# The Art of Data Augmentation and Parameter Expansion in Markov Chain Monte Carlo

by

JIANG Zhangziyan DB928356
GONG Jinqi DC027163

MATH4004: Graduation Project
2023/2024


BSc. in Mathematics (Statistics and Data Science)
Department of Mathematics
Faculty of Science and Technology
University of Macau

Name of Supervisor: LIU Zhi

Faculty/Department: Department of Mathematics, Faculty of Science and Technology

Name of Co-Supervisor (if any):

Faculty/Department:

Approved by

# Abstract

Markov Chain Monte Carlo (MCMC) method plays a crucial role in Bayesian inference but suffers inefficiencies in high-dimensional scenarios. In this report, we summarize recent developments in integrating Data Augmentation (DA) and Parameter Expansion (PE) techniques to enhance MCMC efficiency. By leveraging left-(invariant) Haar measures on locally compact groups, we provide a precise definition of the Parameter Expansion Data Augmentation (PX-DA) algorithm. This novel approach refines the traditional DA methods and exhibits improved convergence properties, as supported by theoretical analysis and extensive simulations, and contributes to advancing Bayesian methods, providing a more robust framework for handling complex models.



**Keywords:** Markov Chain Monte Carlo, Data Augmentation, Parameter Expansion, Haar Measures, Bayesian Inference, MCMC Convergence.

# Contents

# 1 Introduction

In Bayesian statistical analysis, the posterior distribution of parameters often exhibits a complex structure, posing challenges for direct sampling. To efficiently extract samples from these intricate distributions, the Markov Chain Monte Carlo (MCMC) method - particularly the Gibbs sampler - has become an essential tool for researchers due to its adaptability to various complex multidimensional distributions. By iterative sampling from conditional distribution, the Gibbs sampler explores the parameter space globally, approximating the true posterior distribution. However, when faced with high-dimensional parameter spaces or complex dependencies among parameters, the Gibbs sampler may suffer from inefficiency, significantly limiting its practical performance.

Furthermore, handling missing data is critical when dealing with complex statistical models. In real-world data analysis, data incompleteness is common, necessitating the development of Bayesian methods capable of handling incomplete information. Data augmentation (DA) techniques address this issue by transforming the complex target distribution (which involves missing data) into a more manageable form through simulation. The core idea of DA is to jointly consider the observed and missing data, constructing a complete dataset, thereby simplifying the computation and sampling process for the posterior distribution of parameters. This approach not only tackles the missing data problem but also improves the sampling efficiency of the MCMC sampler on complex distributions.

Despite the success of DA methods, dependencies between layers can still lead to suboptimal sampling efficiency. To enhance sampling efficiency further, the Parameter Expansion (PE) method offers a fresh perspective. PE expands the parameter space by introducing additional parameters. These parameters do not directly alter the distribution of observed data but provide the model with more flexibility and degrees of freedom. Through this approach, the PX-DA (Parameter Expansion Data Augmentation) method enhances the algorithm's exploration capability in the parameter space without altering the posterior distribution of observed data, thereby improving the convergence speed and overall performance of MCMC algorithms.

This report delves into two strategies: DA and PX-DA. Leveraging the mathematical tool of left-(invariant) Haar measures on locally compact groups, we rigorously define and discuss the PX-DA algorithm. We demonstrate how PE can enhance DA algorithms, especially when appropriate priors and expansion parameters are used. PX-DA exhibits superior convergence performance compared to traditional DA methods under certain conditions. These findings not only validate the effectiveness of the PX-DA algorithm but also provide new perspectives and methods for the inference of complex models with Bayesian statistics.

Our report builds upon the foundational contributions of Tanner and Wong (1987) and draws inspiration from subsequent studies by Liu and Wu (1999). Through a combination of experimental validation and theoretical analysis, we provide novel insights and practical guidance for data processing and algorithm design within Bayesian statistical analysis. Our goal is to foster the wider adoption and advancement of Bayesian methods across various domains.

# 2 Foundations of Data Augmentation

## 2.1 Bayesian Statistics and Missing Data

In Bayesian statistics, we aim to infer the posterior distribution of unknown parameters by analyzing observed data. However, when data is incomplete (i.e., there are missing data, $Y_{mis}$), the complete posterior distribution $p(\theta \mid Y_{obs}, Y_{mis})$ cannot be directly computed (Kong et al., 1994). To address this issue, data augmentation algorithms approximate the posterior distribution of complete data by simulating the distribution of missing data.

The core of data augmentation lies in iteratively improving the estimate of the predictive distribution $p(Y_{mis} \mid \theta, Y_{obs})$ for missing data, indirectly obtaining an approximation to the posterior distribution of parameters $p(\theta \mid Y_{obs})$. This process can be viewed as a Markov Chain Monte Carlo (MCMC) method, where each sampling step depends on the previous result.

## 2.2 Implementation Strategies of Data Augmentation Algorithms

The fundamental idea of the data augmentation algorithm is to use observed data $Y_{obs}$ and simulated missing data $\hat{Y}_{mis}$ to construct an approximated complete dataset and draw posterior distribution samples for the parameters. This process can be achieved through the following steps.

## 2.3 Monte Carlo Approximation

First, we need to calculate the posterior distribution of the observed data $p(\theta \mid Y_{obs})$. According to Bayes' Theorem, this can be accomplished by integrating over all possible missing data $Y_{mis}$:

$$p(\theta \mid Y_{obs}) \propto \int p(Y_{obs}, Y_{mis} \mid \theta) p(\theta) dP(Y_{mis}).$$

In practice, this integral is often intractable to solve analytically, so we resort to Monte Carlo methods for approximation (Metropolis and Ulam, 1949). Specifically, we can approximate the posterior distribution through the following steps:

1. Draw $m$ parameter samples $\{\theta^{(i)}\}_{i=1}^{m}$ from the prior distribution $p(\theta)$.

2. For each parameter sample $\theta^{(i)}$, draw simulated missing data $\hat{Y}_{mis}^{(i)}$ from the conditional distribution $p(Y_{mis} \mid \theta^{(i)}, Y_{obs})$.

3. Use the simulated complete dataset $(Y_{obs}, \hat{Y}_{mis}^{(i)})$ to update the approximation of the parameter's posterior distribution.

## 2.4 Multiple Imputation and Data Augmentation

Tanner and Wong (1987) suggested that initiating with an approximation $g(\theta)$ to the target distribution $p(\theta \mid y_{obs})$, allows for the generation of $m$ independent instances of missing data $y_{mis}^{(1)}, \ldots, y_{mis}^{(m)}$, derived from the predictive distribution:

$$\tilde{p}\left(y_{mis}\right) = \int p\left(y_{mis} \mid \theta, y_{obs}\right) g(\theta)\, d\theta.$$

This process involves initially drawing $\theta^{(j)}$ from $g(\theta)$ followed by $y_{mis}^{(j)}$ from $p(y_{mis} \mid \theta^{(j)}, y_{obs})$. The resulting $y_{mis}^{(j)}$ are termed "multiple imputations". Utilizing these imputations, we can construct an enhanced approximation of the posterior distribution:

$$g_{new}(\theta) = \frac{1}{m} \sum_{j=1}^{m} p\left(\theta \mid y_{obs}, y_{mis}^{(j)}\right).$$

If $g(\theta)$ accurately represented the true posterior distribution, then $\tilde{p}\left(y_{mis}\right)$ would provide the precise predictive distribution for $Y_{mis}$. However, $g(\theta)$ typically serves as an initial approximation in practical applications, necessitating iterative refinement.

The iterative mechanism of data augmentation and multiple imputation progressively sharpens the accuracy of our posterior distribution estimate. Each iteration is a deliberate attempt to more precisely emulate the missing data, utilizing these simulations to refine the posterior parameter distribution. With increasing iterations, the expectation is that convergence upon the target distribution will occur, culminating in a dependable parameter estimation $\theta$.

## 2.5 Connection with the Gibbs Sampler

Upon further examination of the Data Augmentation (DA) algorithm, it becomes clear that the simulation of multiple copies of missing data in each iteration, denoted by $m$, is not strictly necessary for the convergence of the algorithm. To elucidate this concept, let us focus on the first iteration of the process.

The goal is to generate a new imputed data point, $Y^{(i)}$. Initially, we randomly select a mixture component, which is a step in the DA algorithm referred to as Step 1. This mixture component is associated with the previous iteration's imputed data, $Y^{(i-1)}$. Subsequently, we draw a parameter vector, $\theta^*$, from this mixture component and compute the conditional probability $P(\theta^* \mid Y_{obs}, Y_{mis})$.

In a metaphorical sense, we can view $Y^{(i)}$ as the "offspring" of $Y^{(i-1)}$. Due to the stochastic nature of sampling, a proportion of the initial imputations—approximately $m/2.718$ when $m$ is large—will not produce any "offspring" in the subsequent iterations. This means that these initial imputations do not contribute to the approximation of the posterior distribution in the future.

In essence, we can regard about 37% of the imputations in the zeroth generation as being discarded in a random fashion, and thus, not utilized in the later stages of the algorithm. After a sufficient number of iterations, the lineage of all imputations will be traced back to a single common "ancestor," indicating convergence. This suggests that only one imputed data point from the initial generation contributes

to the final approximation of the posterior distribution. Since the selection of the mixture component, and consequently, the parent $\theta^*$, is entirely random, the selected "ancestor" is chosen without any bias—it is a matter of chance.

Mathematically, the DA algorithm can be seen as equivalent to an algorithm that imputes only a single missing data point $Y_{mis}$, i.e., $k = 1$), in each iteration. This is essentially a Gibbs sampler with two components. Going forward, we will refer to a two-component Gibbs sampler as the "Data Augmentation Scheme."

