posterior distributions. In other words, the resulting point estimates and posterior distributions from the block matrix method will be exactly the same as those resulting from using a full Hessian matrix. Therefore, this subsection will only focus on comparing computation time using the full Hessian matrix and the block matrix to produce the covariance matrix, which operates in the Laplace approximation method. All experiments will be conducted on the same computer to ensure the computer system does not affect the computation time.

To illustrate the computational time, this comparison will consider an increasing of sample sizes from $n = 1000$ to $n = 10,000$ and test of length ($m = 50$). Table 6.25 presents calculation times for inverting the full $\boldsymbol{H}$ matrix (Full H) and inverting the $\boldsymbol{H}$ block utilising the strategy of the 2×2 block matrix for a different number of students. As the results show, inverting the $\boldsymbol{H}$ matrix becomes expensive for larger datasets, particularly a sample size of 5,000 students or more. However, it is not enormously expensive, such as inverting the $\boldsymbol{H}$ matrix of $10,050 \times 10,050$ took only 11 minutes. On the other hand, the computing time can essentially drop to 8 seconds using the block matrix strategy. Therefore, the proposed block matrix strategy can be very efficient in reducing the computational time in the high-dimensional matrix

Table 6.25: Comparison of the computation time between inverting the full Hessian matrix (Full $\boldsymbol{H}$) and using the block matrix method (Block $\boldsymbol{H}$) for a test of length $m = 50$ and different numbers of students.

	Time (in seconds)	
Number of Student	Full $\boldsymbol{H}$	Block $\boldsymbol{H}$
1000	0.75	0.72
2000	6.20	0.28
5000	86.20	1.63
8000	337.86	5.38
10,000	670.00	8.08

6.3.3 Diagonal Laplace Approximation

This section considers another method to address the computational issue of inverting the Hessian matrix in high dimensions. This method aims to calculate the diagonal of the inverse Hessian matrix over the mode to obtain the variance estimates used to operate LA. For example, the approximation posterior distributions of the ability parameters can be obtained as following:

$$\boldsymbol{\theta} \sim \mathcal{N}\left\{\hat{\boldsymbol{\theta}}, diag(-\boldsymbol{H}_{\hat{\boldsymbol{\theta}}})^{-1}\right\}$$

Therefore, inverting the Hessian matrix will not be a full covariance matrix but only gives the vector of all diagonal elements of variance for each parameter. In this setting, we will lose some information from the correlation between the students' abilities and questions' difficulty, as explained earlier in 6.3.1. As a result, this will lead to a change in the posterior distributions. Using the diagonal of the Hessian matrix in LA is commonly used in machine learning and neural networks; see as examples Ritter et al. (2018),Trippe et al. (2019) and Perone et al. (2021). A comparison between using full and diagonal Hessian matrices in the LA method will be carried out in the following subsection. This comparison study will consider the amount of change between the two resulting posterior distributions by measuring the distance between the two distributions.

Comparison Study Between Full and Diagonal LA

In this comparison study, the focus will be on the change in the posterior distributions when using the diagonal of the H matrix instead of the full matrix. Therefore, a small data set will be used to explore and track the change on all ability levels. However, to emphasise, the purpose of using this method is to reduce the computational time in massive datasets and high-dimensional matrix. A comparison of computation time for massive datasets will be considered later.

Simulated Data

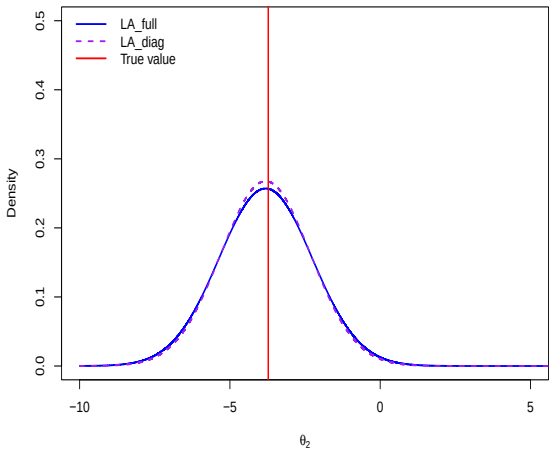
The simulated data used in study 1 (6.2.1) will be utilised in this comparison study. For this dataset, there are 30 students and 10 questions.

Comparison Result

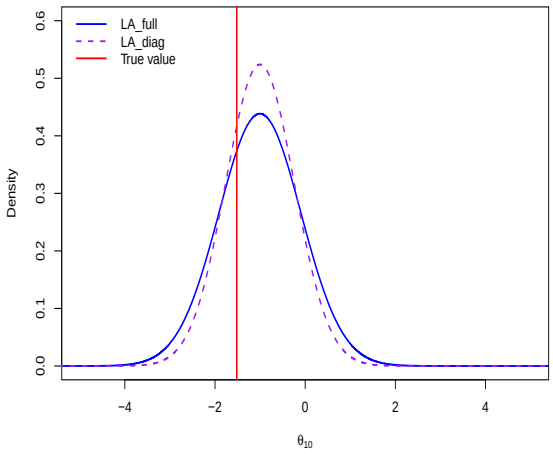
This method will not affect the point estimates results, meaning that the point estimates resulting from full and diagonal LA will be equal. Therefore, the focus will be on comparing the distributions.

Figure 6.21 displays comparisons between the approximate posterior distributions of the full and diagonal LA for five different randomly selected abilities levels. We can see that the variances are slightly underestimated in most cases by using the diagonal LA (LA_diag). However, if a student has a very low ability, with no correct answer, such as θ_2 or very high ability, with all questions answered correctly, such as θ_{28}, the two variances are almost identical. On the other hand, for moderate abilities students such as θ_{10}, θ_{14} and θ_{20}, the approximate posterior distributions resulting from the diagonal LA are narrower than those resulting from full LA, where the variances of the posterior distributions are slightly smaller.

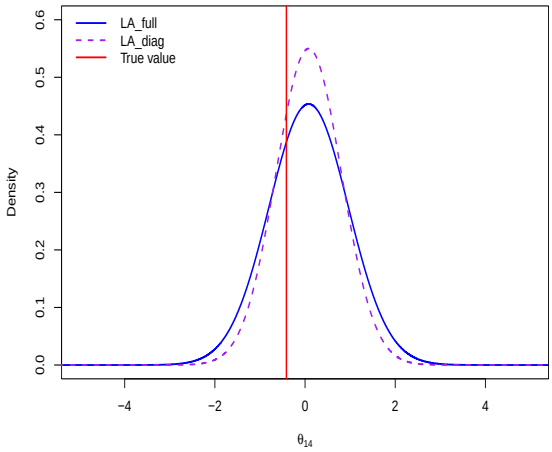
The difference between the resulting posterior distributions from both methods comes from setting all non-diagonal elements in the Hessian matrix to zeros. This setting is valid for $\theta's$ and $b's$ because, as we have seen in 6.3.1, they are conditionally independent. However, this is not the case between $\theta's$ and $b's$, where there is some correlation between them. Figure 6.22 shows the correlation resulting from using the full LA (inverting the entire Hessian matrix), where large/ small circles indicate large/ small correlations. We can notice that for very low/high abilities, where students got all questions wrong or all questions correct, such as θ_2, θ_5, θ_{28} or θ_{30}, the correlations between these abilities and the difficulties of the questions are smaller than those of moderate abilities, such as θ_{10}, θ_{14} or θ_{20}. Therefore, these abilities are less affected by setting non-diagonal elements to zeros in the Hessian matrix. Hence, the variances resulting from both methods are almost the same, and thus the two posterior distributions become practically identical. On the contrary, the correlation between moderate abilities and difficulties of the question is high and hence affected more by this setting.



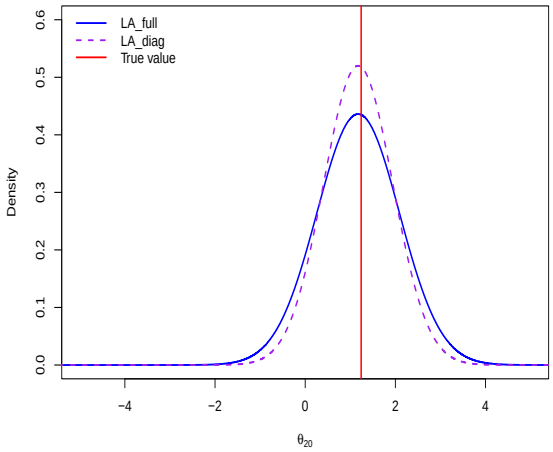
(a) Number of Correct Answers= 0



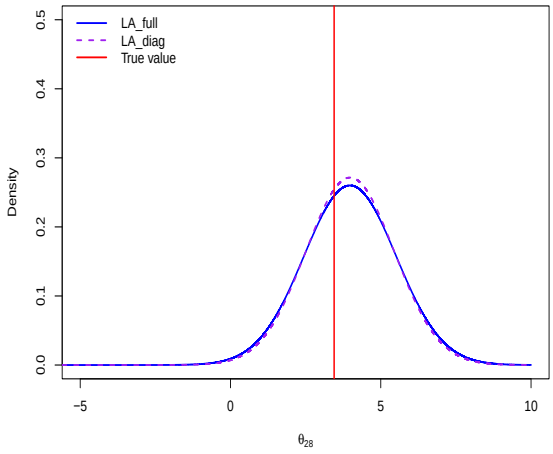
(b) Number of Correct Answers= 3



(b) Number of Correct Answers= 5



(d) Number of Correct Answers = 7



(d) Number of Correct Answers = 10

Figure 6.21: Posterior density resulting form full LA (LA_full) and diagonal LA (LA_diag) methods for selected examinees' abilities with different numbers of correct answers for sample size $n = 30$ and $m = 10$.

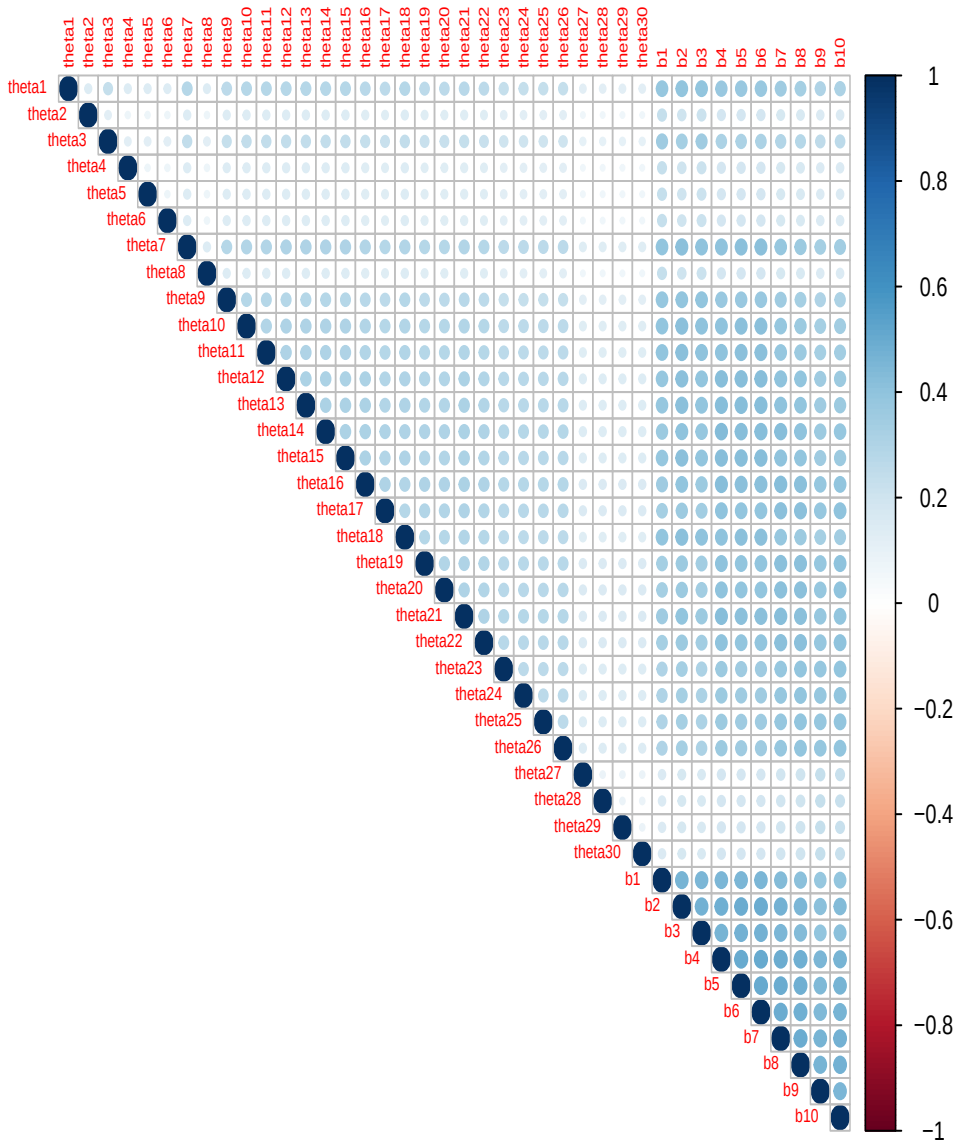


Figure 6.22: Correlation matrix between 1PL model parameters for sample size $n = 30$ and $m = 10$.

Although the diagonal LA method slightly underestimated the variance, the two posterior distributions resulting from both methods are very similar. In this example, the average difference between the diagonal and full LA variances is 0.23, which is still a relatively small difference. The divergence between the two resulting distributions from each method for the 30 students' abilities is calculated using the Jensen-Shannon divergence (JSD) method (0) and presented in Figure 6.23. As we can see, the values of the JSD range between 0.005 and 0.015, indicating that the difference between the two distributions is tiny for all students' abilities. In addition, Figure 6.24 shows 95% credible intervals for the abilities estimates of 30 students, as we can see their uncertainty estimates align closely, with only smaller uncertain intervals range for diagonal LA.

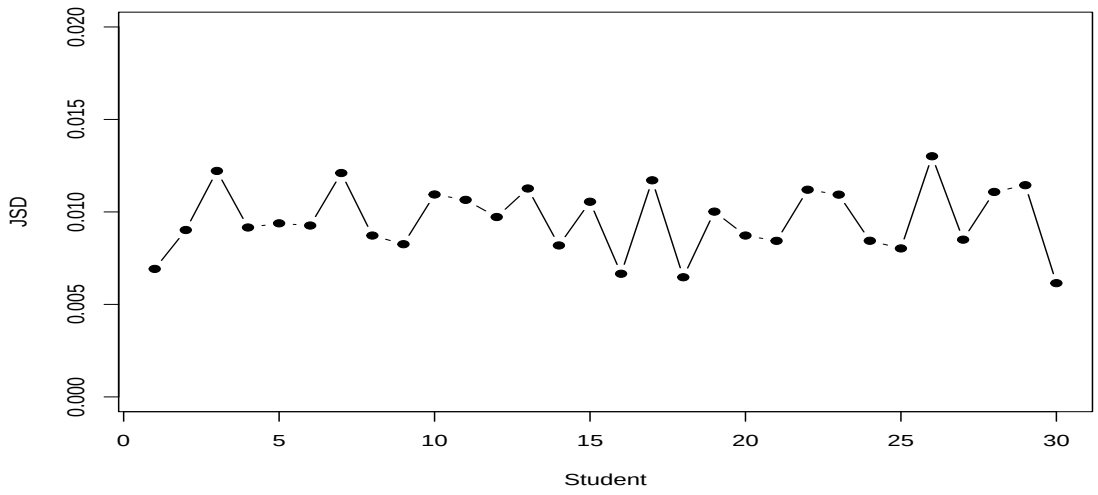


Figure 6.23: Jensen-Shannon divergence (JSD) method for each student's ability parameter obtained from full and diagonal LA for sample size $n = 30$ and $m = 10$.

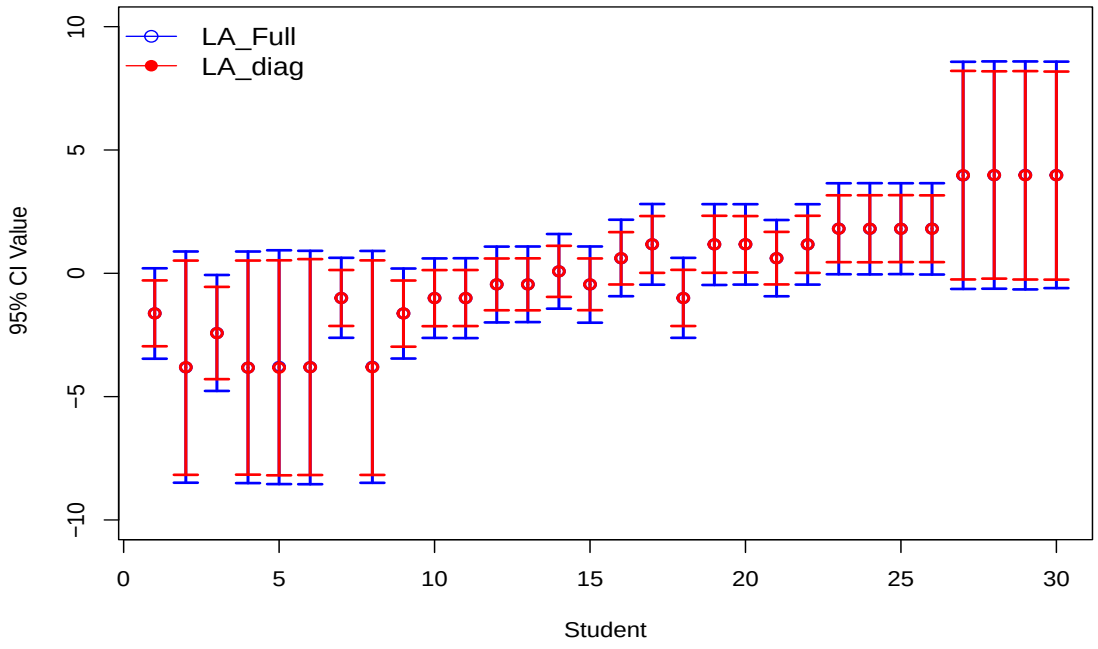


Figure 6.24: Posterior means and 95% credible intervals (CI) of the ability point estimates resulting from full and diagonal LA for sample size $n = 30$ and $m = 10$.

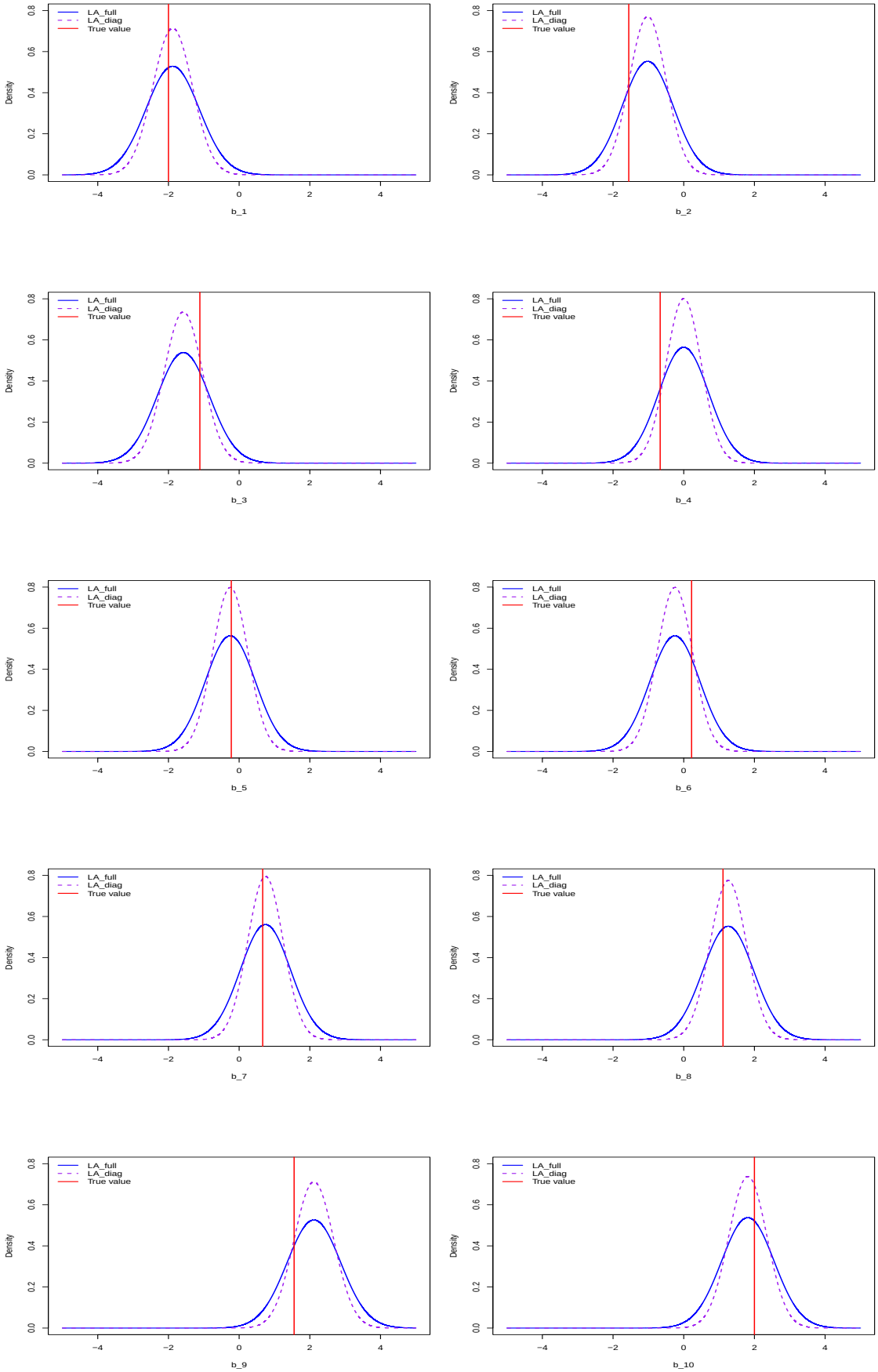


Figure 6.25: Posterior density resulting form full LA (LA_full) and diagonal LA (LA_diag) methods for the difficulty parameter, for sample size $n = 30$ and $m = 10$.

In terms of the difficulty parameter $\boldsymbol{b}$, the posterior distributions resulting from both methods for all questions' difficulties are present in Figure 6.25. This figure shows that the variation between the two distributions is larger than the abilities distributions, where the distributions resulting from diagonal LA are narrower. Also, Figure 6.26 shows 95% credible intervals, as we see the uncertain intervals range for diagonal LA is smaller than full LA. The reason is that the correlation between b's and $\theta's$ and between $b's$ themself are larger, as shown in Figure 6.22. Therefore, by setting non-diagonal to zero, we lose this information. This result is expected since the number of students is three times the number of questions, and hence we have more information about the difficulty of the questions. Furthermore, in this simulation study, the range of correct answers is between 7 and 22. Consequently, there are no extreme point estimates, such as a question answered correctly by all students or a question with no correct answers. However, the JSD values in Figure 6.27, which represent the measurement distance between the two distributions, are pretty small.

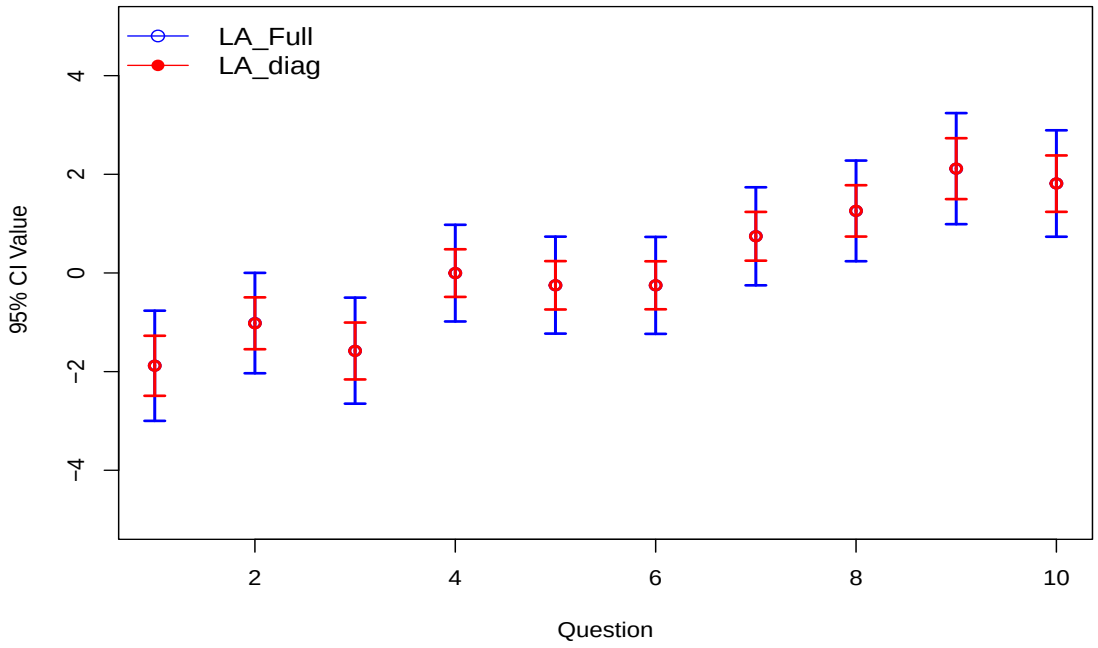


Figure 6.26: Posterior means and 95% credible intervals (CI) of the difficulty point estimates resulting from full and diagonal LA for sample size $n = 30$ and $m = 10$.

In general, the diagonal LA method underestimated the variance due to ignoring the correlations between $\theta's$ and $b's$ and setting the non-diagonal of the Hessian matrix to zero. This result can be seen more clearly for moderate abilities or difficulties. However, as presented in this study, the resulting variance is not massively

off. The following subsection will compare computation time and a general comparison of mean and maximum difference between the variance resulting from full and diagonal LA methods for large datasets.

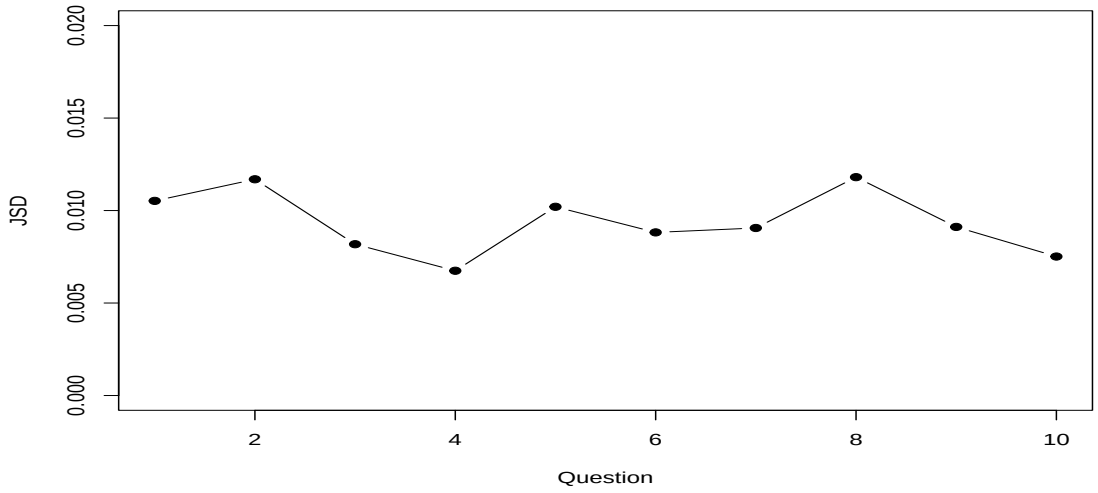


Figure 6.27: Jensen-Shannon divergence (JSD) method for each questions' difficulty parameter obtained from full and diagonal LA for sample size $n = 30$ and $m = 10$.

Comparison of Computation Time for Massive Datasets

The computational time of the diagonal LA method against the large sample size is investigated using five different datasets, the same datasets used in the block matrix 6.25. Therefore, the investigation is on increasing the sample size from $n = 1000$ to $n = 10,000$ with a test of length $m = 50$. Since the main difference between the full and diagonal LA is using the inverse of the diagonal of the Hessian matrix ($\boldsymbol{H}$) instead of the full matrix, the focus will be on computing the time of inverting the diagonal of $\boldsymbol{H}$.

