Stockholms universitet

Comparative Analysis of Bayesian Logistic Regression Using Gibbs and Metropolis-Hastings Sampling with Diverse Prior Distributions

Xin Tang

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2024:3
Matematisk statistik
Januari 2024

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Bachelor Thesis **2024:3**
http://www.math.su.se

Comparative Analysis of Bayesian Logistic Regression Using Gibbs and Metropolis-Hastings Sampling with Diverse Prior Distributions

Xin Tang*

January 2024

Abstract

Investigating the convergence properties of MCMC algorithms is crucial for Bayesian logistic regression, because it is closely related to the accuracy of posterior distribution estimates. Effective convergence is particularly key to enhancing the reliability and precision of Bayesian logistic regression models, which are widely used to solve classification problems in fields such as medicine, biostatistics, finance, and more. This thesis utilizes diverse diagnostic tools to explore and compare the convergence performance of posterior sampling in Bayesian logistic regression using various MCMC algorithms, such as Gibbs Sampling and Metropolis-Hastings random walk, in combination with different prior assumptions, including normal, Student's t, and Cauchy distributions. The primary motivation of this paper is to provide guidance for selecting MCMC algorithms and prior assumptions in Bayesian logistic regression contexts through the experimental results. The primary conclusion of this research is that Gibbs Sampling, in contrast to the Metropolis-Hastings random walk, consistently attains quicker convergence and superior sampling effectiveness across all three of our predetermined prior assumptions. This holds for both low and high correlation scenarios within our simulated data. Moreover, normal priors contribute to higher sampling effectiveness for Gibbs models, and they also lead to faster convergence for Metropolis-Hastings models than Student's t and Cauchy priors.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: bupttaxi@gmail.com. Supervisor: Taras Bodnar, Johannes Heiny.

Acknowledgements

I would like to express my heartfelt thanks to my supervisors, *Taras Bodnar* and *Johannes Heiny*, at the Department of Mathematics, Stockholm University. Their guidance and expertise have been invaluable throughout this project, providing me with the support and knowledge necessary to navigate this complex and challenging journey.

Additionally, I utilized specific technical tools to optimize the syntax and formatting of my paper. I am grateful for the assistance provided by OpenAI's language model in the text proofreading process, which saved me time and enhanced my work efficiency.

Contents

1 Introduction

The article begins with logistic regression. Logistic regression, a key statistical method for binary or categorical outcomes, is especially useful in predicting event probabilities. Widely applied in fields like medicine, social science, marketing, and econometrics, it is used for tasks such as disease risk assessment, voter behavior analysis, and customer purchase propensity prediction.

Bayesian logistic regression, integrating prior knowledge, enhances model interpretability and prediction accuracy. It treats parameters as random variables, offering uncertainty measures in parameter estimates, a feature traditional logistic regression lacks. Additionally, Bayesian approaches adapt well to complex data structures, like missing data or hierarchical data. Precisely for this reason, the ability of Bayesian logistic regression to provide insights into the uncertainty of predictions and to handle complex models with hierarchical structures makes it also valuable for various applications in machine learning.

In Bayesian logistic regression research, a major challenge is the difficulty in analytically calculating the posterior distribution of parameters. This is primarily because, according to Bayes' theorem, the posterior distribution is the product of the likelihood and the prior distribution divided by the marginal density integral, which often results in a complex form. Especially, the marginal density integral in the denominator is hard to compute analytically. Moreover, the involvement of the sigmoid function in logistic regression models further complicates the integral solution. Consequently, MCMC (Markov Chain Monte Carlo, hereinafter referred to as MCMC) algorithms are typically introduced. MCMC avoids the necessity of solving complex integrals directly. This numerical approach is particularly useful in situations where analytical solutions are hard to obtain or computationally expensive.

Given the widespread application of Bayesian logistic regression and the significance of MCMC algorithms in obtaining posterior distributions of parameters, this thesis focuses on comparing different MCMC algorithms combined with various prior information. The comparison will be made in terms of convergence, multi-chain mixing, model complexity, fitting and so on. The primary motivation of this paper is to provide guidance for selecting prior assumptions and MCMC algorithms in Bayesian logistic regression contexts through experimental results. Different combinations of MCMC algorithms and prior assumptions yield varying levels of convergence and reliability. Knowing this variability is crucial for selecting the most effective and reliable sampling methods in Bayesian logistic regression to model real-world problems.

In this paper, I will utilize three commonly used informative priors: the normal distribution, Student's t-distribution, and the Cauchy distribution. The normal distribution prior is generally applied to depict the symmetry and centrality in parameter distributions. In contrast, heavy-tailed distributions such as the t-distribution and the Cauchy distribution are more effective for handling outliers and improving model robustness. Additionally, my research will concentrate on two prevalent MCMC algorithms: Gibbs Sampling by using PG (Pólya-Gamma, hereinafter referred to as PG) augmentation and the MH (Metropolis-Hastings, hereinafter referred to as MH) random walk.

The choice of these two MCMC algorithms is based on several considerations. Firstly, both Gibbs and MH algorithms are fundamental in the realm of MCMC and are widely used in practical applications. Secondly, Gibbs and MH algorithms show unique advantages in handling different types of data and models. For example, Gibbs is suitable for scenarios where conditional distributions are easily manageable, while MH is applicable to a broader range of situations. This makes them suitable for comparative analysis in this paper.

In the upcoming chapters, I will introduce the theoretical background relevant to this paper from *Chapter 2* to *Chapter 6*. *Chapter 7* will be dedicated to introducing the simulated data, followed by *Chapter 8*, which presents the experimental results. *Chapter 9* will be the discussion section of this thesis, covering issues encountered during the experimental process and the subsequent reflections.

It is worth mentioning that due to the lack of existing libraries and function support, I have reimplemented the following content in R. These include: Gibbs sampling by using PG augmentation combined with a Gaussian prior, based on the paper by Polson et al. [12]; Gibbs sampling by using PG augmentation combined with heavy-tailed distributions, namely the Student's t-distribution and the Cauchy distribution, based on the paper by Ghosh et al. [10]; MH random walk algorithms combined with three different priors (normal, Student's t, and Cauchy distributions), based on the book by Held et al. [11]; and the improved Split-$\hat{R}$ method and rank plot, based on the paper by Vehtari et al. [13]. The pseudocode for part of these algorithms will be provided in subsequent chapters.

2 Theoretical Framework of Logistic Regression

In the realm of statistical analysis, logistic regression stands as a pivotal model for the exploration and interpretation of categorical response data. Its utility is most pronounced in scenarios where the response variable under study is dichotomous, thereby encapsulating outcomes in a binary format. This section outlines the basic concepts and mathematical formulation of the logistic regression model, providing a pathway to comprehend the relationship between a binary response and one or more explanatory variables. The content of this section is based on Chapter 5 of the work by Agresti et al. [4].

2.1 Logistic Regression Model

Envision an empirical setting where we are presented with N observations. Each observation is associated with a binary response variable Y, exhibiting possible values of either 0 or 1. Accompanying these responses are M predictor variables, collectively represented by $X_i = (X_{1i}, X_{2i}, \ldots, X_{Mi})$ for the i^{th} observation. The core objective of logistic regression is to model the probability $\pi(X_i)$ that Y_i will equal 1 given the predictors X_i. This relationship is articulated through the equation:

$$\pi(X_i) = P(Y_i = 1|X_i) = \frac{e^{\eta_i}}{1 + e^{\eta_i}}, \tag{1}$$

where $\eta_i = \beta^T X_i$ is the linear predictor. The logistic regression model employs a link function to connect the linear predictor to the probability scale. This link function in the context of logistic regression is the logit function, which expresses the log odds of the binary outcome:

$$\text{logit}(\pi(X_i)) = \log\left(\frac{\pi(X_i)}{1 - \pi(X_i)}\right) = \eta_i. \tag{2}$$

2.2 Link Function in Logistic Regression

In the logistic regression framework, the 'link function' is a critical component that facilitates the modeling of a binary response variable on a continuous scale. It transforms the predicted log odds of the outcome into a probability bounded between 0 and 1. Specifically, the logit link function is defined as the natural logarithm of the odds of $\pi(X_i)$:

$$\log\left(\frac{\pi(X_i)}{1 - \pi(X_i)}\right) = \beta^\top X_i. \tag{3}$$

In logistic regression, the coefficient β represents the change in the log odds of the outcome for a one-unit increase in the corresponding predictor variable. Exponentiating these coefficients transforms them into odds ratios, which describe the relative effect of the predictor variables on the odds of the outcome.

Within the framework of frequentist statistics, the parameters of the logistic regression model are estimated using the method of maximum likelihood estimation (MLE). MLE is aimed at finding the parameter values that maximize the likelihood of the observed data, which, under the frequentist view, is equivalent to finding the parameter values that render the observed outcomes as the most probable given the specified model.

2.3 Likelihood Formulation in Logistic Regression

In logistic regression analysis, the likelihood function is crucial for parameter estimation, serving a fundamental role across both frequentist and Bayesian statistical paradigms. For frequentists, it is instrumental in finding optimal parameters by maximizing the likelihood function. In the Bayesian perspective, the likelihood function is equally essential as it updates prior knowledge to form the posterior probability distribution, where the posterior is proportional to the product of the likelihood and the prior.

Consider a set of N observations with a binary response variable Y_i, each associated with a set of M predictors denoted by X_i. Each response Y_i can be conceptualized as a Bernoulli trial with a success probability $\pi(X_i)$, which is a function of the predictors through the logistic function.

The probability of observing a specific outcome Y_i given X_i is modeled as:

$$P(Y_i|X_i) = \pi(X_i)^{Y_i}(1 - \pi(X_i))^{(1-Y_i)}. \tag{4}$$

The likelihood function $L(\beta)$ for all observations is the product of individual probabilities, representing the joint probability of observing the given set of responses:

$$L(\beta) = \prod_{i=1}^{N} \pi(X_i)^{Y_i}(1 - \pi(X_i))^{(1-Y_i)}. \tag{5}$$

This function is maximized to obtain the estimates for β, which are the coefficients that relate the predictors to the response variable in a logistic regression model.

3 Bayesian Inference in Logistic Regression

Based on Chapter 6 of the work by Held et al. [11], Bayesian inference provides a probabilistic approach to the estimation of a model's parameters. Unlike traditional methods which treat parameters as fixed but unknown quantities, Bayesian inference considers them as random variables with associated probability distributions. This chapter will introduce the Bayes' theorem and the important concepts associated with Bayesian theory, and will lead into the Bayesian logistic regression model.

3.1 Bayesian Theorem

Bayesian inference is rooted in Bayes' theorem, which relates the likelihood and the prior knowledge to form the posterior distribution. For any two events A and B, where $P(B) > 0$, the theorem is defined as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{6}$$

where $P(A|B)$ is the posterior probability of A given B, $P(B|A)$ is the likelihood, $P(A)$ is the prior probability of A, and $P(B)$ is the marginal likelihood of B.

3.2 Prior and Posterior Distributions

In Bayesian statistics applied to logistic regression, our inferential process begins by articulating our initial assumptions about the parameters θ, encapsulated within the prior

distribution $\pi(\theta)$. This prior distribution encompasses our a priori understanding, which can be based on historical data, expert knowledge, or other relevant information.

Upon observing new data, these initial beliefs are updated using Bayes' theorem. The theorem provides a mechanism to revise our beliefs in the light of new evidence, culminating in the posterior distribution $p(\theta|y)$. The fundamental equation of Bayes' theorem in this context is given by:

$$p(\theta|y) = \frac{f(y|\theta)\pi(\theta)}{\int f(y|\theta)\pi(\theta)d\theta}. \tag{7}$$

This equation asserts that the posterior distribution $p(\theta|y)$ is a result of updating the prior distribution $\pi(\theta)$ with the likelihood $f(y|\theta)$, which measures the probability of observing the data y given the parameters θ. The denominator, the integral of the product of the likelihood and the prior over all possible values of θ, serves as a normalizing factor ensuring that the posterior distribution is a true probability distribution.