This process is visually depicted in Figure 1, which provides a schematic representation of the data augmentation scheme.

The abstract formulation of the data augmentation approach can be encapsulated as follows. Suppose we aim at simulating from a distribution $q(\theta)$. We construct an "augmented" system $\pi(\theta, Y_{mis})$ such that the marginal distribution of $\theta$ under this system is equivalent to $q(\theta)$, that is, $\int \pi(\theta, Y_{mis})dY_{mis} = q(\theta)$. If this augmented system facilitates iterative conditional sampling effectively, we can use a Gibbs sampler to simulate from it and extract all necessary information about $q(\theta)$.

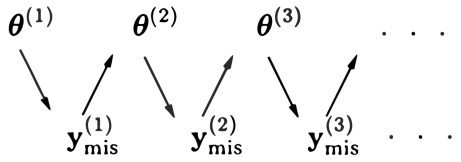


Figure 1: Illustration of the data augmentation scheme.

## 2.6 Example of Data Augmentation: EM Algorithm

### 2.6.1 Introduction

One of the most widely used methods for data augmentation is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), which is an iterative algorithm of finding maximum likelihood estimates of parameters in statistical models when the data is incomplete.

The EM algorithm is based on the idea of augmenting the observed data with unobserved data, and then iteratively applying the expectation (E-step) and maximization (M-step) steps until convergence. Consider a probabilistic model with observed data $Y_{obs}$ and missing data $Y_{mis}$, and parameters $\theta$. The complete data is $Y = (Y_{obs}, Y_{mis})$. The likelihood function for the observed data in terms of the complete data likelihood is:

$$L(\theta; Y_{obs}) = p(Y_{obs} \mid \theta) = \int p(Y_{obs}, Y_{mis} \mid \theta) dY_{mis}.$$

Direct optimization of $L(\theta; Y_{obs})$ is typically difficult due to the integration over $Y_{mis}$. The EM algorithm facilitates this by iterating over the following two steps:

**E-Step:** Compute the expected value of the log-likelihood of the complete data, conditioned on the observed data and the current estimate of the parameters $\theta^{(t)}$:

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{Y_{mis}|Y_{obs},\theta^{(t)}}[\log p(Y_{obs}, Y_{mis} \mid \theta)].$$

This expectation computes the average log-likelihood across the possible completions of the data, weighted according to their current estimated probabilities.

**M-Step:** Maximize the expected log-likelihood obtained in the E-Step:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}).$$

This step updates the parameter estimates to those that maximize the likelihood of the complete data, as averaged over the estimated distribution of the missing data from the E-Step.

**Derivation of the EM-Algorithm**

$$\begin{aligned}
\log P(y_{obs} \mid \theta) &= \log \int_{y_{mis}} P(y_{obs}, y_{mis} \mid \theta) dy_{mis} \\
&= \log \int_{y_{mis}} \frac{P(y_{obs}, y_{mis} \mid \theta)}{g(y_{mis})} \dot{g}(y_{mis}) dy_{mis} \\
&= \log E_{g(y_{mis})} \left[ \frac{P(y_{obs}, y_{mis} \mid \theta)}{g(y_{mis})} \right] \\
&\geq E_{g(y_{mis})} \left[ \frac{P(y_{obs}, y_{mis} \mid \theta)}{g(y_{mis})} \right], \quad \text{from Jensen's inequality}
\end{aligned}$$

The equality holds when $\log \dfrac{P(y_{obs}, y_{mis} \mid \theta)}{g(y_{mis})}$ is a constant, i.e.,

$$\frac{P(y_{obs}, y_{mis} \mid \theta)}{g(y_{mis})} = c \neq 0.$$

$$\implies g(y_{mis}) = \frac{1}{c} p(y_{obs}, y_{mis} \mid \theta).$$

Since $\int_{y_{mis}} g(y_{mis}) dy_{mis} = 1$, we have $\int_{y_{mis}} p(y_{obs}, y_{mis} \mid \theta) dy_{mis} = 1$. Thus

$$c = P(y_{obs} \mid \theta).$$

$$\implies g(y_{mis}) = \frac{P(y_{obs}, y_{mis} \mid \theta)}{P(y_{obs} \mid \theta)} = P(y_{mis} \mid y_{obs}, \theta).$$

Since $g(y_{mis})$ is updating through iteration, it should be

$$g(y_{mis}) = P(y_{mis} \mid y_{obs}, \theta^{(t)}).$$

Then

$$\log P(y_{obs} \mid \theta) = E_{y_{mis} \mid y_{obs}, \theta^{(t)}} \left[ \frac{P(y_{obs}, y_{mis} \mid \theta)}{P(y_{mis} \mid y_{obs}, \theta^{(t)})} \right],$$

where $\theta$ has nothing to do with $\theta^{(t)}$. Hence, our goal is to find

$$\hat{\theta} = \operatorname*{argmax}_{\theta} E_{y_{mis} \mid y_{obs}, \theta^{(t)}} \left[ P(y_{obs}, y_{mis} \mid \theta) \right].$$

### 2.6.2 Application of the EM Algorithm to Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are a probabilistic model assuming that the data are generated from a mixture of several Gaussian distributions, each with its own mean and covariance. This model is widely used for applications such as clustering, density estimation, and pattern recognition.

Consider a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ consisting of $N$ independent observations drawn from a mixture of $K$ Gaussian distributions. The density function of a GMM is given by:

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k),$$

where $\pi_k$'s are the mixing coefficients with $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$, and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$ is the Gaussian distribution for component $k$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\mathbf{\Sigma}_k$.

**E-step:** In the E-step, the algorithm calculates the posterior probabilities (responsibilities) that each data point $\mathbf{x}_i$ belongs to a component $k$, given the current parameter estimates. These responsibilities are defined as:

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)},$$

which are essential for the computation of the expected complete-data log-likelihood.

**M-step:** In the M-step, the algorithm updates the parameters based on the responsibilities calculated in the E-step. The updated parameters are computed as follows:

$$\begin{aligned}
\boldsymbol{\mu}_k^{new} &= \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} \mathbf{x}_i, \\
\mathbf{\Sigma}_k^{new} &= \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{new})(\mathbf{x}_i - \boldsymbol{\mu}_k^{new})^T, \\
\pi_k^{new} &= \frac{N_k}{N},
\end{aligned}$$

where $N_k = \sum_{i=1}^{N} \gamma_{ik}$.

The EM algorithm alternates between these E and M steps until convergence, typically when the increase in the log-likelihood is below a predetermined threshold.

This detailed exposition of the EM algorithm in the context of Gaussian Mixture Models provides a clear example of how advanced mathematical models are fit using iterative algorithms like EM, which are capable of handling latent variables efficiently.

To empirically evaluate the effectiveness of the GMM-EM algorithm, we conducted an experiment using the well-known Iris dataset in the `scikit-learn`, which is a Python library by Pedregosa et al. (2011). This dataset consists of 150 instances, each with four features: sepal length, sepal width, petal length, and petal width.

The dataset is divided into three classes, each corresponding to a different species of Iris flowers.

Our objective is to cluster the instances without prior knowledge of the species. To achieve this, we implement the GMM-EM algorithm as described in Section 2.6.2. We set the number of components to 3, aligning with the known number of species in the dataset, and allowed the algorithm to converge to a solution.

The results of our experiment show that the GMM-EM algorithm achieved an accuracy of 82%.

The visualization of the GMM components (Figure 2) illustrates how the algorithm has successfully identified the natural groupings within the dataset. Each ellipse in the plot represents the region of high probability for a particular Gaussian component, with the data points clustered around these regions. This visual representation provides a clear and intuitive understanding of the clusters formed by the GMM-EM algorithm.

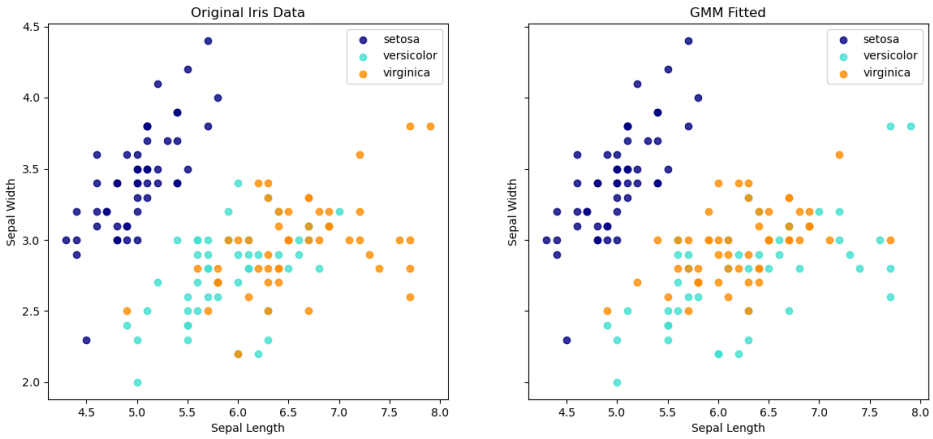


Figure 2: Comparison of the original Iris data (left) and the GMM components (right). The ellipses in the right plot represent the Gaussian components, and the colors correspond to the different species of Iris flowers, demonstrating the algorithm's ability to accurately capture the underlying structure of the dataset.

Overall, the experiment demonstrates the practical applicability of the GMM-EM algorithm for clustering. The combination of quantitative accuracy and qualitative visual analysis provides a comprehensive evaluation of the model's performance and offers insights into the data's structure.