The result of computing the time of inverting the diagonal of $\boldsymbol{H}$ is 0.001 seconds for each sample size. Which is computationally very cheap compared to other methods. As we have seen in 6.25, this can take up to 670 seconds for full LA and 8 seconds for the block LA (for $10,000$).

The more interesting result is that as we increase the sample size, the difference between the variance estimates resulting from the full H matrix and those resulting from the diagonal H matrix becomes negligible for both $\boldsymbol{\theta}$ and $\boldsymbol{b}$. Table 6.26 presents the mean and the maximum difference between the variance estimates resulting from both methods. As we can notice, both the mean and the maximum values are

minimal and only differ on the fourth decimals. This result indicates that the two variances become almost identical for larger datasets.

The main reason for this result is that the correlation between the model's parameters decreases by increasing the sample size. Table 6.27 summarises the maximum correlations between the parameters for the previous five sample sizes. The result shows that the maximum correlations between the abilities ($\boldsymbol{\theta}'s$) are very small (e.g. at $n = 1000$, the max correlation is 0.08) and get smaller by increasing the number of students, which can be negligible at all sample sizes. On the other hand, the maximum correlations between ($\boldsymbol{\theta}'s$) and ($\boldsymbol{b}'\boldsymbol{s}$) are slightly larger but get smaller as well by increasing the sample size and become negligible at $n = 10000$. Finally, the correlations between the difficulties are higher, where the maximum correlation is 0.55 and is not affected so much by increasing the sample sizes. Therefore, from the analysis of the correlations, we can find that setting the non-diagonal of the H matrix to zeros, and hence assuming there are no correlations between the parameters, will not affect the estimation of the variances since the correlations are already minimal in most cases.

Table 6.26: Mean and maximum values of the difference between the estimated variance resulting from diagonal and the full LA for the ability parameter $\boldsymbol{\theta}$ and the difficulty parameter $\boldsymbol{b}$ for different number of students

Number of Student	$\boldsymbol{\theta}$		$\boldsymbol{b}$	
	Mean	Maximum	Mean	Maximum
1,000	0.0093	0.0095	0.0096	0.0097
2,000	0.0048	0.0049	0.0050	0.0051
5,000	0.00195	0.00199	0.00200	0.00202
8,000	0.00116	0.00124	0.00120	0.00124
10,000	0.00092	0.00099	0.00098	0.0.00099

Table 6.27: Maximum correlation between the abilities, the abilities and the difficulties and between the difficulties.

Number of Student	$\boldsymbol{\theta}'\boldsymbol{s}$	$\boldsymbol{\theta}'\boldsymbol{s}$ & $\boldsymbol{b}'\boldsymbol{s}$	$\boldsymbol{b}'\boldsymbol{s}$
1000	0.08	0.21	0.54
2000	0.04	0.15	0.54
5000	0.02	0.10	0.54
8000	0.01	0.08	0.55
10000	0.009	0.07	0.55

Diagonal of Laplace approximation and the Optimisation Method

It is noticeable that, as we increase the sample sizes, the optimisation method that we use to find the maximum vector points of $\boldsymbol{\theta}$ and $\boldsymbol{b}$ and the Hessian matrix as well becomes computationally expensive. Where in this thesis, as mentioned earlier, the BFGS algorithm is used through the **optim** function in R.
Therefore, the main advantage of using the diagonal of the H matrix is that we can use the second derivative of the log posterior distribution and evaluate it at the maximum vector points. In other words, we can use the BFGS algorithm through the **optim** function in R to return only the maximum points, which in this case is computationally cheaper. The diagonal of the Hessian matrix can be found as follows:

$$diag(\mathbf{H}_p) = \left(\frac{\partial^2 p}{\partial \theta_1^2}(\hat{\theta}_1), \cdots, \frac{\partial^2 p}{\partial \theta_n^2}(\hat{\theta}_n), \frac{\partial^2 p}{\partial b_1^2}(\hat{b}_1), \cdots, \frac{\partial^2 p}{\partial b_m^2}(\hat{b}_m) \right),$$

where p is the log posterior distribution of the 1PL model in this case, n is the number of students and m is the number of questions. The resulting of the second derivative is then evaluated at maximum points $(\hat{\theta}_1, \cdots, \hat{\theta}_n, \hat{b}_1, \cdots, \hat{b}_m)$.

Figure 6.28 shows a time comparison of the four methods presented and discussed in this chapter. Full LA refers to using the full Hessian matrix. Block LA refers to utilising the formula of the 2×2 block matrix to invert the $\boldsymbol{H}$ matrix. Diagonal LA refers to using the diagonal of the H matrix returned by the *optim* function. Finally, gradient LA refers to using the second derivative of the log posterior to find the diagonal of the $\boldsymbol{H}$ matrix. The computational time presented in this figure is for running the whole process of the Laplace approximation methods to find the approximate point estimates and the corresponding covariance matrix (full LA and block LA) or the variance estimates (diagonal and gradient LA). The time is calculated in seconds for five different numbers of students, and a test of length $m = 50$.

The results show that as the sample size increases, the difference between the run times of the full LA and the other proposed methods becomes larger. In this case, the optimisation method and inverting the $\boldsymbol{H}$ matrix become computationally expensive. In the block and diagonal LA, the proposed strategies that were used to reduce the computational time of inverting the $\boldsymbol{H}$ matrix worked successfully. However, the optimisation method becomes expensive for a massive dataset, where $n = 5000$ or more. In the gradient LA method, the BFGS algorithm is only used to return the maximum point estimates. Therefore, the total run time becomes very cheap. For example, in a sample size of 10,000, the optimisation method took

approximately 330 seconds to return the maximum point estimates and $\boldsymbol{H}$ matrix. In comparison, it took 53 seconds to produce only maximum point estimates. Hence, from this comparison of the computational time, we can see that we can use the gradient LA method to find comparable results to the full LA in less than one minute for a massive dataset of a sample size of 10,000.

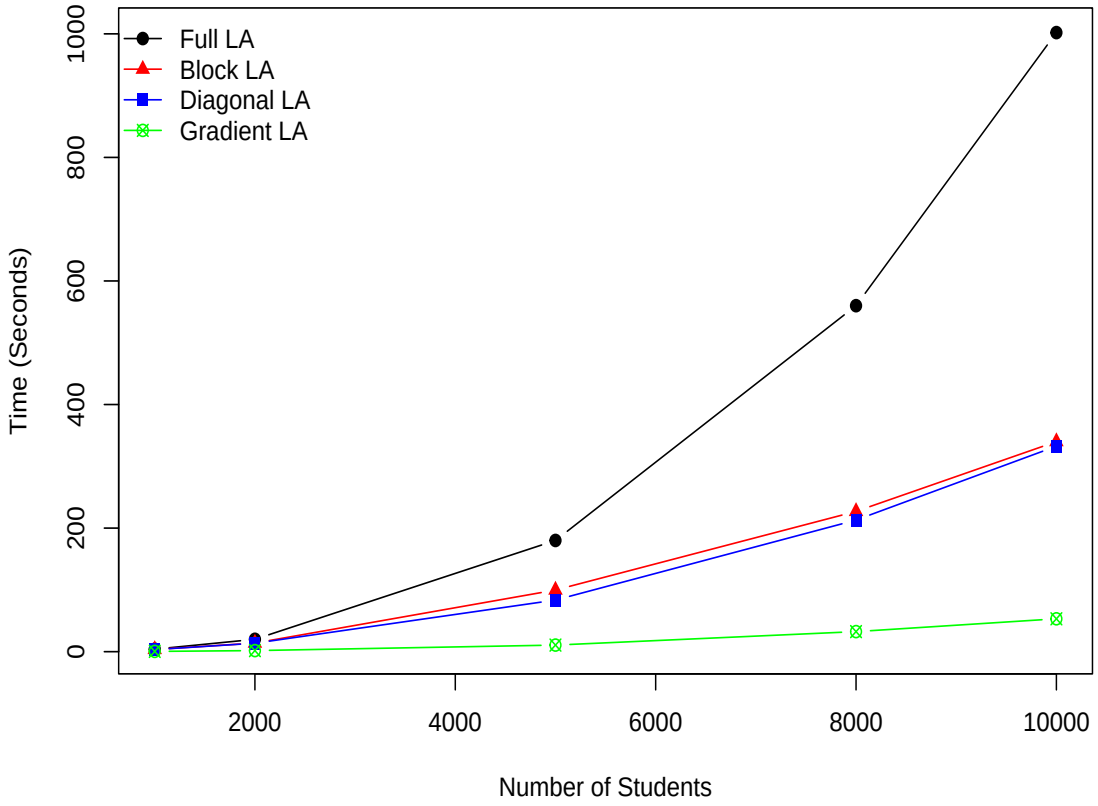


Figure 6.28: Time comparison of the whole process of the four Laplace approximation methods; Full LA: using the full Hessian matrix, Block LA: utilising the formula of the 2×2 block matrix to invert the $\boldsymbol{H}$ matrix, Diagonal LA: using the diagonal of the $\boldsymbol{H}$ matrix returned by the *optim* function and gradient LA: using the second derivative of the log posterior to find the diagonal of the $\boldsymbol{H}$ matrix.

6.4 Laplace Approximation on the Dynamic IRT Models

In educational measurement, we expect the estimation of item parameters in IRT models (e.g. the difficulty parameter $\boldsymbol{b}$) to change over time whenever one or a group of students answer the same test. This section considers the more common situation in a reality where the students complete a test at different times or even

on different days. This section formally introduces and illustrates a novel approach to the Laplace approximation method on the one-parameter dynamic IRT model.

6.4.1 Sequential Update Method

This subsection presents the idea of updating Laplace approximation (LA) sequentially when students finish the same test at different times or on other days.

The procedures of the LA method will be the same as explained earlier. The main difference is in determining the prior distribution of the difficulty parameter $\boldsymbol{b}$. Basically, the setting of the prior distribution will work in the following general way. First, before students answer a given test, it is assumed that there is no information about the difficulties of the questions, so a single weakly informative prior will be set for all questions. Hence, this setting of the prior distribution will be used for the first group of students who will take the test. The Laplace approximation method will be applied in the same way as explained before to estimate the ability of the students $\boldsymbol{\theta}$ and the difficulty of the questions $\boldsymbol{b}$.

This procedure will produce point estimates vector for both ability and difficulty parameters ($\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{b}}$) as well as covariance matrices ($\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{b}}$). In the next update, the resulting points vector of the difficulty parameter ($\hat{\boldsymbol{b}}$) and the covariance matrix ($\hat{\boldsymbol{\Sigma}}_{\boldsymbol{b}}$) will be used as prior distributions for the difficulty parameter. Formally, the prior distribution can be generated from a multivariate normal distribution with a mean equal to the points vector ($\hat{\boldsymbol{b}}$) and covariance matrix ($\hat{\boldsymbol{\Sigma}}_{\boldsymbol{b}}$);

$$\boldsymbol{b} \sim \mathcal{N}(\hat{\boldsymbol{b}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{b}}).$$

In other words, the posterior distributions of the difficulty parameter for every question based on one update based on a group of students becomes the prior distribution for the next update when a new group of students answer the same questions of a test. Thus, the prior distributions become more peaked (have a lower standard deviation) when more students answer the questions. Therefore, it is expected that with each update, the parameter estimates of the questions' difficulty become more tightly concentrated. This idea will be explained in detail with visual examples in the results section 6.4.2.

In the sequential estimation update of ability parameter $\boldsymbol{\theta}$, essentially the same likelihood is used, but, at each sequence, only the data of the group of students in that sequence is utilised. Therefore, the likelihood of 1PL model in this setting can be written as;

$$L(\boldsymbol{x}|\theta_i, b_j) = \prod_{i=1}^{ns} \prod_{j=1}^{m} \pi_{ij}^{x_{ij}} (1 - \pi_{ij})^{1-x_{ij}},$$

where ns is the total number of students in each sequence update, and m is the total number of questions.

The prior distribution for the ability parameter $\boldsymbol{\theta}$ is assigned the same for all students in each update.

The performance of the proposed method will be evaluated on three scenarios of simulated datasets. The simulated data present in study 3 for a sample size of 600 will be used here with three different levels of test length 10, 50 and 100. Each simulated data scenario will be repeated 20 times. Therefore, each test length will have 20 different datasets. Which will ensure there are a variety of students' abilities and questions' difficulties when evaluating the proposed sequential update method. The performance of the proposed method is assessed by comparing the results versus the complete results using full LA. Therefore, the approximation posteriors generated from full LA and the point estimates of both ability ($\boldsymbol{\theta}$) and difficulty ($\boldsymbol{b}$) parameters will be used as baseline results to compare the approximation posteriors and point estimates resulting from the sequential LA method.

6.4.2 Comparison Study of Sequential Update Method

The present study aims to investigate the performance and accuracy of applying the LA in sequential updates. The full LA update's performance and accuracy have been investigated and compared to one of the MCMC methods (M/Gibbs) in the previous section. The goal now is to compare the ability point estimates resulting from the LA sequential updates to those resulting from the LA fully update. The same criteria methods explained in 6.2 for the point estimates will be used; bias, RMSE and Kendall's τ, to evaluate the quality of sequential updates and measure the differences between the point estimates resulting from the two updates.

Simulated Data

The simulated data in comparison study 3 (6.2.1) for a sample size of 600 will be used to investigate the performance of the sequential LA. The study will consider three different test lengths; short ($m = 10$), moderate ($m = 50$) and long ($m = 100$). In each setting, the simulated dataset will be repeated 20 times to ensure there is a variety of student abilities levels. Hence, the average criteria methods will be calculated.

Since it is assumed that students answer a given test in a dynamic system, the total number of students will be divided into groups. For this setting, four

different scenarios of grouping students are assumed. In other words, the total number of students will be divided into four different block sizes. The first scenario is considered to have a block size of 20 students ($ns = 20$). After this group finish the test, they receive the results of their ability immediately, and we get an update on the difficulties of the questions. Consequently, the result of the difficulties of the questions can be stored and then used as prior distributions for the questions when the following 20 students take the test. We keep updating our prior beliefs about the difficulties of the questions and set the same prior distribution for the students abilities until all the students finish the test. The same analysis is repeated for a block size of 50, 100 and 200 students. A summary of simulated datasets is presented in Table 6.28.

Table 6.28: Simulated Data for Sequential Update Method

Variable	Setting
Number of Students	600
Number of Items	$10, 30$ and 100
Bock Sizes	$20, 50, 100$ and 200

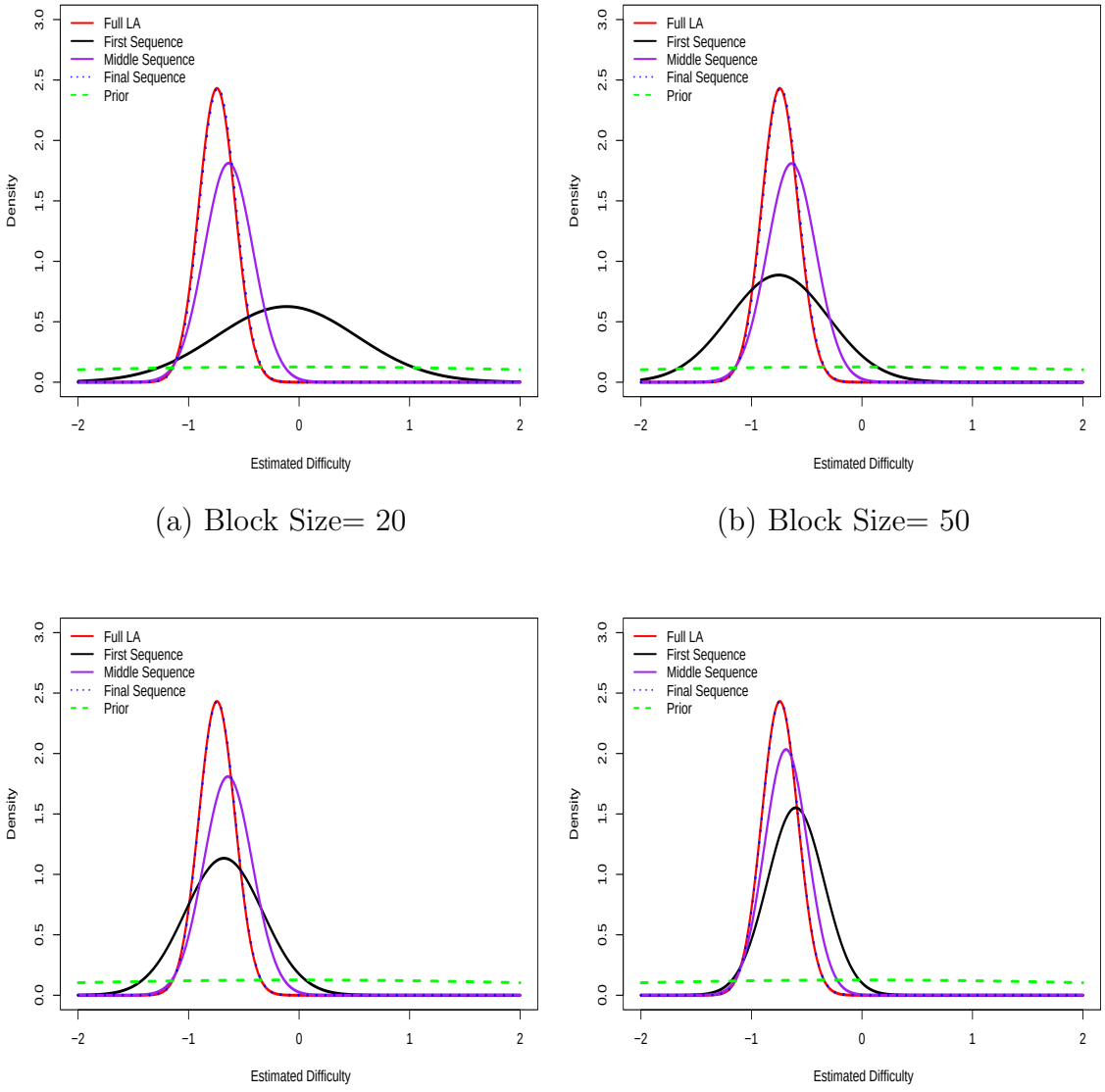
Sequential Update Results

This section discusses the comparison results between the full LA update (6.1) and the proposed sequential LA update (6.4.1). First, the change in the difficulty parameter estimates during the sequence update will be considered in terms of point estimates and posterior distributions. Then, the previously mentioned measurements, bias, RMSE and Kendall's τ will be used to measure the differences between ability point estimates resulting from each update.

Figure 6.29 shows posterior distributions of one randomly selected difficulty parameter estimate for sequential LA update at three different sequence updates and a posterior distribution of full LA. The figure presents four different scenarios of the number of students in each block sizes update. At the beginning of the analysis, there is no information about the difficulty of the parameter, so it is expressed as a weakly non-informative prior distribution, which is almost flat (dashed green line). After the first group of students finish the test, more information about the difficulty of the question becomes available. Hence, the prior, which is the posterior at the current update, becomes more peaked with smaller standard deviation. In addition, we notice that the number of students in each block size affects the prior distribution. For example, the prior distribution at the first sequence (black line) for a block size of 20 students is wider than the one of a block size of 100 students. This

effect is expected because the more students answer the test, the more information about the difficulty of the questions. At the final sequence update, the posterior distribution becomes identical to the posterior distribution resulting from the full LA update. Therefore, in this sequential update, priors of the difficulty of the questions are specified from the data, not from our beliefs and become more informative when more students answer the test.

Regarding the point estimates of all the difficulties of the questions, Figure 6.30 shows the point estimates of the difficulty parameters $\boldsymbol{b}$ for sequential LA update at first, middle and final sequences and full LA update for three different test lengths ($m = 10, 50$ and 100). As an example, the figure presents the case of a block size of 20 students, and more results for other block sizes can be found in the Appendix B. It is clear that the difficulty point estimates in the first sequence, where only one group of students has finished the test, are pretty far from the target point estimates resulting from the full LA in all three test lengths. However, the difficulty level of the questions (high/low) is approximately correct. In addition, the number of questions does not primarily affect the difference between the point estimates in each sequence. For example, the average absolute mean difference between point estimates in first and middle sequences are 0.44, 0.45 and 0.46 for the test of lengths 10, 50 and 100, respectively.



(a) Block Size= 20

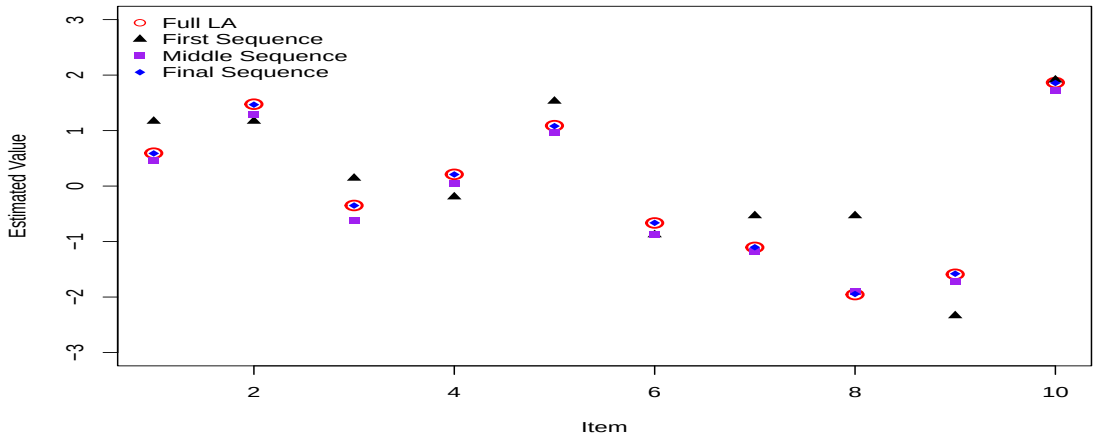
(b) Block Size= 50

(c) Block Size= 100

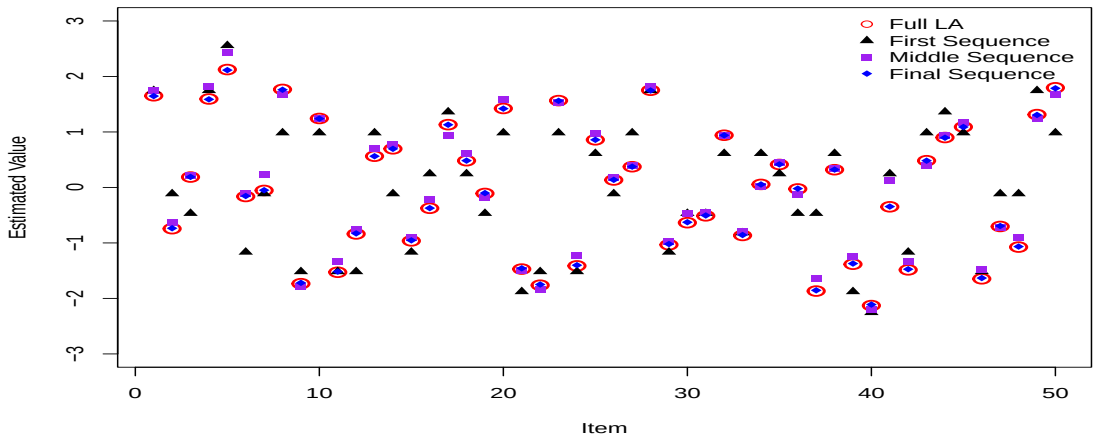
(d) Block Size= 200

Figure 6.29: Posterior distributions of a difficulty parameter estimate for sequential LA update at first (black line), middle (purple line) and final (blue dotted line) sequences and full LA update (red line). The green dashed line represents the prior distribution.

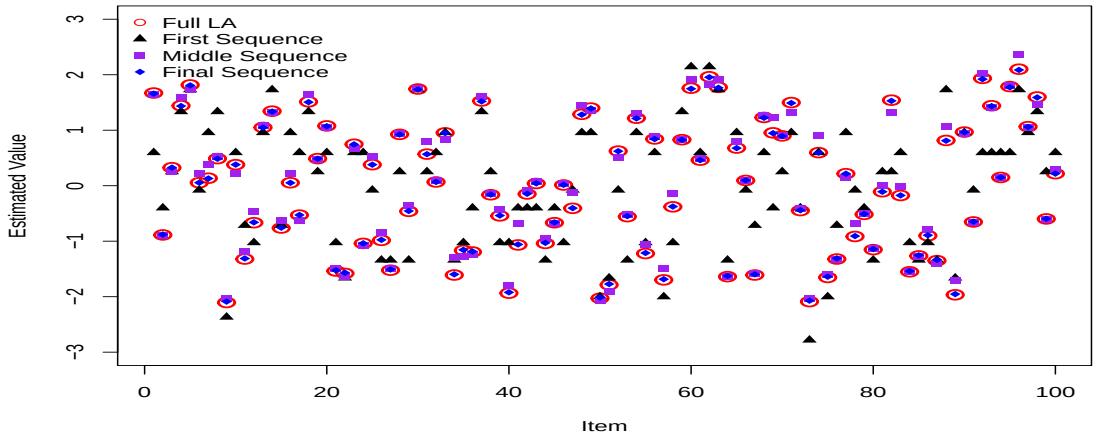
After we have some ideas about updating the estimate of the difficulty parameters, we can compare the ability point estimates resulting from the sequential LA update with those points resulting from the full LA updates. Table 6.29 presents a summary of average bias, RMSE, and Kendalls τ values between the estimated points resulting from both updates for the ability parameter $\boldsymbol{\theta}$ for three different number of questions (10, 50 and 100) and four different block sizes of students (20, 50, 100 and 200). The average biases show that sequential LA is slightly underestimated the students' ability. However, the differences are very small and get smaller by increasing the



Number of Items= 10



Number of Items= 50



Number of Items= 100

Figure 6.30: Point estimates of the difficulty parameters for sequential LA update at first, middle and final sequences and full LA update for three different test lengths ($m = 10, 50$ and 100). The block size of the sequential update is 20.

number of questions. The biases also decrease as the number of students in each block size increases in most cases.

The same pattern was found across all conditions for RMSE values. First, the RMSE values are minimal even for a short test ($m = 10$) and small block size (20); 0.11, indicating that the sequential LA method accurately estimates students' ability. The estimation results get more accurate by increasing the number of questions, as noted in the smaller RMSE values. In addition, the block sizes also affect the values of RMSE, as these values decrease by increasing the number the size of the blocks. Therefore, it is noticeable that the estimation can be improved by increasing the information about the questions in terms of test lengths and block sizes.

In educational settings, teachers or researchers usually are more interested in the qualitative inference for the students' ability than quantitative inference. In other words, the correct order or rank of students' ability is more curious than the exact ability values. This can be measured by Kendall's τ, which is calculated for all conditions and presented in Table 6.29. We can see that Kendall's τ value is very large, even for a small block size (20) and short test ($m = 10$); 0.96. These large values indicate that the sequential LA update method orders students' ability the same way as the full LA method, which almost becomes identical for the test of lengths 50 and 100.

Table 6.29: Average bias, RMSE, and Kendall's τ values between the estimated points resulting from the sequential LA update and the full LA update for the ability parameter $\boldsymbol{\theta}$ for sample size $n = 600$ and different number of items.

Number of Items	Block Size	Bias	RMSE	Kendall's τ
10	20	-0.0087	0.11	0.96
	50	-0.0097	0.10	0.96
	100	-0.0095	0.08	0.97
	200	-0.0080	0.06	0.97
50	20	-0.0071	0.062	0.99
	50	-0.0077	0.062	0.99
	100	-0.0079	0.058	0.99
	200	-0.0082	0.046	0.99
100	20	-0.0032	0.042	0.99
	50	-0.0035	0.041	0.99
	100	-0.0038	0.039	0.99
	200	-0.0045	0.031	0.99

As shown in Figure 6.31, the full and final sequential LA point estimates align very closely for all test lengths, which confirmed the previous numerical results.

Moreover, there is no evidence in the figure if the differences between the two methods occur for very high/low or moderate ability. Also, Figure 6.32 displays the point estimates resulting from each method versus the true values. As shown in this figure, the sequential LA method yields estimate comparable to those from the full LA. The Appendix B shows average bias, RMSE, and Kendall's τ values between the estimated points resulting from the sequential LA update and the true values for the ability parameter $\boldsymbol{\theta}$ across all conditions, which agrees with all numerical and graphical previous results.