It follows from Bayes' theorem that the posterior distribution is directly proportional to the product of the likelihood and the prior distribution:

$$p(\theta|y) \propto f(y|\theta)\pi(\theta). \tag{8}$$

This proportional relationship enables us to understand the posterior distribution qualitatively even without the exact normalization factor. In the Bayesian framework, the posterior distribution represents a synthesis of our existing prior beliefs with the new evidence provided from the data, thus giving a comprehensive probabilistic perspective on our parameters. This approach transforms our view of parameters from fixed unknowns to random variables characterized by a probability distribution. This distribution is continually refined and updated as new evidence emerges, allowing for a more adaptive and informed understanding of the parameters based on both prior knowledge and observed data.

3.3 Posterior Mean and Variance

The posterior mean $E(\theta|y)$ and variance $\text{var}(\theta|y)$ are pivotal in Bayesian inference, summarizing the updated beliefs about the parameters after observing the data. They are defined as follows:

$$E(\theta|y) = \int \theta \, p(\theta|y) \, d\theta, \tag{9}$$

$$\text{var}(\theta|y) = \int (\theta - E(\theta|y))^2 \, p(\theta|y) \, d\theta. \tag{10}$$

The posterior mean is a measure of the central tendency of the parameter values, which is the expected value of the parameters given the data. The posterior variance quantifies the uncertainty around the posterior mean, reflecting the variability of the parameter estimates.

3.4 Model Formulation with Bayesian Logistic Regression

In Bayesian logistic regression, we use a likelihood function similar to that of the traditional logistic regression, combined with a prior distribution for the parameters. The likelihood

function $L(\beta)$ for N observations and M predictors is:

$$L(\beta) = \prod_{i=1}^{N} \pi(x_i)^{y_i}(1 - \pi(x_i))^{(1-y_i)}, \tag{11}$$

where $\pi(x_i)$ is the modeled probability of the i-th observation being a success.

Given that we have multiple model parameters β, we express the prior as a product of density functions over all M parameters to reflect our independent prior beliefs about each parameter:

$$p(\beta) = \prod_{j=0}^{M} f_j(\beta_j), \tag{12}$$

where $f_j(\beta_j)$ is the prior distribution for β_j, the j-th coefficient. This factorial form allows us to combine different prior distributions for each coefficient, catering to our specific prior knowledge or assumptions about the parameter space. Although *Equation 12* presents the priors as a product of individual terms, suggesting independence, it can be extended to include dependencies between parameters. This involves defining priors $f_j(\beta_j)$ that encapsulate these dependencies, possibly through conditioning on other parameters or including dependency-enforcing hyperparameters. Thus, the factorial form can still be compatible with some dependent prior structure.

The posterior distribution $p(\beta|y)$ is then derived by applying Bayes' theorem:

$$p(\beta|y) = \frac{L(\beta) \prod_{j=0}^{M} f_j(\beta_j)}{\int L(\beta) \prod_{j=0}^{M} f_j(\beta_j) d\beta}, \tag{13}$$

which updates our beliefs about the regression coefficients β after considering the evidence from the observed data. The denominator is the integral of the numerator over all possible values of β, ensuring that $p(\beta|y)$ is a proper probability distribution.

In practice, computing the integral in the denominator is often noncomputable, especially as the number of regression coefficients increases. Therefore, we typically use numerical approximation techniques such as MCMC to sample from the posterior distribution.

4 Theoretical Background on MCMC Methods

MCMC methods are a class of algorithms used for sampling from complex probability distributions, especially in scenarios where direct sampling is not tractable due to the absence of analytical solutions. These methods allow for the construction of Markov chains whose stationary distributions are the target posterior distributions $p(\theta|y)$.

The essence of MCMC lies in its numerical sampling prowess, providing a practical approach to estimate distributions that are otherwise mathematically noncomputable. This is particularly valuable when dealing with posterior distributions that do not have closed-form expressions, which is often the case in high-dimensional spaces or with non-conjugate priors. By employing a transition mechanism that respects the Markov property — the future state depends only on the current state and not on the sequence of events that preceded it — MCMC methods iteratively construct a chain of samples. As the number of iterations grows large, the distribution of the samples approximates the true posterior distribution, despite the lack of an analytical form. This iterative approach not only approximates the posterior

but also allows for the estimation of other integrals and expectations with respect to the posterior that are crucial for Bayesian inference and decision making.

The most commonly used MCMC methods include the Gibbs sampler and the MH algorithm. These are also the methods employed in this thesis. In the following subsections, I will introduce the basic principles of Gibbs sampling and the MH random walk algorithm. The highlight of this thesis, Gibbs sampling based on PG augmentation, will be covered in a separate chapter (*Chapter 5*) due to its extensive theoretical content. The content of this section is informed by Chapter 11 of the work by Gelman et al. [7] and Chapter 8.4 of work by Held et al. [11].

4.1 Gibbs Sampling Methodology

Gibbs sampling, according to Gelman et al. [7], is a MCMC algorithm particularly useful in sampling from a multivariate probability distribution. It was introduced by Geman and Geman [9] in 1984 and is also known as alternating conditional sampling. Gibbs sampling is based on the principle that it is simpler to sample from a series of univariate conditional distributions than to sample directly from a joint multivariate distribution.

Consider a parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)$ divided into d subvectors or components. The Gibbs sampler iterates through these subvectors, sampling each one conditional on the current values of all other components.

The algorithm can be described as follows for iteration t:

1. Start with some initial values $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_d^{(0)})$.
2. For each component θ_j at iteration t, update its value by sampling from the conditional distribution $p(\theta_j | \boldsymbol{\theta}_{-j}^{(t-1)}, \mathbf{y})$, where $\boldsymbol{\theta}_{-j}^{(t-1)}$ represents all components of $\boldsymbol{\theta}$ except θ_j at their current values and $\mathbf{y}$ is the observed data.
3. Repeat this process, cycling through each component θ_j, until convergence is achieved or a specified number of iterations is reached.

This procedure generates a Markov chain $\{\boldsymbol{\theta}^{(t)}\}$ where each state is a vector of the parameter values at iteration t. The chain has the property that as $t \to \infty$, the distribution of $\boldsymbol{\theta}^{(t)}$ converges to the joint posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$, assuming proper conditions are met.

For many standard statistical models, the conditional distributions required for Gibbs sampling are often conjugate, which simplifies computation and makes the method particularly attractive. This sampler is widely used in Bayesian statistics for its ease of implementation and its applicability to complex hierarchical models.

This paper adopts the Gibbs sampling algorithm based on PG data augmentation, the details of which will be elaborated in the next chapter.

4.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a cornerstone of the MCMC methods. It allows sampling from complex probability distributions, particularly those in high-dimensional spaces, by generating a Markov chain that converges to the desired posterior distribution $p(\theta|y)$.

According to Chapter 8.4 of work by Held et al. [11], the MH algorithm proceeds by generating a sequence of sample values, where each new sample, θ^*, is drawn from a proposal

distribution $f^*(\theta^*|\theta^{(m)})$, and $\theta^{(m)}$ is the current state of the chain. The proposed value θ^* is accepted as the next state in the Markov chain with a probability α, given by:

$$\alpha = \min\left\{1, \frac{f(\theta^*|y) \cdot f^*(\theta^{(m)}|\theta^*)}{f(\theta^{(m)}|y) \cdot f^*(\theta^*|\theta^{(m)})}\right\}, \tag{14}$$

where $f(\theta|y)$ is the target posterior distribution and $f^*(\theta^*|\theta^{(m)})$ is the proposal density. If θ^* is rejected, then $\theta^{(m+1)}$ is set to $\theta^{(m)}$.

4.3 Metropolis-Hastings Random Walk Algorithm

After introducing the basic principles of Gibbs sampling and the MH algorithm, I will now present the theoretical background of the first MCMC algorithm used in the experimental part of this thesis, namely the MH random walk. The MH random walk algorithm, according to the book of Held et al. [11], is a variation of the MCMC method tailored for sampling from complex probability distributions that are challenging to sample directly.

The MH random walk algorithm differentiates itself from the standard Metropolis-Hastings algorithm primarily through its choice of proposal distribution. While the standard MH algorithm can utilize various forms of proposal distributions, the MH random walk algorithm specifically employs a symmetric distribution centered at the current sample point—typically a zero-mean normal distribution—to generate new candidate points, facilitating exploration within the parameter space.

The MH random walk algorithm updates parameter values by incorporating a random perturbation at each step, where this perturbation is drawn from a symmetric distribution centered on the current parameter value. This stochastic perturbation strategy allows the algorithm to "randomly walk" through the parameter space, potentially reaching and investigating various regions of the target distribution. The random walk proposal enhances the algorithm's exploratory capabilities by enabling the chain to move through different regions of the target distribution, not just towards areas of higher probability, thus improving sampling efficiency and avoiding convergence to local optima. The acceptance probability for moving to a new state in the MH random walk algorithm is given by:

$$\alpha = \min\left\{1, \frac{f(\theta^*|x)}{f(\theta^{(m)}|x)}\right\}, \tag{15}$$

since the random walk proposal is symmetric about the current point, the proposal distribution $f^*(\theta^*|\theta^{(m)})$ (the probability of moving from the current state $\theta^{(m)}$ to the proposed state θ^*) and $f^*(\theta^{(m)}|\theta^*)$ (the probability of moving from the proposed state θ^* back to the current state $\theta^{(m)}$) are equal. This property allows us to disregard the proposal distribution in the calculation of the acceptance probability, as they cancel each other out when calculating the ratio.

4.4 Pseudocode for MH Random Walk Algorithm

The following presents the pseudocode for implementing the Metropolis-Hastings random walk algorithm with normal, Student's t, and Cauchy priors.

Algorithm 1 Metropolis-Hastings Random Walk for Normal, Student's t and Cauchy Priors

Initialize $\beta^{(0)} = \text{MLE}$
Set hyperparameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$
if prior type is "normal" **then**
    Set prior specific parameters for normal distribution
else if prior type is "t"/"Cauchy" **then**
    Set degrees of freedom *df* for Student's t-distribution / Cauchy distribution
end if
for $i \leftarrow 1, \ldots, N_{\text{iter}}$ **do**
    $\boldsymbol{\beta}_i \sim \text{MVN}(\boldsymbol{\beta}_{\text{chain}}^{(i-1)}, \boldsymbol{\Sigma})$ ▷ Draw from multivariate normal
    Calculate acceptance probability α
    if prior type is "normal" **then**
        $\alpha \leftarrow \min\{1, \frac{\text{post\_normal}(\boldsymbol{\beta}_i, \mathbf{Y}, \mathbf{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\text{post\_normal}(\boldsymbol{\beta}_{\text{chain}}^{(i-1)}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\}$
    else if prior type is "t" (df = 7) or "Cauchy" (df = 1) **then**
        $\alpha \leftarrow \min\{1, \frac{\text{post\_t}(\boldsymbol{\beta}_i, \mathbf{Y}, \mathbf{X}, df)}{\text{post\_t}(\boldsymbol{\beta}_{\text{chain}}^{(i-1)}, \mathbf{Y}, \mathbf{X}, df)}\}$
    end if
    if $\alpha > \text{runif}(1)$ **then**
        $\boldsymbol{\beta}_{\text{chain}}^{(i)} \leftarrow \boldsymbol{\beta}_i$
    else
        $\boldsymbol{\beta}_{\text{chain}}^{(i)} \leftarrow \boldsymbol{\beta}_{\text{chain}}^{(i-1)}$
    end if
    Store $\boldsymbol{\beta}_{\text{chain}}^{(i)}$
end for
return $\boldsymbol{\beta}_{\text{chain}}$

In this pseudocode:

- $\boldsymbol{\beta}_{\text{chain}}^{(i)}$ represents the current values for the regression coefficients at iteration i.
- $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and covariance matrix for the prior distribution, and $\boldsymbol{\Sigma}$ is also used as the covariance matrix for the proposal distribution.
- N_{iter} is the number of iterations for the algorithm.
- Prior type indicates whether a normal, Student's t-distribution or Cauchy distribution is used as the prior.
- For the Student's t-prior, *df* represents the degrees of freedom for the distribution, which is set to 7. In special case $df = 1$, it represents Cauchy distribution.
- $\mathbf{Y}$ is the binary response vector in the dataset, consisting of 0 and 1 values, generated from the simulation for Bayesian logistic regression.
- $\mathbf{X}$ is the matrix of covariates.
- α is the acceptance probability calculated at each iteration based on the prior type.
- $\boldsymbol{\beta}_{\text{chain}}$ stores the posterior sampling of the parameters across iterations.
- **post_normal** and **post_t** functions calculate the posterior probability for Bayesian logistic regression. The **post_normal** function uses a normal distribution as the prior, while **post_t** is designed for cases where the prior is a Student's t-distribution, including its special case of the Cauchy distribution with one degree of freedom.