### 2.6.3 Convergence Properties

The convergence properties of the EM algorithm are grounded in the concept of the likelihood function and the information matrix. Let $\boldsymbol{\theta}$ denote the parameter vector of interest, and let $L(\boldsymbol{\theta})$ represent the likelihood function based on the observed data $\mathbf{Y}_{obs}$. The observed information matrix is defined as $I(\boldsymbol{\theta}) = -E[\frac{\partial^2}{\partial \boldsymbol{\theta}^T \partial \boldsymbol{\theta}} \log L(\boldsymbol{\theta}) | \mathbf{Y}_{obs}]$.

**Ascent Property of the EM Algorithm** The ascent property is a fundamental characteristic of the EM algorithm. This property guarantees that the observed log-

likelihood $\log p(Y_{obs} \mid \theta)$ monotonically increases with each iteration, thus ensuring convergence to a local maximum of the likelihood function.

**Proposition 1.** $\log p(Y_{obs} \mid \theta^t) \leq \log p(Y_{obs} \mid \theta^{t+1})$.

*Proof.* To prove this property, consider the EM algorithm, which iteratively applies the E-step and the M-step to update the parameter estimates. We start by expressing the log-likelihood of the observed data $Y_{obs}$ as an integral involving the complete data log-likelihood:

$$\begin{aligned}\log p(Y_{obs} \mid \theta) &= \log \frac{p(Y_{mis}, Y_{obs} \mid \theta)}{p(Y_{mis} \mid Y_{obs}, \theta)} \\ &= \log p(Y_{mis}, Y_{obs} \mid \theta) - \log p(Y_{mis} \mid Y_{obs}, \theta).\end{aligned}$$

Take expectations on both sides with respect to $Y_{mis} \mid Y_{obs}, \theta^t$, we have

$$\begin{aligned}\log p(Y_{obs} \mid \theta) &= \int p(Y_{mis} \mid Y_{obs}, \theta^t) \log \frac{p(Y_{obs}, Y_{mis} \mid \theta)}{p(Y_{mis} \mid Y_{obs}, \theta)} dY_{mis} \\ &= \int p(Y_{mis} \mid Y_{obs}, \theta^t) \log p(Y_{obs}, Y_{mis} \mid \theta) dY_{mis} \\ &\quad - \int p(Y_{mis} \mid Y_{obs}, \theta^t) \log p(Y_{mis} \mid Y_{obs}, \theta) dY_{mis} \\ &= Q(\theta, \theta^t) - H(\theta, \theta^t),\end{aligned}$$

where $Q(\theta, \theta^t)$ is the expectation of the complete data log-likelihood with respect to the distribution of $Y_{mis}$ given $Y_{obs}$ and $\theta^t$, and $H(\theta, \theta^t)$ is the corresponding entropy term.

**E-step:** In the E-step, the algorithm calculates:

$$Q(\theta^t, \theta^t) = \int p(Y_{mis} \mid Y_{obs}, \theta^t) \log p(Y_{obs}, Y_{mis} \mid \theta^t) dY_{mis}.$$

**M-step:** In the M-step, the algorithm finds $\theta^{t+1}$ such that:

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t).$$

By definition, this maximization ensures:

$$Q(\theta^{t+1}, \theta^t) \geq Q(\theta^t, \theta^t).$$

To complete the proof, consider the entropy change from $\theta^t$ to $\theta^{t+1}$:

$$\begin{aligned}H(\theta^{t+1}, \theta^t) - H(\theta^t, \theta^t) &= \int p(Y_{mis} \mid Y_{obs}, \theta^t) \log \frac{p(Y_{mis} \mid Y_{obs}, \theta^{t+1})}{p(Y_{mis} \mid Y_{obs}, \theta^t)} dY_{mis} \\ &= -KL\left(p(Y_{mis} \mid Y_{obs}, \theta^{t+1}) \parallel p(Y_{mis} \mid Y_{obs}, \theta^t)\right) \leq 0,\end{aligned}$$

where $KL$ denotes the Kullback-Leibler divergence, which is non-negative and equals zero if and only if $p(Y_{mis} \mid Y_{obs}, \theta^{t+1}) = p(Y_{mis} \mid Y_{obs}, \theta^t)$.

Thus, combining the increase in $Q$ and the non-positive change in $H$, we have:

$$\log p(Y_{obs} \mid \theta^t) = Q(\theta^t, \theta^t) - H(\theta^t, \theta^t) \leq Q(\theta^{t+1}, \theta^t) - H(\theta^{t+1}, \theta^t) = \log p(Y_{obs} \mid \theta^{t+1}),$$

demonstrating the ascent property of the EM algorithm where $\log p(Y_{obs} \mid \theta^{t+1}) \geq \log p(Y_{obs} \mid \theta^t)$ and thereby ensuring that each iteration step improves or maintains the likelihood of the observed data under the model.

□

**Convergence to a Stationary Point** Normally, such as the existence of second-order derivatives and the boundedness of the parameter space, the EM algorithm converges to a stationary point of the likelihood function. This is formalized in the following statement.

**Proposition 2.** If $\boldsymbol{\theta}_0$ is an initial estimate such that $Q(\boldsymbol{\theta}_0)$ is finite, and the model satisfies the regularity conditions of Louis, then the sequence $\{\boldsymbol{\theta}^{(t)}\}_{t=0}^{\infty}$ generated by the EM algorithm converges to a limit $\boldsymbol{\theta}^*$ that is a stationary point of $Q(\boldsymbol{\theta})$, i.e., $Q(\boldsymbol{\theta}^*) \geq Q(\boldsymbol{\theta}')$ for all $\boldsymbol{\theta}'$ in the parameter space.

*Proof.* Let $\boldsymbol{\theta}^{(t)}$ denote the parameter estimate at iteration $t$, and $\boldsymbol{\theta}^*$ be the limit of the sequence. We aim to show that $Q(\boldsymbol{\theta}^*) \geq Q(\boldsymbol{\theta}')$ for all $\boldsymbol{\theta}'$ in the parameter space.

Given that $Q(\boldsymbol{\theta}^{(t)})_{t=0}^{\infty}$ is monotonically increasing and converges to $Q(\boldsymbol{\theta}^*)$, we have:

$$\lim_{t \to \infty} Q(\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}^*).$$

Furthermore, since $Q(\boldsymbol{\theta})$ is concave with respect to $\boldsymbol{\theta}$, the limit $\boldsymbol{\theta}^*$ must satisfy the first-order condition for optimality:

$$\nabla Q(\boldsymbol{\theta}^*) = \mathbf{0}.$$

This implies that $\boldsymbol{\theta}^*$ is a stationary point of $Q(\boldsymbol{\theta})$.

Therefore, for any parameter value $\boldsymbol{\theta}'$ in the parameter space, we have:

$$Q(\boldsymbol{\theta}^*) \geq Q(\boldsymbol{\theta}'), \quad \text{since} \quad Q(\boldsymbol{\theta}^*) = \lim_{t \to \infty} Q(\boldsymbol{\theta}^{(t)}).$$

Thus, the convergence to a stationary point is established. □

**Rate of Convergence** The convergence rate of the EM algorithm is influenced by several factors, including the initial estimate, the structure of the model, and the nature of the missing data. The rate can be characterized by the rate of convergence of the information matrix towards its inverse as the sample size increases.

Under certain regularity conditions, if the true parameter value $\boldsymbol{\theta}^0$ lies in the interior of the parameter space, and the observed information matrix $I(\boldsymbol{\theta}^0)$ is positive definite, then the EM algorithm converges at a superlinear rate.

This result implies that the EM algorithm is not only consistent but also asymptotically efficient, achieving a faster convergence rate as the sample size grows.

**Effect of Data Augmentation** Data augmentation can significantly affect the convergence rate of the EM algorithm. By introducing auxiliary variables that are functionally related to the missing data, the effective sample size is increased, which can lead to faster convergence and more accurate estimates. The choice of data augmentation scheme is thus crucial for the efficiency of the EM algorithm.

# 3 Bayesian Curve Fitting via Data Augmentation

## 3.1 Introduction to Bayesian Curve Fitting

In Bayesian curve fitting, the objective is to estimate the parameters of a model that best describes the relationship between a set of predictors and an observed response variable (Fan et al., 2010). This is often achieved by positing a smooth function, such as a spline, which can be flexibly adjusted to accommodate the data's underlying structure. The Bayesian approach is particularly advantageous as it allows for the incorporation of prior knowledge and the quantification of uncertainty in the estimates.

## 3.2 The Use of Auxiliary Variables

Auxiliary variables are introduced to circumvent the challenge of dealing with an unknown number of knots in a spline model. To fit the curve $f$, we have a general and powerful non-parametric approach, which is via spline functions of a given degree, $P \geq 1$. Then, $f$ can be written as the linear combination of basis functions:

$$f(x) = \alpha_0 + \sum_{j=1}^{P} \alpha_j x^j + \sum_{k=1}^{K} \eta_k (x - \gamma_k)_+^P, \quad x \in [a, b]$$

where $(x - \gamma_k)_+ = \max\{0, x - \gamma_k\}$ and $\gamma_k$, for $k = 1, \ldots, K$, represent the locations of the knot points, where $(x - \gamma_k)_+$ is the positive part of the difference $x - \gamma_k$. To facilitate the Bayesian treatment, we introduce binary auxiliary variables $z_k$ such that $z_k = 1$ if a knot is present at $\gamma_k$ and $z_k = 0$ otherwise. This representation allows the model to adaptively select the number of knots and their locations based on the data.