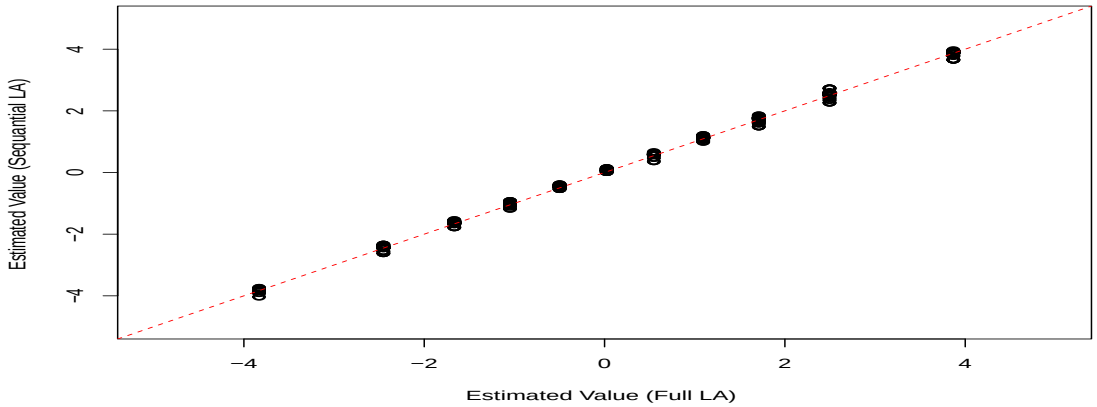
Table 6.30: Average absolute mean and maximum values of the difference between the estimated points resulting from the sequential LA update and the full LA update for the ability parameter $\boldsymbol{\theta}$ for sample size $n = 600$ and different number of items.

Number of Items	Block size	Mean difference	Maximum difference
10	20.00	0.076	0.39
	50.00	0.069	0.31
	100.00	0.059	0.20
	200.00	0.041	0.12
50	20.00	0.050	0.168
	50.00	0.049	0.160
	100.00	0.045	0.130
	200.00	0.340	0.091
100	20.00	0.034	0.110
	50.00	0.032	0.100
	100.00	0.030	0.085
	200.00	0.024	0.063

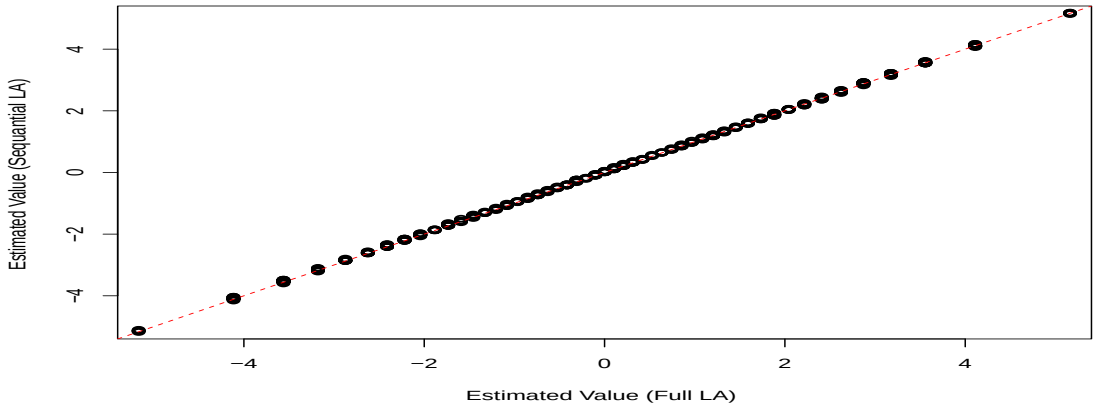
To investigate more about the main differences between the ability point estimates resulting from the two methods, Table 6.30 shows the average absolute mean and maximum differences across all conditions. The average of the absolute mean differences between the ability point estimates resulting from each method ranges between 0.076 and 0.024 across all conditions. We can see that these values are decreased incrementally by increasing the number of questions and the block sizes. These differences are very small, even when there is less information about the test for a short test ($m = 10$) and a block of size 20. The average absolute maximum differences range between 0.39 and 0.063 across all conditions. The largest values appear for the shortest test and a block of size 20, which is still acceptable given the amount of information about the test.

The largest considerable maximum differences occur in the early sequences updates in most cases. For example, Figure 6.33 shows one ability parameter estimate at the first sequence and another ability parameter estimate at the last sequence update,

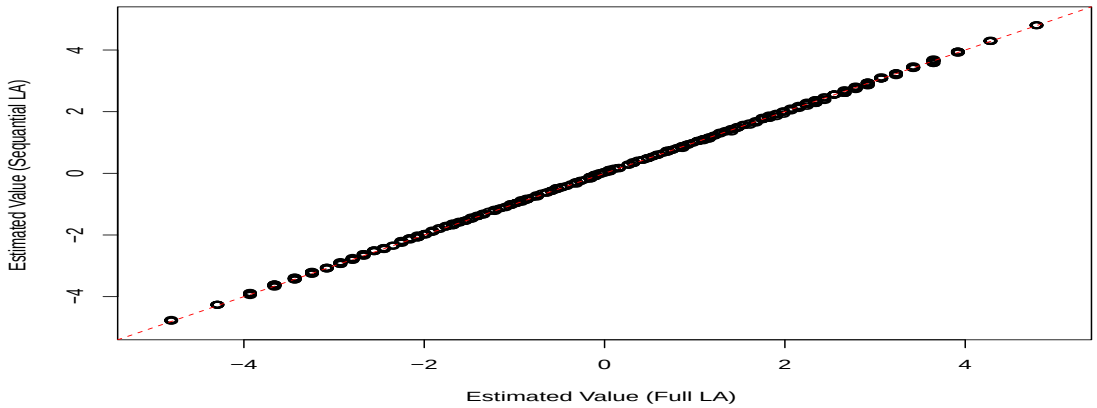
where both parameters are randomly selected. In the first sequence, we can see that the mode of the posterior distribution (point estimate) resulting from the sequential update is not equal to the mode of the posterior distribution resulting from the full update. Moreover, the estimation result varies between the block sizes. However, as we can notice, these differences are not quite large. The posterior distributions resulting from sequential LA are only shifted slightly to the right, with almost the same variance for all the block sizes. On the other hand, the two posterior distributions from both updates become identical for all the block size cases in the last update. More examples about the ability parameters estimates at the first sequence can be found in the Appendix B (Figure B.8).



Number of Items= 10

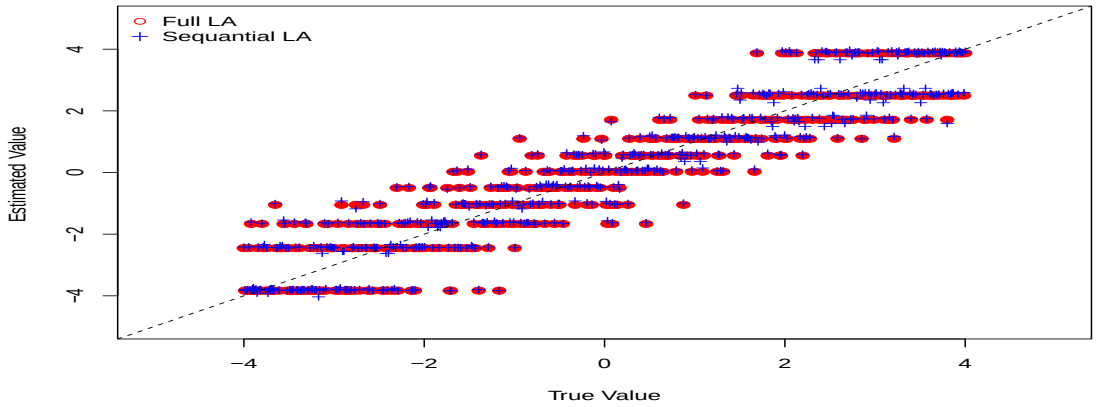


Number of Items= 50



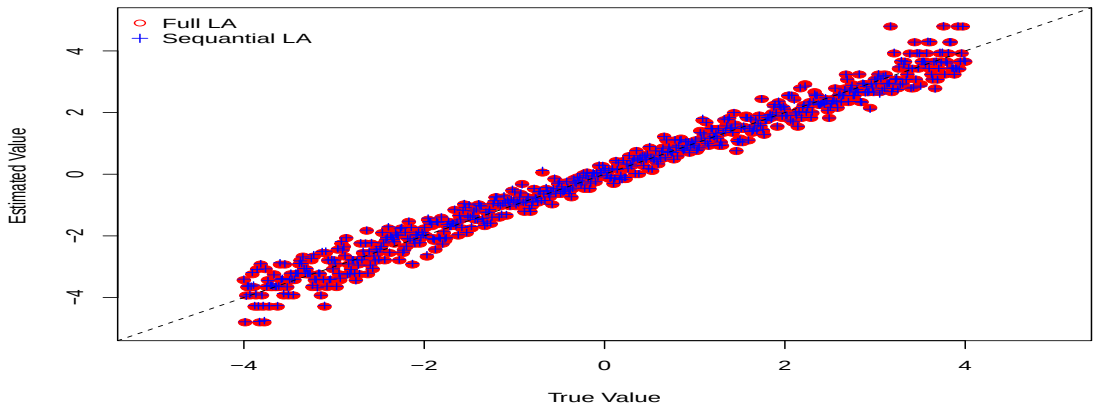
Number of Items= 100

Figure 6.31: Point estimates of the students' abilities resulting from full LA update method versus sequential LA method, for a block size of 20. The red dashed line illustrates the equality line.



Number of Items= 10

Number of Items= 50



Number of Items= 100

Figure 6.32: Point estimates of the students' abilities resulting from full and sequential LA method update methods versus true values, for a block size of 20. The black dashed line illustrates the equality line.

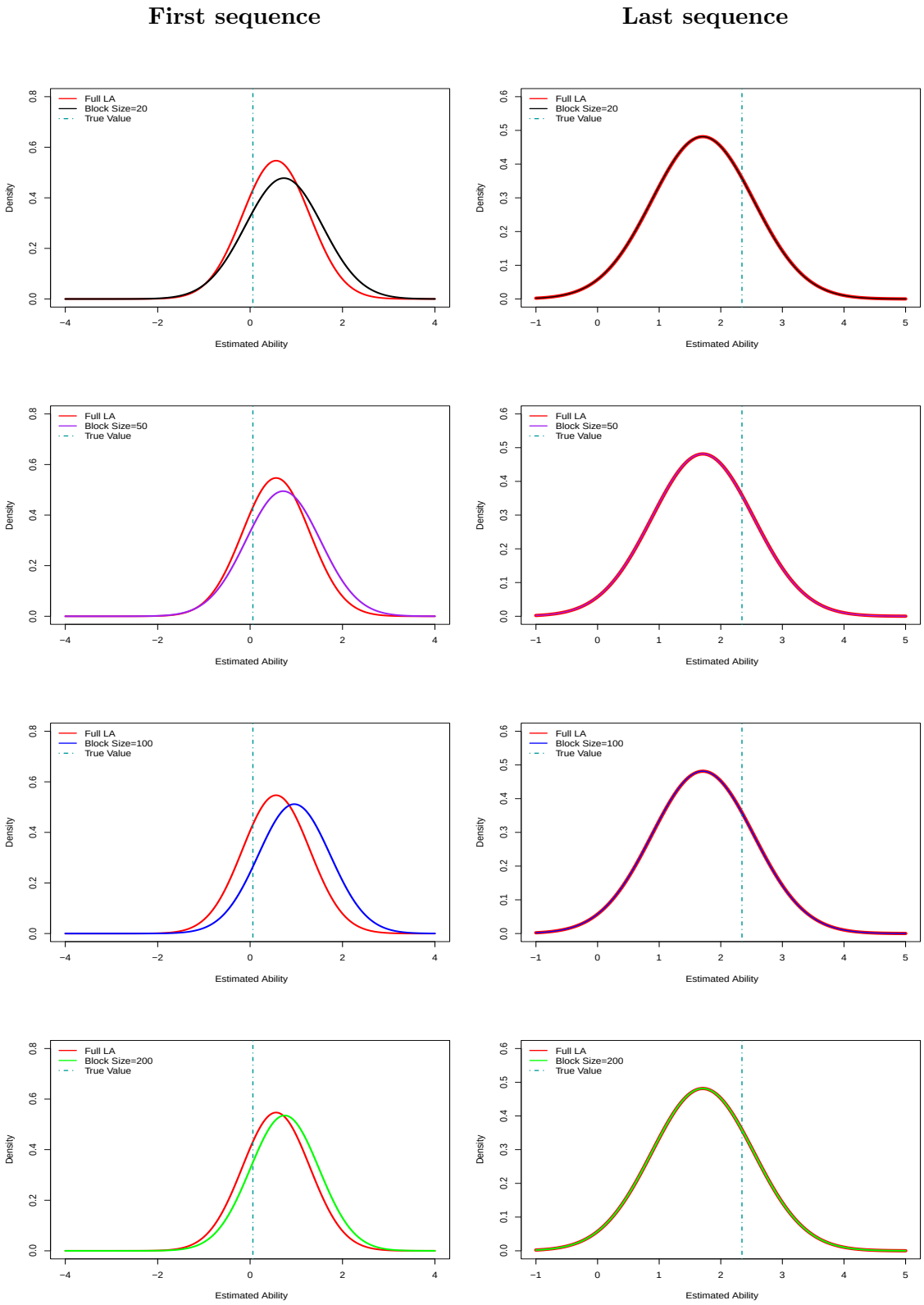


Figure 6.33: Posterior distributions of an ability parameter at the first sequence update and an ability parameter at the last sequence update for four different block sizes.

6.4.3 Summary and Discussion

This section has introduced a novel approach for the sequential Laplace approximation method in a Dynamic IRT model. The main idea of this method is that the difficulty parameter estimates are updated sequentially from the data every time new students answer the test. Hence, the results are used as prior distributions for the difficulty parameters to estimate the following students' abilities. The idea of this method has been illustrated by the comparison study to the full LA update method. Based on the criterion measurements presented in this study, the sequential LA method resulted in ability point estimates comparable to those from the full LA. However, the most considerable differences between the point estimates resulting from each method appeared for the early sequence updates.

The advantage of this method is that it can be a helpful tool for research problems for big data or online inference. As described earlier, the procedure of sequential LA is to summarise current information regarding difficulty parameters and store the data in terms of means ($\hat{\boldsymbol{b}}$) and covariance matrix ($\hat{\boldsymbol{\Sigma}}_{\boldsymbol{b}}$) to use these as prior distributions when new students answer the test. Therefore, we only need to store the $(m * m)$ covariance matrix in this setting, where the number of questions in the most real-life scenario is not very massive. Hence, this procedure will help avoid storing large data sets in computer memory since keeping the previous student's information is unnecessary. Moreover, the likelihood only needs to be calculated for the new students to estimate their abilities, making this method very cheap for online inference.

The next chapter will consider the application of the Laplace approximation to the non-dynamic and dynamic 1PL IRT model for the real dataset.

Chapter 7

General Aptitude Test Case Study

7.1 Data

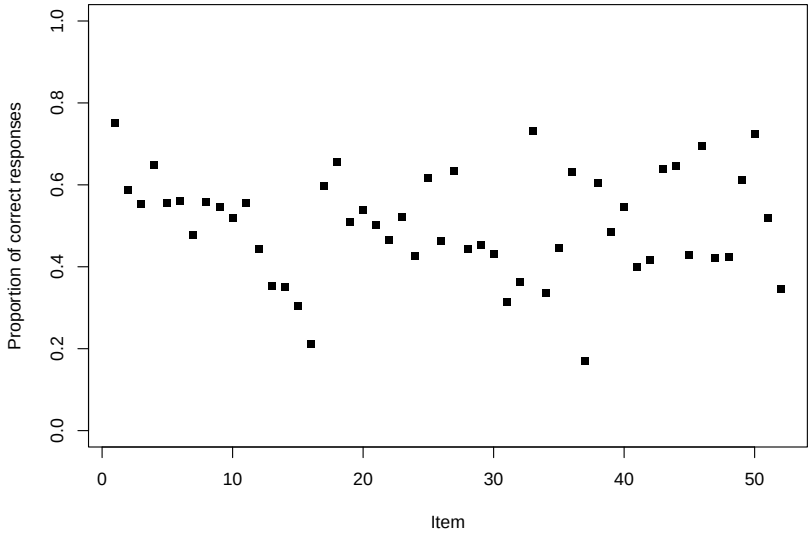
This case study considers an application of the Laplace approximation method explained in Chapter 6 on a real dataset. This data was obtained from the General Aptitude Test (GAT), a test that targets high school graduates and is used for university admission purposes in Saudi Arabia. See Alghamdi and Al-Hattami (2014) and Dimitrov and Shamrani (2015) for more information about the GAT test. As described in the test website (`https://etec.gov.sa/en/productsandservices/Qiyas/Education/GeneralAbilities/Pages/default.aspx`), the GAT tests the general ability to learn regardless of any specific skills in a particular subject or topic. The test consists of two parts: verbal and quantitative. The verbal section (GAT-V) consists of 52 multiple-choice items divided into four domains: reading comprehension, sentence completion, verbal analogy, and contextual error. The quantitative section (GAT-Q) consists of 44 multiple-choice items, which focus on mathematical problems and are divided into five domains: arithmetic, geometry, algebra, statistical and analytical and comparison questions. In each section, the questions are supposed to be arranged in order of difficulty level from easiest to most difficult. The data includes a sample of 4000 students and the 96 dichotomously scored multiple-choice questions, but there are 652 students who left some questions unanswered. Hence the analysis is based on 3348 students who completed all 96 questions.

In an initial exploration of the GAT data, Figure 7.1 presents the proportions of correct responses by the number of items for the two sections of the GAT. We can see that item 1 seems to be the easiest question with the highest proportion of correct responses for both sections. On the other hand, item 37 appears to

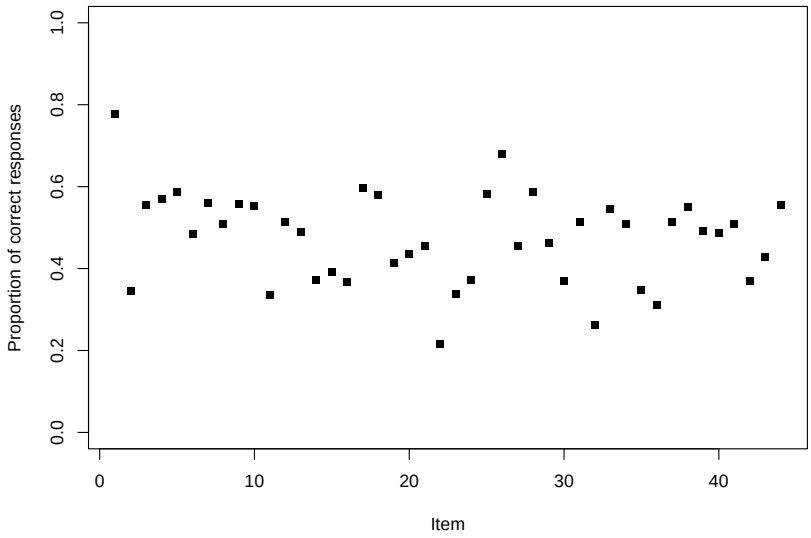
be the most difficult question for the verbal section (GAT-V), with only 16 % of students answering this question correctly, and item 22 is the hardest question for the quantitative section (GAT-Q) with only 21% of students answering correctly. The average proportions of correct responses are 50% for the GAT-V and, 47 % for the GAT-Q.

Figure 7.2 shows the histogram of the students' total number of correct answers for the GAT data set with all questions (GAT-all), the verbal section (GAT-V) and the quantitative section (GAT-Q). As mentioned early, the total number of questions for GAT-V is 52. We see that the peak of the histogram is between 20 to 25, with about 600 students getting that number of correct questions. The total number of GAT-Q questions is 44, and we can see that the peak of the histogram with almost 800 students is between 10 to 15. Moreover, in the GAT-Q, about 300 students answered only 10 questions or less, while in GAT-V, about 100 students answered 10 questions or less. For GAT-all, where the total number of questions is 96, Figure 7.2 shows that the total number of questions answered correctly in the peak of the histogram ranges between 20 to 60, where these total number is noticeably decreased after 60 questions. From the initial analysis, we can assume that the ability of students to answer the GAT- V section is higher than answering the GAT-Q section. However, we cannot directly compare them since the total number of questions is different in each section.

In the following sections, a formal analysis will be applied to the General Aptitude Test data set to estimate the difficulty of the questions for each section (GAT-V and GAT-Q), the abilities of the students for each section separately and the general students' abilities (GAT-all).



GAT-V



GAT-Q

Figure 7.1: Proportions of correct responses by item for the General Aptitude Test (GAT) dataset. The upper panel represents the proportions of correct responses for the verbal section (52 questions), and the lower panel represent the quantitative section (44 questions).

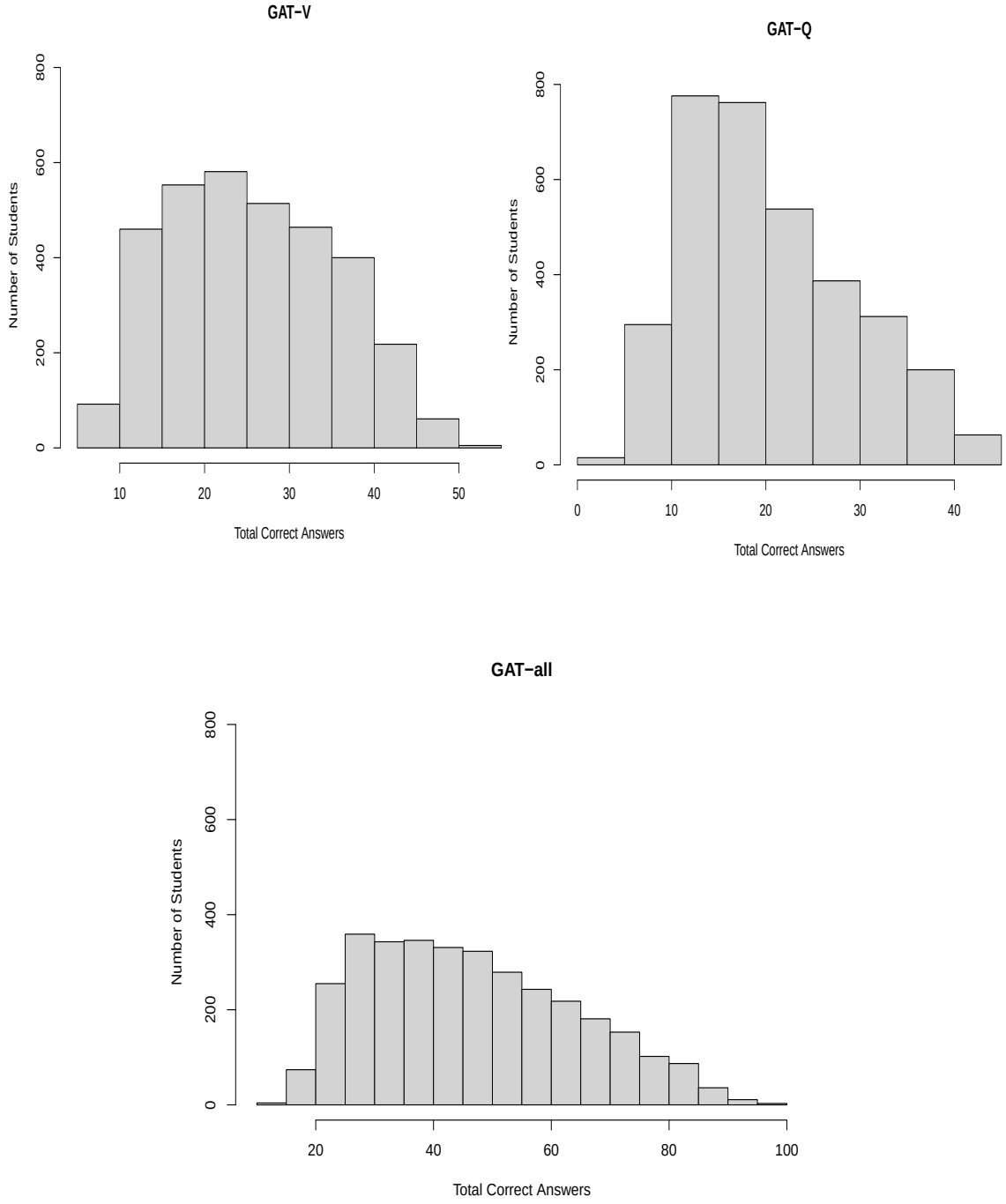


Figure 7.2: Histogram of the students' total number of correct answers for the General Aptitude Test data set with all questions (GAT-all), the verbal section (GAT-V) and the quantitative section (GAT-Q).

7.2 Method

The main objective of this study is to estimate students' ability for the GAT dataset in a reasonable time, assuming that same inference and the conclusion must be made in real-time. In reality, both teachers and students are interested in immediate test

results. Compared to both MCMC (Chapter 4) and the sequential Monte Carlo method SMC (Chapter 5), Laplace approximation allows for computing results much more quickly. From the simulation and comparison studies in Chapter 6, we can see that Laplace approximation is an effective inference method in terms of accuracy and time consumed for small and large datasets for the IRT model. Therefore, the Laplace approximation will be applied to the GAT dataset to find the estimates of the ability of the students and the difficulty of the questions. It has been mentioned earlier that the frequentist approaches require a large sample size. Therefore, to use frequentist approaches in online inference, the student who has answered the test questions has to wait for other students to finish the test to get feedback on their ability. Since, in this case study, there is a large sample size, and we have all the data at one point, the Maximum likelihood approach will be implemented in this section beside the MCMC method for comparison purposes to check the accuracy of the Laplace approximation method.

To explore the performance of the Laplace approximation method (LA) in the GAT data set, the Rasch model (1PL) described in Chapter 3 will be used to model the data. The equation that expresses the probability of correct answer as a function of examinees' abilities (θ_i) and items' difficulties (b_j), can be written as:

$$p(X_{ij} = 1) = \frac{exp(\theta_i - b_j)}{(1 + exp(\theta_i - b_j))},\ \theta_i\ and\ b_j \in \mathbb{R},$$

where X_{ij} is a binary response given by student i to item (1 for correct, 0 for incorrect). The subscript $j = 1, 2, ..., m$ represents the number items, and m is the total number of items. The subscript $i = 1, 2, ..., n$ represents the number of examinees where n is the total number of examinees, which is 3348 in this data set. In this model, it is assumed that all items measure a single ability θ. In terms of questions, the study will be carried out in three different settings; estimating the ability parameters for the verbal section (52 items), estimating the ability parameters for the quantitative section (44 items) and estimating the ability parameters for all GAT test questions (96 items). Therefore, if one is interested in diagnostic testing according to which type of skill is mastered or not, the student's ability for each section can be compared. Then, students can know immediately which section they have the low ability, and how it can affect their overall ability.

To explore the LA results, the output is compared to the ability parameters' estimation resulting from a frequentist fitting of the Rasch model. In the frequentist IRT, the item parameters and person's ability parameters are estimated in two different steps (Feuerstahler, 2018). In the first step, item parameters are estimated

using one of the most common methods; marginal maximum likelihood (MML) (Bock and Aitkin, 1981). The process starts by assuming that the ability parameter θ follow a distribution, usually standard normal. Hence, given the initial θ distribution and the item response, the marginal likelihood of the item parameters is estimated. The next object is to find the item parameters (difficulty) estimates where the likelihood function reaches a maximum point. The second step is to estimate the ability parameters given the previously estimated result of item parameters. There are three common approaches: maximum likelihood estimation (MLE), expected a posteriori (EAP), and maximum a posteriori (MAP). The resulting ability estimates from all three methods are point estimates. For more details about the IRT parameters estimate in frequentist see DeMars (2010). In this case study, the difficulty parameters will be estimated using MML, and then the ability parameters will be estimated using MLE. The process will be implemented using a R package **irtoys** (Partchev et al., 2017).

Furthermore, the MCMC method is used to evaluate the accuracy of the approximation results obtained from the Laplace approximation in this real data set. The Gibbs sampler within the Metropolis algorithm (M/Gibbs), which is explained in detail in Chapter 4, will be used in this case study. To improve the accuracy of MCMC estimates, the M/Gibbs algorithm will run for a large number of draws. The MCMC point estimates will be based on the average of these draws' values. See Luo (2018) for an example of applying and comparing MML and MCMC methods in the GAT-V data for a generalized partial credit model which is a polytomous IRT model.

7.3 Results

This section provides parameter estimation results for the Rasch model for the GAT data set for the three methods MLE, MCMC and LA. The result is based on implementing the three methods in three different scenarios; estimating the ability and the difficulty parameters for GAT-V, GAT-Q and GAT-all. In each setting, the three methods are executed independently, and the numerical and graphical results are recorded.