5 Gibbs Sampling for Bayesian Logistic Regression

In this chapter, we will delve into the core of the theoretical segment of our thesis, exploring the foundational principles of the algorithms presented in two key pieces of literature and reimplementing their algorithmic components. These two algorithms are the PG-augmented Gibbs sampling for prior distributions with Gaussian distribution(Polson et al. [12]), and the PG-augmented Gibbs sampling for prior distributions with heavy-tailed distributions, specifically the Cauchy and t-distributions (Ghosh et al. [10]).

5.1 Pólya-Gamma Augmentation Gibbs Sampling with Gaussian prior for Bayesian Logistic Regression

Bayesian logistic regression poses computational challenges due to the non-conjugacy of the binomial likelihood with the Gaussian prior. This non-conjugacy leads to intractable posterior distributions, making direct sampling methods infeasible. To address this, Polson et al. [12] introduced a novel PG data augmentation strategy in 2013, which offers a tractable approach to sampling from the posterior distribution.

5.1.1 The Pólya-Gamma Augmentation Scheme

The key innovation in Polson et al. [12] is the introduction of PG latent variables, transforming the logistic regression likelihood into a form that resembles a Gaussian model. This transformation is facilitated by the PG distribution's property of being a scale mixture of Gaussians, allowing for an efficient Gibbs sampling scheme. The approach diverges from that of Albert and Chib [5] by utilizing a scale mixture rather than a location mixture, and by replacing truncated normals with PG variables.

5.1.2 Advantages of the Pólya-Gamma Method

The PG method is particularly advantageous due to its conjugacy properties when combined with a Gaussian prior. The integral identity proved in *Theorem 1* in the 2013 paper by Polson et al. [12],

$$\frac{(e^{\psi})^a}{(1+e^{\psi})^b} = 2^{-b} e^{\kappa\psi} \int_0^{\infty} e^{-\omega\psi^2/2} p(\omega) d\omega, \tag{16}$$

where $p(\omega)$ denotes the density of the random variable ω following a Pólya-Gamma distribution, $\omega \sim PG(b, 0)$, for $b > 0$. Here, $\psi = x_i^T \beta$ represents the log odds of success, and β is assumed to have a Gaussian prior, $\beta \sim N(\phi, B)$. The term $\kappa = a - \frac{b}{2}$ demonstrates how the logistic likelihood can be expressed in a Gaussian form, allowing direct sampling from the posterior. The PG augmentation simplifies the posterior to be Gaussian conditional on the PG variables, leading to the Gibbs sampling steps:

$$(\omega_i|\beta) \sim PG(n_i, x_i^T \beta), \tag{17}$$

and

$$(\beta|y, \omega) \sim N(m_\omega, V_\omega), \tag{18}$$

with $V_\omega = (X^T \Omega X + B^{-1})^{-1}$ and $m_\omega = V_\omega(X^T \xi + B^{-1}\phi)$, where X is the design matrix and Ω is the diagonal matrix whose diagonal elements are ω_i. Furthermore, ϕ and B represent

the mean vector and variance-covariance matrix of the Gaussian prior, respectively. The vector ξ is defined as $\xi = (y_1 - n_1/2, \ldots, y_N - n_N/2)$, where y_i is the number of successes, n_i is the number of trials, and $y_i \sim \text{Binom}(n_i, \frac{1}{1+e^{-\psi_i}})$.

In this paper, we focus on a specific scenario where $a = b = 1$ and $n_i = 1$. Under these conditions, the formulas previously discussed can be reformulated as follows:

Considering the integral identity in *Equation 16*, with $a = b = 1$, it simplifies to:

$$\frac{e^{\psi}}{(1+e^{\psi})} = 2^{-1} e^{\kappa\psi} \int_0^{\infty} e^{-\omega\psi^2/2} p(\omega) d\omega, \tag{19}$$

where $\kappa = a - b/2$ becomes $\kappa = 1 - 1/2 = 1/2$.

In the PG augmentation context, especially for binary logistic regression, we set $n_i = 1$ since it corresponds to the number of trials or observations for the ith data point. In this scenario, each data point represents a single Bernoulli trial, making n_i naturally equal to 1. The Gibbs sampling steps become:

$$(\omega_i|\beta) \sim PG(1, x_i^T\beta), \tag{20}$$

and

$$(\beta|y, \omega) \sim N(m_\omega, V_\omega), \tag{21}$$

where the expressions for V_ω and m_ω remain the same as the general case mentioned earlier (see *formulas 17* and *18*), but the sampling for ω_i and the value of κ are adjusted as per the specified conditions.

5.1.3 Gaussian Representation of the Posterior

To derive our Gibbs sampler, we turn to *Theorem 1* in the 2013 paper by Polson et al. [12] and express the likelihood contribution of observation i as follows:

$$\begin{aligned} L_i(\beta) &= \left(\frac{\exp X^T\beta}{1+\exp X^T\beta}\right)^{y_i} \left(\frac{1}{1+\exp X^T\beta}\right)^{1-y_i} = \frac{\{\exp(x_i^T\beta)\}^{y_i}}{1+\exp(x_i^T\beta)} \\ &\propto \exp(\kappa_i x_i^T\beta) \int_0^{\infty} \exp\left(-\omega_i \frac{(x_i^T\beta)^2}{2}\right) p(\omega_i|n_i, 0) d\omega_i, \end{aligned} \tag{22}$$

where $\kappa_i = y_i - n_i/2$, the value of y_i can be either 0 or 1, and $n_i = 1$. Furthermore, $p(\omega_i|n_i, 0)$ is the density of a Pólya-Gamma random variable with parameters $(n_i, 0)$. By combining the terms from all n data points, we obtain the following expression for the conditional posterior of β, given $\omega = (\omega_1, ..., \omega_n)$:

$$p(\beta|\omega, y) \propto p(\beta) \prod_{i=1}^{N} L_i(\beta|\omega_i) = p(\beta) \prod_{i=1}^{N} \exp\left(\kappa_i x_i^T\beta - \omega_i \frac{(x_i^T\beta)^2}{2}\right). \tag{23}$$

This simplifies to a Gaussian form(Polson et al. [12]), which is amenable to direct sampling:

$$p(\beta|\omega, y) \propto p(\beta) \exp\left(-\frac{1}{2}(z - X\beta)^T \Omega (z - X\beta)\right), \tag{24}$$

where $z = (\kappa_1/\omega_1, ..., \kappa_n/\omega_n)$, and $\Omega = \text{diag}(\omega_1, ..., \omega_n)$. This yields a conditionally Gaussian likelihood in β, with working responses z, design matrix X, and diagonal covariance matrix

Ω^{-1}. Since the prior $p(\beta)$ is Gaussian, a simple linear-model calculation leads to the Gibbs sampler defined above.

Writing the posterior distribution in the form of a Gaussian, as shown in *Equation 24*, is particularly sampling-friendly because the Gaussian distribution is one of the most well-understood and widely used probability distributions. Its linear and symmetric properties enable efficient and accurate sampling using standard statistical software packages. This is crucial as it allows us to avoid more computationally intensive and less efficient numerical methods. Moreover, the mathematical properties of the Gaussian distribution, especially its conjugacy, ensure that the posterior has a closed-form solution, greatly simplifying the sampling process. Each sampling step can be explicitly expressed as a mathematical formula without the need for approximations. This simplification leads to significant improvements in computational speed and convergence, facilitating efficient Bayesian inference in complex multi-parameter spaces.

5.1.4 Sampling from the Posterior

To iterate the Gibbs sampler, one samples from the PG distribution for ω_i given β and then from the Gaussian posterior for β given ω_i and y. This procedure creates a Markov chain that converges to the true posterior distribution. The sampler's simplicity and efficiency make it suitable for a wide range of Bayesian applications, particularly where Gaussian priors are employed due to their conjugacy with the Gaussian likelihood, allowing closed-form updates for the posterior distributions.

In summary, the Pólya-Gamma method provides a computationally efficient and theoretically sound approach to Bayesian logistic regression, overcoming the challenges posed by the traditional non-conjugate models. The method's ability to convert the binomial likelihood into a Gaussian form by integrating out the PG variables justifies its preference, particularly when the Gaussian prior is in place.

5.1.5 Pseudocode for Gibbs Sampling with Normal Priors

The following presents the pseudocode for the PG augmented Gibbs Sampling with Gaussian priors.

Algorithm 2 PG augmented Gibbs Sampling with Gaussian priors

Initialize $\beta^{(0)}$ to MLE
for $t = 1, 2, \ldots, T$ **do**
    for $i = 1, 2, \ldots, N$ **do**
        Generate $\omega_i^{(t)} \sim PG(n_i, x_i^T \beta^{(t-1)})$
    end for
    Compute $\Omega^{(t)}$ as a diagonal matrix with entries $\omega_i^{(t)}$
    Compute $V(\omega) = (X^T \Omega^{(t)} X + B^{-1})^{-1}$
    Compute $m(\omega) = V(\omega)(X^T k + B^{-1} b)$
    Generate $\beta^{(t)} \sim \mathcal{N}(m(\omega), V(\omega))$
end for
return $\{\beta^{(1)}, \ldots, \beta^{(T)}\}$

In this pseudocode:

- $\beta^{(0)}$ is the initial estimate for the regression coefficients (Setting MLE in this project).
- T is the total number of iterations.
- N is the number of observations.
- n_i is the number of trials (set to 1 if each observation corresponds to a single trial).
- X^T is the transpose of the design matrix X.
- $\omega_i^{(t)}$ are the Pólya-Gamma random variables generated at iteration t.
- $\Omega^{(t)}$ is the diagonal matrix of $\omega_i^{(t)}$ values.
- $V(\omega)$ and $m(\omega)$ are the posterior variance and mean for the regression coefficients.
- $\beta^{(t)}$ is the sample of the regression coefficients at iteration t.
- PG, $\mathcal{N}$ indicate Pólya-Gamma and normal distributions, respectively.
- k is the vector of $Y - \frac{1}{2}$.
- B and b are the prior covariance and mean of β.

5.2 Gibbs Sampling with Heavy-Tailed Priors via PG Augmentation

5.2.1 Review of Existing Literature

The 2018 paper by Ghosh et al. [10] primarily aimed to address the challenges posed by the separation problem in logistic regression. Separation occurs when a linear combination of predictors can perfectly classify observations, leading to infinite maximum likelihood estimates for regression coefficients. The heavy-tailed nature of the Cauchy distribution, as a prior, contributes to more robust posterior estimation, especially in scenarios with extreme values or separation. This methodology effectively stabilizes the model estimation under challenging conditions.

In my thesis, while acknowledging the contributions of 2018 paper by Ghosh et al. [10], I diverge in my research focus. Rather than addressing the separation problem and its associated instability in logistic regression models, my work centers on exploring the efficacy of different combinations of priors and MCMC algorithms. Specifically, I draw upon the approach mentioned in Ghosh et al. [10], which is the use of Gibbs sampling with PG data augmentation and employing heavy-tailed distributions such as Cauchy and Student-t as priors.

The primary objective of my thesis is to compare and contrast these methodological variations in Bayesian logistic regression. This involves examining different choices of priors, posterior sampling algorithms, and the strengths and weaknesses of the models, offering insights into their relative performances rather than delving into the problem of model instability due to separation.