## 3.3 Prior Settings

The prior distribution for the model parameters plays a crucial role in the Bayesian analysis. In the context of spline regression with auxiliary variables, the following priors are commonly used:

- **Spline Coefficients** $\eta_k$: A natural choice is to assume a normal prior for the spline coefficients, $\eta_k \sim \mathcal{N}(0, \tau^2)$, where $\tau^2$ is a hyperparameter that controls the variance of the coefficients. This prior encourages smoothness in the spline function.

- **Knot Locations** $\gamma_k$: The knot locations can be assigned a uniform prior on a predefined interval, say $[0, 1]$, reflecting our lack of prior knowledge about the specific locations of the knots.
- **Auxiliary Variables** $z_k$: The prior for the auxiliary variables can be a Bernoulli distribution with parameter $p$, representing the probability that a knot is present at a given location. This choice allows for a flexible number of knots to be included in the model.
- **Variance of the Observations** $\sigma^2$: A common choice for the observation noise is an inverse-gamma prior, $\sigma^2 \sim \text{IG}(a, b)$, where $a$ and $b$ are hyperparameters that can be set based on prior knowledge about the variability in the data.

The use of these priors allows the Bayesian curve fitting model to incorporate both the data-driven information and the analyst's prior beliefs about the smoothness and structure of the underlying function.

## 3.4 Data Augmentation in Bayesian Curve Fitting

As we mentioned data augmentation is a technique used to improve the efficiency of MCMC algorithms by augmenting the observed data with additional, unobserved variables. In the context of Bayesian curve fitting we can augment the parameter space by adding auxiliary variables, in this way, data augmentation can be used to enhance the sampling of the knot locations and the spline coefficients.

The augmented data model includes the observed data $Y$, the latent knot locations $\gamma$, and the latent auxiliary variables $Z$. The joint distribution of these quantities can be expressed as:

$$p(Y, \gamma, Z \mid \eta, \sigma^2) = p(Y \mid \gamma, \eta, \sigma^2)p(\gamma \mid Z)p(Z)p(\eta \mid \tau^2)p(\sigma^2),$$

where $p(Y \mid \gamma, \eta, \sigma^2)$ is the likelihood of the observed data given the spline model, $p(\gamma \mid Z)$ is the prior for the knot locations, and $p(Z)$, $p(\eta \mid \tau^2)$, and $p(\sigma^2)$ are the priors for the auxiliary variables, spline coefficients, and observation noise variance, respectively.

During the MCMC sampling process, data augmentation allows for the generation of the latent variables from their conditional distributions, which can be more easily sampled than the original parameters. This leads to more efficient exploration of the parameter space and improved estimates of the posterior distribution.

The combination of auxiliary variables and data augmentation techniques provides a powerful framework for Bayesian curve fitting. By allowing for a flexible and data-driven selection of knots and efficient MCMC sampling, this approach can lead to more accurate and robust estimates of the underlying curve.

## 3.5 Simulation Studies

We consider the cubic spline model by setting $P = 3$ in the spline function

$$Y_i = \eta\left(x_i\right) + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is the white noise, and

$$\eta(x_i) = \sum_{j=0}^{3} \alpha_j x^j + \sum_{k=1}^{K_{\max}} \zeta_k \psi_k (x_i - \gamma_k)_+^3,$$

where $\zeta_k = 0$ or 1 which is the auxiliary variable. Define

$$|\zeta| = \sum_{k=1}^{K_{\max}} \zeta_k.$$

The priors on the parameters in the model:

$$\begin{gathered}
\sigma^2 \sim \Gamma^{-1}(0.1, 0.1), \\
\alpha_j \sim \mathcal{N}(0, 10), 0 \leq j \leq 3, \\
\psi_k \sim \mathcal{N}(0, 10), 0 \leq k \leq K, \\
\gamma_k \sim \mathcal{U}(\mathrm{I}_k), 0 \leq j \leq K.
\end{gathered}$$

Besides, we let

$$\begin{gathered}
p(\zeta \mid |\zeta|) = \begin{pmatrix} K_{\max} \\ |\zeta| \end{pmatrix}^{-1} = \frac{|\zeta|!\,(K_{\max} - |\zeta|)!}{K_{\max}!}, \\
p(\ |\zeta|\ \mid \lambda) \propto \frac{\lambda^{|\zeta|}}{|\zeta|!} \mathbf{1}_{\{|\zeta| \leq K_{\max}\}}, \\
\lambda \sim \mathrm{Bin}(K_{\max}, p = 0.5), p(\lambda) \propto \frac{K_{\max}!}{\lambda!\,(K_{\max} - \lambda)!}.
\end{gathered}$$

**Sampling scheme**
**Initialize:**
The initial sample is produced according to the following formula

$$\begin{gathered}
\sigma^{2,1} \leftarrow \Gamma^{-1}(0.1, 0.1), \\
\alpha_j^1 \leftarrow \mathcal{N}(0, 10), 0 \leq j \leq 3, \\
\psi_k^1 \leftarrow \mathcal{N}(0, 10), 0 \leq k \leq K, \\
\gamma_k^1 \leftarrow \mathcal{U}(\mathrm{I}_k), 0 \leq j \leq K, \\
\lambda^1 \leftarrow \mathrm{Bin}(K_{\max}, p = 0.5), \\
|\zeta|^1 \leftarrow p\left(\ |\zeta|\ |\lambda^1\right),
\end{gathered}$$

Then, Randomly select $|\zeta|^1$ elements from $\zeta_{1:K_{\max}}^1$ and assign a value of 1 to these elements, while assigning a value of 0 to the remaining elements. The superscript of the variable is used to denote the index of the sample. Subsequently, we denote $l$ as the index of the sample. (set $l = 1$ during this step)

**Step 1:** First, we let $l \leftarrow l + 1$. Besides, denote

$$\Phi^{l-1}=\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \left(x_1-\gamma_{k_1}^{l-1}\right)_+^3 & \cdots & \left(x_1-\gamma_{k_K}^{l-1}\right)_+^3 \\ 1 & x_2 & x_2^2 & x_2^3 & \left(x_2-\gamma_{k_1}^{l-1}\right)_+^3 & \cdots & \left(x_2-\gamma_{k_K}^{l-1}\right)_+^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \left(x_n-\gamma_{k_1}^{l-1}\right)_+^3 & \cdots & \left(x_n-\gamma_{k_K}^{l-1}\right)_+^3 \end{bmatrix}, \mathbf{Y}=\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix},$$

where $K=\left|\zeta^{l-1}\right|$, and $\zeta_{k_1}^{l-1}=1,\zeta_{k_2}^{l-1}=1,\cdots,\zeta_{k_K}^{l-1}=1$. The following equation is used to generate $\left[\alpha_1^l,\alpha_2^l,\alpha_3^l,\psi_1^l,\cdots,\psi_K^l\right]$

$$\left[\alpha_0^l,\alpha_1^l,\alpha_2^l,\alpha_3^l,\psi_{k_1}^l,\cdots,\psi_{k_K}^l\right]\sim\mathcal{N}\left(A^{-1}\Phi^{l-1\top}\mathbf{Y},\sigma^{2,l-1}A^{-1}\right),$$

where

$$A\triangleq\Phi^{l-1\top}\Phi^{l-1}+0.1\sigma^{2,l-1}I.$$

Furthermore, for any $s\in\left\{s\mid\zeta_s^{l-1}=0\right\},\psi_s^l\leftarrow\mathcal{N}(0,10)$.

**Step 2:** Using the MH to generate $\gamma_k^l$. First, for any $s\in\left\{s\mid\zeta_s^{l-1}=0\right\}$, assign $\gamma_s^l\leftarrow\mathcal{U}\left(\mathrm{I}_s\right)$. Furthermore, for any $r\in\left\{r\mid\zeta_r^{l-1}=1\right\}$, set $\gamma_r^*\leftarrow\mathcal{U}\left(\mathrm{I}_r\right)$, and define

$$e_i^*=\sum_{j=0}^{3}\alpha_j^l x_i^j+\sum_{k=1}^{K_{\max}}\zeta_k^{l-1}\psi_r^l\left(x_i-\gamma_r^*\right)_+^3-Y_i,$$

$$e_i^l=\sum_{j=0}^{3}\alpha_j^l x_i^j+\sum_{k=1}^{K_{\max}}\zeta_k^{l-1}\psi_r^l\left(x_i-\gamma_r^{l-1}\right)_+^3-Y_i$$

$$a=\min\left\{1,\exp\left[\sum_{i=1}^{n}\frac{\left(e_i^l\right)^2}{2\sigma^{2,l-1}}-\sum_{i=1}^{n}\frac{\left(e_i^*\right)^2}{2\sigma^{2,l-1}}\right]\right\}.$$

Subsequently, generate $u\sim\mathcal{U}(0,1)$. If $u<a$, then assign $\gamma_r^l=\gamma_r^*$ for $r\in\left\{r\mid\zeta_r^{l-1}=1\right\}$. Otherwise, set $\gamma_r^l=\gamma_r^{l-1}$.