In the Bayesian framework, for MCMC and LA application, the choice of the prior distributions could be made based on the previous GAT test results, since the test is repeated several times during the year, and the difficulty levels of the questions are assumed to be equal for all tests. However, due to the lack of information in this study about the previous test results and the privacy of the test questions, the prior

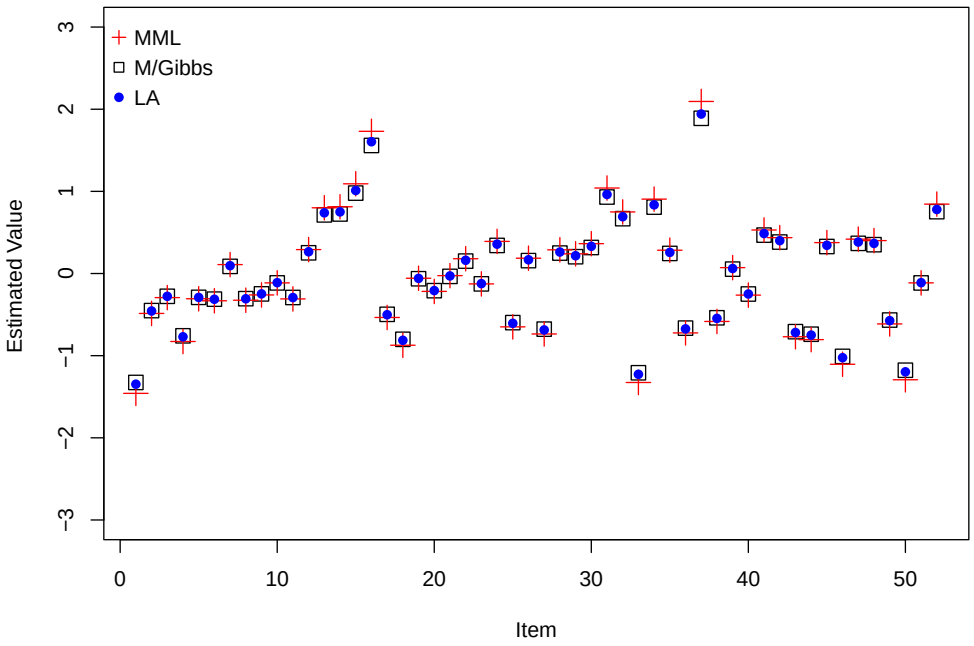
distributions for items difficulty and students ability are chosen to be the same as in the simulation studies. The prior distributions are defined as follows:

$$\theta_i \sim N(0, \sigma_\theta^2 = 10) \qquad \forall i$$

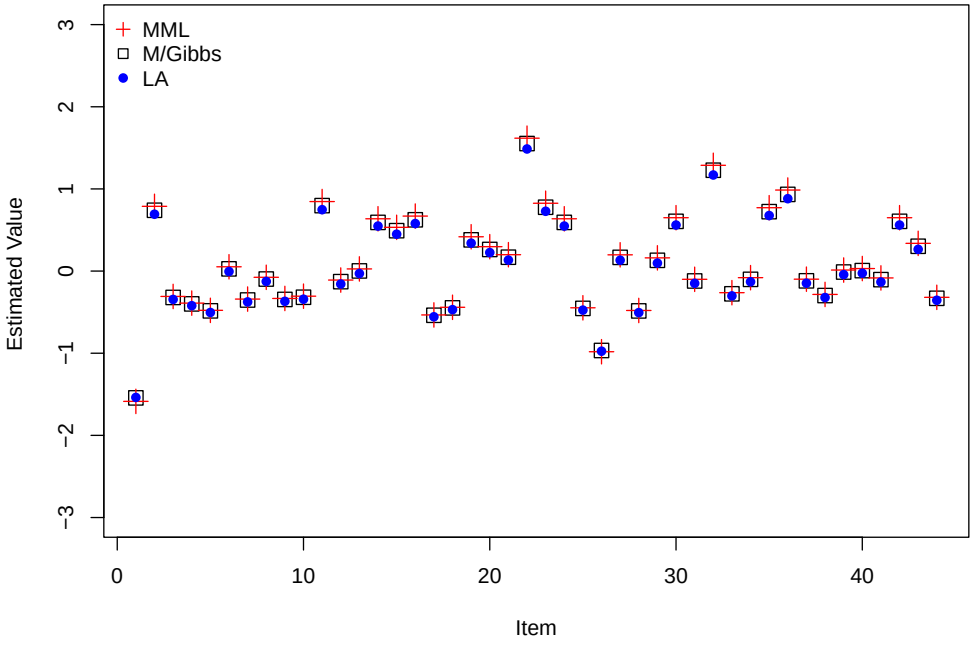
and

$$b_j \sim N(0, \sigma_b^2 = 10) \qquad \forall j$$

The result of the MCMC method is based on running the Gibbs sampler within the Metropolis algorithm for a run of a hundred thousand samples. Hence, the mean of the posterior distribution for each parameter is calculated by taking the average of the drawn values after burn-in.



GAT-V



GAT-Q

Figure 7.3: Difficulty Estimates for the General Aptitude Test (GAT) questions, where MML refers to the result of using marginal maximum likelihood, M/Gibbs refers to the result of using the Gibbs sampler within the Metropolis algorithm for MCMC method, and LA indicates the Laplace approximation method. The upper panel represents the estimate result of the verbal section (52 questions), and the lower panel represent the quantitative section (44 questions).

Figure 7.3 provides a visual presentation of the difficulty parameter estimates for the verbal section (GAT-V) and the quantitative section (GAT-Q) based on all participating students. All three methods seem to estimate the difficulty parameter in the same order. The resulting difficulty estimates are almost arranged between -2 to 2, with most of them between -1 to 1. In the GAT-V, question 1 appears to be the easiest question, with the difficulty estimation being -1.33, -1.45 and -1.34 for M/Gibbs, MML and LA respectively. In this section, question 37 is the hardest question, with difficulty estimation being 1.88, 2.09 and 1.94 for M/Gibbs, MML and LA respectively. On the other hand, In the GAT-Q, question 1 is the easiest question, with the difficulty estimation being -1.54, -1.58 and -1.53 for M/Gibbs, MML and LA, respectively. The hardest question for this section is question 22, with difficulty estimation being 1.55, 1.61 and 1.48 for M/Gibbs, MML and LA, respectively. The resulting estimation from MML is more extreme because the use of the prior distribution pulls both M/Gibbs and LA toward zero (the mean of the prior distribution). This can be more useful when there are some extreme points where the students answer all questions right or wrong. In this case, it can be challenging to find the maximum point using MML, where the results may go to infinity. However, the conservative estimates, in this case, do not matter since these estimates are usually used for ranking students or ordering the difficulty of the questions. For that purpose, Figure 7.3 shows that the ordering of the questions' difficulties is similar in all the methods.

We can notice that the results of the descriptive analysis in Figure 7.1 are also validated by the estimation resulting from the three methods. Where items 1 and 37 are the most difficult and the easiest, respectively, for the GAT-V, and items 1 and 22 are the most difficult and the easiest, respectively, for the GAT-Q. However, the estimation result of the items' difficulty does not agree with the assumption that questions are arranged according to the difficulty, from the easiest to the more difficult in each section. For example, we can see in the GAT-V section that question 16 is the second hardest question, while questions 33 and 50 are the second and third easiest questions. Similarly, in the GAT-Q, the difficulty of the questions has a random arrangement, where for example, question 44 seems to be easiest than question 32.

Regarding the correlations between the points estimates, Figure 7.5 displays the scatter plots of the difficulty parameters' point estimates for all the GAT test questions that have been answered by 3348 students across the three methods; M/Gibbs, MML and LA. The Figure shows that the three sets of estimates are not identical but highly similar. The red dotted equality line ($x = y$) in this figure

also indicates the slight difference between the three methods: as we can notice, most points lie on this line or tightly cluster around it.

The scatter plot of M/Gibbs against LA shows that the point estimates look close to being equal. The mean absolute difference between M/Gibbs and LA is 0.011, the maximum absolute difference is 0.03 and the minimum is 0. The maximum difference appears in particular for more extreme point estimates (i.e., especially for the most challenging questions when a few students answer these questions correctly). For example, the maximum difference is for question 37, where we have seen that this is the most difficult item for all GAT questions in both sections, and only 566 students have answered this question correctly. Unlike the maximum difference, the minimum difference between M/Gibbs and LA appears for the easiest questions, when a lot of students answer this question correctly. For example, the minimum difference between these two methods is for questions 46, 78 and 33, where these questions answered correctly by 2328, 2278 and 2445, respectively.

The scatter plot of MML against M/Gibbs and LA in Figure 7.5 shows more deviations, especially for more extreme point estimates (for easiest and hardest questions). The mean absolute difference between M/Gibbs and MML is 0.081, the maximum absolute difference is 0.32 and the minimum is almost 0. Moreover, the mean absolute difference between LA and MML is 0.088, the maximum absolute difference is 0.35 and the minimum is 0.006. The minimum difference here is for the moderate questions, such as the lowest difference between M/Gibbs and MML is for question 51 with difficulty estimate being -0.13 and 1738 students answered this question correctly.

Moving to the students' ability estimates, which is the primary goal of this study, Figure 7.4 displays box plots of parameters ability estimate of $\boldsymbol{\theta}$ for the three methods; MLE, M/Gibbs and LA. The upper panel represents the estimation results of 3348 students' ability for answering the verbal section (52 questions), and the lower panel represents the quantitative section (44 questions). As displayed in this figure, all three methods agree that the majority of the students' abilities range between -2 to 2.5 for both sections, with only a few students having abilities higher than 3. The average abilities estimates for all the students in GAT-V is 0.010, 0 and 0.018 for MLE, M/Gibss and LA, respectively. Similarly, in the GAT-Q, the average abilities estimates is 0.014, 0 and -0.034 for MLE, M/Gibbs and LA, respectively.

To find the main differences between the three methods, Figure 7.6 shows the scatter plots of the total ability point estimates resulting from the answer to all

GAT-V and GAT-Q questions. The same analysis can be applied to each section separately to find the difference between the three estimation methods. Appendix C displays the comparison of scatter plots for each section. As displayed in Figure 7.6, the ability parameter estimates of all methods are pretty similar except for more extreme point estimates when students perform very well or poorly. This difference is more apparent for the higher-level ability; the ability that higher than 3, where we can see these points diverge from the equality line. The mean absolute difference between M/Gibbs and LA is 0.013, the maximum absolute difference is 0.52 and the minimum is almost 0. It is clear that the difference between M/Gibss and MLE is larger, where the maximum absolute difference is 2.03, with the mean equals 0.13 and 0 minimum. Similarly, the maximum absolute difference between LA and MLE is 1.50, with the mean equals 0.14 and 0 minimum. The maximum difference appears for students who answered all questions correctly (i.e. student 1 and student 2626). The minimum difference between M/Gibbs and LA is when students answer approximately 30% of the questions correctly (between 25 to 29 questions). In the difference between MLE and the other two methods, the minimum appears when students answer approximately 50% of the questions correctly (between 45 to 48 questions).

Furthermore, we can check the divergence between the three methods using a ranking measurement distance method. The idea is that we look at the order of the student ability estimates resulting from each method, then we find the distance measure between the two sets of ordering. The Kendall's τ distance method, explained in Chapter 6 6.2, could also be used here to measure the differences.

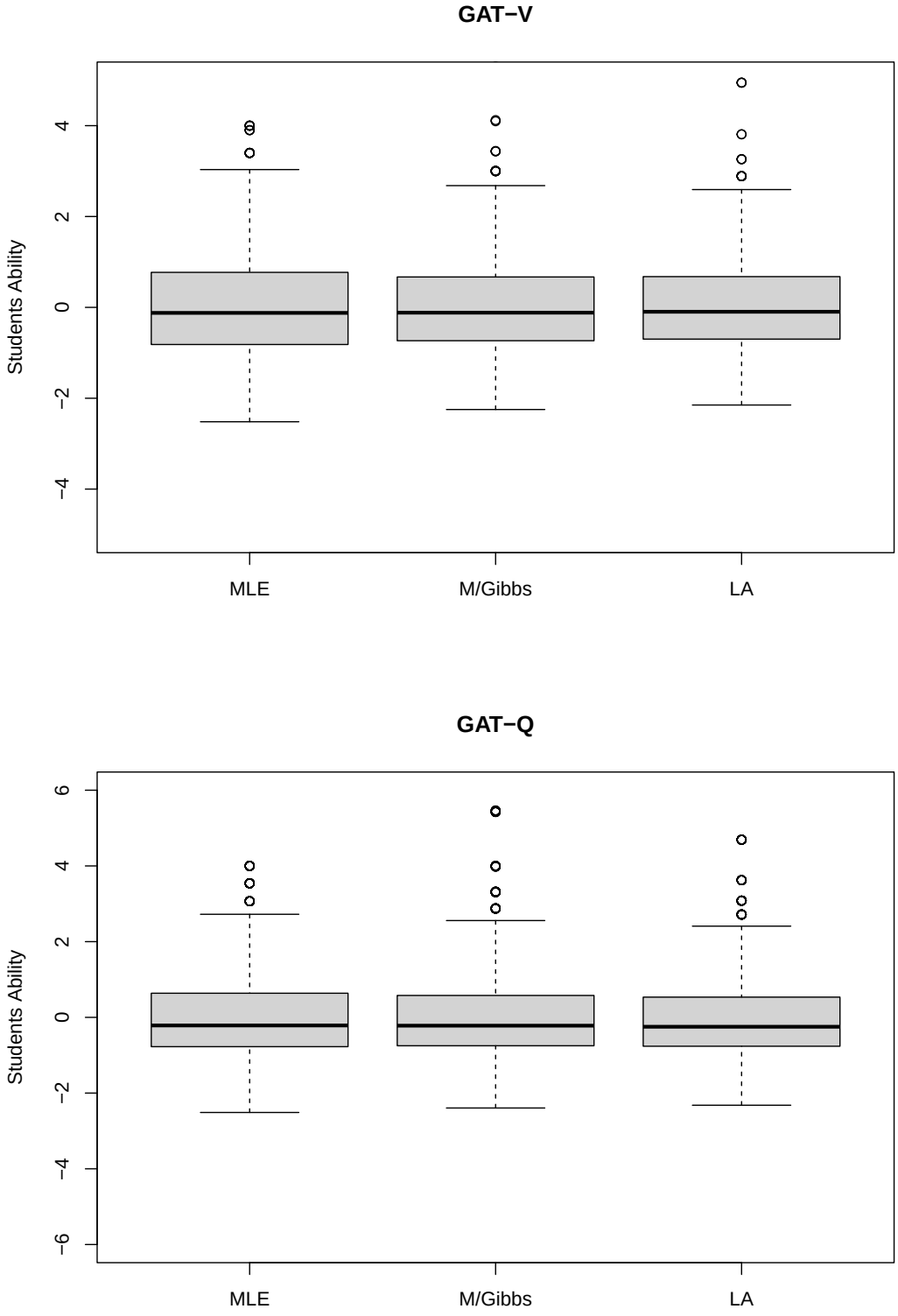


Figure 7.4: Box plots of parameters ability estimate θ for the three methods; MLE, M/Gibbs and LA. The upper panel represents the students' ability estimates in the verbal section (52 questions), and the lower panel represent the quantitative section (44 questions).

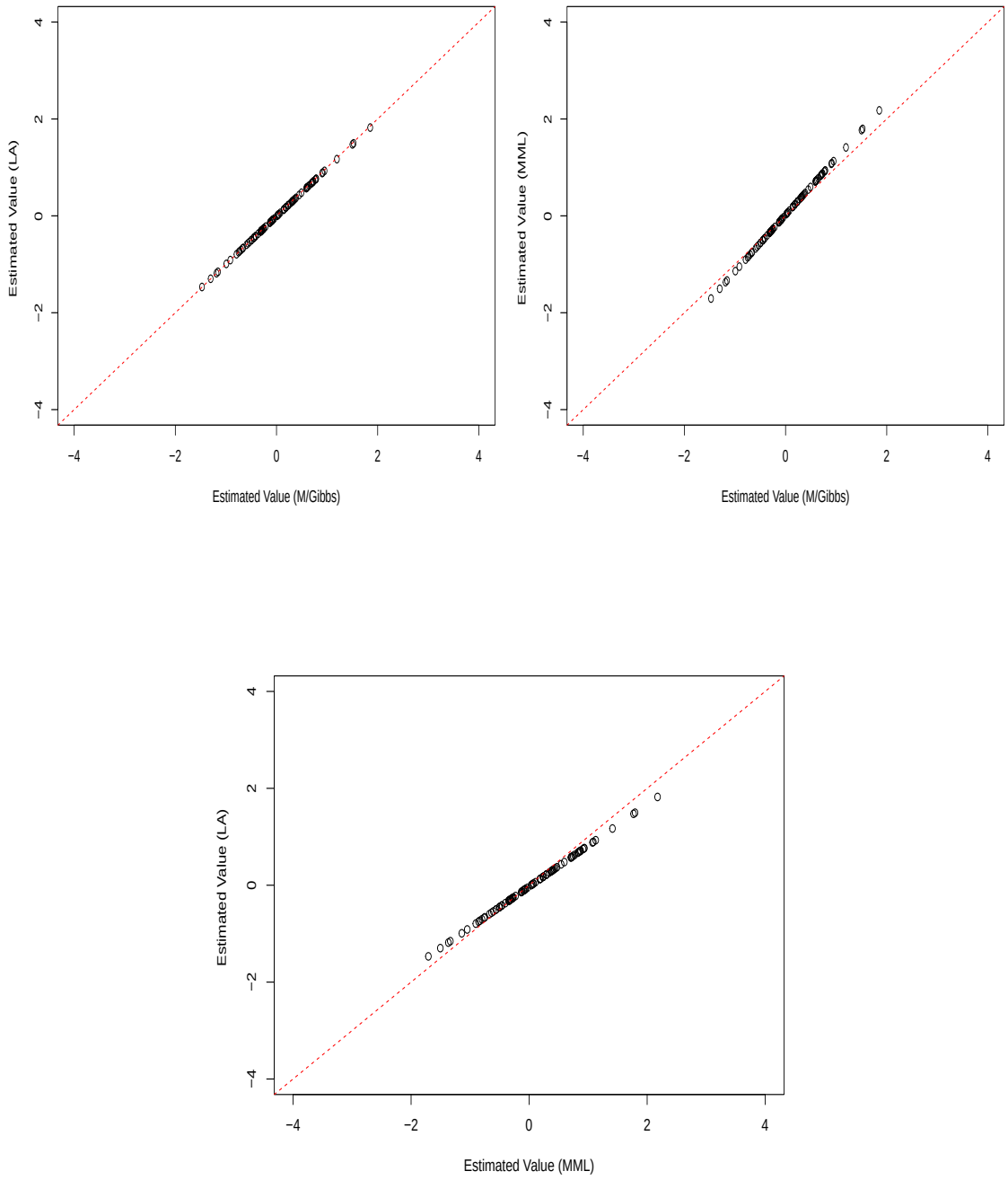


Figure 7.5: Comparison between the points estimates of the difficulty parameters for all GAT test questions (96 items) across the three methods; M/Gibbs, MML and LA. The red line illustrates the equality line.

This method counts the number of pairs agreed in the same order in each of the two orders and which are not agreed in reverse order. If C is the number of agreements, and D is the number of disagreements,

$$\tau = \frac{C - D}{C + D}$$

The value of τ ranges from -1 to 1, where 1 means the two rankings are identical, and -1 means one is opposite of the other. Table 7.1 represents the τ values that measure the differences between the ordered abilities resulting from three methods. It is clear that the resulting order abilities from the three methods are almost identical.

Table 7.1: Kendall's τ distance values between the three methods; M/Gibbs, MLE and LA.

Method	M/Gibbs	MLE	LA
M/Gibbs	1	0.98	0.99
MLE	0.98	1	0.98
LA	0.99	0.98	1

Although all the three methods approximately present similar estimation results, the proposal approximation method, LA, appears to be a suitable method for this type of model and data in terms of accuracy and time. Furthermore, we have seen that LA distinguished between extreme points, where students have high abilities, better than MLE. Moreover, MLE is time-consuming to be applied in real-time, where estimation takes two different steps for difficulty and ability parameters. The MCMC running time for the GAT-V section took 8 hours and 17 minutes, and 5 hours and 12 minutes for the GAT-Q sections. When all the questions (96) were used, the algorithm was run for 9 hours and 48 minutes, making this method too slow for real-time inference. On the other hand, LA took only less than 2 minutes to estimate the difficulties of the 96 questions and the abilities of 3348 students at the same time.

After evaluating the accuracy of the Laplace approximation (LA) by comparing the estimation results to the MCMC method (M/Gibbs) and the frequentist method (MLE), the next step is to further explore the LA results in the real data set. Figure 7.7 shows the histograms of student abilities θ resulting from LA for each GAT test section (GAT-V and GAT-Q) and the total student abilities to answer both sections (GAT-all). The histogram of GAT-V shows the θ values mainly distributed in the range [-2,5]. Most of the students in this section have abilities between 0 to -1, about 1300 students, and the next group has ability level 0.5. There are a few students who have abilities less than -1, about 300 students have almost -1.5 ability level, and 50 students have -2 ability level. In the same way, a few students in this section have an abilities level of 1.5 or more. On the other hand, the histogram of GAT-Q

shows the θ values distributed in the range [-2.5,5]. We can see that the performance of the students in this section is less than the GAT-V section, where most of the students answered only 20 questions or less as described in Figure 7.2, and hence they got fewer abilities levels.

Running LA for each GAT section could help students find their abilities' weaknesses based on the difficulties of the questions, not only the total points of correct answers. Table 7.2 shows as examples of the ability estimates for the first 20 students resulting from LA for all GAT questions (GAT-all), verbal section questions (GAT -V), and quantitative section questions (GAT- Q). We can see for example student 1 has almost equal high ability for both sections, which result in average of high ability for all test question. Student 9 has higher ability in the GAT-Q (0.79) than GAT-V (-0.11), while for example students 13 and 20 have a noticeable higher abilities in the GAT-V than the GAT-Q.

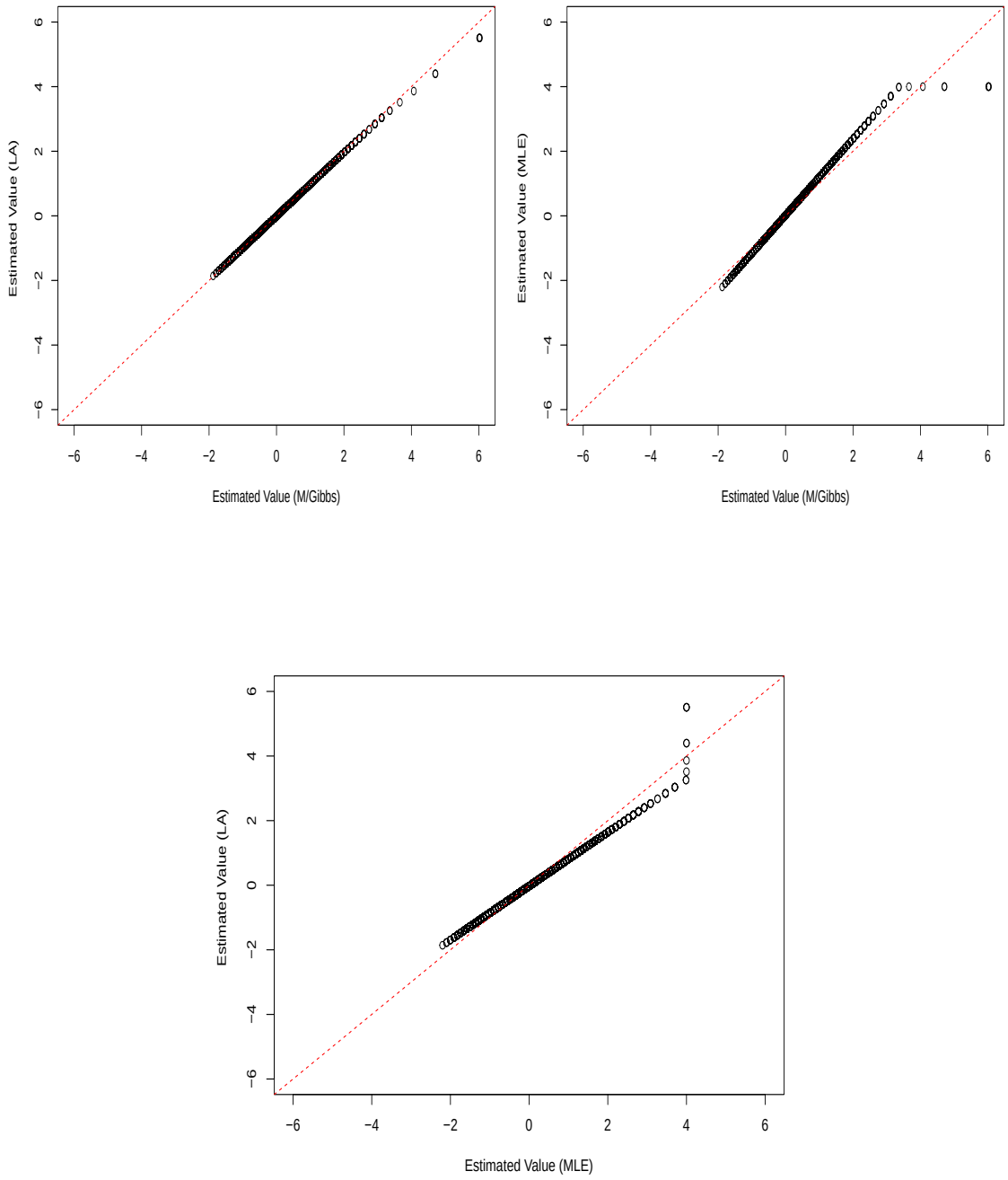


Figure 7.6: Comparison between the points estimates of the ability parameters (θ) based on all GAT test questions (96 items) across the three methods; M/Gibbs, MML and LA. The red line illustrates the equality line.

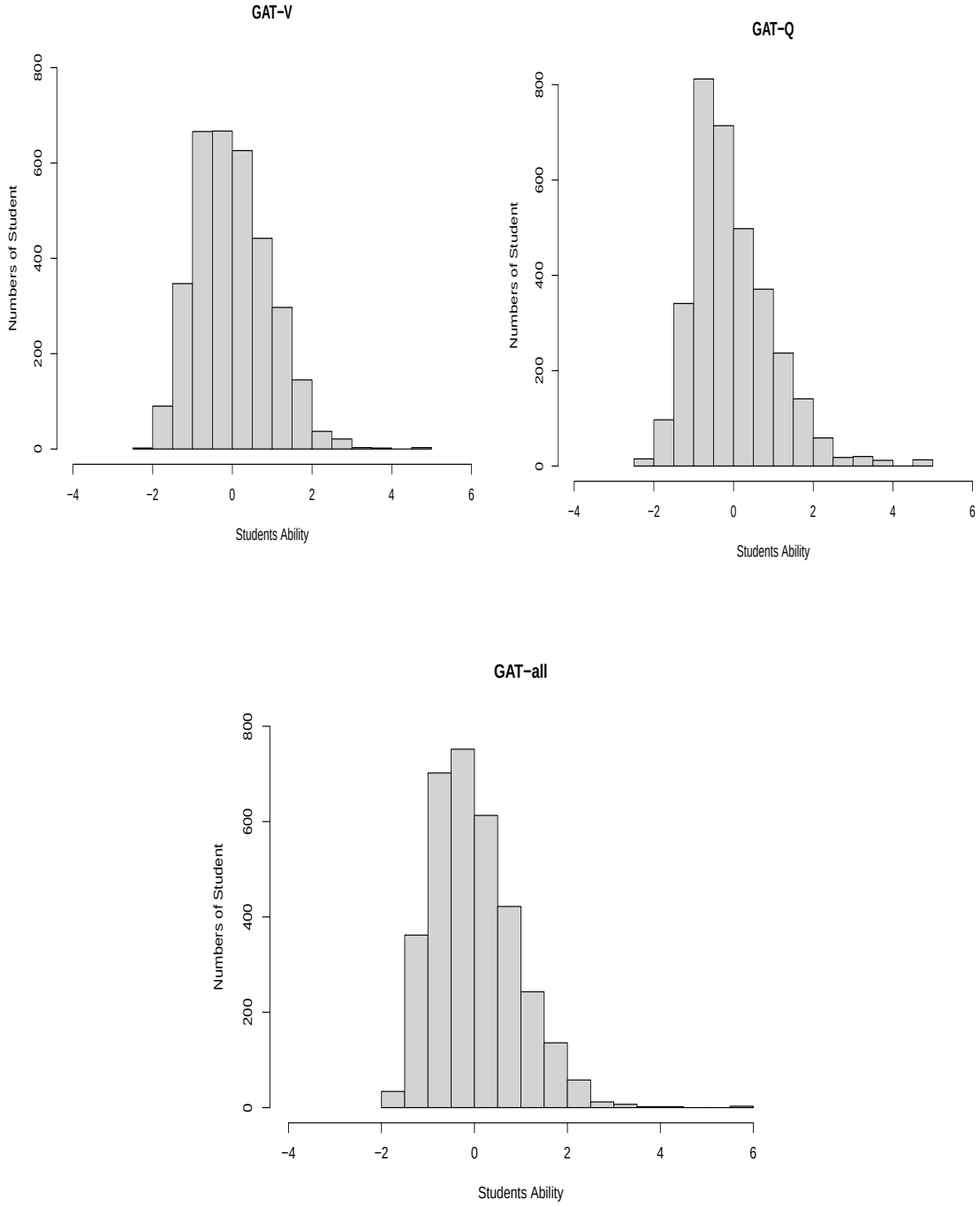


Figure 7.7: Histograms of student abilities for each GAT test sections (GAT-V and GAT-Q), and the total student abilities for answering both sections (GAT-all), resulting from Laplace approximation method (LA).