Polson et al. [12] in their 2013 research have found Gibbs sampling algorithms for logistic regression using latent variables. Polson and colleagues demonstrated that the likelihood function for logistic regression could be expressed as a mixture of normals by incorporating Pólya-Gamma distributed latent variables, thus facilitating Gibbs sampling. Building on this result, Choi and Hobert [6] developed a uniformly ergodic Gibbs sampler in their 2013 work. Further advancements were made by Ghosh et al. [10] in 2018, who complemented the methodology with a Student-t prior distribution. They extended the approach by integrating independent Student-t priors, thus evolving a Gibbs sampler specifically designed for logistic regression models.This signifies that their work proposed a theoretical method using latent variables to simplify posterior sampling for logistic regression models, paired with Student-t priors. The following theoretical part of this section is based on the literature by Ghosh et al. [10] from 2018.

5.2.2 Theoretical Basis and Sampling Steps

We define a random variable U as the weighted sum of an infinite sequence of exponential random variables W_l with rate parameter 1, expressed as

$$U = \left(\frac{2}{\pi^2}\right) \sum_{l=1}^{\infty} \frac{W_l}{(2l-1)^2}, \tag{25}$$

where each W_l is independently and identically distributed.

The density function of U, denoted as $h(u)$, is an alternating series given by

$$h(u) = \sum_{l=0}^{\infty} (-1)^l \frac{(2l+1)}{\sqrt{2\pi u^3}} \exp\left(-\frac{(2l+1)^2}{8u}\right), \quad 0 < u < \infty. \tag{26}$$

Utilizing this function, we construct the Pólya-Gamma distribution for a non-negative parameter k through an exponential tilting of $h(u)$, resulting in the density

$$p(u;k) = \cosh\left(\frac{k}{2}\right)\exp\left(-\frac{k^2}{2}u\right)h(u), \quad 0 < u < \infty, \tag{27}$$

such that a random variable with this density follows a $PG(1,k)$ distribution.

For modeling purposes, we consider the Student-t distribution with ν degrees of freedom, location parameter 0, and scale parameter σ_j. It is known that the Student-t distribution can be represented as an inverse-gamma (IG) scale mixture of normal distributions. This relationship is denoted as

$$\beta_j|\gamma_j \sim \mathcal{N}(0,\gamma_j), \quad \gamma_j \sim IG\left(\frac{\nu}{2}, \frac{\nu\sigma_j^2}{2}\right). \tag{28}$$

For the Bayesian logistic regression model, we assume β and $\Gamma = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_p)$. The dependent random vectors $(y_1, z_1), (y_2, z_2), \ldots, (y_n, z_n)$ are such that each y_i follows a Bernoulli distribution with success probability $\exp(\mathbf{x}_i^T\beta)/(1+\exp(\mathbf{x}_i^T\beta))$ and each z_i is independently drawn from a $PG(1, |\mathbf{x}_i^T\beta|)$ distribution. The augmented posterior density is $p(\beta, \Gamma, Z_D|y)$, where $Z_D = \text{diag}(z_1, z_2, \ldots, z_n)$.

To sample from this posterior, we iteratively cycle through the following Gibbs sampling steps:

1. Update β from the conditional distribution

$$\beta|\Gamma, Z_D, y \sim \mathcal{N}\left((\mathbf{X}^T Z_D \mathbf{X} + \Gamma^{-1})^{-1}\mathbf{X}^T\tilde{\mathbf{y}}, (\mathbf{X}^T Z_D \mathbf{X} + \Gamma^{-1})^{-1}\right), \tag{29}$$

   where $\tilde{y_i} = y_i - \frac{1}{2}$ and $\tilde{\mathbf{y}} = (\tilde{y_1}, \tilde{y_2}, ..., \tilde{y_n})^T$.

2. Update each γ_j independently from the conditional distribution

$$\gamma_j|\beta, Z_D, y \sim IG\left(\frac{\nu+1}{2}, \frac{\beta_j^2 + \nu\sigma_j^2}{2}\right), \tag{30}$$

   for $j = 1, 2, \ldots, p$.

3. Update each z_i independently from the conditional distribution

$$z_i|\beta, \Gamma, y \sim PG(1, |X_i^T\beta|), \tag{31}$$

   where $i = 1, 2, \ldots, n$, X_i represents the covariate vector for the i-th observation, and z_i realizes the data augmentation for the logistic regression model.

Steps 1 (*Equation 29*) and 3 (*Equation 31*) of the Gibbs sampler utilize the properties of the normal and Pólya-Gamma distributions, respectively, for conditional sampling. Step 2 (*Equation 30*) leverages the inverse-Gamma distribution as the conditional distribution for the precision parameters γ_j, aligning with the Student-t representation of the regression coefficients β. This sequence of updates forms a Markov chain that converges to the target joint posterior distribution $p(\beta, \Gamma, Z_D|y)$ and generates a posterior sampling chain for the parameter β, allowing for Bayesian inference of the model parameters.

5.2.3 Pseudocode for Gibbs Sampling with Student's t and Cauchy Priors

The following shows the pseudocode for the PG Data Augmentation Gibbs Sampling algorithm with Student's t and Cauchy priors.

Algorithm 3 PG augmented Gibbs Sampling with Student's t and Cauchy priors

Initialize $\beta^{(0)}$ to MLE
Set hyperparameters $v = 1, 7$, $\sigma = \text{rep}(1, 6)$, $T = 20000$
for $t = 1, 2, \ldots, T$ **do**
    Compute $\tilde{y} = Y - \frac{1}{2}$
    Initialize γ vector for scale parameters
    for $j = 1, 2, \ldots, D$ **do**
        $\gamma[j] \leftarrow$ draw from Inv-Gamma $\left(\frac{v+1}{2}, \frac{\beta_i[j]^2 + v\sigma[j]^2}{2}\right)$
    end for
    Construct diagonal matrix Γ with γ
    $X_\beta \leftarrow X\beta^{(t-1)}$
    Initialize vector z for auxiliary variables
    for $i = 1, 2, \ldots, N$ **do**
        $z[i] \leftarrow$ draw from PG distribution using $\text{rpg}(n_i, X_\beta)$
    end for
    Construct diagonal matrix Z with z
    $\beta_{\text{mean}} \leftarrow \left(X^T Z X + \Gamma^{-1}\right)^{-1} X^T \tilde{y}$
    $\beta_{\text{cov}} \leftarrow \left(X^T Z X + \Gamma^{-1}\right)^{-1}$
    $\beta^{(t)} \leftarrow$ draw from $\mathcal{N}(\beta_{\text{mean}}, \beta_{\text{cov}})$
    Store $\beta^{(t)}$ in β_{chain}
end for
return β_{chain}

In this pseudocode:

- β_i represents the current estimate for the regression coefficients at iteration i.
- D is the number of covariates (or predictors) in the matrix X.
- N is the number of observations in the dataset.
- T is the total number of iterations.
- n_i is set to 1, indicating that each observation corresponds to a single Bernoulli trial.
- $\tilde{y}$ is the adjusted response vector, $Y - \frac{1}{2}$.
- γ is a vector of scale parameters drawn from the inverse-gamma distribution.
- Γ is a diagonal matrix with elements of γ on its diagonal.
- X_β is the matrix X multiplied by the regression coefficients vector β_i.
- z_i is a vector of augmentation variables from the Pólya-Gamma distribution.
- Z is a diagonal matrix with elements of z_i on its diagonal.
- v is the degrees of freedom for the Student's t prior distribution. In our experiments, setting $v = 1$ corresponds to a Cauchy prior, while $v = 7$ indicates a Student's t-prior.
- σ is a vector where each element represents the scale parameter for the prior distribution of the corresponding regression coefficient's variance in the Student's t model.
- β_{mean} is the mean of the posterior distribution for the regression coefficients.
- β_{cov} is the covariance of the posterior distribution for the regression coefficients.
- β_{chain} stores the sampled regression coefficients across iterations.

6 Diagnostic Tools for Convergence Check and Model Selection

This chapter will introduce several commonly used diagnostic tools for assessing the convergence and stability of MCMC algorithms, as well as criteria for model selection. This includes some relatively new diagnostic methods, such as the improved Split-$\hat{R}$ and rank plot introduced by Vehtari et al. [13] in 2021. These methods and tools will be used in the simulated experiments of this thesis. Through these means, we can evaluate the differences in chain convergence, chain mixing, and the effectiveness of sample information exhibited by the MCMC algorithms when combined with different priors, and the respective advantages and disadvantages of the models.

6.1 Trace Plot

A Trace Plot, also known as a trajectory plot, is a graphical tool used to assess the convergence of parameters in MCMC methods. It illustrates the variation trend of model parameters during the MCMC sampling process by plotting the iteration values of the parameters.

In MCMC processes, for each parameter, the Trace Plot plots all its iteration values. A key feature of such a plot is its ability to visually show the variation of parameter values over iterations, helping to identify whether the parameter has reached a steady distribution. If a parameter's trace plot shows a stable, trendless "cloudy" distribution, it generally indicates that the parameter has converged. The content of this subsection is inspired by and organized from the ideas found in Gelman et al. [7].

6.1.1 The Burn-in Phase

Understanding the burn-in phase is very important for interpreting Trace Plots. The initial part of the Trace Plot, known as the burn-in phase, represents the period where MCMC samples typically do not reflect the target distribution due to the influence of initial conditions. In the Trace Plot, the burn-in phase usually appears as an unstable region at the beginning of the chart, where parameter values may fluctuate significantly or show certain trends. When these fluctuations gradually decrease and the parameter values begin to stabilize and oscillate around a certain constant or region, it usually indicates that the burn-in phase has ended and the sampling chain is entering a state of convergence. Before conducting further statistical analysis, we typically discard the samples from this phase.

6.1.2 Evaluating Convergence Using Trace Plots

Trace Plots are primarily used to check for convergence in the following two aspects:

- **Stability**: The trace plot should exhibit a stable distribution with no significant trends, indicating that the chain has reached equilibrium.
- **Mixing**: Trace plots of multiple chains should overlap and interweave with each other, showing good mixing, which indicates consistency between different chains.

While Trace Plot is one of the key tools for understanding and diagnosing MCMC models, they also have limitations. It is difficult to accurately judge the convergence of complex models based solely on trace plots. Therefore, it is generally recommended to use them in

conjunction with other diagnostic tools (such as Gelman-Rubin diagnostics, etc.) for a more comprehensive assessment of convergence.

6.2 Potential Scale Reduction Factor (PSRF) also known as $\hat{R}$

Gelman and Rubin [8] in their work introduced the Potential Scale Reduction Factor, which is abbreviated as PSRF and is also known as $\hat{R}$, as a diagnostic for assessing the convergence of MCMC method. The method involves running $m > 1$ parallel chains and is predicated on the convergence of these chains, which is achieved when the chains are indistinguishable from one another. The diagnostic utilizes the within-chain and between-chain variances, denoted by W and B, respectively. The PSRF is then estimated by the formula:

$$\hat{R} = \sqrt{\frac{d+3}{d+1} \cdot \frac{\hat{V}}{W}}, \tag{32}$$

where $\hat{V}$ is a pooled estimate of variance, considering both within-chain and between-chain variances and d represents degrees of freedom estimated by the method of moments:

$$d = \frac{2\hat{V}^2}{\text{Var}(\hat{V})}, \tag{33}$$

where $\hat{V}$ is the variance estimate mentioned above and $\text{Var}(\hat{V})$ is its variance.

When the Potential Scale Reduction Factor (PSRF) is close to 1, more precisely, according to Gelman and Rubin [8], when PSRF < 1.1, it indicates that all MCMC chains have converged to the target posterior distribution, implying a consistent estimation of the model parameters across the chains. On the other hand, a PSRF greater than 1.1 suggests that there is a substantial difference between the chains, implying that they may not have converged to the same distribution or that convergence is occurring slowly, and more iterations may be required to ensure convergence. In the experiment of this paper, I utilized the `gelman.diag` function from the `coda` R package to calculate the PSRF values for the posterior samples. For more information about PSRF, refer to the R documentation [2].