**Step 3:** We define

$$\eta^l\left(x_i\right)=\sum_{j=0}^{3}\alpha_j^l x^j+\sum_{r\in\left\{r\mid\zeta_r^l=1\right\}}\psi_r^l\left(x_i-\gamma_r^l\right)_+^3.$$

Generate $\sigma^{2,l}$ based on the formula below:

$$\sigma^{2,l}\leftarrow\Gamma^{-1}\left(0.1+\frac{n}{2},0.1+\frac{1}{2}\sum_{i=1}^{n}\left[Y_i-\eta^l\left(x_i\right)\right]^2\right).$$

**Step 4:** Using the MH to generate $\zeta_k^l$. To begin with, let $\zeta_k^l\leftarrow\zeta_k^{l-1}$, and calculate

$$e_i=\sum_{j=0}^{3}\alpha_j^l x_i^j+\sum_{k=1}^{K_{\max}}\zeta_k^{l-1}\psi_r^l\left(x_i-\gamma_r^l\right)_+^3-Y_i.$$

Further, For $k=1,2,\cdots,K_{\max}$ :

1. assign $\zeta_k^* = 1 - \zeta_k^{l-1}, M = 1 - 2\zeta_k^{l-1}$,
2. calculate $\tilde{e}_i = e_i + M\psi_r^l \left(x_i - \gamma_r^l\right)_+^3, i = 1, 2, \cdots, n$,
3. calculate $J_1 = \exp\left\{\left[\sum_{i=1}^n (e_i)^2 - \sum_{i=1}^n (\tilde{e}_i)^2\right]/2\sigma^{2,l}\right\}$, and $J_2 = \left(\frac{\lambda^{l-1}}{|\zeta^l|+\zeta_k^*}\right)^M$,
4. calculate $a = \min(1, J_1 J_2)$, and generate $u \leftarrow \mathcal{U}(0,1)$,
5. If $u < a$, then assign $\zeta_k^l \leftarrow \zeta_k^*$ and $e_i \leftarrow \tilde{e}_i$, otherwise $\zeta_k^l \leftarrow \zeta_k^{l-1}$.

**Step 5:** Generate $\lambda^* \sim \text{Bin}(K_{\max}, p = 0.5), u \sim \mathcal{U}(0,1)$. If $u < \min\left[1, \left(\frac{\lambda^l}{\lambda^{l-1}}\right)^{|\zeta^l|}\right], \lambda^l \leftarrow \lambda^*$, other wise $\lambda^{l-1}$. Go to the step 1.

### 3.5.1 Comparison of the performance of with and without using DA

Now, we plot the ACF plots to show the effect of implementing data augmentation.

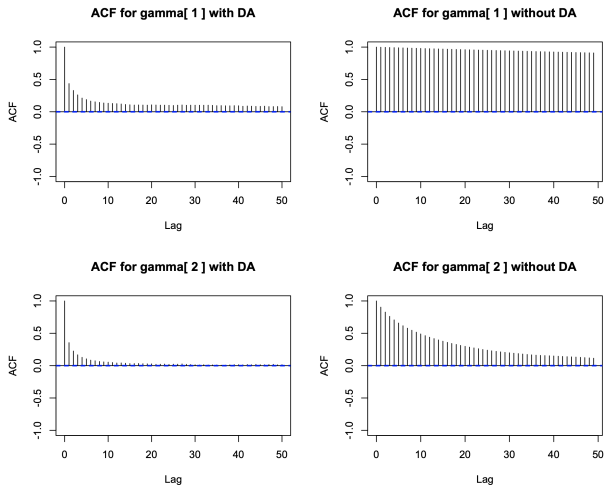


Figure 3: ACF plot: with and without using DA

From Figure 3, we conclude that the value of ACF decreases much faster as the lag increases, which indicates that DA is effective.

| Parameter | With DA | Without DA |
|---|---|---|
| $\alpha_0$ | 86151.681 | 663.32267 |
| $\alpha_1$ | 92356.746 | 355.65203 |
| $\alpha_2$ | 48477.154 | 352.47682 |
| $\alpha_3$ | 29862.479 | 95.66642 |
| $\psi_0$ | 100000.000 | 94.38433 |
| $\psi_1$ | 100000.000 | 15139.96309 |
| $\sigma^2$ | 91872.256 | 89309.96309 |
| $\gamma_0$ | 4979.635 | 97.25944 |
| $\gamma_1$ | 17673.547 | 2485.57896 |

Table 1: Values of Prarameters

Table 1 shows iterated values of the parameters.

# 4 Pólya-Gamma Augmentation in Bayesian Inference

As we mentioned data augmentation is a powerful tool in Bayesian statistics that enhances computational efficiency by introducing latent variables to reparameterize the model. In the realm of logistic regression, the Pólya-Gamma augmentation stands out as an innovative solution to the computational bottleneck posed by the logistic likelihood. By recasting the logistic model through the lens of the Pólya-Gamma distribution, Polson et al. (2013) enabled a more straightforward Gibbs sampling approach, thus facilitating Bayesian inference in logistic models. This section presents the integration of the Pólya-Gamma latent variables into logistic regression, illustrating how data augmentation streamlines the Bayesian analysis workflow. The discussion will highlight the methodological insights and computational strategies that arise from this augmentation, showcasing its utility in handling the logistic model's complexity.

## 4.1 Pólya-Gamma Distribution

**Definition** A Pólya-Gamma random variable, denoted as $\omega$, is important for data augmentation in Bayesian models with binomial likelihood. It is defined as

$$\omega \sim \mathrm{PG}(b, c) \iff \omega = \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + c^2/4\pi^2},$$

where $g_k \sim \mathrm{Gamma}(b, 1)$ are independent Gamma random variables, $b > 0$ is a parameter often representing the number of trials in the binomial model, and $c$ is a real number related to the logistic regression parameters.

**Properties**

- **Infinite Divisibility:** The Pólya-Gamma distribution is infinitely divisible, essential for modeling in a Bayesian framework.
- **Conjugacy:** This distribution results in posterior distributions that are easier to sample from due to its conjugate properties in logistic regression when parameterized by log-odds.

## 4.2 Theoretical Justification

**Theorem 1.** For $b > 0$ and any real $\psi$, the following integral identity holds:

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

where $k = a - b/2$ and $\omega \sim \mathrm{PG}(b, 0)$. Furthermore, the conditional distribution

$$p(\omega \mid \psi) = \frac{e^{-\omega\psi^2/2} p(\omega)}{\int_0^{\infty} e^{-\omega\psi^2/2} p(\omega) d\omega},$$

which implies $(\omega \mid \psi)$ is also in the Pólya-Gamma class, i.e., $(\omega \mid \psi) \sim \text{PG}(b, \psi)$.

Here we give the proof of the integral identity, and the derivation of the conditional distribution is provided in the work of Polson et al. (2013).

*Proof.* We aim to establish a relationship between the Pólya-Gamma distribution and our logistic likelihood expression. Let $\psi$ be a real number and consider the following steps.

First, we recognize that the hyperbolic cosine can be expressed as

$$\cosh\left(\frac{\psi}{2}\right) = \frac{e^{\frac{\psi}{2}} + e^{-\frac{\psi}{2}}}{2} = \frac{\left(e^{\frac{\psi}{2}} + e^{-\frac{\psi}{2}}\right) \cdot e^{\frac{\psi}{2}}}{2e^{\frac{\psi}{2}}} = \frac{e^{\psi} + 1}{2e^{\psi}},$$

$$\implies 1 + e^{\psi} = 2e^{\frac{\psi}{2}} \cosh\left(\frac{\psi}{2}\right).$$

Substituting this into our original expression, we get

$$\frac{(e^{\psi})^a}{(1+e^{\psi})^b} = \frac{(e^{\psi})^a}{(2e^{\frac{\psi}{2}})^b (\cosh(\frac{\psi}{2}))^b} = 2^{-b} e^{k\psi} \left(\frac{1}{\cosh(\frac{\psi}{2})}\right)^b,$$

where $k = a - \frac{b}{2}$.

Next, we apply the definition of the Pólya-Gamma distribution where $\omega \sim \text{PG}(b, 0)$ and take the expectation with respect to $\omega$, denoted as $E_\omega$, to obtain

$$2^{-b} e^{k\psi} E_\omega \left[e^{-\frac{\omega\psi^2}{2}}\right] = 2^{-b} e^{k\psi} \int_0^\infty e^{-\frac{\omega\psi^2}{2}} p(\omega)\, d\omega.$$

where $p(\omega)$ is the probability density function of the Pólya-Gamma distribution $\text{PG}(b, 0)$.

Thus, we have reformulated the logistic likelihood expression into a combination of a normal distribution and a Pólya-Gamma distribution, which is a significant step in the Bayesian analysis of logistic regression models. □

And the conditional (posterior) distribution $p(\omega \mid \psi)$ as being in the Pólya-Gamma class enables us to construct efficient Markov Chain Monte Carlo (MCMC) algorithms for Bayesian inference in logistic regression models.

Now, to derive our Gibbs sampler, we first write the likelihood contribution of observation $i$ as

$$\begin{aligned} L_i(\beta) &= \frac{\{\exp(x_i^T\beta)\}^{y_i}}{1+\exp(x_i^T\beta)} \\ &\propto \exp(k_i x_i^T \beta) \int_0^\infty \exp\{-\omega_i (x_i^T\beta)^2/2\}\; p(\omega_i \mid n_i, 0)\,, \end{aligned}$$

where $k_i = y_i - n_i/2$, and $p(\omega_i \mid n_i, 0)$ is the density of a Pólya-Gamma random variable with parameters $(n_i, 0)$.

Combining the terms from all $n$ data points gives the following expression for the conditional posterior of $\beta$, given $\omega = (\omega_1, \ldots, \omega_N)$:

$$\begin{aligned} p(\beta \mid \omega, y) \propto p(\beta) \prod_{i=1}^{N} L_i(\beta \mid \omega_i) &= p(\beta) \prod_{i=1}^{N} \exp\left\{k_i x_i^T \beta - \omega_i (x_i^T \beta)^2/2\right\} \\ &\propto p(\beta) \prod_{i=1}^{N} \exp\left\{\frac{\omega_i}{2}(x_i^T \beta - k_i/\omega_i)^2\right\} \\ &\propto p(\beta) \exp\left\{-\frac{1}{2}(z - X\beta)^T D (z - X\beta)\right\}, \end{aligned}$$

where $z = (k_1/\omega_1, \ldots, k_n/\omega_N)$, and where $D = \text{diag}(\omega_1, \ldots, \omega_N)$.