7.4 Further Analysis

The challenge arises in real-life scenarios when students take a test at different times or on other days. This section deals with the case of updating the students' abilities

Table 7.2: Ability Estimates for the first 20 students resulting from Laplace approximation for all GAT questions (GAT-all), verbal section questions (GAT -V), and quantitative section questions (GAT- Q).

Student	GAT-V	GAT-Q	GAT-all
1	4.90	4.78	5.42
2	-0.81	-1.07	-0.93
3	-1.57	-1.07	-1.34
4	-1.57	-0.83	-1.21
5	-1.21	-0.95	-1.09
6	-1.21	-0.95	-1.09
7	-0.81	-0.20	-0.52
8	-0.91	-1.65	-1.21
9	-0.11	0.79	0.30
10	-0.54	-0.01	-0.29
11	-1.01	-0.40	-0.72
12	-1.21	-1.20	-1.21
13	0.57	-0.30	0.16
14	-1.32	-0.83	-1.09
15	-0.81	-0.95	-0.87
16	-1.11	-0.40	-0.77
17	-1.11	-0.83	-0.98
18	-0.91	-1.07	-0.98
19	-0.37	-1.20	-0.72
20	0.86	-0.51	0.21

sequentially in the GAT dataset. From the previous sections, we can see that both the MLE method and MCMC are unsuitable for this type of analysis. The reason is that MLE needs to be done in two different steps to estimate the difficulty of the questions and then the students' abilities. Moreover, MCMC algorithms must be restarted each time a new student takes the test. Thus, this can take a long time for the purpose of immediate results in real-time. For educational use, we have seen from the simulation studies in Chapter 6 that Laplace approximation is a quick method, and we can easily store the information from the previous inference.

In this setting, the analysis is started by setting the same prior distribution for all students and the same prior for all questions as described in section 7.3. The total number of students is divided into groups, and four different scenarios of grouping students are assumed. Each group of students is supposed to answer all 96 questions and then estimate their abilities (the same analysis can be applied for each test section separately). The first scenario is considered to have a block size of 50 students. After this group finish the test, they receive the results of their ability immediately, and we get an update on the difficulties of questions. Consequently, the result of the difficulties of the questions can be stored and then used as prior

distributions for the questions in the next 50 students. We keep updating our prior beliefs about the difficulties of the questions and set the same prior distribution for the student's abilities until all the student finish the test. The same analysis is repeated for a block size of 200, 500 and 1000 students.

The goal now is to compare the result of these four different scenarios of estimating the abilities parameters sequentially to the result of estimating all students' abilities in a single update. Table 7.3 shows the Kendall's τ values between the student's abilities resulting from LA by updating all students at one time and updating the abilities of the students sequentially for different numbers of students in each block size, where 1 means the two sets of group are identical, and -1 means one is opposite of the other. The values of τ for all order sets are higher than 0.3, which indicates a strong association between the two ranking sets (Walker and Beretvas, 2003). Moreover, all the τ values are 0.97 or higher, indicating that the abilities estimates resulting from these sequentially updated are almost identical to the abilities estimates resulting from updating all abilities once.

Table 7.3: Kendall's τ values between the students abilities resulting from LA by updating all students at one time and updating the abilities of the students sequentially for different numbers of students in each block size.

Block size	Kendall's τ
50.00	0.97
200.00	0.97
500.00	0.98
1000.00	0.99

Table 7.4 displays the average and maximum values of the absolute difference between abilities estimates resulting from updating all abilities once and sequentially for different block sizes. We can see that the highest average difference is for using 50 students in the block size and updating them sequentially. The average values decreased slightly for the block of 200 students (from 0.0385 to 0.0373), and more noticeable decreases for the block size of 500 and 1000. The maximum absolute difference appears to be higher for small block sizes; 50 and 200 and slightly smaller for larger block sizes; 500 and 1000. These maximum differences seem to appear frequently in the first and second sequences for each block size and for the students of higher abilities, who answer all the 96 questions correctly. This might happen because of the lack of information about the difficulty of the questions in early sequences. Figure 7.8 provides a visual presentation of the absolute average values of the difference between the difficulties of the questions estimates resulting from the updating of all the difficulties once and sequentially for different block sizes (50,

200, 500 and 1000) at the first sequence, middle sequence and final sequence. We can see that the average difference is relatively high for the first sequence in all four scenarios when the same prior distribution is used for all questions. After that, this difference decreases gradually by updating the prior distributions for the difficulty of the questions until it becomes almost zero in the final sequence. Furthermore, the average difference is increased by the number of block sizes, where it is below 0.1 for the block size of 50 and increase to 0.35 for the block size of 1000.

Table 7.4: Average and maximum values of the absolute difference between abilities estimates resulting from updating all abilities once and updating the abilities sequentially for different block sizes.

Block size	Average absolute difference	Maximum absolute difference
50.00	0.0385	0.1894
200.00	0.0373	0.1907
500.00	0.0296	0.1342
1000.00	0.0216	0.1428

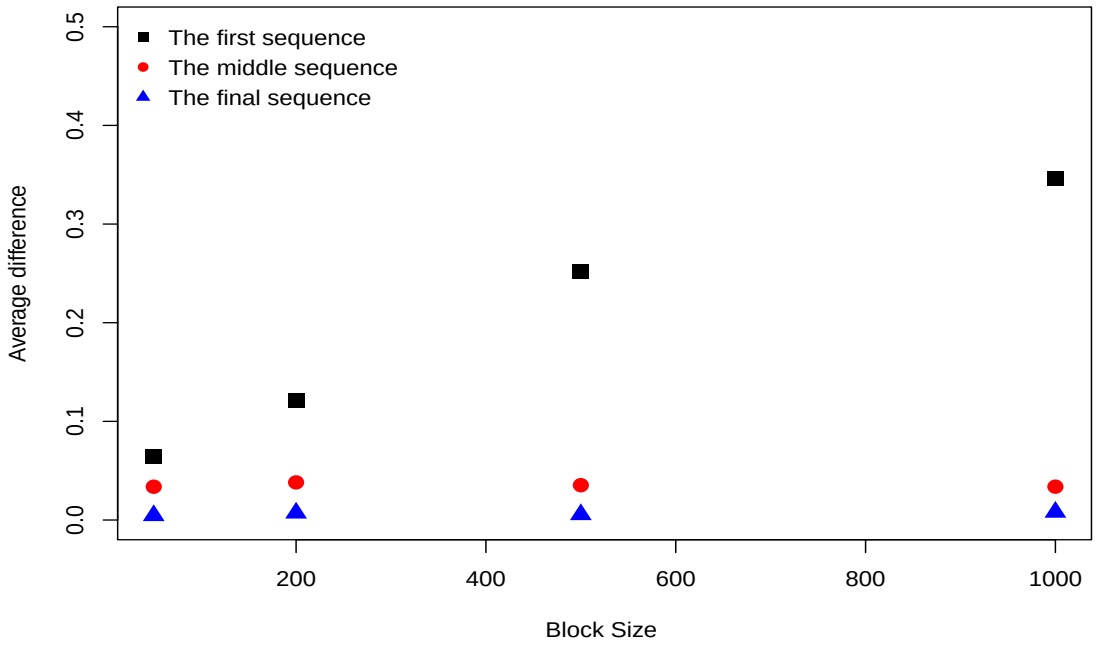


Figure 7.8: Average absolute values of the difference between the difficulties of the GAT_all (96) questions estimates resulting from the updating of all the difficulties once (LA_all) and updating the difficulties sequentially for different block sizes (50, 200, 500 and 1000) at the first sequence, middle sequence and final sequence.

Table 7.5 and Table 7.6 compare the differences between estimating students' ability in different block sizes (50, 200, 500 and 1000) updated sequentially and

updated once for the first and last 10 students, where these students belong to the first and last sequences (blocks), respectively. We can see in the first 10 students (Table 7.5), where we have no information about the difficulty of the questions, there are some differences between the abilities estimate in each sequence. It seems that the abilities for students 2 to 10 in each sequence are slightly overestimated. These differences between estimating the abilities in each sequence decrease incrementally as we add more data (blocks), and they become approximately identical, as shown in Table 7.6.

Table 7.5: Comparison of the differences between estimating students' ability for the first 10 students in different block sizes (50, 200,500 and 1000) sequentially updates and a single update.

Student	LA_seq_50	LA_seq_200	LA_seq_500	LA_seq_1000	LA_all
1	5.42	5.50	5.43	5.37	5.51
2	-0.86	-0.76	-0.84	-0.90	-0.94
3	-1.26	-1.16	-1.24	-1.30	-1.35
4	-1.14	-1.04	-1.12	-1.18	-1.22
5	-1.03	-0.92	-1.00	-1.06	-1.11
6	-1.03	-0.92	-1.00	-1.06	-1.11
7	-0.47	-0.36	-0.44	-0.49	-0.54
8	-1.14	-1.04	-1.12	-1.18	-1.22
9	0.34	0.45	0.37	0.32	0.28
10	-0.24	-0.13	-0.21	-0.26	-0.31

Table 7.6: Comparison of the differences between estimating students' ability for the last 10 students in different block sizes (50, 200, 500 and 1000) sequentially updates and a single update.

Student	LA_seq_50	LA_seq_200	LA_seq_500	LA_seq_1000	LA_all
3339	0.81	0.81	0.81	0.81	0.81
3340	-0.78	-0.78	-0.78	-0.78	-0.78
3341	-0.68	-0.68	-0.68	-0.68	-0.68
3342	-0.34	-0.35	-0.35	-0.35	-0.35
3343	0.43	0.42	0.43	0.42	0.42
3344	0.47	0.47	0.47	0.47	0.47
3345	0.24	0.24	0.24	0.24	0.24
3346	0.62	0.61	0.62	0.61	0.61
3347	-0.12	-0.12	-0.12	-0.12	-0.12
3348	2.08	2.08	2.08	2.08	2.07

To ensure that there is no effect of choosing the first group on each block size, the experiment was repeated 10 times randomly reordering the students in the data set each time. Each experiment was run independently to estimate the difficulties of the questions and the student's abilities using sequential Laplace approximation for block sizes of 50, 200, 500, and 1000 and one update. Table 7.7 shows the average of the Kendall's τ values between the student's abilities for the 10 different experiments. As we can see that the ordering abilities estimates resulting from these sequentially updated are almost identical to that if the abilities estimates resulting from updating all abilities once. We notice that the average Kendall's τ is higher than Kendall's τ values in the original experiment due to the effect of having a high-ability student, who answered all questions correctly in the first group. The experiment was run again while keeping the three more competent students, who answered all questions correctly, in the first group to investigate the effect of having

more than one high ability student in the early group. The Kendall's τ values in Table 7.8 shows that having three very high ability students in the first sequence have the same effect as having one high ability student.

Table 7.7: Average Kendall's τ values between the students abilities resulting from LA by updating all students at one time and updating the abilities of the students sequentially for different numbers of students in each block size for 10 different experiments.

Block size	Kendall's τ
50.00	0.98
200.00	0.98
500.00	0.99
1000.00	0.99

Table 7.8: Kendall's τ values between the students abilities resulting from LA by updating all students at one time and updating the abilities of the students sequentially for different numbers of students in each block size with very competent students in the first sequence.

Block size	Kendall's τ
50.00	0.97
200.00	0.97
500.00	0.98
1000.00	0.99

7.5 Conclusion

This case study has proposed using the Laplace approximation (LA) method to estimate the difficulty of the questions and the students' ability for the one-parameter (1PL) item response theory (IRT) model. The experiment has conducted on the General Aptitude Test (GAT) dataset, which includes 3348 students answered 96 questions divided into two sections: the verbal section (GAT-V) and the quantitative section (GAT-V). For comparison purposes in terms of accuracy and computational time, the same experiment has conducted using one of the frequentist approaches (Maximum Likelihood) and one of the Markov chain Monte Carlo (MCMC) methods (Gibbs sampler within the Metropolis algorithm). The three methods have been implemented for three different settings; estimating the ability parameters for the verbal section (52 items), the quantitative section (44 items) and all GAT test questions (96 items) to compare the students' abilities in each section to the total abilities. The novel approach explained in Chapter 6 (6.4), where the sequential Laplace approximation method can be applied in a dynamic IRT model, has also been carried out in the GAT data set.

Experimental results confirmed that the proposed LA method could produce precise approximation results for students' abilities and questions' difficulties in a short time. The point estimates for both students' abilities and questions difficulties resulting from the three methods were highly similar. However, the LA and M/Gibbs were almost identical due to the use of the prior distribution in both settings. The time required to estimate the difficulties and abilities parameters of the GAT dataset using M/Gibbs was between 5 to 9 hours. While this time was only two minutes or less when implemented the same experiment using the standard LA method. On the other hand, implementing the MLE method requires estimating the difficulties and abilities parameters in two different steps, which can be time-consuming when real-time inference is required. In this particular experiment, the MLE produced the final results in approximately 4 minutes.

With regard to point estimates of the difficulty of the questions for the GAT data set, most of the questions were in a reasonable range between -2 to 2 for both sections (GAT-V and GATQ). Although the estimated difficulties do not agree with the assumption that questions are arranged from easiest to the more difficult in each section, they are distributed across the desired range; -2 to 2 (DeMars, 2010). A few questions were so easy that between 2400 to 2604 students were able to answer these questions correctly; questions 1, 33 and 50 for GAT-V and only question 78 for GAT-Q. The most challenging questions were questions 37 and 16 for GAT-V

and questions 47 and 84 for GAT-Q, where only between 566 to 880 answered these questions correctly.

Considering the point estimates of the students' abilities, approximately 96 % of the students' abilities were between -2 to 2,5. There were three students able to answer all the questions correctly to achieve high abilities equal to 5.5. The average of the abilities was approximately 0, and the standard deviation was 0.9.

The analysis result of updating Laplace approximation (LA) sequentially showed a perfect match to update LA once. Moreover, even if starting with a small block size of 50 students, Kendall's τ measurement had a value of 0.97, indicating that the two sets are almost identical in terms of ranking the students' abilities. The experiment was repeated several times by randomly ordering the students in the GAT data set to get a different range of students' abilities in the first blocks. The average Kendall's τ of these experiments indicated that the two sets of students' abilities resulting from the sequential LA and updating the LA once are almost identical.

The analysis presented here is considered the case of a single skill model, which could be extended in a multi-skill model (Multidimensional IRT Approach MIRT). Using the Laplace approximation in MIRT models for real-time inference could give students immediate feedback on their abilities and which abilities they should improve at a reasonable time. However, we have seen that MCMC and MLE methods are expensive for one skill model. Moreover, MLE usually requires a large data set, so in real-time inference, the students who finish the test have to wait until other students finish. Therefore, if the goal is to give immediate feedback, the LA is a suitable method, where we have seen in sequential LA that the number of students does not affect the abilities estimates.

Chapter 8

Conclusion

8.1 Summary and Conclusion

One of the most challenging tasks in the educational field is to evaluate students' ability levels accurately. Item response theory (IRT) provides a valuable theoretical framework for educational measurement, focusing on the response pattern, not the total score. However, inferring students' ability and item difficulty or discrimination can be a technical challenge, as one needs to achieve a balance between speed and accuracy. A larger challenge arises when one needs to estimate these parameters in a dynamic system or for massive datasets. In many real-life scenarios, teachers and students are interested in immediate test results and feedback for evaluating students' abilities. However, the speed and volume can present considerable challenges in applying a common method such as Markov chain Monte Carlo (MCMC) to the IRT model when real-time inference or large-scale datasets is required. The main objective of this thesis is to develop approximate Bayesian inference based on the Laplace approximation method (LA), which allows faster inference for IRT models and matches MCMC's accuracy.

This thesis's focus was mainly on estimating students' ability in a reasonably fast time, considering the case that real-time inference is required. Although it did not intend to provide a complete overview of other inference approaches for estimating IRT model parameters, some of these methods were briefly reviewed in Chapter 3, giving more details for the MCMC method in Chapter 4. The results of applying two MCMC methods in Chapter 4: Metropolis-Hastings within Gibbs (M/Gibbs) and Hamiltonian Monte Carlo (HMC), to two-parameter logistic IRT model, showed that these methods are not usable for real-time inference due to their computational expense. Despite the fact that the HMC method is more efficient with a high effective sample size per second than M/Gibbs and is twice as fast, this method is still too

expensive for online inference or massive datasets. Moreover, the computational cost remained high even using a smaller number of iterations and small datasets.

To reduce the computational cost, Chapter 5 provided an application of the sequential Monte Carlo Method (SMC), the most common method used in the literature for a dynamic system. Applying the SMC to the IRT model was considered first in the use of classical SMC method (SMC1), where the likelihood is added gradually to overcome the difference between the prior and the posterior distributions. Although this method was successfully applied to the 1PL IRT model and compared to M/Gibbs, the result showed that using this method may require more effort for educational use, such as estimating the students' ability in real time. The reason is that the efficiency of this algorithm depends on the user settings, e.g. number of particles, the variance of the proposal distribution and the number of intermediate distributions between prior and posteriors. All these factors can directly affect the ESS; hence, small ESS will require a re-sample step, which increases the time cost.

A different setting of SMC algorithms (SMC2) was also presented in Chapter 5 to reduce the additional computational time that occurred by resampling steps and increase the SMC's efficiency by using fewer particles. In this setting, data is added sequentially in the likelihood instead of sampling from the sequence of intermediate distributions, eliminating the time to introduce the data. Moreover, an additional MCMC move step was added to increase particle diversity and overcome problems such as sample impoverishment. This method was successfully applied to the 1PL IRT model, and can also be straightforward in other models. The comparison result of the time showed that SMC2 was about 7 (using 10,000) to 700 (using 500,000) times speed up over SMC1. However, for a particular experiment presented in Chapter 5, the approximation results were improved by increasing the number of particles to $500,000$, which was 248 seconds for very small data ($n = 10$ and $m = 5$). Moreover, same as SMC1, the performance of the SMC2 depends on the user setting, including the number of particles, introducing of the sequence of the data, choosing the proposal density in the MCMC step, and choosing resampling methods. These settings can be complicated for real-time inference or non-professional users.

Therefore, from these experiences conducted in Chapters 4 and 5, we can see that MCMC and SMC are still too slow in real-time inference. Therefore, simulation methods are suitable for smaller data sets and when more precise inference samples are required. However, for educational use to estimate students' ability, for most teachers, it is a qualitative inference, not quantitative inference. That means the exact ability point estimates (e.g. 4.70 or 4.72) may not be necessary, it is enough

to get them roughly correct particularly the order. Therefore, this suggested using an alternative method to address the computational cost by using Bayesian approximation methods based on Laplace approximation. This method is relatively simple to compute and can derive helpful information about the models' parameters.

The Laplace approximation (LA) method was applied successfully for the 1PL and extended to the more complex; 2PL model. Chapter 6 provided comprehensive comparison studies between the MCMC method using M/Gibbs and the LA method. Regarding the 1PL IRT model, the comparison studies were carried out considering three different simulated data: small, moderate and large and various test lengths from very small (m=10) to very large (m=100). In terms of the 2PL model, the comparison study was carried out for a relatively large number of students ($n = 600$) and moderate test length ($m = 50$). Based on several comparison criteria, such as Jensen-Shannon divergence (JSD), Kendall's $\boldsymbol{\tau}$, Bias and RMSE, the LA method is faster than the MCMC algorithm, with 120 to 900 times speedup, without losing accuracy. Regarding the point estimates of ability parameter $\boldsymbol{\theta}$, the biases and the RMSE for the LA were generally smaller than those from M/Gibbs, with a few exceptions. On the other hand, Kendall's τ values were generally larger for the LA method, with the most marked differences occurring with smaller sample sizes and shorter test lengths, indicating that the LA ordered the actual abilities more accurately than M/Gibbs. However, Kendall's τ values became almost identical for more extended tests and larger sample sizes under both methods. Regarding the difference between the posterior distribution generated by both methods, the JDS measurement values and the plots of the posterior densities suggested that the largest difference between the two resulting posteriors appeared for more extreme point estimates (very low/high ability). The analysis of the difficulty point estimates confirmed this conclusion also. Therefore, the results showed that the LA method could provide very accurate approximations in very cheap computational time. Thus, the LA method seems useful for researchers interested in obtaining real-time estimates of students' abilities or for massive datasets.

The Laplace approximation requires calculating the Hessian matrix around a mode and inverting the results to obtain the covariance matrix. In high-dimensional problems, the covariance matrix elements will be large according to the number of examinees and items. Hence, this will increase the time cost of the LA method. Two proposed solutions were discussed in section 6.3 to reduce the time. The first method was to use the idea of the block matrix and some linear algebra strategies. This method does not require changes in process or optimisation. It can be implemented in a few lines of code to achieve a reasonable time reduction, providing the same

result as standard LA. This method was successfully applied to the LA method and reduced the time for massive datasets, such as 10,000 students and 50 items, from 670 seconds to almost 8.0 seconds.

The second and most straightforward solution is to take a diagonal approximation of the Hessian and invert it to obtain the variance rather than using the entire matrix. The proposed approach was investigated by an extensive comparison study between full and diagonal LA to estimate students' ability and questions' difficulty uncertainty. The result showed that, the diagonal LA method underestimated the variance in general, which appeared more clearly for moderate ability or difficulty levels. Nevertheless, this difference became negligible for a huge dataset. The time cost was comparable to the first approach. However, the main advantage of using the diagonal LA is that we can use the second derivative of the log posterior distribution directly to obtain the diagonal of the Hessian matrix rather than the optimisation method. Considering this strategy, the cost time was considerably dropped, which allowed estimating the ability of 10,000 students and the difficulty of 50 questions in 53 seconds. This study demonstrated that a diagonal Hessian approximation not only can reduce the computational cost of massive data problems but also can overcome the limitation of computer memory.

In a common real-life scenario, students can take the test on different days or other times. Hence, it is expected that the estimation of item parameters in IRT models (e.g. the difficulty parameter $\boldsymbol{b}$) to change over time whenever one or a group of students answer the same test. This thesis provided a novel approach in section 6.4 to the sequential Laplace approximation method on the one-parameter dynamic IRT model, which can also be straightforward to apply to other models. The main idea was that the difficulty parameter estimates are updated sequentially from the data every time new students answer the test and then used the results as prior distributions for the difficulty parameters to estimate the following students' abilities. The idea of this method was illustrated by the comparison study to the full LA update method in three different simulated datasets. Based on several criterion measurements, the sequential LA method resulted in ability point estimates comparable to those from the full LA. The most considerable differences between the point estimates of each method unsurprisingly appeared for the early sequence updates. The biggest advantage of this method is that it can be a helpful tool for research problems for big data taking into account online inference. This method can help avoid storing large data sets in computer memory since keeping the previous student's information is unnecessary. Moreover, the posterior distribution only needs to be calculated for the new students to estimate their abilities, making this method

very fast for online inference.

The Laplace approximation method was applied to a real dataset, taking into consideration non-dynamic and dynamic IRT models. This data was obtained from the General Aptitude Test (GAT), which is used for university admission in Saudi Arabia. The analysis was based on 3,348 students who completed all 96 questions, where the questions were divided into two sections; the verbal section (GAT-V) and the quantitative section (GAT-Q). The MCMC method based on M/Gibbs and marginal maximum likelihood (MML) was applied to this data set for comparison purposes to investigate the performance of LA. The three methods were carried out for three different settings; estimating the ability parameters for the verbal section (52 items), the quantitative section (44 items) and all GAT test questions (96 items) to compare the students abilities in each section to the ordered abilities. The results showed that the LA method produced approximate results for students abilities and questions difficulties comparable to M/Gibbs in a short time; using M/Gibbs was cost between 5 to 9 hours and only two minutes or less for full LA. The analysis result of updating the LA sequentially showed a perfect match to the full LA. Moreover, even if starting with a small block size, Kendall's τ had a value of 0.97, indicating that the two sets are almost identical in ranking the students abilities.

The Bayesian approximation method based on the LA demonstrates a simple application to unidimensional item response theory without requiring extra tuning of parameters like the SMC and MCMC methods. Moreover, it produces fast estimation results even for the massive dataset, which appeared from different experiments in this thesis that the results are comparable to the MCMC results. The following section will provide a brief guide to some directions for future work.

8.2 Future work

The analysis presented in this thesis could be extended in various directions. First, one could fit more complex models such as polytomous IRT models that take into account more than two options for the questions' answers.

Although this thesis successfully applied to real data for the UIRT model, assuming a single ability is sometimes insufficient to distinguish variation in examinees' responses. Hence, the Laplace approximation could be applied to dynamic or non-dynamic multidimensional item response theory, carrying two or more abilities.

Prior distributions play an essential role in Bayesian inference. Prior elicitation, mentioned briefly in 4.2, has gained more attention to Bayesian inference. However, the application of this method to the IRT models is still limited. The performance of the Laplace approximation can be investigated more using the prior elicitation; it is expected in education that experts or teachers have some knowledge about the IRT models' parameters, especially item parameters.

Considering online inference for education measurement, this thesis has no more motivations to do further analyses in sequential Monte Carlo methods. However, this method could be investigated further for other measures, such as online ranking. For example, the performance of the SMC can be primarily affected by the choice of the proposal density; hence one can investigate the effect of different proposals.

8.3 Software Implementation

The algorithms employed in this work were implemented using R. I have my own implementations codes of the M/Gibbs, HMC, SMC1, SMC2, full LA, block LA, diagonal LA, and sequential LA algorithms, which are available on request.