6.3 Improved Split-$\hat{R}$ Diagnostic

In the previous section, we introduced the $\hat{R}$ (PSRF) diagnostic method and its standards, originally proposed by Gelman and Rubin in 1992. However, in their 2021 study, Vehtari et al. [13] indicated that PSRF has serious flaws. Specifically, traditional $\hat{R}$ fails to correctly diagnose convergence failures when the chain exhibits heavy tails or when the variance varies across chains. Consequently, Vehtari et al. proposed an improved diagnostic method known as Split-$\hat{R}$.

In this study, we adopt the improved Split-$\hat{R}$ diagnostic introduced by Vehtari et al. [13] in 2021 and reimplemented in R. The Split-$\hat{R}$ diagnostic is particularly adept at identifying non-stationarity and inadequate mixing within chains, which are critical indicators of convergence.

The Split–$\hat{R}$ diagnostic functions by partitioning each Markov chain into two halves. It then assesses the convergence of each segment independently, evaluating the variances both within and between these partitions of the chains. Specifically, the between-chain variance, denoted as B, and the within-chain variance, denoted as W, are calculated as follows:

$$B = \frac{N}{M-1}\sum_{m=1}^{M}\left(\bar{\theta}^{(.m)} - \bar{\theta}^{(..)}\right)^2, \text{ where } \bar{\theta}^{(.m)} = \frac{1}{N}\sum_{n=1}^{N}\theta^{(nm)}, \bar{\theta}^{(..)} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}^{(.m)}; \quad (34)$$

$$W = \frac{1}{M}\sum_{m=1}^{M}s_m^2, \quad \text{where} \quad s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}\left(\theta^{(nm)} - \bar{\theta}^{(.m)}\right)^2. \quad (35)$$

In the equations above, N is the number of draws per chain, M is the number of chains, $S = MN$ is the total number of draws from all chains, $\theta_{(nm)}$ is the n-th draw of the m-th chain, $\theta_{(.m)}$ is the average of draws from the m-th chain, and $\theta_{(..)}$ is the average of all draws.

The improved diagnostic combines a weighted average of within-chain variance (W) and between-chain variance (B), providing an estimate for the variance of the quantity being estimated in the posterior distribution. The refinement lies in the Split-$\hat{R}$'s ability to account for an initial overestimation of variance in the initial chain distributions, achieved through the following formulation:

$$\hat{Var}^{+}(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B. \quad (36)$$

In the end, the diagnostic process finalizes with the computation of the Split-$\hat{R}$ value:

$$\hat{R} = \sqrt{\frac{\hat{Var}^{+}(\theta|y)}{W}}. \quad (37)$$

This value, for an ergodic process, approaches 1 as N goes to infinity, signaling convergence. The document (Vehtari et al. [13]) suggests a more stringent threshold for the improved convergence diagnostic $\hat{R}$, proposing improved Split-$\hat{R} < 1.01$ as opposed to the earlier recommendation of $\hat{R} < 1.1$. This tighter threshold reflects more rigorous standards is supported by the results detailed in an appendix of the paper (Vehtari et al. [13]). The updated criterion is part of an effort to address the limitations of the traditional $\hat{R}$ in identifying convergence, especially in the presence of heavy-tailed distributions or variable variances across chains. This adaptation also ensures that the Split-$\hat{R}$ statistic remains robust against the initial sampling biases, which could lead to an underestimation of variability, providing a more reliable measure for the convergence of MCMC samplings.

Because both PSRF ($\hat{R}$) and the improved Split-$\hat{R}$ reflect the convergence and mixing of the posterior sampling chains, we will use both of these diagnostic tools in the upcoming *Results* section and compare their diagnostic outcomes.

6.4 Effective Sample Size (ESS)

A key challenge in MCMC sampling is the autocorrelation in the generated samples, which reduces the effective information each sample carries about the posterior distribution. This leads to the necessity of using the Effective Sample Size (ESS) as a metric (see R documentation [1]).

The ESS quantifies the number of independent-like samples within the correlated MCMC chain. It is critical for assessing the efficiency and reliability of the MCMC algorithm. Higher autocorrelation results in a lower ESS, suggesting that a greater number of iterations may be required for reliable estimates. Conversely, a high ESS value is the desired outcome,

as it implies that the sampling within the chain has lower autocorrelation, better chain mixing, and better convergence performance. In the experiments of this paper, I used the `effectiveSize` function from the `coda` R library to calculate the effective sample size. For more information, please refer to the relevant R documentation [1].

6.5 Rank Plot

In our statistical analysis practice, to effectively assess the convergence of multiple Markov chains to the same posterior distribution, we have adopted rank plots as an innovative graphical diagnostic tool (see Vehtari et al. [13]) to complement or even replace traditional trace plots. Rank plots visualize the relative positioning of each chain within the overall posterior distribution by ranking the posterior sample values across all chains and independently plotting histograms for each chain. Uniform distributions in the rank plots across all chains indicate that the chains are sampling from the same posterior distribution without any chain-specific biases. Similarity and absence of significant skewness in the rank distributions across chains usually signify a good chain mixing.

The advantage of rank plots is particularly evident in the assessment of inter-chain mixing. Unlike trace plots, which can become visually cluttered with long chains, rank plots provide clearer visual feedback on the quality of mixing. Trace plots may become difficult to interpret due to the overlay of multiple chains, whereas rank plots, by displaying the distribution of ranks for each parameter value, allow for direct observation of the uniformity of mixing. Thus, rank plots emerge as a powerful tool for evaluating the quality of mixing and convergence across chains, especially when dealing with complex models with multiple parameters and extended chains. In this paper, I reimplemented the section on rank plots from Vehtari et al. [13] using R code.

6.6 Ranking Plot for WAIC

The Ranking Plot is a crucial visualization tool used in the evaluation of model selection criteria, specifically focusing on WAIC (Watanabe-Akaike Information Criterion, hereinafter referred to as WAIC) or DIC (Deviance Information Criterion). This section delves into the theoretical underpinnings and practical applications of WAIC criteria (see Watanabe [14]), as well as how it is represented in Ranking Plots. In the experiment, I used the `WAIC` function from the `LaplacesDemon` package to calculate the WAIC values for each model. For detailed information, please refer to the R documentation on the WAIC function [3].

6.6.1 Watanabe-Akaike Information Criterion (WAIC)

WAIC is a fully Bayesian criterion for model selection, which estimates the out-of-sample prediction error. It is defined as:

$$\text{WAIC} = -2(\text{LPPD} - \text{pWAIC}), \tag{38}$$

where LPPD is the log pointwise predictive density for the model, and pWAIC represents the effective number of parameters, serving as a penalty term to avoid overfitting. WAIC is particularly useful for comparing models with different numbers of parameters or different structures.

When the WAIC is low, it indicates better model fitting, improved predictive performance, and lower model complexity. Conversely, a higher WAIC suggests poor model

fitting, potential overfitting, and subpar predictive performance. We can say that models with lower WAIC values are our better choices. For this reason, WAIC serves as a criterion for Bayesian model selection.

6.6.2 Ranking Plot

Ranking Plots are used to visually compare models using WAIC. Each model is plotted with its criterion value (WAIC), allowing for an easy comparison across models. Ranking Plots are particularly useful in scenarios where several models are compared, as they offer a clear, visual representation of which models perform better according to the WAIC criterion. The application of Ranking Plots in model selection is powerful as it provides a straightforward method to assess the trade-off between model complexity and fit.

7 Setup of the Simulation

This chapter will introduce the specific details of the simulated dataset used in this thesis.

7.1 Predictor Variables

The simulated dataset in this thesis comprises five predictor variables, denoted as x_1, x_2, x_3, x_4, x_5, along with an intercept term (a column of ones). These correspond to six parameters: $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$. The predictor variables are not independent; instead, correlation among them is introduced by setting different correlation coefficients, with $\rho = 0.1, 0.3, 0.6, 0.9$.

Our specific setup involves predictor variables following a multivariate normal distribution, with means set to $(-3, 2, 0, 1, -1)^T$, and the covariance matrix defined as $(1-\rho) \times I + \rho \times J$, where I is the identity matrix and J is an all-ones 5×5 matrix.

The reason for simulating the predictor variables X in this manner is that, within the context of Bayesian logistic regression, it is a common practice to simulate predictor variables from a multivariate normal distribution with predefined means and covariance matrices. The choice of means and covariance structure is arbitrary in this context, and by doing so, we can simplify and effectively control the generation of the simulated dataset (X, Y) for the Bayesian logistic regression model.

Additionally, we introduce correlation among the predictor variables for two main reasons. Firstly, we aim to better replicate real-world data, as correlations are commonly observed in real datasets. Secondly, we aim to study whether the utilization of different prior distributions in scenarios with varying levels of correlation leads to significant differences in the convergence performance of distinct MCMC algorithms.

7.2 The Sample Size of the Simulated Dataset

In this experiment, the sample size is set to 100 because we aim to avoid the implications of the *Bernstein-von Mises theorem*, and we wish to observe the effect of the chosen prior. Therefore, our data sample size is not very large.

The Bernstein-von Mises theorem is an important result in Bayesian statistics. It essentially states that, under certain regularity conditions, the posterior distribution of a parameter converges to a normal distribution as the sample size increases, regardless of the form of the prior distribution. This convergence is centered around the MLE, with the variance of the normal distribution inversely proportional to the sample size. In simpler terms, the theorem implies that for large sample sizes, Bayesian inference (using the posterior distribution) and frequentist inference (relying on point estimates like the maximum likelihood estimator) will give similar results. This is because the posterior distribution becomes more concentrated around the maximum likelihood estimate as the sample size grows. However, for smaller sample sizes, the choice of the prior can have a significant impact on the results, and the posterior distribution may not approximate a normal distribution. In such cases, the specific characteristics of the prior distribution play a more prominent role in Bayesian analysis, which is also the core objective of this paper, to check effect of priors in different MCMC algorithms.

7.3 Iterations

In this thesis, we set the number of iterations for the MCMC algorithm to 20,000 based on the following considerations:

1. **Convergence Assurance**: MCMC methods are iterative and can take a significant number of steps to converge to the target distribution, especially in complex models or high-dimensional spaces. A higher number of iterations increases the likelihood that the chain has reached convergence.

2. **Accuracy and Stability**: More iterations can lead to more accurate and stable estimates of the parameters. It allows for a better exploration of the parameter space, ensuring that less frequent regions of high posterior probability are visited.

3. **Burn-in Period**: A portion of the initial iterations (often called the 'burn-in' period) may be discarded because the chain might not have reached its stationary distribution yet. A large total number of iterations ensures that there is still a substantial amount of data left for analysis after removing the burn-in.

4. **Actual Testing Performance**: In addition, I conducted multiple iteration tests on the simulated dataset. The test results indicated that setting the iteration count to 20,000 is a relatively ideal choice. This is because at this point, Gibbs Sampling and MH random walk exhibit significant differences in terms of convergence, which aligns with the objectives of our paper. Additionally, the choice of iteration count also takes into consideration the trade-off with computer computational power.

Furthermore, it is worth mentioning that we have implemented a thinning strategy (selecting every 4th sample) when using the MH random walk algorithm. This is done to reduce the autocorrelation in the posterior sampling results. Additionally, our sampling begins from the 1001st iteration, as we consider the first 1000 iterations to be the burn-in phase. This approach is adopted because the MH random walk algorithm exhibits more pronounced autocorrelation and has a slower convergence rate in MCMC. Therefore, for the MH random walk algorithm, the actual number of iterations amounts to 81,000.

7.4 Starting Value for the MCMC Chain

In this thesis, we set the initial value for the MCMC sampling to the MLE. This is because we aim to accelerate the convergence phase of the sampling process. Despite our limited sample size, which might impede the MLE from being a perfect estimate of the true β values, this approach is still considered to be a more rational choice for the initial value compared to an all-zero initialization.

7.5 The Number of Chains

In the simulation, we sampled the posterior distribution of each model parameter by generating 4 independent MCMC chains. The primary reason for choosing multiple chains is to reduce the risk of bias arising from a single chain. A single chain might be influenced by random fluctuations and chance, potentially failing to fully represent the entire posterior distribution. By independently generating multiple chains, we can more effectively test and ensure the convergence of the model. This means if all chains converge to the same distribution, demonstrating good mixing and consistent convergence, we can be more confident that our results closely approximate the true posterior distribution.