Let $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$ be the vector of regressors, $y_i$ be the number of successes, and $n_i$ be the number of trials, where $i = 1, ..., N$. Let $\beta \sim \mathcal{N}(b, B)$, and $y_i \sim \text{Bin}(n_i, 1/(1 + e^{-\psi_i}))$, where $\psi_i = x_i^\top \beta$ are the log odds of success.

Then we can use Pólya-Gamma method to sample from the posterior distribution by simply iterating two steps:

$$(\omega_i \mid \beta) \sim \text{PG}(n_i, x_i^\top \beta),$$

$$(\beta \mid y, \omega) \sim \mathcal{N}((X^\top D X + b^{-1})^{-1}(X^\top k + B^{-1} b), (X^\top D X + b^{-1})^{-1}),$$

where $k = (y_1 - n_1/2, ..., y_N - n_N/2)$ and $D$ is the diagonal matrix of $\omega_i$'s.

## 4.3 Implementation in Bayesian Logistic Regression

In the Bayesian logistic regression context, outcomes $y$ are modeled as

$$y_i \mid p_i \sim \text{Bernoulli}(p_i), \quad \text{logit}(p_i) = x_i^T \beta.$$

Using Pólya-Gamma augmentation, the conditional distributions are simplified:

- **Augmentation Step:** $\omega_i \mid \beta, x_i \sim \text{PG}(1, x_i^T \beta)$.
- **Sampling Step for $\beta$:** Given $\omega$ and the data, the posterior distribution of $\beta$ can be sampled using Gaussian distributions, simplifying the MCMC process.

## 4.4 Simulation Studies

### 4.4.1 The PG$(1, z)$ sampler

**Background and Relation to Jacobi Distribution** The $PG(1, z)$ distribution can be related to a tilted version of the Jacobi distribution, denoted $J^*(1, z)$, through the transformation

$$PG(1, z) = \frac{1}{4} J^*(1, z/2).$$

The Jacobi distribution $J^*(1, 0)$ relates to the Jacobi theta function, and its exponentially tilted version is $J^*(1, z)$.

**Density Representation** The density $f(x \mid z)$ of $J^*(1, z)$ is defined by an exponential tilt of the base Jacobi density:

$$f(x \mid z) = \cosh(z)e^{-\frac{xz^2}{2}} f(x),$$

where $f(x)$ is the density of $J^*(1)$.

When $f(x) = \sum_{n=0}^{\infty}(-1)^n a_n(x)$, where the coefficients $a_1(x) < a_2(x) < \cdots < a_n(x)$ for all $n \in \mathbb{N}_0$, then the partial sum $S_n x = \sum_{i=0}^{n}(-1)^i a_i(x)$, satisfy

$$S_0(x) > S_2(x) > \cdots > f(x) > \cdots > S_3(x) > S_1(x).$$

**Series Representation and Sampling Strategy** The density of $J^*(1, z)$ can be represented as an infinite alternating sum, suitable for sampling using an accept-reject algorithm with series expansion methods.

**Accept-Reject Sampling Procedure**

- **Proposal Distribution:** Construct a proposal density $g(x|z)$ that approximates the target density well. For the Jacobi distribution, the proposal is a mixture of an inverse-Gaussian for small values of $x$ and an exponential distribution for larger $x$, each conditioned on $z$.
- **Sampling Steps:**
  1. Draw $X \sim g(x \mid z)$.
  2. Draw $U \sim \mathcal{U}(c(z)g(X \mid z))$, where $c(z)$ is a normalizing constant ensuring the envelope property.
  3. Compute the series sum $S_n(X \mid z)$ iteratively for the alternating sum representation.
  4. Accept $X$ if $U < S_n(X \mid z)$ for an odd $n$, and reject $X$ and repeat from step 1 if $U > S_n(x)$ for some even $n$.

**Conversion to PG(1, z)** Once $X$ is sampled from $J^*(1, z/2)$, convert it to a sample from $\mathrm{PG}(1, z)$ by scaling:

$$Y = X/4.$$

### 4.4.2 Comparison of the performance between two samplers

Now, we compare our sampling scheme with the Metropolis-Hasting sampler. Based on the ACF plots presented in Figure 4, the comparison between the Metropolis-Hasting (MH) sampler and the Pólya-Gamma (PG) augmented sampling method indicates a distinct performance advantage in favor of the PG approach. For each of the four components, the ACF values for the PG method decrease more rapidly as the lag increases. This is indicative of better mixing and faster convergence to the stationary distribution for the PG sampler.

A more rapid decline in ACF values suggests that the samples generated by the PG method are less autocorrelated. This implies that successive samples are more independent of each other, which is a desirable property in Markov Chain Monte Carlo (MCMC) sampling. It leads to more efficient sampling because it reduces the number of samples needed to estimate quantities of interest with a given level of accuracy.

The effectiveness of the PG method is particularly notable at higher lags, where the ACF values approach zero much quicker than those from the MH sampler. This enhanced performance could be attributed to the PG method's augmentation strategy, which may be better suited to the underlying structure of the data or model being sampled.

In summary, the ACF plots strongly suggest that the PG method outperforms the traditional MH sampler in terms of efficiency and convergence speed, making it a compelling choice for practitioners seeking robust and effective MCMC sampling strategies.

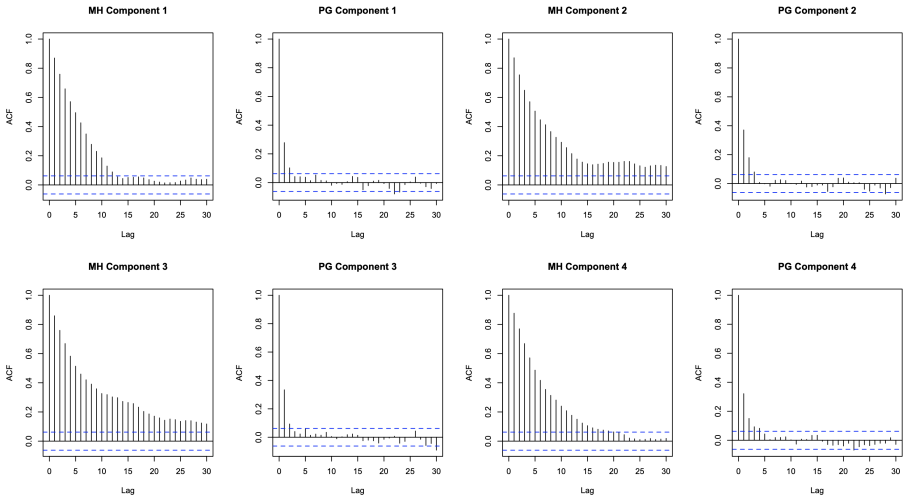


Figure 4: ACF plot: MH vs PG

# 5 Parameter Expansion for Data Augmentation

In the realm of Bayesian Statistics, the accurate representation of posterior distributions is paramount. However, this task is often complicated by the presence of missing data, which necessitates the use of robust methods to handle such incomplete information. Data Augmentation (DA) algorithms are a staple in this regard, yet they may sometimes suffer from slow convergence, particularly when dealing with highly structured or correlated data (Liu and Liu, 2001). This section explores the concept of Parameter Expansion (PE) as a means to enhance the efficiency of DA algorithms.

## 5.1 Autocorrelation in Data Augmentation

The DA algorithm, as originally proposed, iterates by imputing missing data and updating the parameter of interest. However, this straightforward approach can lead to a high degree of autocorrelation in the Markov Chain, which in turn hampers the algorithm's mixing properties. Specifically, the DA algorithm's two-step process involves first drawing missing data $Y_{mis}$ from its conditional distribution given the observed data $Y_{obs}$ and the current estimate of the parameter $\theta$, and then updating $\theta$ using the newly imputed data (Van Dyk and Meng, 2001). This method, while intuitive, may not fully exploit the structure of the data to guide the search through the parameter space.

## 5.2 Parameter Expansion

To address the issue of slow convergence, the concept of Parameter Expansion (PE) is introduced (Liu and Wu, 1999). PE involves augmenting the parameter space by introducing additional parameters that capture certain characteristics of the missing data distribution. These auxiliary parameters, denoted by $a$, are chosen to be part of a transformation group that acts on the missing data $Y_{mis}$, allowing for more global adjustments and potentially improving the exploration of the posterior distribution.

The introduction of $a$ as an expansion parameter enables the DA algorithm to transition from a local representation of the missing data to a more comprehensive one that accounts for the underlying structure of the data. By doing so, PE aims at reducing the autocorrelation observed in the standard DA algorithm and thereby accelerates the convergence to the target distribution.

## 5.3 Theoretical fundations for Parameter Expansion (PE)

The Markov Chain Monte Carlo (MCMC) method is a powerful tool for sampling complex probability distributions, particularly those that are difficult to access through traditional methods. At the heart of MCMC lies the metaphor of a "ghost point" moving through its sample space, with the sampler's output representing the trajectory of this point's movement.