Appendices

Appendix A

Additional result from MCMC

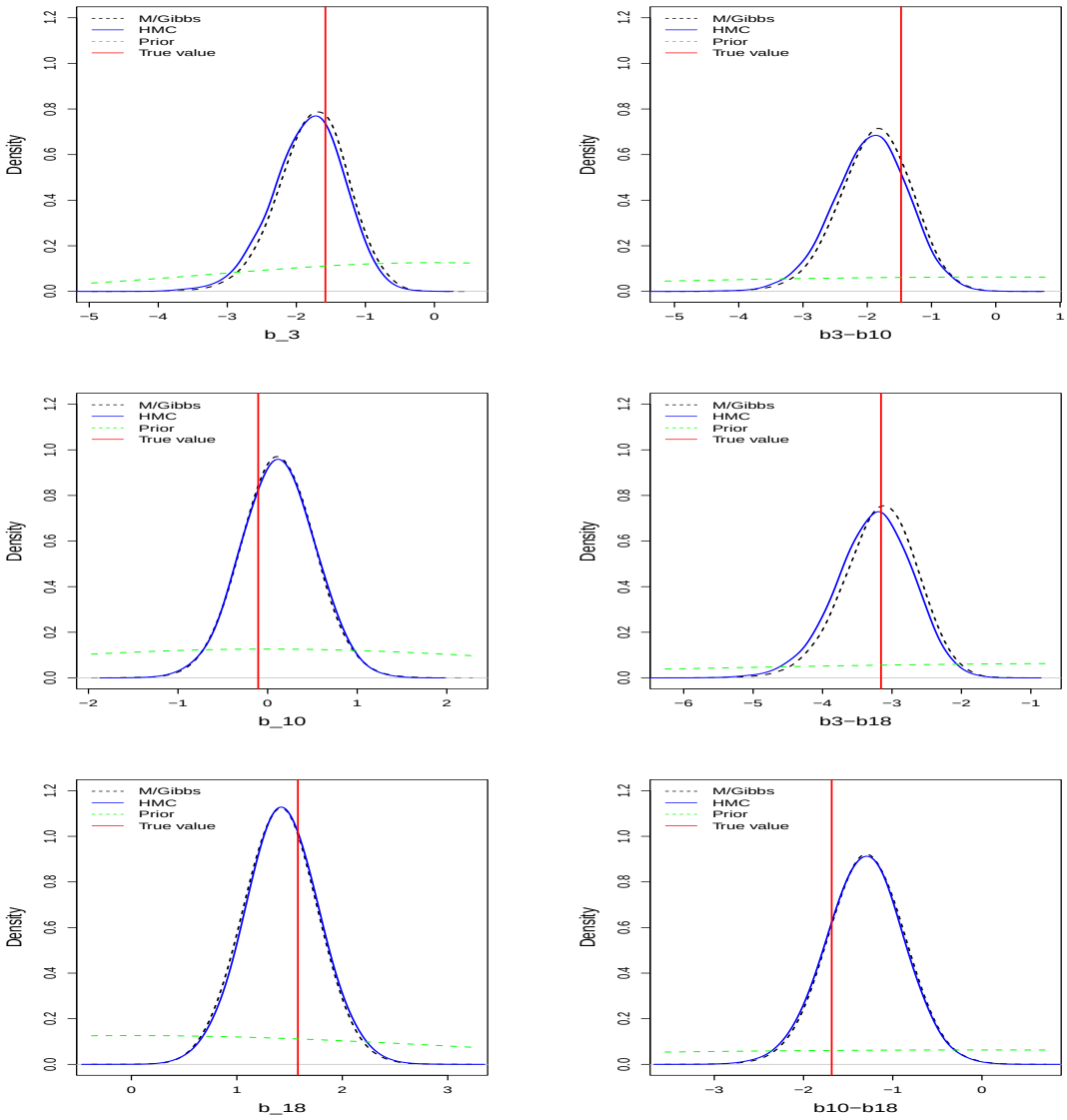


Figure A.1: Posterior density plots for M/Gibbs and HMC methods of three levels of selected questions' difficulties

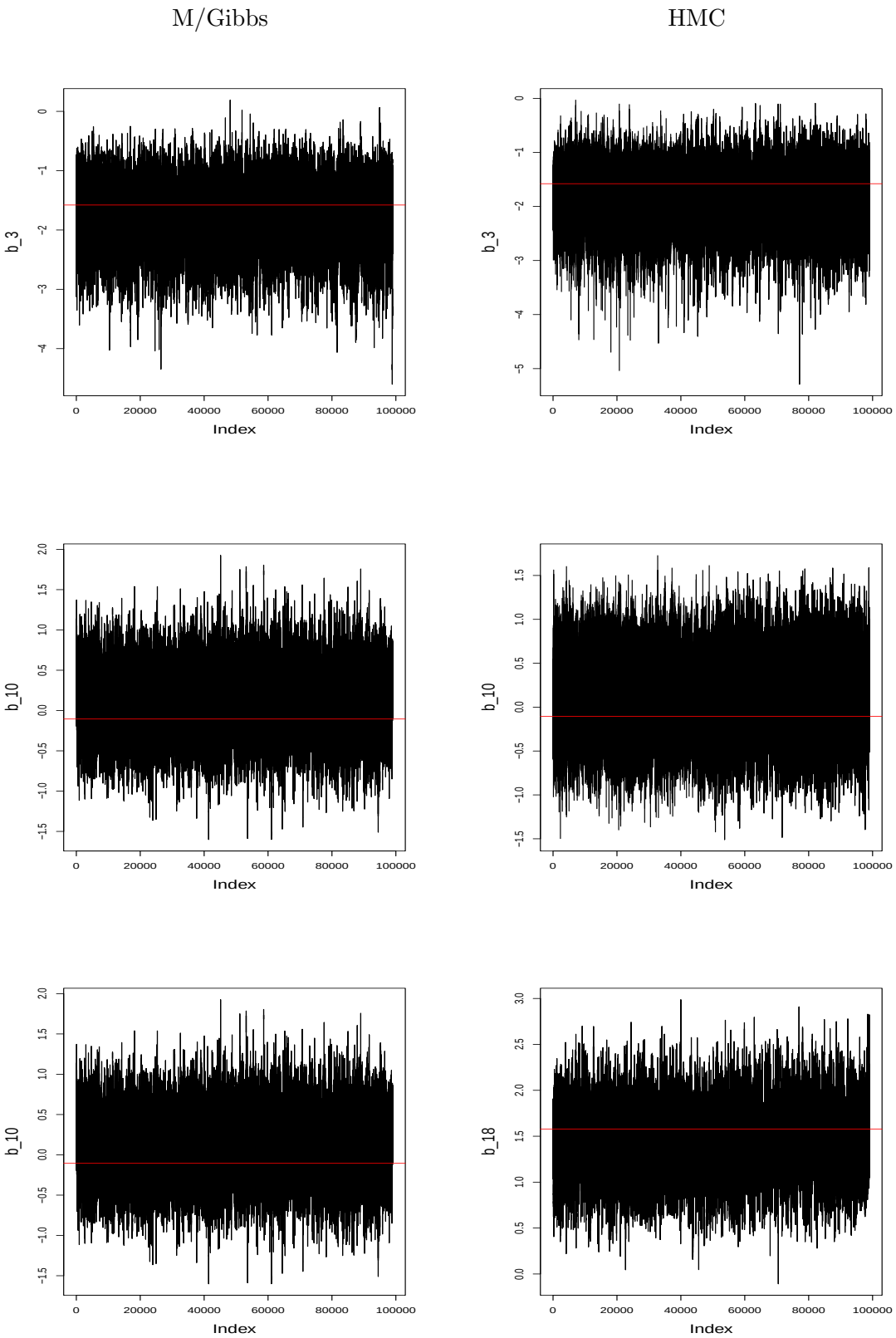


Figure A.2: Trace plots of three levels of randomly selected questions' difficulties obtained from M/Gibbs (left) and HMC (right). The red line indicates the true parameter value.

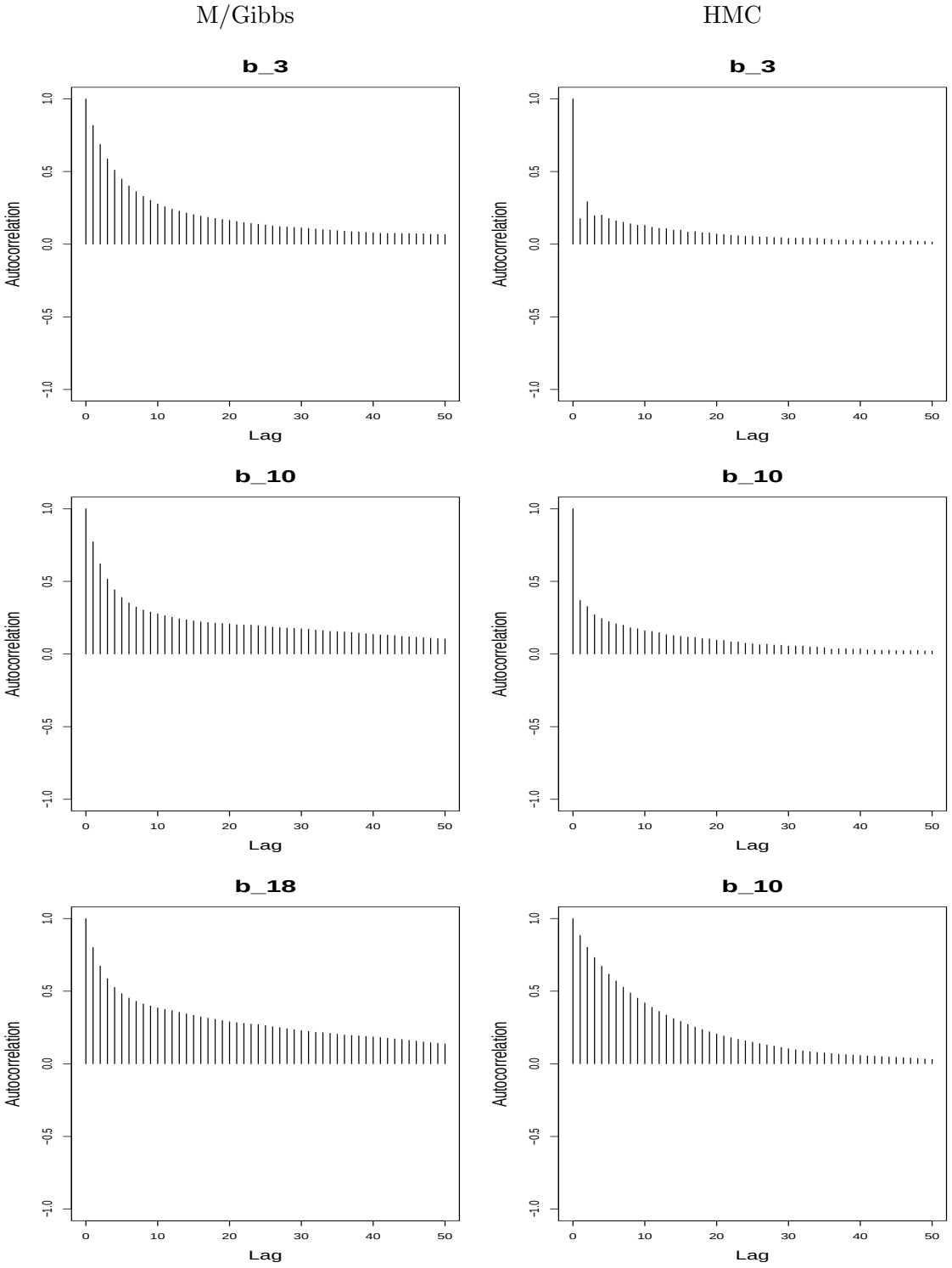


Figure A.3: Autocorrelations between the samples returned by M/Gibbs (left) and HMC (right) for three levels of randomly selected questions' difficulties.

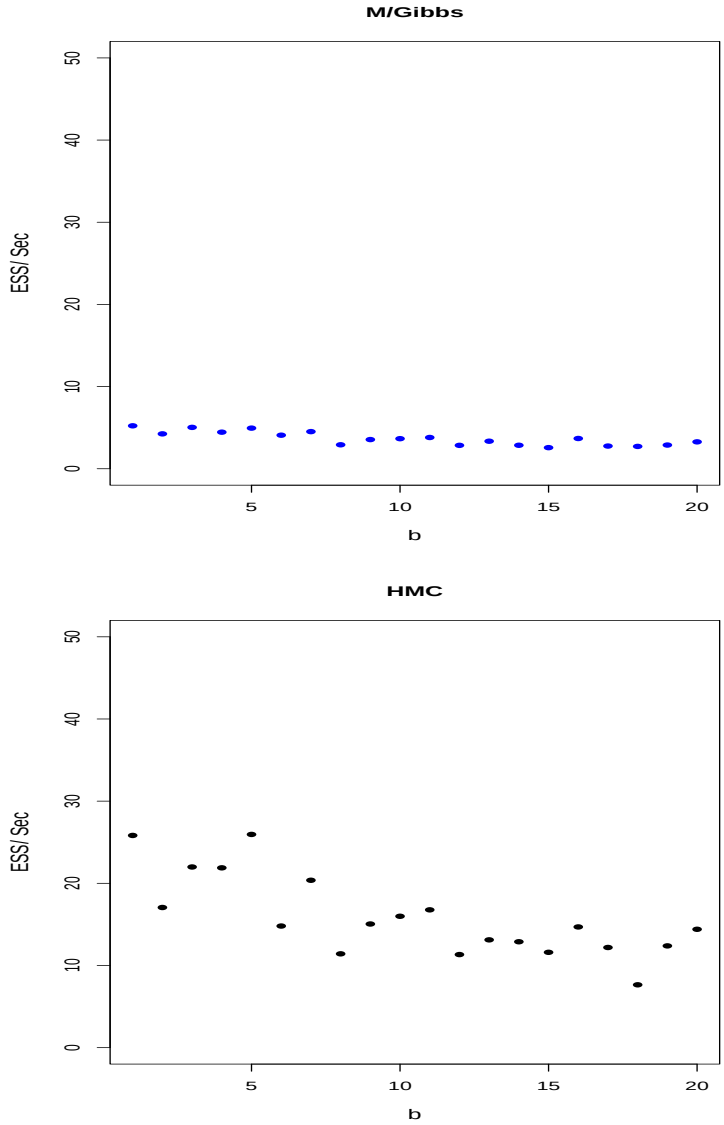


Figure A.4: ESS per second from the performance of M/Gibbs (blue) and HMC (black) for questions' difficulties.

Appendix B

Additional Results for LA

Additional Results of Comparison Studies

Table B.1: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$ with sample size $n = 1000$ and a different number of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
10	MCMC	0.00025	1.02	0.76
	LA	0.000001	0.85	0.82
30	MCMC	0.0007	0.705	0.86
	LA	-0.000002	0.587	0.88
50	MCMC	-0.00047	0.555	0.89
	LA	0.0011	0.478	0.90
70	MCMC	0.00079	0.468	0.90
	LA	0.0017	0.410	0.91
100	MCMC	0.00016	0.372	0.91
	LA	0.0016	0.339	0.92

Table B.2: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$ for sample size $n = 1000$ and different numbers of items.

Number of Items	Method	Bias	RMSE	Kendalls τ
10	MCMC	-0.0607	0.119	1
	LA	-0.0045	0.09	1
30	MCMC	0.0015	0.099	0.98
	LA	0.00026	0.95	0.98
50	MCMC	0.00068	0.092	0.96
	LA	0.0012	0.088	0.97
70	MCMC	-0.0026	0.091	0.96
	LA	-0.0013	0.088	0.96
100	MCMC	-0.0036	0.096	0.96
	LA	-0.0018	0.094	0.96

Table B.3: Comparison of the computation time between M/Gibbs method and LA method for sample size $n = 1000$ and different numbers of items.

	Time (in seconds)	
Number of Items	Gibbs/M	LA
10	3288	1.80
30	4260	3.30
50	5320	5.00
70	6040	6.80
100	7143	9.82

Table B.4: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the ability parameter $\boldsymbol{\theta}$ with sample size $n = 2000$ and a different number of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
10	MCMC	-0.0008	1.011	0.76
	LA	0.0003	0.846	0.82
30	MCMC	-0.0026	0.70	0.85
	LA	-0.0026	0.586	0.88
50	MCMC	0.00028	0.56	0.89
	LA	-0.00084	0.48	0.90
70	MCMC	-0.0003	0.46	0.90
	LA	-0.0002	0.41	0.91
100	MCMC	-0.00034	0.39	0.92
	LA	0.00024	0.35	0.92

Table B.5: Average bias, average RMSE, and Kendall's τ values between the estimated points and the true values for the difficulty parameter $\boldsymbol{b}$ for sample size $n = 2000$ and different numbers of items.

Number of Items	Method	Bias	RMSE	Kendall's τ
10	MCMC	0.005	0.09	1
	LA	0.005	0.7	1
30	MCMC	0.0058	0.07	0.99
	LA	0.0022	0.066	0.99
50	MCMC	0.002	0.07	0.98
	LA	0.001	0.06	0.98
70	MCMC	-0.00026	0.068	0.97
	LA	-0.00037	0.065	0.97
100	MCMC	-0.002	0.06	0.97
	LA	-0.001	0.06	0.97

Table B.6: Comparison of the computation time between M/Gibbs method and LA method for sample size $n = 2000$ and different numbers of items.

	Time (in seconds)	
Number of Items	Gibbs/M	LA
10	6354	10
30	7604	16
50	9089	23
70	10119	31
100	11743	41

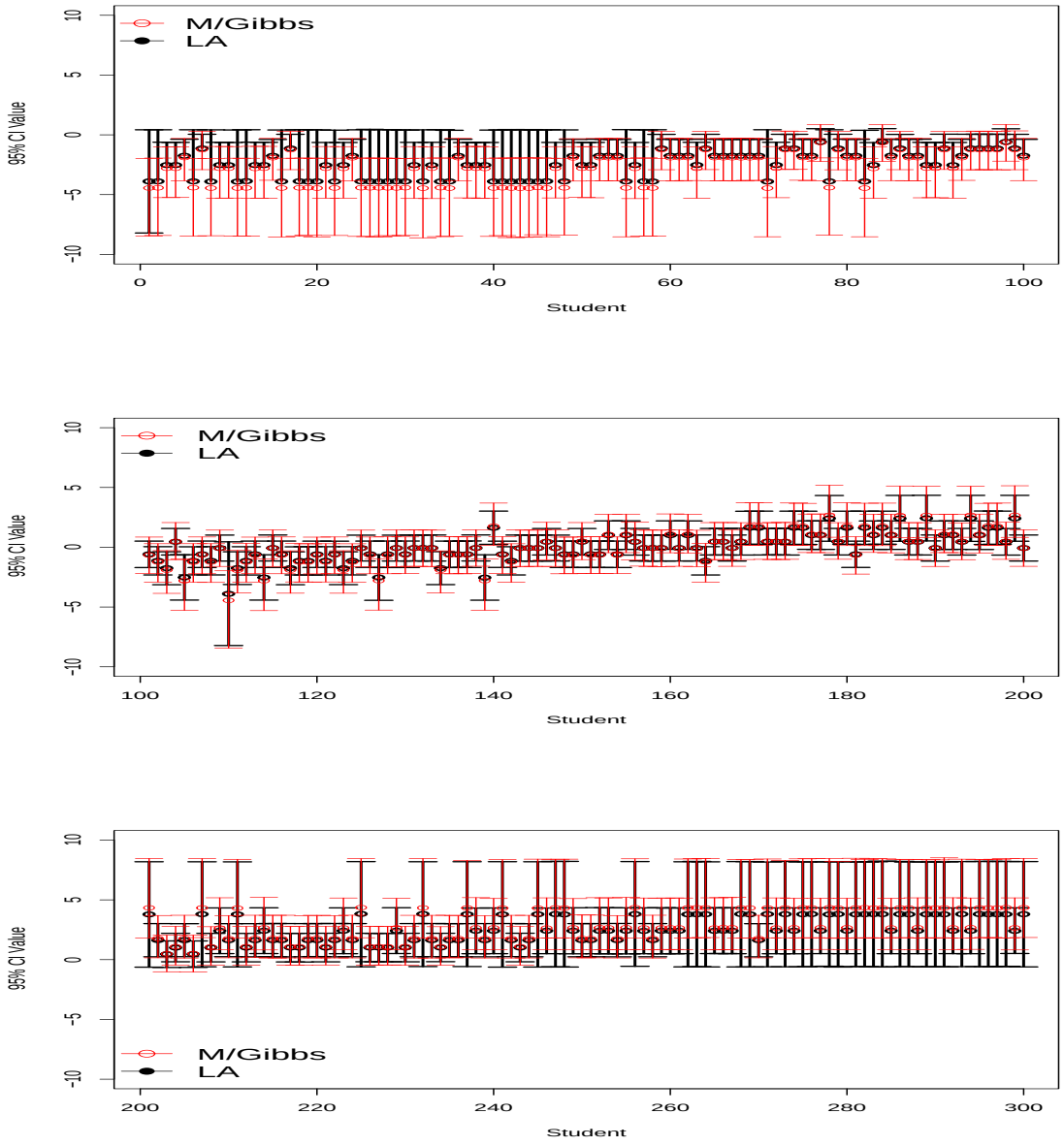


Figure B.1: Posterior means and 95% credible intervals (CI) of the point estimates resulting from M/Gibbs and approximation method LA for sample size $n = 300$ and $m = 10$.

Additional Results of High-Dimensional Problem

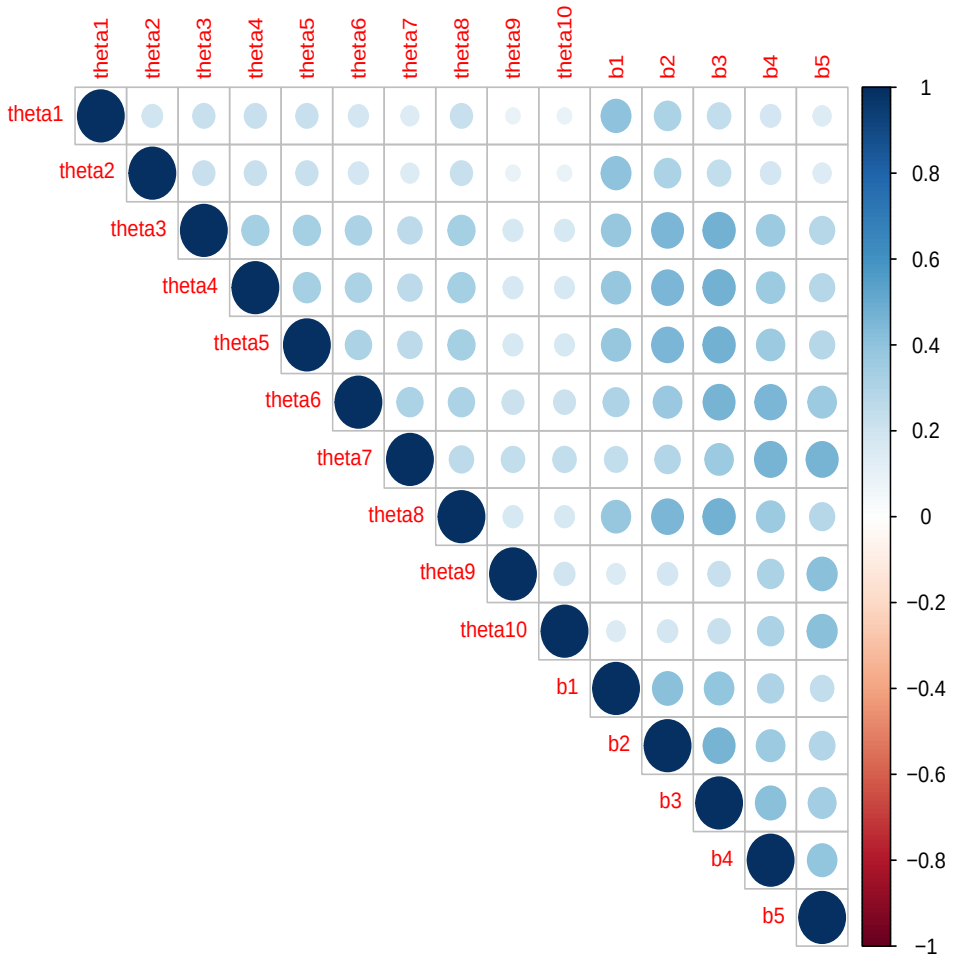


Figure B.2: Correlation matrix between 1PL model parameters for $n = 10$ and $m = 5$.

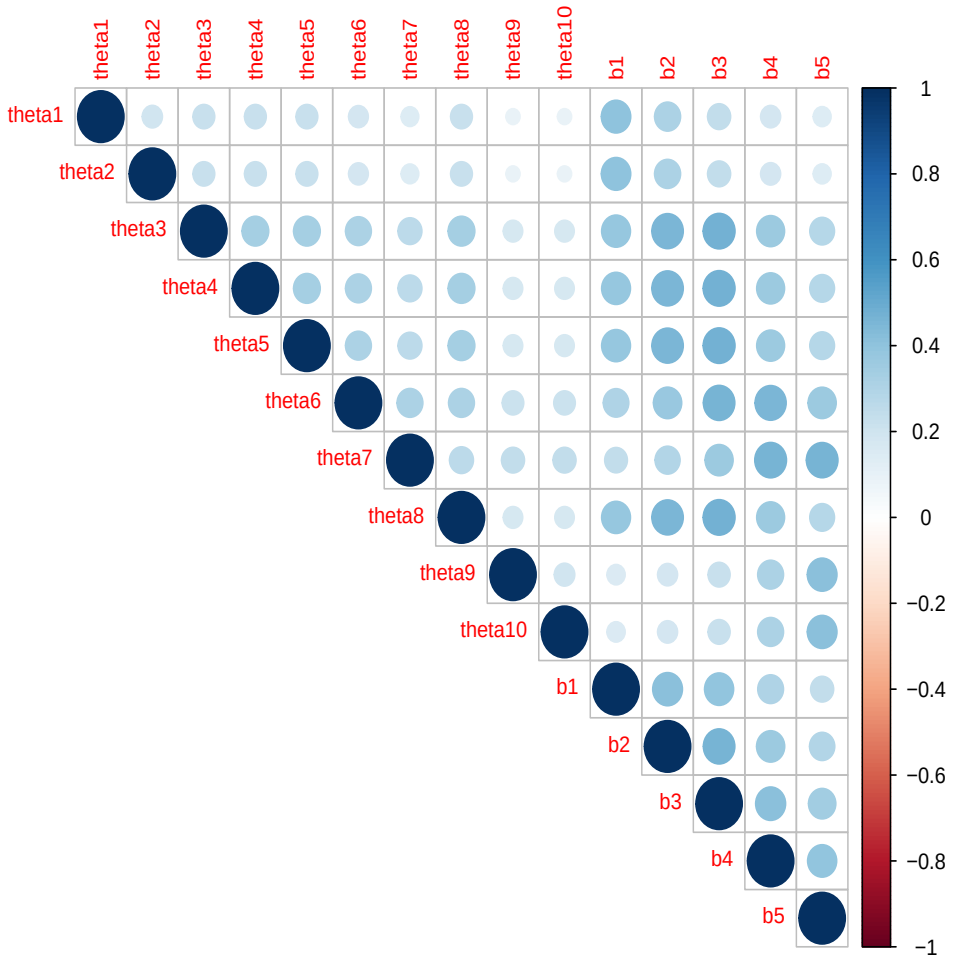


Figure B.3: Correlation matrix between 1PL model parameters for $n = 10$ and $m = 10$.