7.6 Prior Assumptions

The following introduces the settings of the prior distributions involved in this paper. Since the relevant literature does not provide explicit recommendations for the mean and covariance matrix of the normal prior, no matter for Gibbs sampling or for MH random walk, this study adopts standard settings based on conventional practice. Meanwhile, specific settings for the location and scale parameters of the Student's t and Cauchy priors for Gibbs sampling are based on recommendations from Ghosh et al. [10]. Additionally, to ensure the reliability of the simulated experimental results, the same prior assumptions are applied to both the PG-augmented Gibbs Sampling algorithm and the Metropolis-Hastings random walk algorithm (hereinafter referred to as "the two algorithms") in comparative experiments. The prior settings for the two algorithms are as follows:

1. **Normal Prior Assumption**
    - **Mean vector**: Set to $(0, 0, 0, 0, 0, 0)^T$. This implies that a prior mean of 0 is specified for each parameter β_i $(i = 0, 1, 2, \ldots, 6)$.
    - **Covariance matrix**: Set as a 6×6 diagonal matrix with diagonal elements of 1. This indicates that each parameter β_i $(i = 0, 1, 2, \ldots, 6)$ has a variance of 1, and there is no prior correlation between variables.
2. **Student's t and Cauchy Prior Assumption**
    - **Location parameter vector**: Set uniformly to $(0, 0, 0, 0, 0, 0)^T$. This reflects the central location of both the Student's t and Cauchy priors set at 0, consistent with the settings in Ghosh et al. [10].
    - **Scale parameter vector**: Set to $(10, 2.5, 2.5, 2.5, 2.5, 2.5)^T$. This indicates that the scale parameter for the intercept β_0 is 10, while the scale parameters for the remaining coefficients β_1 to β_5 are all 2.5 and they are independent. A larger scale parameter means more data dispersion and thicker tails. Here, the distribution of intercept β_0 has a wider tail than β_1 to β_5. The scale parameter settings also align with recommendations from the 2018 literature by Ghosh et al. [10].

7.7 The Proposal Distribution

The proposal distribution in the MH random walk algorithm is a multivariate normal distribution. It takes the sampling value from the previous step as the mean vector and the identity matrix as the covariance matrix. This is a common and well-established setting in practice and for our parameter configurations. Our focus was on comparing convergence performance between MH and Gibbs sampling, as well as examining the impact of factors such as prior distributions. Given the scope of our research, the use of a conventional proposal distribution was a pragmatic choice that allowed us to focus on these key aspects of the analysis.

7.8 The True Values of β

To simplify, I employed `runif(6, -3, 3)`, which means randomly drawing 6 real numbers as the true values for β, each with equal probability, within the range from -3 to 3.

8 Results

In this chapter, I will present the results of simulated experiments conducted on the simulated dataset, as introduced in *Chapter 7*. This research involves an in-depth diagnostic analysis of the results using Bayesian logistic regression models and various MCMC algorithms. The main diagnostic and comparative aspects include:

1. **Convergence and Mixing of Chains**: Trace plots are used to analyze the convergence of the chains, and rank plots are introduced for more effective verification of chain mixing. Additionally, I employ both the improved Split-$\hat{R}$ and the traditional PSRF to assess multi-chain mixing.
2. **Auto-correlation and Effectiveness of Samples**: The ESS (Effective Sample Size) is utilized to determine the effectiveness of samples and their impact on chain convergence.
3. **Model Evaluation and Selection**: A ranking plot based on the WAIC is used to assess the fittning and predictive performance of the models, and to discuss issues related to model selection.

Before delving into the experimental results, 3 key points need to be clarified:

1. The MCMC algorithms used in the simulated experiments are PG-augmented Gibbs sampling and MH random walk. In the discussions that follow, unless specifically stated otherwise, all references to Gibbs sampling refer to PG-augmented Gibbs sampling, and all mentions of the MH algorithm refer to the MH random walk.
2. The model includes six parameters, including the intercept parameter, ranging from β_0 to β_5. While part of our discussion in this chapter primarily focuses on the single parameter β_1, it should be noted that the results for the other parameters also exhibited outcomes similar to β_1, allowing us to draw a set of broad conclusions that are applicable to these parameters as well.
3. In the subsequent paragraphs and tables, the notation `MCMC + prior` is used to denote a model that incorporates a specific MCMC method combined with a prior assumption. For instance, `MH+t` refers to a model that employs the Metropolis-Hastings random walk with a Student's t-distribution as its prior.

8.1 Verification of Posterior Distribution Integrals

Before performing convergence diagnostics, I first calculated the integrals of the chains from the posterior sampling results in the simulated experiments. This step is crucial for validating the effectiveness of the MCMC samplings, as in Bayesian statistics, the integral of the posterior distribution must be 1 to ensure it is a valid probability density function (PDF).

According to our results (*Table 1*), for all models under different values of ρ, the integrals of the posterior distributions are close to 1, suggesting that the MCMC sampling might have converged to the true posterior distribution. The consistent results across different ρ settings strengthen the credibility of the model's posterior distributions. However, the integral being 1 is only a necessary condition and does not fully confirm the accuracy of the sampling. Further diagnostics are necessary to validate the accuracy and convergence of the MCMC sampling.

ρ	Method	β_0	β_1	β_2	β_3	β_4	β_5
0.1	Gibbs+normal	1.000978	1.000978	1.000978	1.000978	1.000978	1.000981
	Gibbs+t	1.000978	1.000978	1.000978	1.000978	1.000979	1.000980
	Gibbs+Cauchy	1.000978	1.000978	1.000978	1.000978	1.000978	1.000982
	MH+normal	1.000939	1.000977	1.000975	1.000977	1.000975	1.000978
	MH+t	1.000969	1.000978	1.000978	1.000976	1.000978	1.000975
	MH+Cauchy	1.000970	1.000975	1.000976	1.000977	1.000977	1.000974
0.3	Gibbs+normal	1.000978	1.000978	1.000978	1.000978	1.000980	1.000983
	Gibbs+t	1.000978	1.000978	1.000978	1.000978	1.000979	1.000982
	Gibbs+Cauchy	1.000978	1.000978	1.000978	1.000978	1.000980	1.000975
	MH+normal	1.000976	1.000976	1.000977	1.000978	1.000978	1.000977
	MH+t	1.000978	1.000978	1.000977	1.000978	1.000978	1.000978
	MH+Cauchy	1.000978	1.000978	1.000978	1.000978	1.000977	1.000978
0.6	Gibbs+normal	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	Gibbs+t	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	Gibbs+Cauchy	1.000978	1.000980	1.000978	1.000978	1.000978	1.000978
	MH+normal	1.000977	1.000978	1.000978	1.000977	1.000978	1.000978
	MH+t	1.000978	1.000978	1.000977	1.000978	1.000976	1.000978
	MH+Cauchy	1.000978	1.000978	1.000977	1.000978	1.000978	1.000978
0.9	Gibbs+normal	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	Gibbs+t	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	Gibbs+Cauchy	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	MH+normal	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	MH+t	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978
	MH+Cauchy	1.000978	1.000978	1.000978	1.000978	1.000978	1.000978

Table 1: The posterior integral of β for different models across various ρ values

8.2 Analysis of Estimations

To ensure the accuracy of posterior sampling, this section compares the means of posterior distributions, MLE, and true parameter values (see *Table 2* and *Table 3*). Ideally, the mean of the posterior distribution should be close to the true parameter values with minimal error, reflecting the high credibility of posterior sampling.

As we have observed, for a limited sample size (n=100), the posterior means obtained through MCMC sampling methods are similar to the MLE. Additionally, the deviation of the posterior distribution's mean from the true parameter values is not significant and is acceptable.

	β_1	MLE	Posterior mean					
			Gibbs+normal	Gibbs+t	Gibbs+Cauchy	MH+normal	MH+t	MH+Cauchy
$\rho = 0.1$	1.339022	0.9054439	0.6852904	0.9626719	0.9400584	0.6844759	0.6911559	0.6862541
$\rho = 0.3$	1.339022	1.1819641	0.7615924	1.2477004	1.2153344	0.7442198	0.7749092	0.8239267
$\rho = 0.6$	1.339022	1.9241888	1.0288777	1.9737189	1.9376159	0.9996676	1.0449753	1.3151916
$\rho = 0.9$	1.339022	1.9811506	0.8459840	1.7582428	1.7104387	0.8348128	0.8496338	0.8851541

Table 2: True Values, MLE, and Posterior Means of β_1 for Various ρ Levels

	β_1	$\lvert\text{Error}_{\text{MLE}}\rvert$	$\lvert\text{Error}_{\text{Posterior Means}}\rvert$					
			Gibbs+normal	Gibbs+t	Gibbs+Cauchy	MH+normal	MH+t	MH+Cauchy
$\rho = 0.1$	1.339022	0.433578	0.653732	0.376350	0.398964	0.654546	0.647866	0.652768
$\rho = 0.3$	1.339022	0.1570579	0.577430	0.091322	0.123688	0.594802	0.564113	0.515095
$\rho = 0.6$	1.339022	0.5851668	0.310144	0.634697	0.598594	0.339354	0.294047	0.023830
$\rho = 0.9$	1.339022	0.6421286	0.493038	0.419221	0.371417	0.504209	0.489388	0.453868

Table 3: Absolute Error for MLE and Posterior Means of β_1 for Various ρ Levels

8.3 Trace plot

The provided *Figure 1* and *Figure 2* in this chapter display the convergence behavior and sampling characteristics of a single parameter β_1 for six statistical models at two different levels of correlation coefficients ($\rho = 0.1$ and $\rho = 0.9$) by using trace plot. In these two extreme cases of correlation, the three models using Gibbs sampling (`Gibbs+normal`, `Gibbs+t`, `Gibbs+Cauchy`) all show stable trace lines without any apparent burn-in period. This indicates that the Gibbs sampling method is capable of achieving rapid and effective convergence for data with both low and high correlation.

In contrast, the trace plots of the MH method at $\rho = 0.1$ exhibit a lower effective sampling rate, as evidenced by long periods where sample points remain unchanged, which indicates poor sampling efficiency.

At the higher correlation level of $\rho = 0.9$, the noticeable trend and fluctuations in the trace lines of the MH random walk related models, particularly MH with Cauchy prior, suggest that the posterior samples may be highly autocorrelated, indicating a lack of independence and effectiveness in the samples. This also implies that the chains obtained by the MH algorithm may still not have converged.

For the sake of brevity, I have omitted the trace plots for ρ values of 0.3 and 0.6 in this section. This is because the results at $\rho = 0.3$ are very similar to those at $\rho = 0.1$, and the trace plots at $\rho = 0.6$ are highly similar to those at $\rho = 0.9$. This observation suggests that at lower correlations (below 0.5), the MH-related models primarily exhibit poor sampling efficiency in their trace plots, while at higher data correlations (above 0.5), the trace plots of these models primarily demonstrate issues of high autocorrelation in sampling.

It is important to note that since plotting multiple chains' traces in the same traceplot can lead to visual confusion, the traceplots in this chapter only involve a single chain for β_1. For an check of multi-chain's mixing and consistency, I will elaborate in the next subsections.

8.4 Rank Plot

To more clearly assess the mixing and convergence consistency of the chains, I introduced the rank plot. As shown in *Figure 3*, rank plots for the four chains of β_1 in the six models at $\rho = 0.1$ were constructed, where the `Gibbs+normal`, `Gibbs+t`, and `Gibbs+Cauchy` models displayed uniformly distributed ranks across their four chains. This indicates that the combination of Gibbs sampling with these three prior distributions demonstrates good mixing and consistency.

However, the performance of models related to the MH algorithm in the rank plots was less satisfactory, which showed evident non-uniform distributions, indicating either an abundance or scarcity of certain ranks in sampling. This could suggest insufficient inter-chain mixing or that they may not have converged to the same distribution.

Additionally, I have chosen to omit the rank plot analysis for ρ values of 0.3, 0.6, and 0.9, as it has been verified that the rank plots exhibit similar outcomes to those at $\rho = 0.1$ when the correlation coefficient assumes other values.