### 5.3.1 The Ghost Point and MCMC Moves

Consider the operation of a Metropolis-Hastings algorithm, where at each step, a new position for the ghost point is proposed based on its current location. This tentative position is evaluated using an acceptance-rejection rule, which determines whether to move the ghost point to the proposed position or to keep it at its current location. This process is analogous to a random-scan Gibbs sampler, which selects a direction in the parameter space at random and moves the ghost point along this direction to a location drawn from a conditional distribution appropriate for the chosen direction.

### 5.3.2 Invariance and the Design of MCMC Schemes

The key principle in designing effective MCMC schemes is the preservation of the target distribution $\pi(\mathbf{x})$ under the proposed moves. This means that the probability of the ghost point being at any point in the sample space should remain unchanged after a move. The partial resampling principle discussed below is a broad strategy for ensuring this invariance, but it often lacks the specificity and practicality required for efficient sampling.

**Partial Resampling Principle in MCMC Methods**

Partial Resampling is a strategy used in Markov Chain Monte Carlo (MCMC) methods to enhance the sampling process from complex multivariate distributions. It is particularly useful when dealing with high-dimensional or intractable probability distributions. The key idea is to leverage the structure of the target distribution to design more efficient sampling algorithms.

**Gibbs Sampling and Partial Resampling** Gibbs sampling is a special case of Partial Resampling where the random vector $\mathbf{x}$ of interest is decomposed into components, and samples are drawn sequentially from their conditional distributions. Specifically, the vector $\mathbf{x}$ is split into two parts: $\mathbf{x} = (x_1, \mathbf{x}_{-1})$, where $x_1$ is a single component and $\mathbf{x}_{-1}$ represents the remaining components. The conditional distribution $\pi(x_1|\mathbf{x}_{-1})$ is then used to update $x_1$ with a new sample $x^*$. The invariance of the conditional distribution under the transition rule $A(x_1 \to x^*)$ ensures that the target distribution $\pi(\mathbf{x})$ remains unchanged after the update.

$$
\begin{aligned}
&\int \pi(x_1, \mathbf{x}_{-1}) A(x_1 \to x_1^* \mid \mathbf{x}_{-1})\, dx_1 \\
&= \pi(\mathbf{x}_{-1}) \int \pi(x_1 \mid \mathbf{x}_{-1}) A(x_1 \to x_1^* \mid \mathbf{x}_{-1})\, dx_1 \\
&= \pi(\mathbf{x}_{-1}) \pi(x_1^* \mid \mathbf{x}_{-1}) \\
&= \pi(x_1^*, \mathbf{x}_{-1}) \\
&= \pi(\mathbf{x}^*).
\end{aligned}
$$

This property allows the Gibbs sampler to make proper moves and explore the sample space effectively.

**Fiber Decomposition and Conditional Sampling** A more general approach to Partial Resampling involves partitioning the sample space $\mathcal{X}$ into disjoint subsets, known as fibers, such that $\mathcal{X} = \bigcup_{a \in A} \mathcal{X}_a$ and $\mathcal{X}_a \cap \mathcal{X}_b = \emptyset$ for $a \neq b$. Each fiber $\mathcal{X}_a$ is associated with a conditional distribution $\pi_a(\mathbf{x})$, which is the distribution of $\mathbf{x}$ given that it lies in $\mathcal{X}_a$. The overall distribution $\pi(\mathbf{x})$ can then be decomposed as an integral over the fibers:

$$
\pi(\mathbf{x}) = \int_{a \in A} \pi_a(\mathbf{x}) \upsilon_a(\mathrm{d}a),
$$

where $v_a(\mathrm{d}a)$ is a measure over the parameter space $A$. The invariance of $\pi(\mathbf{x})$ under transitions on the fiber is crucial for the Partial Resampling strategy. A transition rule $A(\mathbf{x} \to \mathbf{x}')$ that leaves $\pi_a(\mathbf{x})$ invariant will also leave $\pi(\mathbf{x})$ invariant:

$$\int \pi_a(\mathbf{x}) A(\mathbf{x} \to \mathbf{x}' \mid \mathcal{X}_a) \mathrm{d}\mathbf{x} = \int \pi_a(\mathbf{x}') A(\mathbf{x} \to \mathbf{x}' \mid \mathcal{X}_a) \mathrm{d}\mathbf{x}'.$$

This property enables the construction of MCMC algorithms that can efficiently sample from complex distributions by focusing on lower-dimensional fibers and using conditional moves that preserve the local structure of the target distribution.

**Challenges in Fiber Decomposition and Conditional Measures** In the context of Partial Resampling, two primary challenges arise. Firstly, the process of constructing a fiber decomposition for a given sample space lacks straightforward guidelines. The fiber decomposition is essential as it segments the sample space into subsets, each associated with a conditional distribution. However, without clear directives, this task can become intricate and arduous.

Secondly, even when a fiber decomposition is successfully defined, deriving the corresponding conditional measure $v_a(x)$ presents its own set of difficulties. The conditional measure is pivotal for the Partial Resampling process, as it governs the transition probabilities within the MCMC scheme. Despite the importance of $v_a(x)$, its determination can be a non-trivial problem, often requiring sophisticated mathematical techniques and a deep understanding of the underlying distribution.

### 5.3.3 Transformation Groups and Constructive Solutions

To address the limitations of partial resampling, we turn to the concept of transformation groups. These groups provide a structured way to describe the moves of the ghost point in the sample space. Specifically, we consider transformations that can be applied to the current position of the ghost point to obtain a new position, with the goal of maintaining the invariance of the target distribution.

Suppose at time $t$, the ghost point is located at $\mathbf{x}^{(t)} = \mathbf{x}$. At the subsequent time step $t+1$, the ghost point is "moved" to a new location $\mathbf{x}^{(t+1)} = \mathbf{x}'$ through the application of a transformation $\gamma$ from a set of transformations $\Gamma$. This move is designed to leave the target distribution $\pi(\mathbf{x})$ invariant, ensuring that the MCMC sampler explores the sample space in a manner consistent with the underlying probability distribution.

The use of transformation groups allows for a more explicit and constructive approach to MCMC sampling. By carefully selecting the group $\Gamma$ and the associated distribution for $\gamma$, we can design MCMC schemes that are not only theoretically sound but also computationally efficient. This approach forms the basis for the Parameter Expansion technique, which has been shown to improve the mixing and convergence properties of MCMC algorithms.

### 5.3.4 Transformation Groups and MCMC Moves

Suppose we can represent the transition from a current state $\mathbf{x}$ to a new state $\mathbf{x}'$ in an MCMC sampler through the application of a transformation $\gamma$, chosen from a

set of transformations $\Gamma$. This transformation acts as a "mover" that facilitates the transition:

$$\mathbf{x}' = \gamma(\mathbf{x}).$$

For instance, in a random-scan Gibbs sampler, the current position $\mathbf{x} = (x_i, \mathbf{x}_{[-i]})$, where $x_i$ is a single coordinate and $\mathbf{x}_{[-i]}$ represents all other coordinates, is updated by moving only the $i$-th coordinate:

$$\mathbf{x}' = (x_i', \mathbf{x}_{[-i]}).$$

This can be interpreted as a translation transformation applied to $\mathbf{x}$:

$$\left(x_i, \mathbf{x}_{[-i]}\right) \to \left(x_i + \gamma, \mathbf{x}_{[-i]}\right), \quad \gamma \in \mathbb{R}.$$

Here, the set of all eligible $\gamma$ values forms a group under the usual addition operation, and we denote this group as $\Gamma$.

### 5.3.5 Invariance and the Choice of Transformation Distribution

The crucial aspect of choosing $\gamma$ from $\Gamma$ is to ensure that the target distribution $\pi$ remains invariant under such transformations. In other words, the distribution of $\mathbf{x}' = \gamma(\mathbf{x})$ should be identical to the original distribution $\pi$. This leads to the general problem formulation: given a random variable $\mathbf{x}$ distributed according to $\pi$ and a set of transformations $\Gamma$, what distribution should we use to draw $\gamma$ from $\Gamma$ so that the transformed variable $\mathbf{x}'$ follows the same distribution $\pi$?

Later we will provide a clear answer to this question when the set of transformations $\Gamma$ forms a locally compact group. It offers a method to select $\gamma$ such that the distribution of the transformed variable is consistent with the target distribution, thereby ensuring the invariance property required for effective MCMC sampling.

### 5.3.6 Definition of a Locally Compact Group

Suppose $\pi(\mathbf{x})$ is a probability distribution of interest defined on the sample space $\mathcal{X}$. A set $\Gamma = \{\gamma\}$ of transformations on $\mathcal{X}$ is called a **locally compact group** (or a topological group) if:

- $\Gamma$ is a locally compact space.
- The elements in $\Gamma$ form a group with respect to the usual operation for composing two transformations, i.e., $\gamma_1\gamma_2(\mathbf{x}) = \gamma_1\left(\gamma_2(\mathbf{x})\right)$.
- The group operations $(\gamma_1, \gamma_2) \to \gamma_1\gamma_2$ and $\gamma \to \gamma^{-1}$ are continuous functions.

### 5.3.7 Haar Measures

For any measurable subset $B \subset \Gamma$ and element $\gamma_0 \in \Gamma$, the action of $\gamma_0$ on $B$ defines another subset of $\Gamma$. A measure $L$ is called a **left-(invariant) Haar measure** if for every $\gamma_0$ and measurable set $B \in \Gamma$, the following holds:

$$L(B) = \int_B L(d\gamma) = \int_{\gamma_0 B} L(d\gamma) = L\left(\gamma_0 B\right).$$

Similarly, a **right-Haar measure** can be defined. Under mild conditions, these measures exist and are unique up to a positive multiplicative constant.