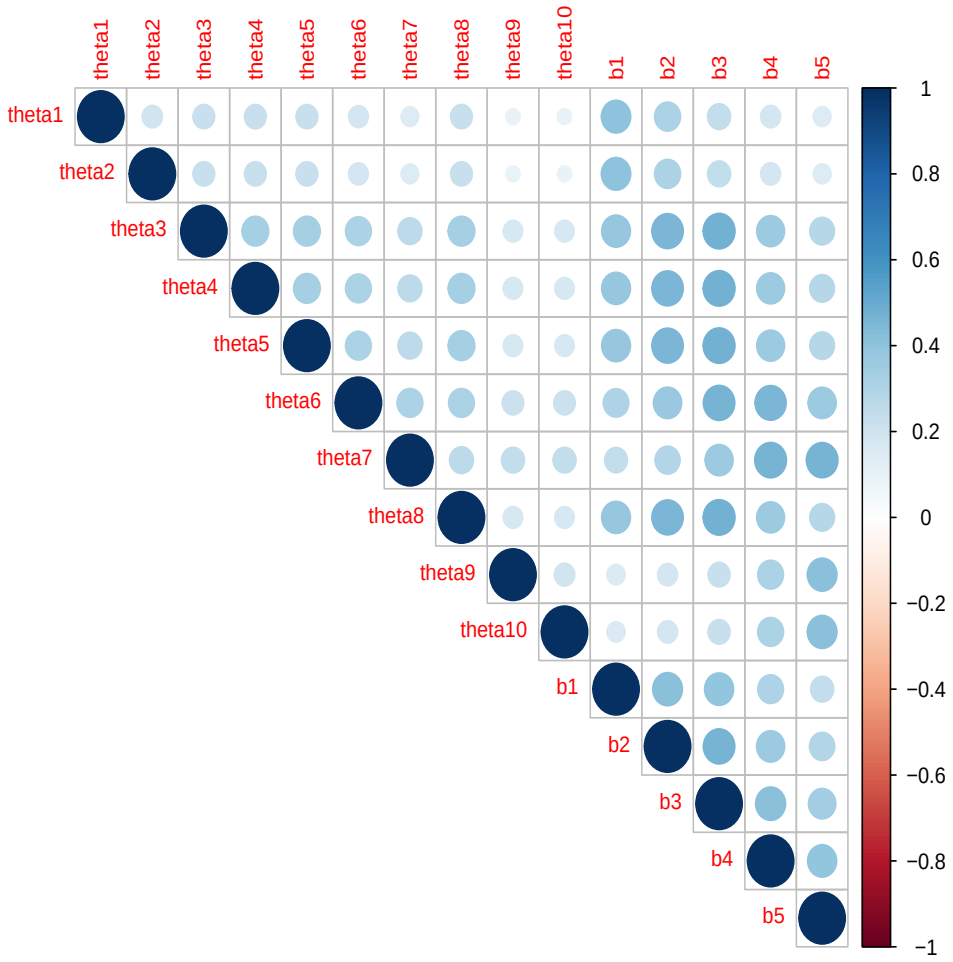
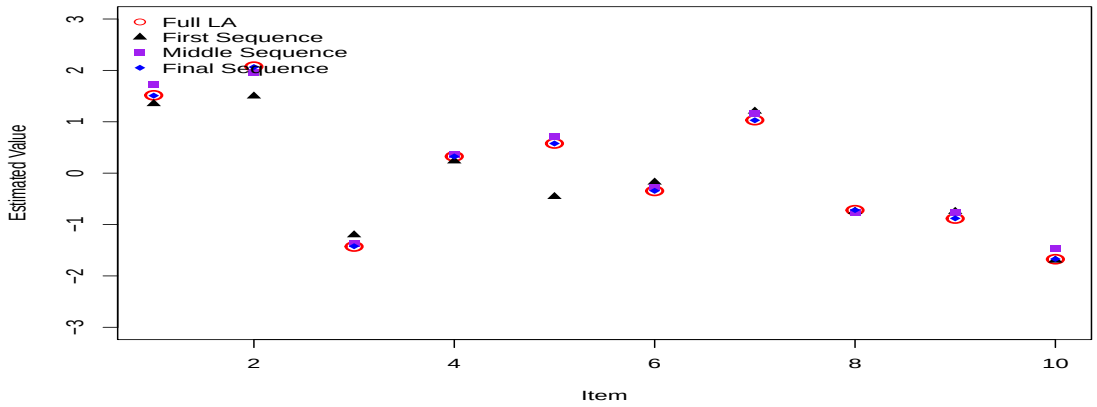
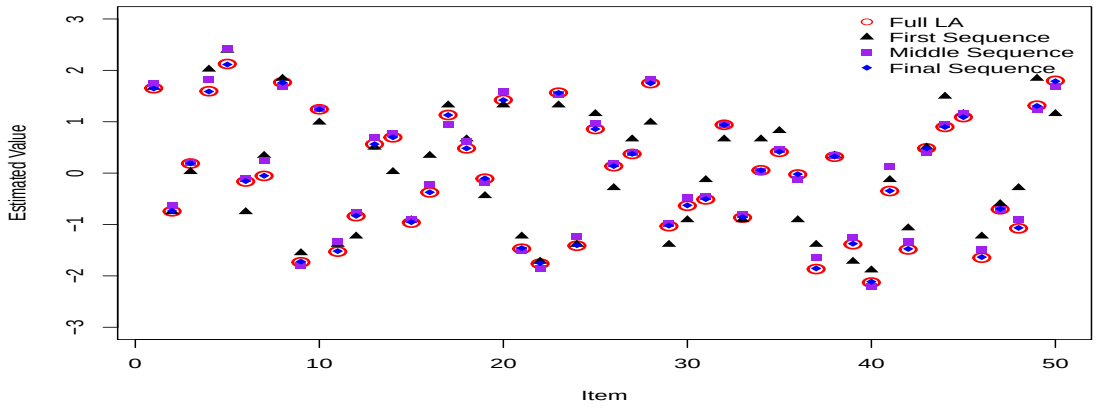


Figure B.4: Correlation matrix between 1PL model parameters for $n = 10$ and $m = 20$.

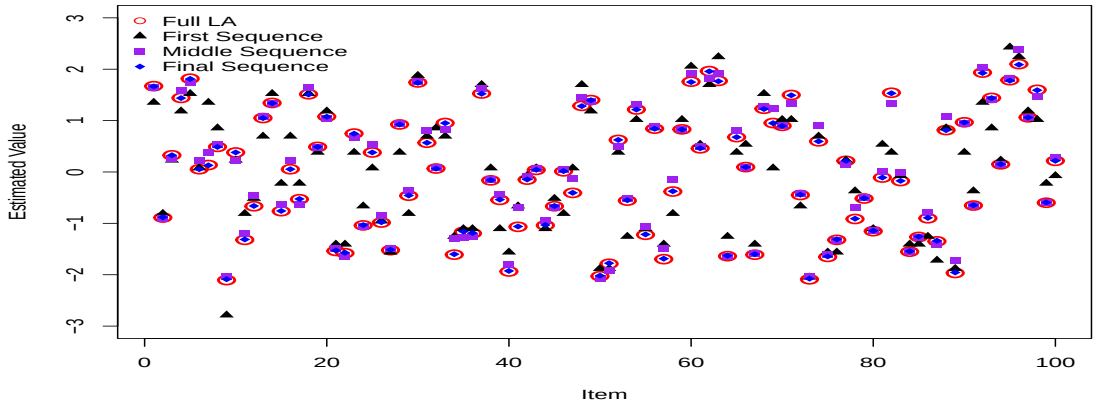
Additional Results of Sequential LA



Number of Items= 10

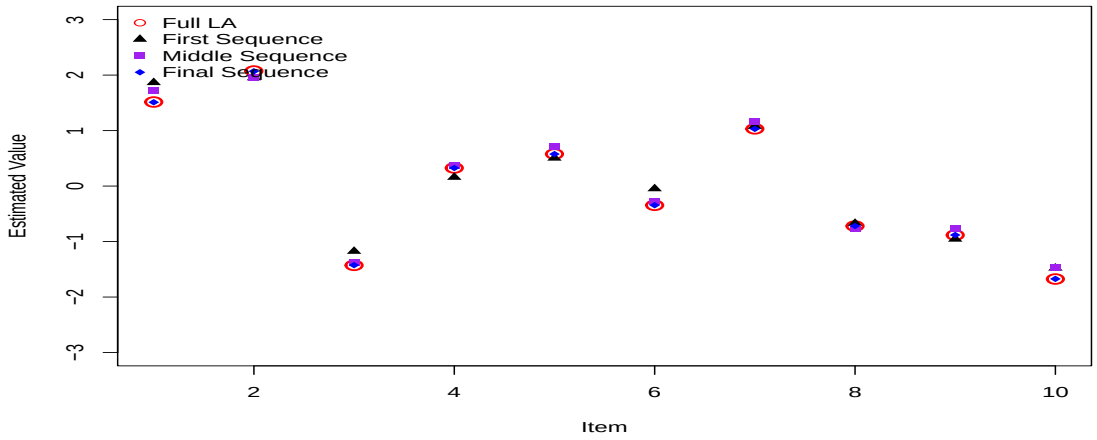


Number of Items= 50

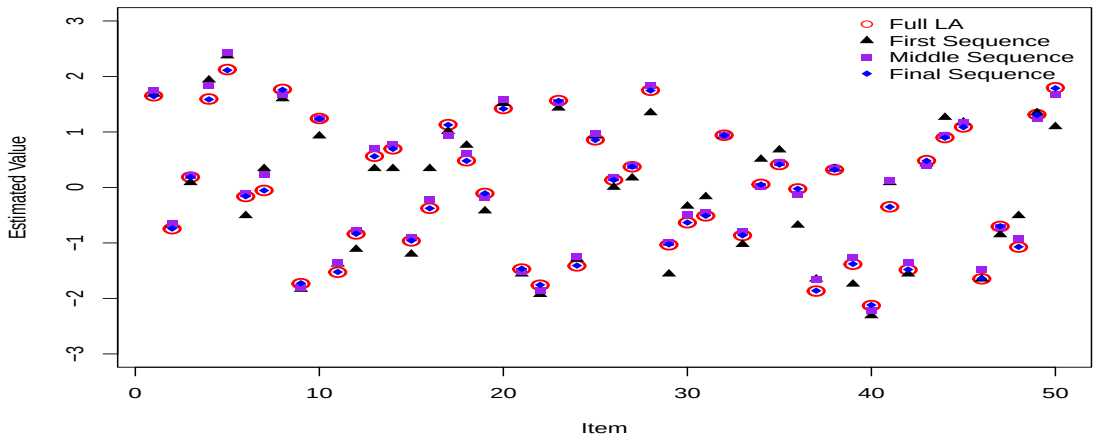


Number of Items= 100

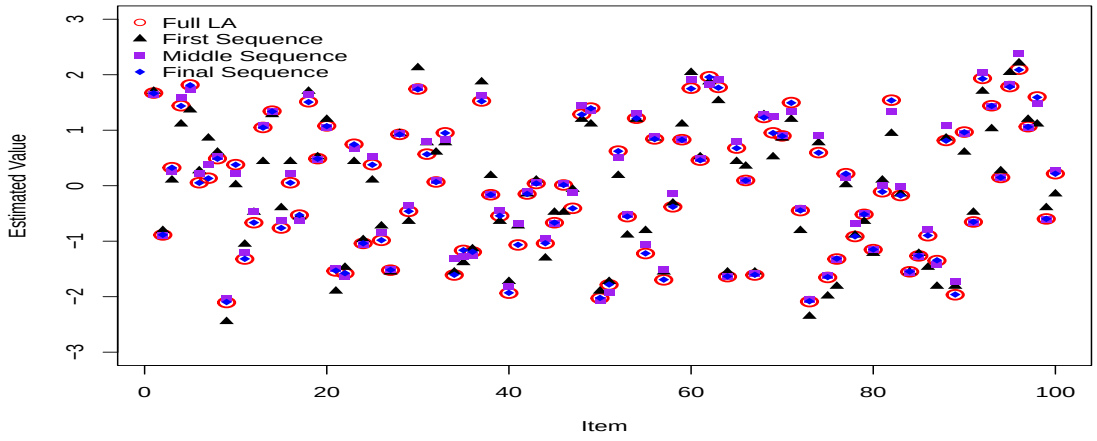
Figure B.5: Point estimates of the difficulty parameters for sequential LA update at first, middle and final sequences and full LA update for three different test lengths ($m = 10, 50$ and 100). The block size of the sequential update is 50.



Number of Items= 10

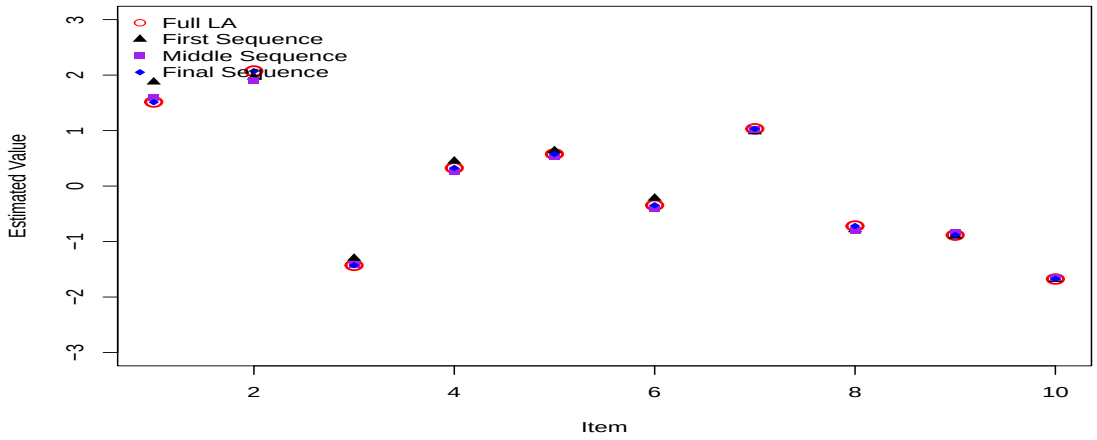


Number of Items= 50

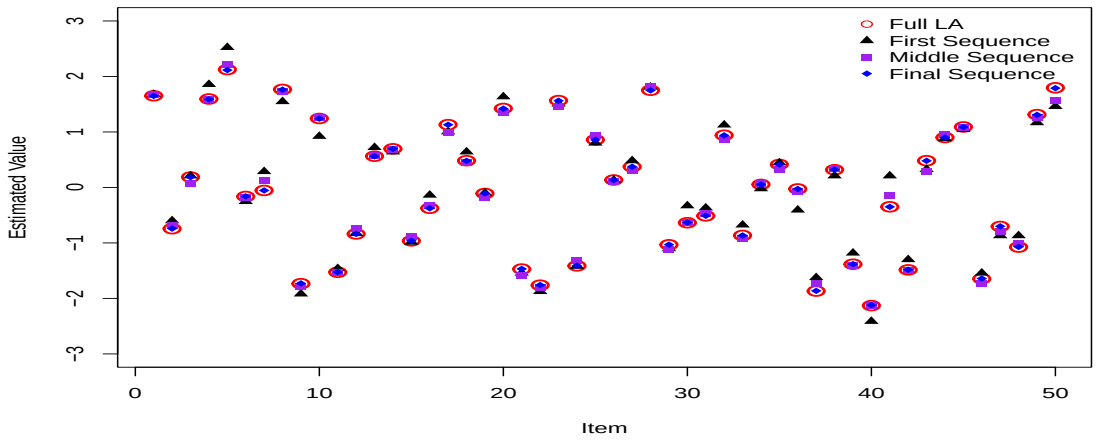


Number of Items= 100

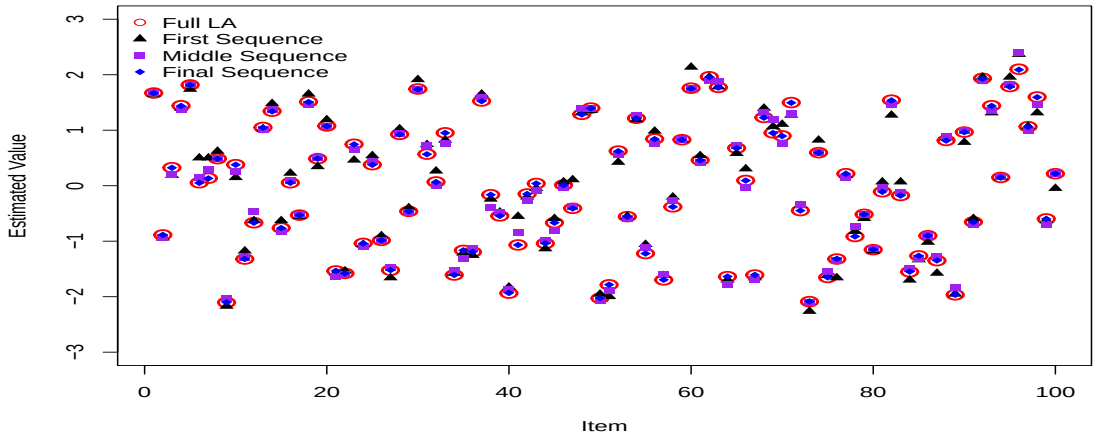
Figure B.6: Point estimates of the difficulty parameters for sequential LA update at first, middle and final sequences and full LA update for three different test lengths ($m = 10, 50$ and 100). The block size of the sequential update is 100.



Number of Items= 10



Number of Items= 50



Number of Items= 100

Figure B.7: Point estimates of the difficulty parameters for sequential LA update at first, middle and final sequences and full LA update for three different test lengths ($m = 10, 50$ and 100). The block size of the sequential update is 200.

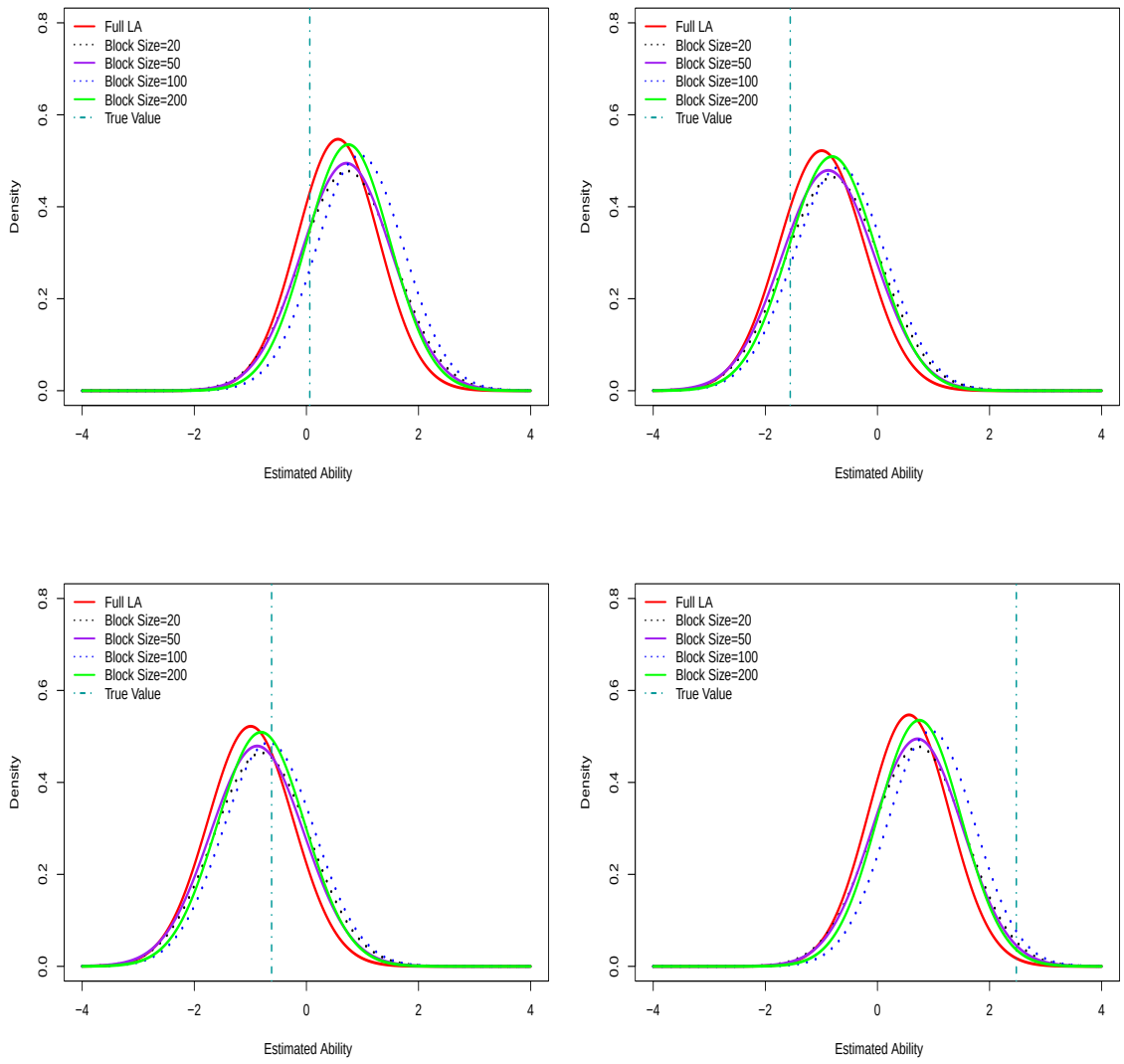
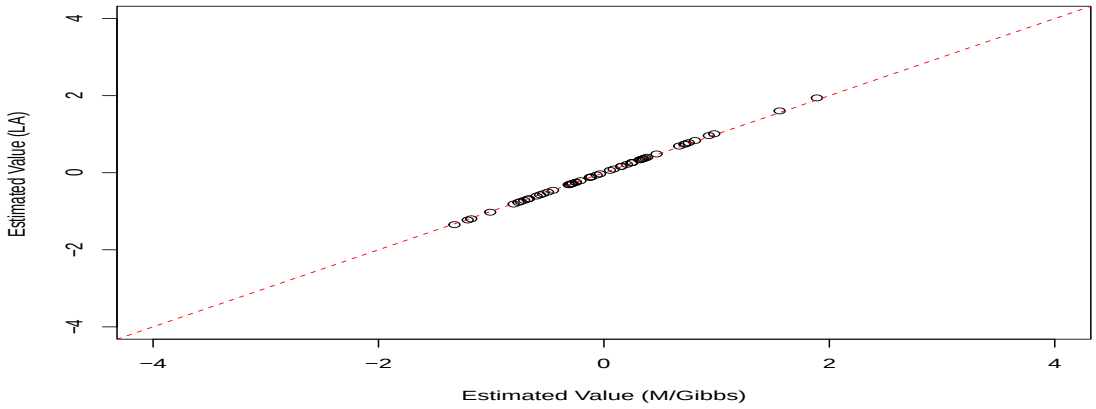


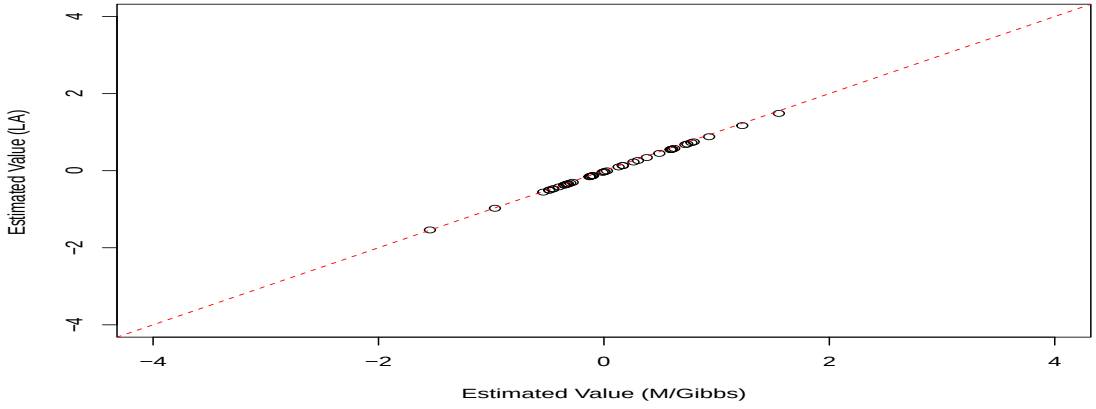
Figure B.8: Posterior distributions of ability parameters at the first sequence update for four different block sizes

Appendix C

Additional Results for the Case Study



GAT-V



GAT-Q

Figure C.1: Comparison of the points estimates of the difficulty parameters for MCMC method (M/Gibbds) against the LA. The red line illustrates the equality line The upper panel represents the estimate result of the verbal section (52 questions), and the lower panel represent the quantitative section (44 questions).

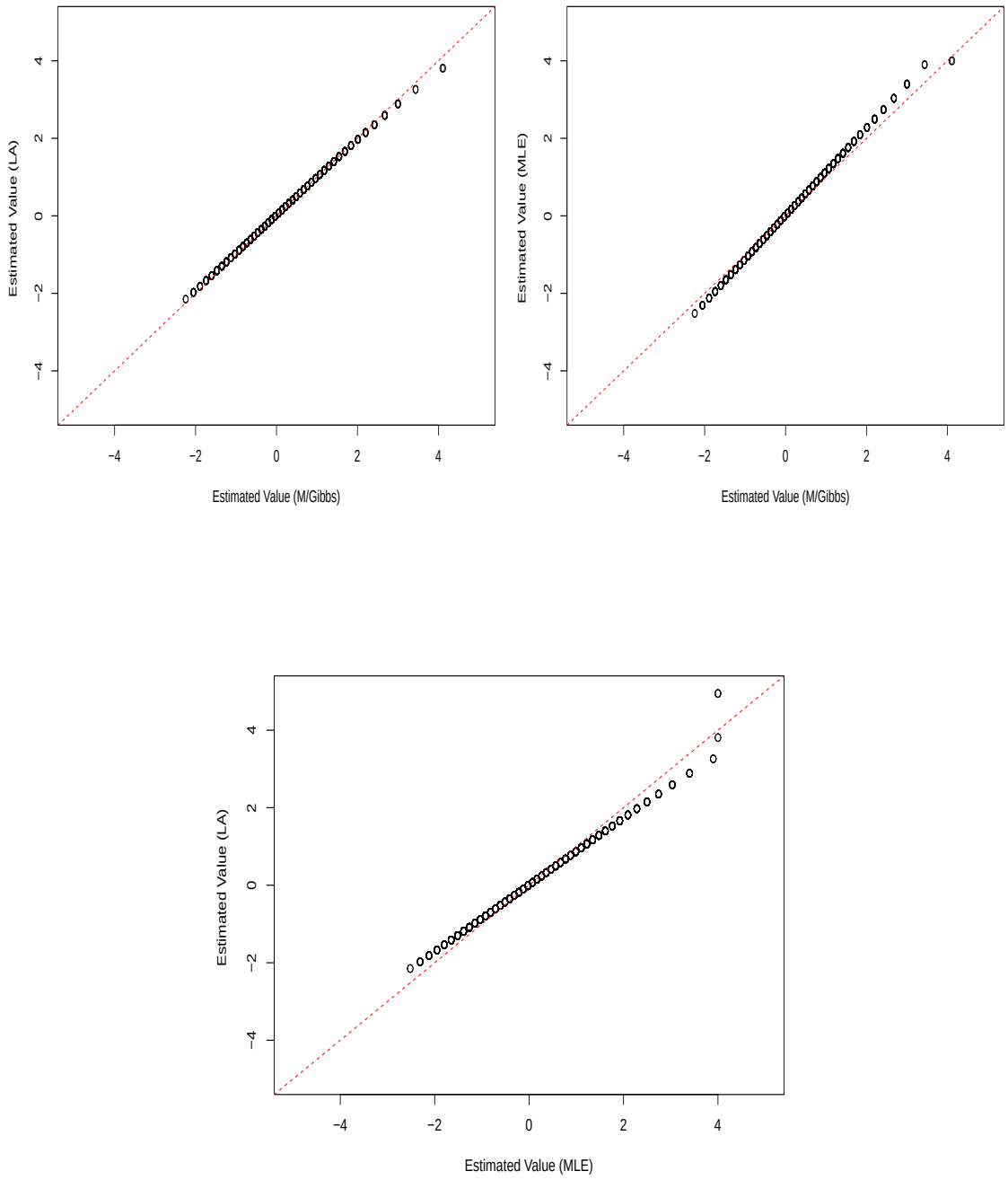


Figure C.2: Comparison between the point estimates of the ability parameters (θ) based on the verbal section (GAT -V) test questions (52 items) across the three methods; M/Gibbs, MML and LA. The red line illustrates the equality line.

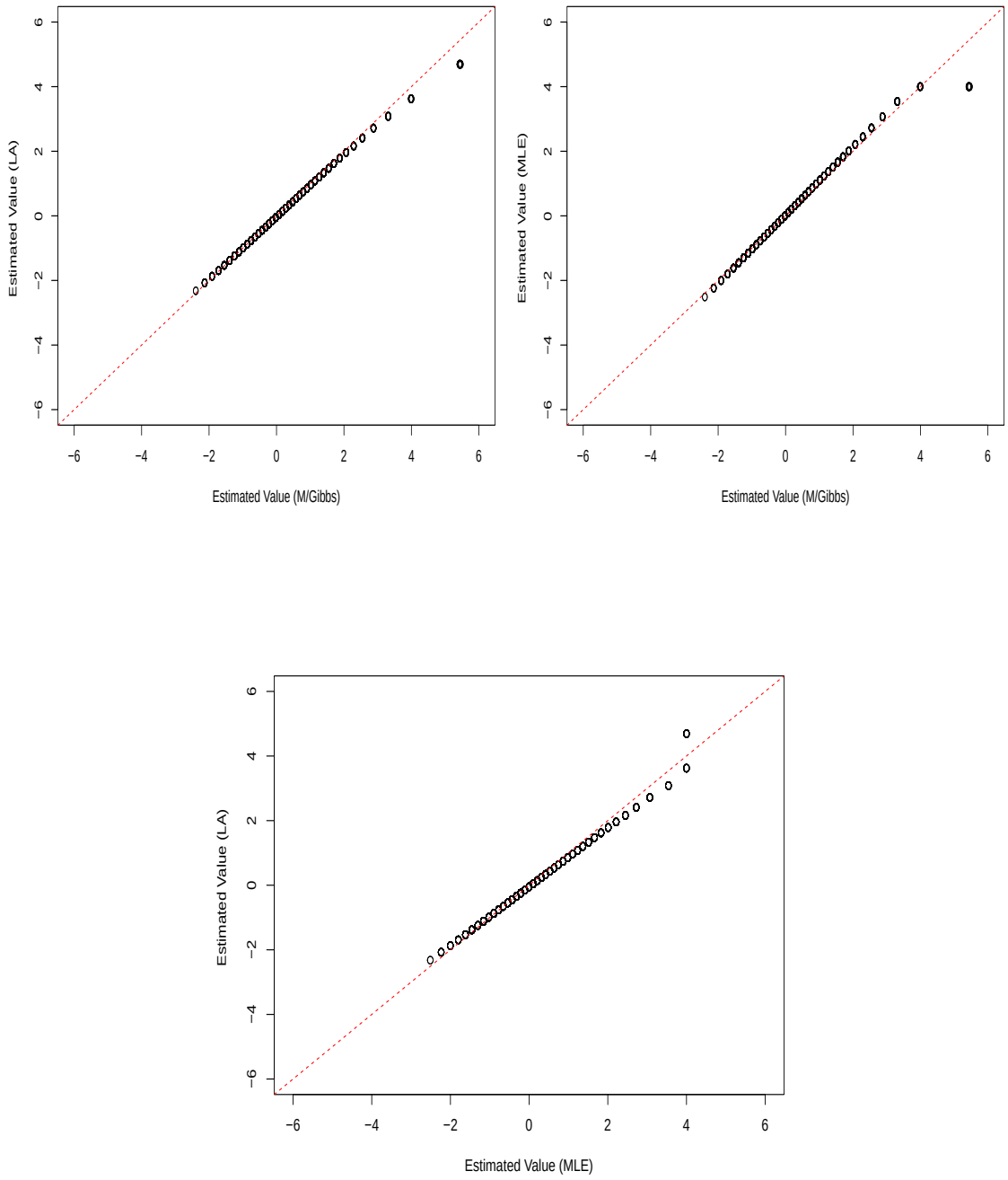


Figure C.3: Comparison between the point estimates of the ability parameters (θ) based on the quantitative section (GAT -Q) test questions (44 items) across the three methods; M/Gibbs, MML and LA. The red line illustrates the quality line.