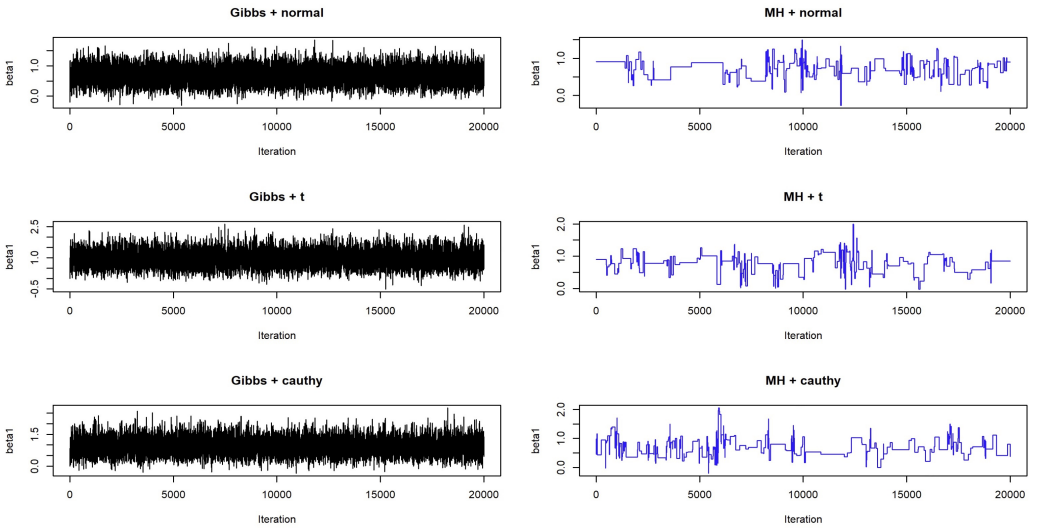

Figure 1: Trace Plot: Gibbs Sampling vs Metropolis-Hastings Randow Walk for β_1 ($\rho = 0.1$)

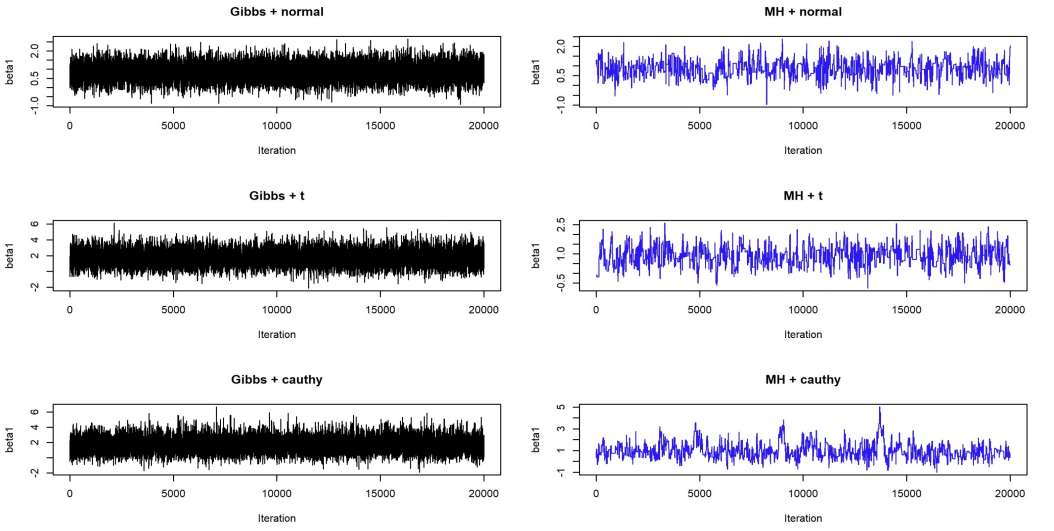

Figure 2: Trace Plot: Gibbs Sampling vs Metropolis-Hastings Randow Walk for β_1 ($\rho = 0.9$)

Figure 3: Rank plot of β_1 across 4 chains for different models at $\rho = 0.1$

8.5 Analysis of Split-$\hat{R}$ and PSRF

In this chapter, I will analyze the convergence behavior of MCMC samplings by calculating the Potential Scale Reduction Factor (PSRF) and Split-$\hat{R}$ values for different statistical models at multiple correlation coefficient (ρ) levels. Theoretically, when the Potential Scale Reduction Factor (PSRF) is below 1.1 or the Split-$\hat{R}$ is below 1.01, it is indicative that all chains generated by the MCMC algorithm for a model may have converged to the same distribution.

Let us first explore the differences among the models in terms of the Split-$\hat{R}$ index. As shown in *Figure 4*, with any prior assumption, the Gibbs sampling technique maintains a Split-$\hat{R}$ value below 1.01 from the start, significantly indicating the high convergence efficiency of the Gibbs sampling algorithm. In contrast, the MH random walk, regardless of the prior used, exhibits a relatively slower convergence rate, generally requiring 2000 to 5000 iterations to lower the $\hat{R}$ value below 1.01. Particularly, with data correlation as high as 0.9, the MH random walk combined with the Cauchy prior necessitates up to 12500 iterations to achieve convergence. Worse still, when ρ equals 0.1, the combinations of `MH+t` and `MH+Cauchy`, even after 20,000 iterations, have their Split-$\hat{R}$ values remaining slightly above the convergence threshold of 1.01. Additionally, it is noteworthy that in most cases, the fastest convergence rate with MH random walk is observed when the prior is a normal distribution. Conversely, using Cauchy or Student's t-distributions as priors tends to slow down the convergence. However, there is an exception: when the data correlation coefficient ρ is 0.3, the convergence rate of MH random walk with a normal prior is slightly slower than

with Student's t or Cauchy priors.

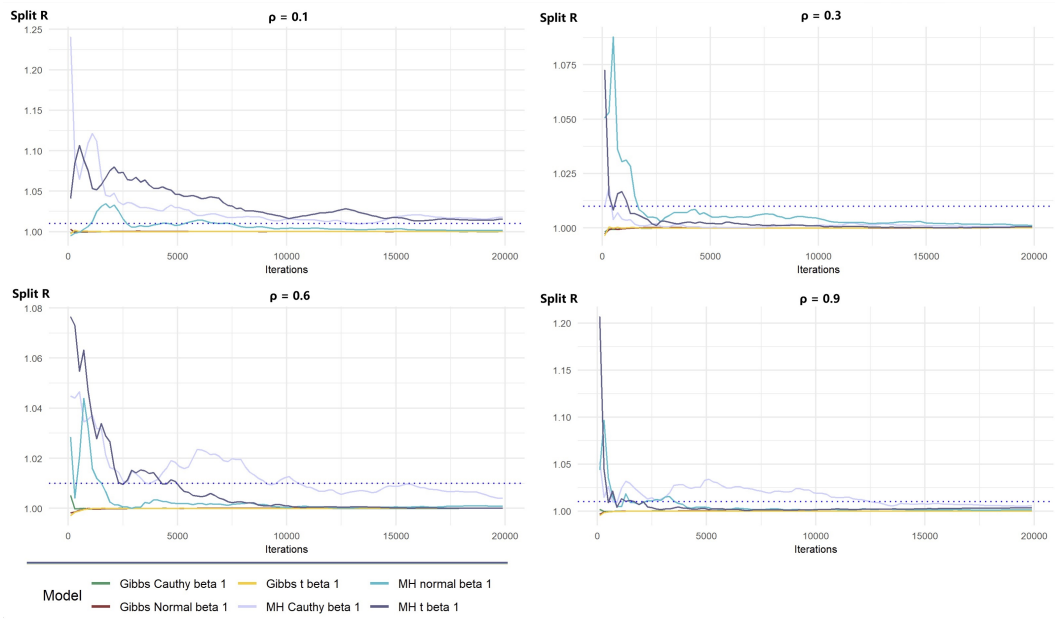

Figure 4: Split-$\hat{R}$ trends for different models of β_1 under varying correlations

We also explored the use of the PSRF by utilizing the `gelman.diag` function from the `coda` library, known as the traditional $\hat{R}$, for generating similar graphs. However, we encountered errors during the computation process, attributable to the calculation method of the `gelman.diag` function. This issue warrants further discussion, which we will reserve for the discussion section (*Chapter 9*). Consequently, I opted to employ a tabular approach to compare the Split-$\hat{R}$ and PSRF after completing 20,000 iterations.

As observed in *Table 4*, overall, both Gibbs sampling and MH models remain below the convergence threshold for PSRF and Split-$\hat{R}$ across different levels of ρ. Only two outliers are observed, namely the `MH+t` and `MH+Cauchy` models at $\rho = 0.1$, where their Split-$\hat{R}$ values exceed the threshold of 1.01.

This indicates that the conclusions derived from PSRF and Split-$\hat{R}$ are largely consistent. After completing 20,000 iterations, the sampling chains of all models essentially reached convergence. However, a divergence in conclusions between PSRF and Split-$\hat{R}$ emerged when ρ is 0.1. The PSRF indicated that `MH+t` and `MH+Cauchy` had achieved convergence (with values of 1.0798 and 1.0704, both below the PSRF threshold of 1.1), whereas Split-$\hat{R}$ suggested that these combinations were only nearing convergence (with values of 1.0329 and 1.0243), not fully reaching the stringent convergence threshold of 1.01. This subtly suggests that Split–$\hat{R}$ may be more stringent than PSRF.

ρ	Method	Gibbs+normal	Gibbs+t	Gibbs+Cauchy	MH+normal	MH+t	MH+Cauchy
0.1	PSRF	1.0001	1.0002	1.0002	1.0222	1.0798	1.0704
	Split-$\hat{R}$	1.0000	1.0000	1.0000	1.0041	1.0329	1.0243
0.3	PSRF	1.0001	1.0002	1.0005	1.0062	1.0373	1.0184
	Split-$\hat{R}$	1.0000	1.0000	1.0001	1.0006	1.0067	1.0017
0.6	PSRF	1.0004	1.0003	1.0002	1.0159	1.0034	1.0126
	Split-$\hat{R}$	1.0000	1.0000	1.0000	1.0014	1.0004	1.0056
0.9	PSRF	1.0001	1.0000	1.0000	1.0096	1.0480	1.0221
	Split-$\hat{R}$	1.0000	1.0000	1.0000	1.0009	1.0085	1.0055

Table 4: PSRF and Split-$\hat{R}$ of β_1 for different models across various ρ values

8.6 ESS across Different Models and Correlation Levels

Figure 5 presents a comparison of the Effective Sample Size (ESS) for six models based on different MCMC algorithms and prior distributions at various levels of correlation. It is observed that the Gibbs sampling model with a normal prior consistently maintains the highest ESS values (approximately between 6500 and 8800, equating to about 32.5% to 44% sample effectiveness) across all correlation settings. This indicates that the `Gibbs+normal` model obtained the most actual independent samples in our set of 20,000 iteration samplings. Although the effective sample sizes of `Gibbs+t` and `Gibbs+Cauchy` are slightly lower than `Gibbs+normal`, they still reach about 4000 to 7500 in 20,000 iterations, approximately 20% to 37.5% effectiveness. In contrast, the models using MH random walk show significantly lower ESS values, with effective samples consistently below 500. This demonstrates that even after applying a thinning strategy of sampling every four steps, MH random walk may require more iterations to achieve a comparable number of effective samples as Gibbs sampling. Therefore, Gibbs sampling, regardless of the prior assumption, provides higher sampling efficiency, greater sample independence, and better effectiveness, especially the combination with a normal prior. Additionally, it is observed that at $\rho = 0.6$ or 0.9, the sample effectiveness when MH is combined with the Cauchy prior is lower than when combined with normal or Student's t-priors. This implies that, among the six combinations, the MH combined with Cauchy prior has the smallest number of effective samples in cases of high data correlation.

It is worth mentioning that my design involves first calculating the ESS values for individual chains in MCMC and then computing the average effective sample size (ESS) across multiple chains. The advantage of this approach is that it helps avoid randomness and increases the credibility of the experimental results.

8.7 Ranking Plot with WAIC

As shown in *Figure 6*, the WAIC values of all 24 models are very close to each other, indicating that the convergence of the chains and the sampling efficiency are not always directly related to the WAIC values. For example, as we have previously observed, although the models using the MH algorithm did not perform well in terms of chain mixing and effective sample size (ESS), their WAIC values do not differ significantly from those models using Gibbs sampling. We will delve further into this point in the discussion section(see *Chapter 9.2*).

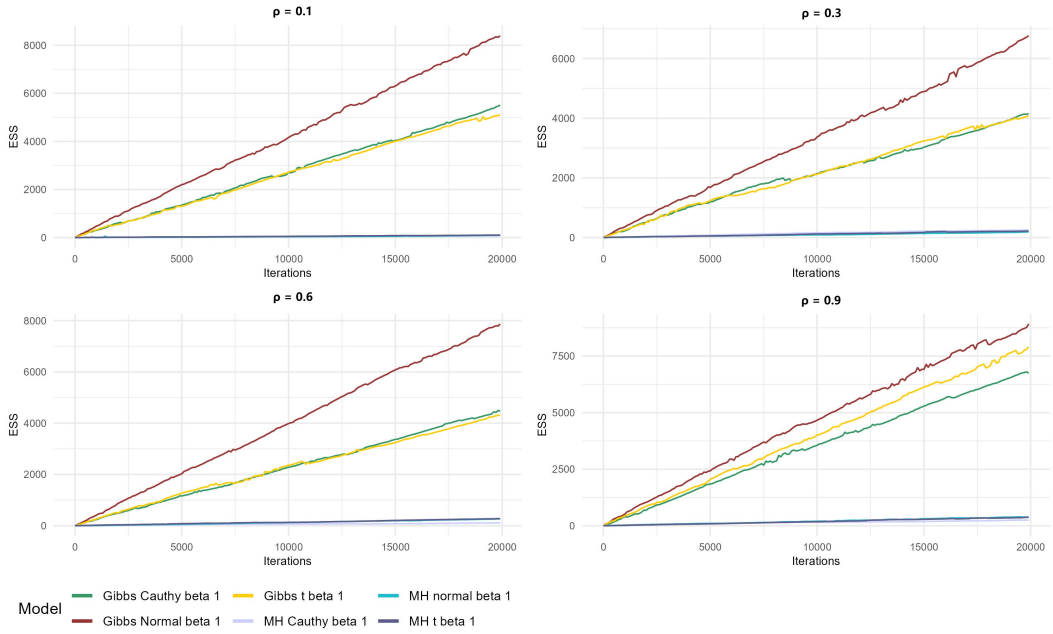

Figure 5: Effective sample size trends for different models under varying correlations

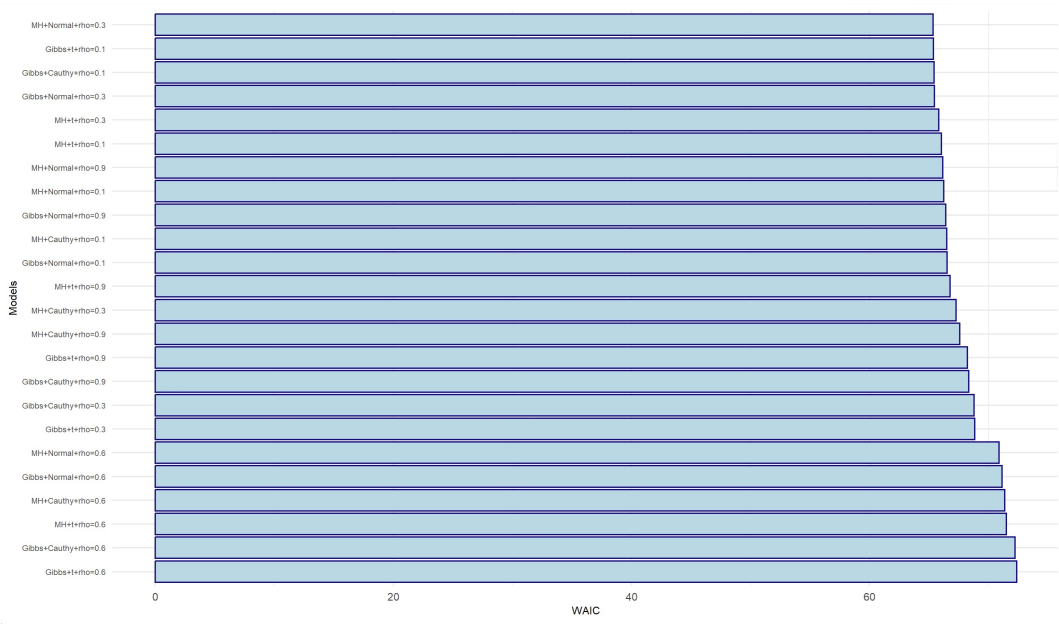

Figure 6: WAIC Ranking plot for all models

9 Summary and Discussion

9.1 Summary of Results

Based on the experimental results, it is observed that after 20,000 iterations, the posterior distribution integrals obtained by combining Gibbs Sampling and the Metropolis-Hastings random walk algorithm with three different prior distributions are all very close to 1. Additionally, the posterior mean accuracy of all models is generally acceptable.

By analyzing the trace plots and rank plots, it was found that the posterior single chains obtained using the MH algorithm did not show obvious signs of convergence, but instead had issues with low sampling efficiency and autocorrelation. Additionally, the rank plots also indicated poor multi-chain mixing and inconsistent convergence when using the MH algorithm. Conversely, the chains sampled using Gibbs sampling exhibit rapid and stable convergence, as evidenced by their trace plots. Additionally, the rank plots for these chains demonstrate effective mixing across multiple chains. In the validation using Split-$\hat{R}$, we observed a consistent conclusion: regardless of the chosen prior assumption, models employing Gibbs sampling exhibited significantly superior convergence efficiency compared to the MH random walk algorithm. Notably, the combination of MH method with a normal prior demonstrated better convergence efficiency than other priors. This may be partly attributed to the fact that the simulated dataset X also follows a normal distribution. In the evaluation of Effective Sample Size (ESS), the conclusion was further supported: models using Gibbs sampling exhibited significantly higher posterior sampling effectiveness compared to those using the MH algorithm. Among them, the Gibbs model combined with a normal prior had the highest number of effective samples. In contrast, models employing MH random walk had a significantly lower number of effective samples, indicating high autocorrelation and low sampling effectiveness.

Finally, after ranking the models using WAIC, we did not find significant differences among them. However, it is important to note that this does not directly lead to the conclusion that "the predictive ability of the models are theoretically similar." This is because when the chains have not fully converged, the estimation of parameters becomes inaccurate, rendering the WAIC values unreliable. On the other hand, this also suggests that a model's WAIC is not directly linked to the chain's convergence performance. I will discuss this point in further detail in the following subsection.

9.2 Discussion

The results of the simulated experiments conducted in this thesis are closely related to a variety of factors, including the sample size of the simulated dataset, the number of parameters, the setting of prior parameters, the number of iterations in the MCMC algorithm, the step length in MH sampling and the choice of the proposal distribution. Therefore, these experimental results are not universally applicable. For example, when the sample size is set to 50 and other conditions remain unchanged, the results indicate that the Cauchy prior significantly affects the convergence of the chain. Regardless of which MCMC algorithm is used, its posterior sampling is difficult to converge effectively, as evidenced by multiple indicators such as trace plot, Split-$\hat{R}$, and ESS. I also tried different parameter values from those recommended in the 2018 literature for Cauchy and Student's t-prior distributions, and the results showed that the differences from the recommended configuration were not significant. However, some predictable differences do exist. For instance, significantly increasing the sampling step length in the Thinning strategy of the MH algorithm can increase the

effectiveness of MH algorithm's posterior samples and reduce autocorrelation in sampling. But it is important to recognize that due to the increased step length, the total number of iterations significantly increases, indirectly indicating that the MH algorithm is less efficient in convergence than Gibbs sampling, a conclusion consistent with our experimental findings.

It's also important to note that the convergence performance of the MH algorithm is closely related to the choice of proposal distribution. The MH algorithm requires careful tuning and optimization to achieve optimal performance, which is not the main focus of this paper. The selection of a common proposal distribution was motivated by the desire for simplicity, computational efficiency, and the scope of our research objectives. Certainly, we can seek to find a more optimal proposal distribution by adjusting parameters. As per experience, the MH random walk algorithm typically selects acceptance rate in the range of 20% to 50% to prevent excessively small step sizes and, consequently, insufficient exploration of the parameter space. However, our experimental results have already indicated that the effective sample proportion for the Gibbs models falls within the same range of 20% to 50%. This suggests that even if all samples the MH random walk algorithm generates are effective, its effective sample size remains similar to that of the Gibbs models. This results comes at the cost of computational and time expenses. Therefore, in practice, even with the adoption of an optimal proposal distribution, our comparative results may still demonstrate the superiority of Gibbs sampling.

In *Section 8.5*, we discussed the inability to calculate PSRF values at lower iteration counts, attributed to failures in the Cholesky decomposition of the covariance matrix during the computation process by the `gelman.diag` function. This issue arises when MCMC chains do not converge adequately, or their mixing is insufficient, leading to a singular or near-singular covariance matrix, which in turn impedes the Cholesky decomposition. As a result, the PSRF values cannot be successfully calculated in the early stages when chain samples are insufficient, preventing the creation of PSRF trend graphs similar to those depicted in *Figure 4*. This also underscores the relative computational instability of PSRF compared to Split-$\hat{R}$.

In our study, it is crucial to note that the WAIC does not have a direct relationship with the MCMC convergence efficiency of models. WAIC focuses on the overall predictive ability of a model, rather than on the process of estimating model parameters or their convergence. MCMC convergence efficiency refers to the number of iterations required to reach the target posterior distribution. Given that WAIC and MCMC convergence efficiency play different roles in Bayesian model evaluation, in our experiments, models using the MH algorithm exhibited similar WAIC values to other models, even with a lower ESS (Effective Sample Size, see *Figure 5*), a slower convergence process (see *Figure 4*) and poor parameter mixing(see *Figure 3*). However, we must not overlook the fact that a good MCMC convergence is a prerequisite for an effective WAIC evaluation. If MCMC sampling has not converged, even an apparently favorable WAIC value might be misleading due to inaccurately estimated parameters. Therefore, before employing WAIC for model evaluation, it is essential to ensure that MCMC sampling has indeed converged.

References

[1] Effective sample size for estimating the mean. R Documentation. Available at: `https://search.r-project.org/CRAN/refmans/coda/html/effectiveSize.html` [Accessed 28 December 2023].

[2] Gelman and rubin's convergence diagnostic. R Documentation. Available at: `https://search.r-project.org/CRAN/refmans/coda/html/gelman.diag.html` [Accessed 28 December 2023].

[3] Widely applicable information criterion. R Documentation. Available at: `https://search.r-project.org/CRAN/refmans/LaplacesDemon/html/WAIC.html` [Accessed 28 December 2023].

[4] Alan Agresti. *Categorical data analysis*, volume 792. John Wiley & Sons, 2012.

[5] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical association*, pages 669–679, 1993.

[6] Hee Min Choi and James P Hobert. The Pólya-Gamma Gibbs sampler for Bayesian logistic regression is uniformly ergodic. *Electronic Journal of Statistics*, 2013.

[7] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. Bayesian data analysis. *Bayesian Data Analysis*, 2014.

[8] Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[9] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.

[10] Joyee Ghosh, Yingbo Li, and Robin Mitra. On the use of Cauchy prior distributions for Bayesian logistic regression. *Project Euclid*, 2018.

[11] Leonhard Held and Daniel Sabanés Bové. Applied statistical inference. *Springer, Berlin Heidelberg*, 2014.

[12] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya-Gamma latent variables. *Journal of the American statistical association*, 108(504):1339–1349, 2013.

[13] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian analysis*, 16(2):667–718, 2021.

[14] Sumio Watanabe and Manfred Opper. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of machine learning research*, 11(12), 2010.