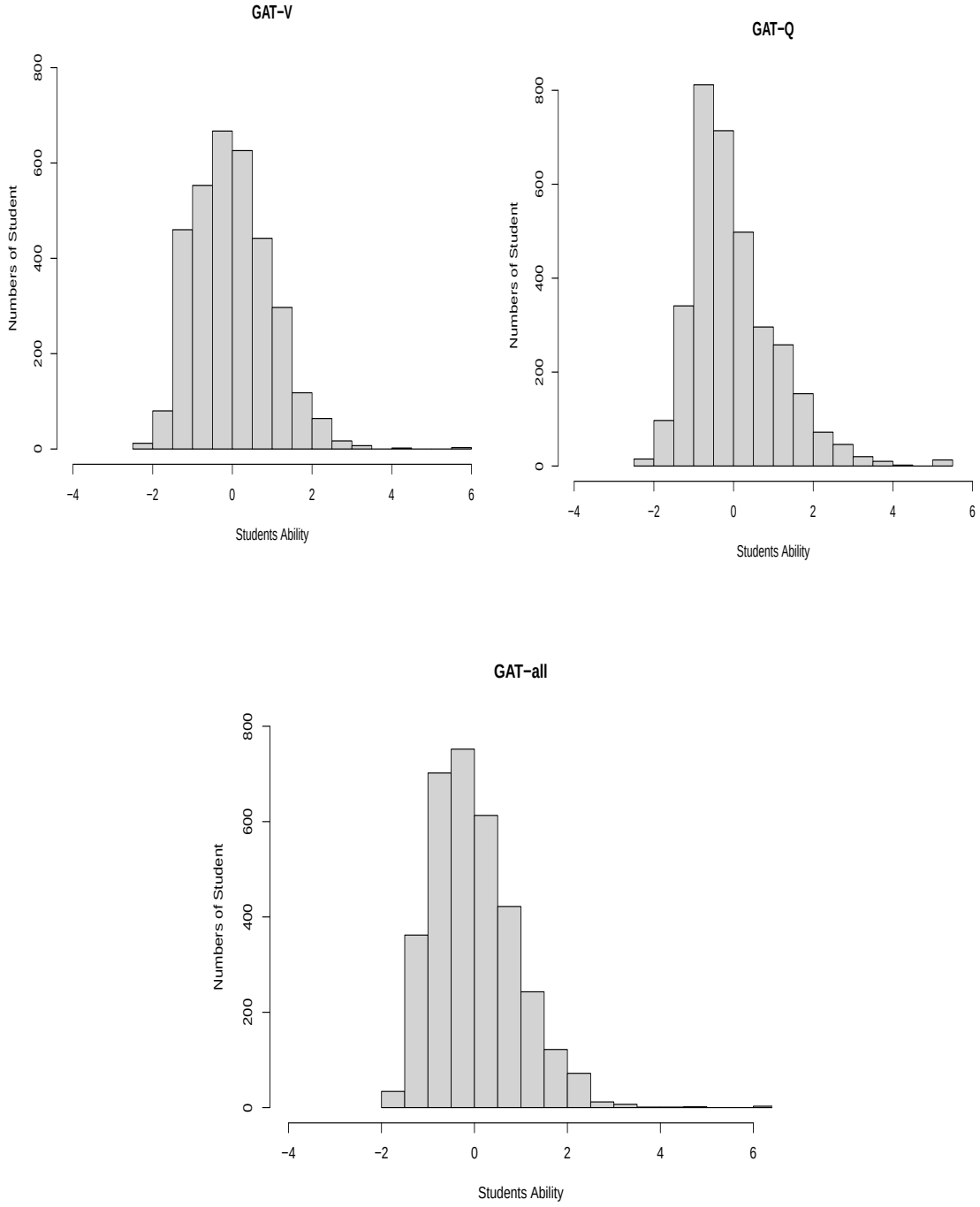


Figure C.4: Histograms of student abilities for each GAT test sections (GAT-V and GAT-Q), and the total student abilities for answering both sections (GAT-all), resulting from M/Gibbs.

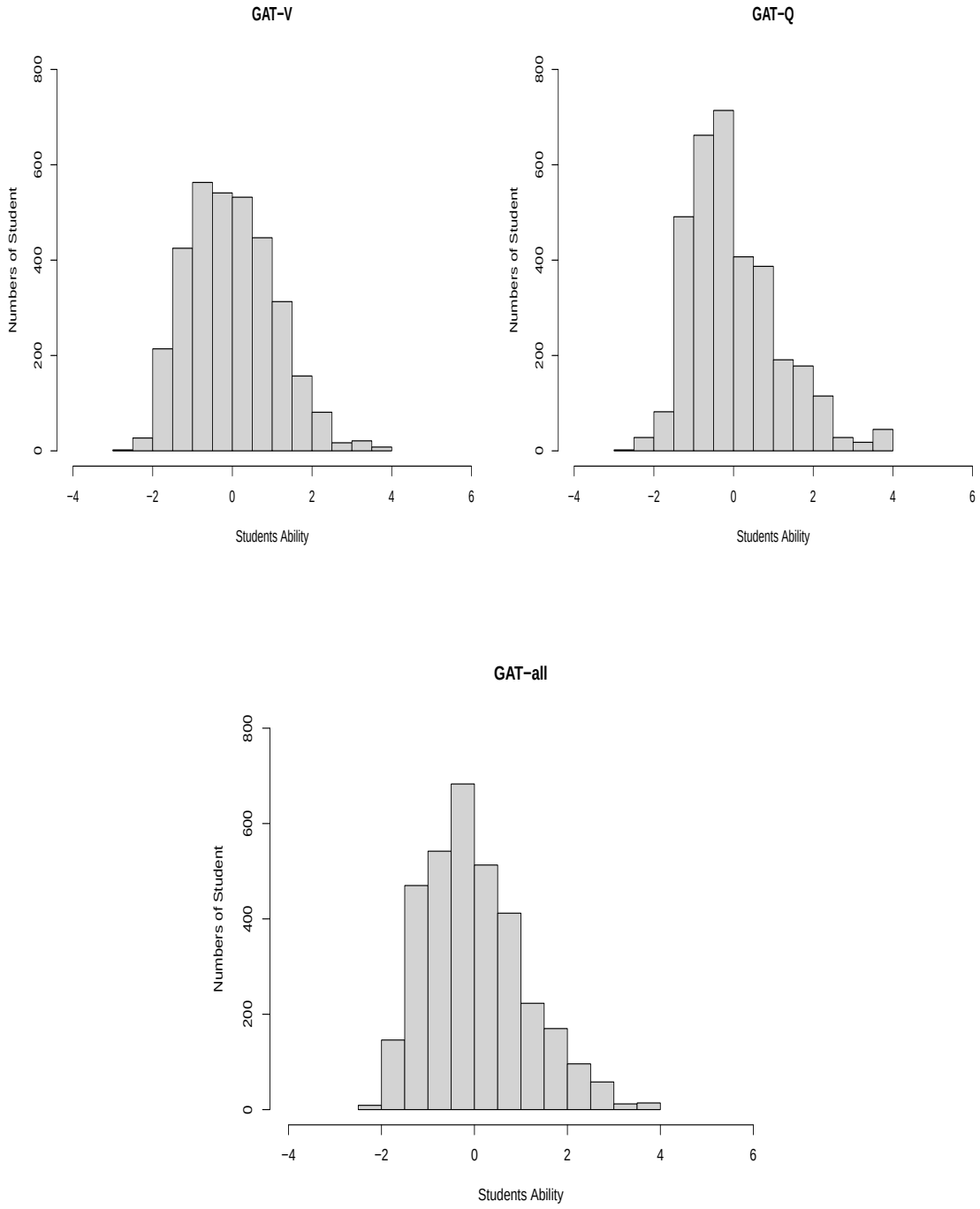


Figure C.5: Histograms of student abilities for each GAT test sections (GAT-V and GAT-Q), and the total student abilities for answering both sections (GAT-all), resulting from MLE.

Table C.1: Comparison of the differences between estimating students' ability for the middle 10 students in different block sizes (50, 200,500 and 1000) sequentially updates and a single update.

Student	LA_seq_50	LA_seq_200	LA_seq_500	LA_seq_1000	LA_all
2001	-0.10	-0.09	-0.08	-0.07	-0.08
2002	-1.07	-1.07	-1.06	-1.04	-1.05
2003	0.54	0.55	0.56	0.57	0.56
2004	-0.01	-0.00	0.01	0.02	0.01
2005	-0.19	-0.18	-0.17	-0.16	-0.17
2006	0.17	0.18	0.19	0.20	0.19
2007	-0.51	-0.51	-0.50	-0.49	-0.49
2008	0.79	0.79	0.80	0.81	0.81
2009	0.74	0.74	0.75	0.76	0.76
2010	1.87	1.88	1.89	1.89	1.89

Bibliography

Stan Development Team (2022). Stan users guide. `https://mc-stan.org/docs/2_29/stan-users-guide-2_29.pdf`.

Abdi, H. (2007). The kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, 508–510.

Albert, J. (2015). Introduction to Bayesian item response modelling. *International Journal of Quantitative Research in Education 2*(3-4), 178–193.

Alghamdi, A. K. H. and A. A. Al-Hattami (2014). The accuracy of predicting university students academic success. *J. Saudi Educ. Psychol. Assoc 1*, 1–8.

Ames, A. and E. Smith (2018). Subjective priors for item response models: Application of elicitation by design. *Journal of Educational Measurement 55*(3), 373–402.

Ames, A. J. (2018). Prior sensitivity of the posterior predictive checks method for item response theory models. *Measurement: Interdisciplinary Research and Perspectives 16*(4), 239–255.

Ames, A. J. and C. H. Au (2018). Using Stan for item response theory models. *Measurement: Interdisciplinary Research and Perspectives 16*(2), 129–134.

Andrade, J. and J. Gosling (2018). Expert knowledge elicitation using item response theory. *Journal of Applied Statistics 45*(16), 2981–2998.

Baker, F. B. (1961). Empirical comparison of item parameters based on the logistic and normal functions. *Psychometrika 26*(2), 239–246.

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

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

Bayes, T. (1763). Lii. An essay towards solving a problem in the doctrine of chances. by the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFRS. *Philosophical Transactions of the Royal Society of London* (53), 370–418.

Bernardo, J. M. and A. F. Smith (2009). *Bayesian theory*, Volume 405. John Wiley & Sons.

Berzuini, C., N. G. Best, W. R. Gilks, and C. Larizza (1997). Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association 92*(440), 1403–1412.

Beskos, A., N. Pillai, G. Roberts, J.-M. Sanz-Serna, A. Stuart, et al. (2013). Optimal tuning of the Hybrid Monte Carlo algorithm. *Bernoulli 19*(5A), 1501–1534.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

Binh, H. T. and B. T. Duy (2016). Student ability estimation based on IRT. In *2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pp. 56–61. IEEE.

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

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

Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics 6*(1), 76–90.

Bürkner, P.-C. (2019). Bayesian item response modeling in R with brms and Stan. *arXiv preprint arXiv:1905.09501*.

Bürkner, P.-C. (2020). Analysing standard progressive matrices (spm-ls) with Bayesian item response models. *Journal of Intelligence 8*(1), 5.

Cai, L. (2010). Metropolis-hastings robbins-monro algorithm for confirmatory item factor analysis. *Journal of Educational and Behavioral Statistics 35*(3), 307–335.

Cappé, O., S. J. Godsill, and E. Moulines (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE 95*(5), 899–924.

Carpenter, J., P. Clifford, and P. Fearnhead (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation 146*(1), 2–7.

Casella, G. and R. L. Berger (2021). *Statistical inference*. Cengage Learning.

Chaloner, K. and I. Verdinelli (1995). Bayesian experimental design: A review. *Statistical Science*, 273–304.

Chen, W.-H., W. Lenderking, Y. Jin, K. W. Wyrwich, H. Gelhorn, and D. A. Revicki (2014). Is rasch model analysis applicable in small sample size pilot studies for assessing item characteristics? An example using promis pain behavior item bank data. *Quality of life research 23*(2), 485–493.

Chen, W.-H. and D. Thissen (1997). Local dependence indexes for item pairs using item response theory. *Journal of Educational and Behavioral Statistics 22*(3), 265–289.

Chib, S. and E. Greenberg (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician 49*(4), 327–335.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika 89*(3), 539–552.

Cowles, M. K. and B. P. Carlin (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association 91*(434), 883–904.

Creal, D. (2012). A survey of sequential Monte Carlo methods for economics and finance. *Econometric reviews 31*(3), 245–296.

Daviet, R. (2018). Inference with Hamiltonian sequential Monte Carlo simulators. *arXiv preprint arXiv:1812.07978*.

De Ayala, R. (2009). Methodology in the social sciences. *The theory and practice of item response theory. New York*.

De Ayala, R. J. (2013). *The theory and practice of item response theory*. Guilford Publications.

Debelak, R. and I. Koller (2020). Testing the local independence assumption of the rasch model with q 3-based nonparametric model tests. *Applied Psychological Measurement 44*(2), 103–117.

Del Moral, P., A. Doucet, and A. Jasra (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(3), 411–436.

DeMars, C. (2010). *Item response theory*. Oxford University Press.

Dimitrov, D. M. and A. R. Shamrani (2015). Psychometric features of the general aptitude test–verbal part (GAT-V) a large-scale assessment of high school graduates in saudi arabia. *Measurement and Evaluation in Counseling and Development 48*(2), 79–94.

Do, H. (2021). *Parameter Recovery for the Four-Parameter Unidimensional Binary IRT Model: A Comparison of Marginal Maximum Likelihood and Markov Chain Monte Carlo Approaches*. Ph. D. thesis, Ohio University.

Douc, R. and O. Cappé (2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pp. 64–69. IEEE.

Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing 10*(3), 197–208.

Doucet, A., N. J. Gordon, and V. Krishnamurthy (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Srocessing 49*(3), 613–624.

Doucet, A. and A. M. Johansen (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering 12*(656-704), 3.

Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth (1987). Hybrid Monte Carlo. *Physics letters B 195*(2), 216–222.

Edwards, M. C., C. R. Houts, and L. Cai (2018). A diagnostic procedure to detect departures from local independence in item response theory models. *Psychological Methods 23*(1), 138.

Elfring, J., E. Torta, and R. van de Molengraft (2021). Particle filters: A hands-on tutorial. *Sensors 21*(2), 438.

Embretson, S., S. Reise, and S. Reise (2000). Item response theory for psychologists. lawrence earlboum associates. *Inc., NJ*.

Evans, M. and H. Moshonov (2006). Checking for prior-data conflict. *Bayesian Analysis 1*(4), 893–914.

Everitt, R. G., R. Culliford, F. Medina-Aguayo, and D. J. Wilson (2020). Sequential Monte Carlo with transformations. *Statistics and Computing 30*(3), 663–676.

Fan, Y., D. S. Leslie, and M. P. Wand (2008). Generalised linear mixed model analysis via sequential Monte Carlo sampling. *Electronic Journal of Statistics 2*, 916–938.

Fearnhead, P. (2002). Markov chain Monte Carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics 11*(4), 848–862.

Fearnhead, P. and B. M. Taylor (2013). An adaptive sequential Monte Carlo sampler. *Bayesian analysis 8*(2), 411–438.

Feuerstahler, L. M. (2018). Sources of error in IRT trait estimation. *Applied psychological measurement 42*(5), 359–375.

Finch, H. and B. F. French (2019). A comparison of estimation techniques for IRT models with small samples. *Applied Measurement in Education 32*(2), 77–96.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal 13*(3), 317–322.

Flora, D. B. and P. J. Curran (2004). An empirical evaluation of alternative methods of estimation for confirmatory factor analysis with ordinal data. *Psychological methods 9*(4), 466.

Fox, C. W. and S. J. Roberts (2012). A tutorial on variational bayesian inference. *Artificial Intelligence Review 38*(2), 85–95.

Fox, J.-P. (2010). *Bayesian item response modeling: Theory and applications*. Springer Science & Business Media.

Fu, Z., S. Zhang, Y.-H. Su, N. Shi, and J. Tao (2021). A Gibbs sampler for the multidimensional four-parameter logistic item response model via a data augmentation scheme. *British Journal of Mathematical and Statistical Psychology 74*(3), 427–464.

Fuglede, B. and F. Topsoe (2004). Jensen-Shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pp. 31. IEEE.

Gamerman, D. and H. F. Lopes (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC.

Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American statistical Association 95*(452), 1300–1304.

Gelfand, A. E. and A. F. Smith (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association 85*(410), 398–409.

Gelman, A. (2002). Prior distribution. *Encyclopedia of Environmetrics 3*(4), 1634–1637.

Gelman, A. and J. Hill (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University press.

Gelman, A., D. B. Rubin, et al. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science 7*(4), 457–472.

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

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

Ghosh, M., A. Ghosh, M.-H. Chen, and A. Agresti (2000). Noninformative priors for one-parameter item response models. *Journal of Statistical Planning and Inference 88*(1), 99–115.

Gilks, W. R. and C. Berzuini (2001). Following a moving targetmonte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(1), 127–146.

Glynn, P. W. and D. L. Iglehart (1989). Importance sampling for stochastic simulations. *Management Science 35*(11), 1367–1392.

Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of Computation 24*(109), 23–26.

Gordon, N. J., D. J. Salmond, and A. F. Smith (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (Radar and Signal Processing)*, Volume 140, pp. 107–113. IET.

Hammersley, J. (2013). *Monte Carlo methods*. Springer Science & Business Media.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications.

Hespanhol, L., C. S. Vallio, L. M. Costa, and B. T. Saragiotto (2019). Understanding and interpreting confidence and credible intervals around effect estimates. *Brazilian Journal of Physical Therapy 23*(4), 290–301.

Hobbhahn, M. and P. Hennig (2021). Laplace matching for fast approximate inference in generalized linear models. *arXiv preprint arXiv:2105.03109*.

Hoffman, M. D., A. Gelman, et al. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res. 15*(1), 1593–1623.

Hol, J. D., T. B. Schon, and F. Gustafsson (2006). On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing workshop*, pp. 79–82. IEEE.

Isard, M. and A. Blake (1996). Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pp. 343–356. Springer.

Jiang, Z. and J. Templin (2019). Gibbs samplers for logistic item response models via the pólya–gamma distribution: A computationally efficient data-augmentation strategy. *Psychometrika 84*(2), 358–374.

Junker, B. W., R. J. Patz, and N. M. VanHoudnos (2016). Markov chain monte carlo for item response models. *Handbook of item response theory, volume two: statistical tools 21*, 271–325.

Kass, R. E., L. Tierney, and J. B. Kadane (1991). Laplace's method in Bayesian analysis. *Contemporary Mathematics 115*, 89–99.

Kim, D., R. De Ayala, A. A. Ferdous, and M. L. Nering (2011). The comparative performance of conditional independence indices. *Applied Psychological Measurement 35*(6), 447–471.

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

Kitagawa, G. and S. Sato (2001). Monte carlo smoothing and self-organising state-space model. In *Sequential Monte Carlo methods in practice*, pp. 177–195. Springer.

Levy, R., R. J. Mislevy, and J. T. Behrens (2011). MCMC in educational research. *Handbook of Markov chain Monte Carlo: Methods and applications*, 531–545.

Li, T., M. Bolic, and P. M. Djuric (2015). Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal processing magazine 32*(3), 70–86.

Link, W. A. and M. J. Eaton (2012). On thinning of chains in MCMC. *Methods in Ecology and Evolution 3*(1), 112–115.

Liu, J. S. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association 93*(443), 1032–1044.

Liu, Y. and A. Maydeu-Olivares (2013). Local dependence diagnostics in IRT modeling of binary data. *Educational and Psychological Measurement 73*(2), 254–274.

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

Lu, T.-T. and S.-H. Shiou (2002). Inverses of 2× 2 block matrices. *Computers & Mathematics with Applications 43*(1-2), 119–129.

Luo, Y. (2018). Parameter recovery with marginal maximum likelihood and Markov chain Monte Carlo estimation for the generalized partial credit model. *arXiv preprint arXiv:1809.07359*.

Luo, Y. and H. Jiao (2018). Using the Stan program for Bayesian item response theory. *Educational and Psychological Measurement 78*(3), 384–408.

MacKay, D. J., D. J. Mac Kay, et al. (2003). *Information theory, inference and learning algorithms*. Cambridge University press.

Marcoulides, K. M. (2018). Careful with those priors: A note on Bayesian estimation in two-parameter logistic item response theory models. *Measurement: Interdisciplinary Research and Perspectives 16*(2), 92–99.

Maris, G. and T. Bechger (2009). On interpreting the model parameters for the three parameter logistic model. *Measurement 7*(2), 75–88.

Martin, A. D. and K. M. Quinn (2002). Dynamic ideal point estimation via Markov chain Monte Carlo for the US Supreme Court, 1953–1999. *Political Analysis 10*(2), 134–153.

McLean, M. W., C. J. Oates, and M. P. Wand (2017). Real-time semiparametric regression via sequential Monte Carlo.

Menéndez, M., J. Pardo, L. Pardo, and M. Pardo (1997). The jensen-shannon divergence. *Journal of the Franklin Institute 334*(2), 307–318.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*(6), 1087–1092.

Metropolis, N. and S. Ulam (1949). The Monte Carlo method. *Journal of the American Statistical Sssociation 44*(247), 335–341.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing 11*(2), 125–139.

Neal, R. M. et al. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo 2*(11), 2.

Nelder, J. A. and R. Mead (1965). A simplex method for function minimization. *The computer journal 7*(4), 308–313.

Nielsen, F. (2019). On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy 21*(5), 485.

Noether, G. E. (1967). Elements of nonparametric statistics. *Elements of Nonparametric Statistics*.

O'Hagan, A., C. E. Buck, A. Daneshkhah, J. R. Eiser, P. H. Garthwaite, D. J. Jenkinson, J. E. Oakley, and T. Rakow (2006). *Uncertain judgements: eliciting experts' probabilities*. John Wiley & Sons.

Partchev, I., M. I. Partchev, and M. Suggests (2017). Package irtoys. *A collection of functions related to item response theory (IRT)*.

Patz, R. J. and B. W. Junker (1999a). 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.

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

Perone, C. S., R. P. Silveira, and T. Paula (2021). L2m: Practical posterior Laplace approximation with optimization-driven second moment estimation. *arXiv preprint arXiv:2107.04695*.

Plummer, M., N. Best, K. Cowles, and K. Vines (2006). Coda: convergence diagnosis and output analysis for MCMC. *R news 6*(1), 7–11.

Rasch, G. (1960). Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests.

Ritter, H., A. Botev, and D. Barber (2018). Online structured Laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems 31*.

Robert, C. P., G. Casella, and G. Casella (2010). *Introducing Monte Carlo methods with R*, Volume 18. Springer.

Robert, C. P., I. CREST, and P. G. Casella (1998). Monte Carlo statistical methods.

Roberts, G. O., A. Gelman, W. R. Gilks, et al. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability 7*(1), 110–120.

Rosenthal, J. S. et al. (2011). Optimal proposal distributions and adaptive MCMC. *Handbook of Markov Chain Monte Carlo 4*(10.1201).

Sahin, A. and D. Anil (2017). The effects of test length and sample size on item parameters in item response theory.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*.

San Martín, E., J.-M. Rolin, and L. M. Castro (2013). Identification of the 1PL model with guessing parameter: parametric and semi-parametric results. *Psychometrika 78*(2), 341–379.

Schäfer, C. and N. Chopin (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing 23*(2), 163–184.

Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation 24*(111), 647–656.

Shariati, M. M., I. Korsgaard, and D. Sorensen (2009). Identifiability of parameters and behaviour of MCMC chains: a case study using the reaction norm model. *Journal of Animal Breeding and Genetics 126*(2), 92–102.

Sheng, Y. (2010). A sensitivity analysis of Gibbs sampling for 3pno irt models: Effects of prior specifications on parameter estimates. *Behaviormetrika 37*(2), 87–110.

Sinharay, S. (2006). Bayesian item fit analysis for unidimensional item response theory models. *British journal of Mathematical and Statistical psychology 59*(2), 429–449.

Smith, A. (2013). *Sequential Monte Carlo methods in practice*. Springer Science & Business Media.

South, L. F., A. N. Pettitt, C. C. Drovandi, et al. (2019). Sequential Monte Carlo samplers with independent markov chain monte carlo proposals. *Bayesian Analysis 14*(3), 753–776.

Speekenbrink, M. (2016). A tutorial on particle filters. *Journal of Mathematical Psychology 73*, 140–152.

Su, C.-L., S.-H. Chang, and R. C.-H. Weng (2018). A note on item response theory modeling for online customer ratings. *The American Statistician*.

Swaminathan, H. and J. A. Gifford (1982). Bayesian estimation in the Rasch model. *Journal of Educational Statistics 7*(3), 175–191.

Swaminathan, H., R. K. Hambleton, S. G. Sireci, D. Xing, and S. M. Rizavi (2003). Small sample estimation in dichotomous item response models: Effect of priors based on judgmental information on the accuracy of item parameter estimates. *Applied Psychological Measurement 27*(1), 27–51.

Tierney, L. and J. B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association 81*(393), 82–86.

Trippe, B., J. Huggins, R. Agrawal, and T. Broderick (2019). Lr-glm: High-dimensional Bayesian inference using low-rank data approximations. In *International Conference on Machine Learning*, pp. 6315–6324. PMLR.

Ulitzsch, E. and S. Nestler (2022). Evaluating Stan's variational Bayes algorithm for estimating multidimensional IRT models. *Psych 4*(1), 73–88.

Uyigue, A. V. and M. U. Orheruata (2019). Test length and sample size for item-difficulty parameter estimation in item response theory.

Van Ravenzwaaij, D., P. Cassey, and S. D. Brown (2018). A simple introduction to Markov chain Monte–Carlo sampling. *Psychonomic Bulletin & Review 25*(1), 143–154.

Walker, C. M. and S. N. Beretvas (2000). Using multidimensional versus unidimensional ability estimates to determine student proficiency in mathematics.

Walker, C. M. and S. N. Beretvas (2003). Comparing multidimensional and unidimensional proficiency classifications: Multidimensional IRT as a diagnostic aid. *Journal of Educational Measurement 40*(3), 255–275.

Wang, C., G. Xu, and X. Zhang (2019). Correction for item response theory latent trait measurement error in linear mixed effects models. *Psychometrika 84*(3), 673–700.

Wang, X., J. O. Berger, and D. S. Burdick (2013). Bayesian analysis of dynamic item response models in educational testing. *The Annals of Applied Statistics 7*(1), 126–153.

Weng, R. C.-H., D. S. Coad, et al. (2018). Real-time Bayesian parameter estimation for item response models. *Bayesian Analysis 13*(1), 115–137.

Wu, M., R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman (2020). Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*.

Yen, W. M. (1984). Effects of local item dependence on the fit and equating performance of the three-parameter logistic model. *Applied Psychological Measurement 8*(2), 125–145.

Yuen, K.-V. (2010). *Bayesian methods for structural dynamics and civil engineering*. John Wiley & Sons.

Zaman, A., A.-U.-R. Kashmiri, M. Mubarak, A. Ali, et al. (2008). Students ranking, based on their abilities on objective type test: Comparison of CTT and IRT.