### 5.3.8 Transformation Groups and Gibbs Sampling

Generally, any move from $\mathbf{x}$ to $\mathbf{x}'$ in the sample space $\mathcal{X}$ can be achieved by a transformation $\gamma$ chosen from a suitable group $\Gamma$. For instance, if $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{x}' = (x_1', x_2, \ldots, x_d)$, the move can be accomplished by a translation group $\Gamma$ acting on $\mathbf{x}$ as follows:

$$\Gamma = \{\gamma \in \mathbb{R} : \gamma(\mathbf{x}) = (x_1 + \gamma, x_2, \ldots, x_d)\}.$$

An appropriate sampling distribution for $\gamma$ leads to the Gibbs sampling update of $x_1$ to $x_1'$. To construct a complete Gibbs sampling chain, one must use a combination of translation groups, one for each coordinate of $\mathbf{x}$. If this combination is transitive, the resulting Markov chain is irreducible.

### 5.3.9 Invariant Sampling Distribution

Given a group $\Gamma$, it is crucial to determine an appropriate sampling distribution for $\gamma$ to ensure that $\pi$ remains invariant under the transformation $\mathbf{x}' = \gamma(\mathbf{x})$. The theorem below provides a specific form for the sampling distribution.

### 5.3.10 Generalized Gibbs Sampling

**Theorem 2: Generalized Gibbs Sampling** Let $\Gamma$ be a locally compact group of transformations acting on the sample space $\mathcal{X}$, and let $L$ denote its left-invariant Haar measure. Let $\pi$ be an arbitrary probability measure defined on $\mathcal{X}$. Assume $\mathbf{x} \sim \pi(\mathbf{x})$, and let $\gamma$ be sampled from $\Gamma$ with the probability distribution

$$p_{\mathbf{x}}(\gamma) \propto \pi(\gamma \cdot \mathbf{x}) \, |J_\gamma(\mathbf{x})| \, L(d\gamma),$$

where $J_\gamma(\mathbf{x}) = \det\left(\frac{\partial \gamma \cdot \mathbf{x}}{\partial \mathbf{x}}\right)$ is the Jacobian determinant of the transformation $\gamma$. Then, the transformed sample $\mathbf{x}' = \gamma \cdot \mathbf{x}$ is distributed according to $\pi$. The theorem is proved by Liu and Sabatti (2000).

The standard Gibbs sampler can be obtained as a special case by selecting $\Gamma$ to be the group corresponding to translations along each coordinate axis. An immediate extension of this standard sampler involves choosing $\Gamma$ to be the group of translations in an arbitrary direction, characterized by

$$\Gamma = \{\gamma \in \mathbb{R} : \gamma(\mathbf{x}) = \mathbf{x} + \gamma \boldsymbol{e} \equiv (x_1 + \gamma e_1, \ldots, x_d + \gamma e_d)\},$$

where $\boldsymbol{e} = (e_1, \ldots, e_d)$ is a predetermined unit vector. The appropriate distribution for sampling $\gamma$ is then derived from Theorem 2 as $p_{\mathbf{x}}(\gamma) \propto \pi(\mathbf{x} + \gamma \boldsymbol{e})$.

For any locally compact transformation group $\Gamma$, a generalized Gibbs step can be delineated as follows:

1. Draw $\gamma \in \Gamma$ from the distribution $p_{\mathbf{x}}(\gamma) \propto \pi(\gamma \cdot \mathbf{x}) \, |J_\gamma(\mathbf{x})| \, L(d\gamma)$;
2. Update the state to $\mathbf{x}' = \gamma \cdot \mathbf{x}$.

The key insight provided by Theorem 2 is that the action of $\Gamma$ on $\mathbf{x}$ allows us to transition from one point to another along the orbit in a manner that is consistent with the target probability distribution $\pi$. This transition is achieved by sampling

a transformation $\gamma \in \Gamma$ according to the derived distribution, which inherently accounts for the changes in the probability measure induced by the transformation.

In essence, Theorem 2 provides a mathematical framework that allows for the simulation of a Gibbs sampling scheme where the reparameterization is implicitly defined by the group action. This approach circumvents the need for an explicit reparameterization, offering a flexible and generalizable method for constructing Markov Chain Monte Carlo (MCMC) algorithms that can efficiently explore the sample space according to the desired distribution $\pi$.

### 5.3.11 Scale-Transformation Group in Gibbs Sampling

Within the framework of Gibbs sampling, the selection of an appropriate transformation group is crucial for the efficiency and versatility of the sampling algorithm. The scale transformation group, alongside the affine and orthonormal transformation groups, stands out as a particularly potent choice for updating sample points in a high-dimensional space. These groups offer a structured approach to modifying the sample space, which can enhance the exploration capabilities of the Gibbs sampler.

To elaborate on the scale transformation group, consider the operation that updates a sample point $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ to a new point $\mathbf{x}' = (\gamma x_1, \ldots, \gamma x_d)$, where $\gamma \in \mathbb{R} \setminus \{0\}$ is a scaling factor. This operation can be formalized by defining the scale-transformation group $\Gamma$ as the set of all scaling transformations:

$$\Gamma = \left\{ \gamma\mathbf{x} : \gamma \in \mathbb{R} \setminus \{0\}, \mathbf{x} \in \mathbb{R}^d \right\},$$

where $\gamma\mathbf{x}$ represents the component-wise scaling of the vector $\mathbf{x}$ by the factor $\gamma$.

The choice of the distribution from which to sample the scaling factor $\gamma$ is pivotal. In the context of a generalized Gibbs sampling algorithm, $\gamma$ should be drawn from a distribution that is proportional to $|\gamma|^{d-1}\pi(\gamma\mathbf{x})$. This choice ensures that the scaling operation respects the target probability distribution $\pi$ and maintains detailed balance, which is a necessary condition for the Markov chain to be stationary and converge to the desired distribution.

The proportionality to $|\gamma|^{d-1}$ accounts for the change in the volume element under the scaling transformation, which is essential for preserving the invariant measure of the Markov chain. Specifically, when $\gamma$ scales the sample point $\mathbf{x}$, the Jacobian of the transformation contributes a factor of $|\gamma|^d$. By sampling $\gamma$ in proportion to $|\gamma|^{d-1}$, we effectively cancel out the $|\gamma|^d$ term, ensuring that the overall transition probability is invariant to the scaling of $\mathbf{x}$.

This approach to sampling the scaling factor $\gamma$ is a key component of the generalized Gibbs sampling algorithm. It allows for a controlled exploration of the sample space by scaling the current sample point, which can lead to more efficient mixing and faster convergence to the target distribution $\pi$. The scale-transformation group $\Gamma$, therefore, provides a mathematically principled and computationally feasible strategy for updating sample points in Gibbs sampling schemes.

### 5.3.12 Transformation-Invariant Markov Transition Function

**Theorem 3. Transformation-Invariant Markov Transition Function** Suppose $A_{\mathbf{x}}(\gamma, \gamma')\, L(d\gamma)$ is a Markov transition function that leaves the distribution

$$p_{\mathbf{x}}(\gamma)d\gamma \propto \pi(\gamma(\mathbf{x}))\,|J_{\gamma}(\mathbf{x})|\, L(d\gamma),$$

invariant and satisfies the following transformation-invariant property:

$$A_{\mathbf{x}}(\gamma, \gamma') = A_{\gamma_0^{-1}\mathbf{x}}(\gamma\gamma_0, \gamma'\gamma_0),$$

for all $\gamma, \gamma', \gamma_0 \in \Gamma$. If $\mathbf{x} \sim \pi$ and $\gamma \sim A_{\mathbf{x}}(\gamma_{\mathrm{id}}, \gamma)$, then $w = \gamma(\mathbf{x})$ follows $\pi$.

The inference is straightforward that the local transition function $A_{\mathbf{x}}$ ought to remain unaffected by the choice of reference point $\mathbf{x}$ on the group's "orbit" $\{y : y = \gamma(\mathbf{x}), \gamma \in \Gamma\}$. This stipulation is fulfilled if $A_{\mathbf{x}}(\gamma, \gamma')$ takes the form $g\{p_{\mathbf{x}}(\gamma), p_{\mathbf{x}}(\gamma')\}$.

*Proof.* We want to show that if $A_{\mathbf{x}}(\gamma, \gamma')$ is of the form $g\{p_{\mathbf{x}}(\gamma), p_{\mathbf{x}}(\gamma')\}$, then the transformation-invariant property holds.

Let $\mathbf{x} \sim \pi$ and let $\gamma, \gamma', \gamma_0 \in \Gamma$ be arbitrary transformations. We have the following equality by the definition of $A_{\mathbf{x}}$:

$$\begin{aligned} A_{\mathbf{x}}(\gamma, \gamma') &= g\{p_{\mathbf{x}}(\gamma), p_{\mathbf{x}}(\gamma')\} \\ &= g\left\{p_{\gamma_0^{-1}\mathbf{x}}(\gamma\gamma_0), p_{\gamma_0^{-1}\mathbf{x}}(\gamma'\gamma_0)\right\} \\ &= A_{\gamma_0^{-1}\mathbf{x}}(\gamma\gamma_0, \gamma'\gamma_0), \end{aligned}$$

where the last equality follows from the property of the function $g$ being dependent only on the probability densities evaluated at the transformations.

This shows that $A_{\mathbf{x}}(\gamma, \gamma')$ is invariant under the transformation of the reference point $\mathbf{x}$, which completes the proof. □

## 5.4 Parameter Expansion

The primary objective of parameter expansion is to construct an augmented model that encompasses the original model while incorporating the extra parameter $\alpha$. This expanded model is given by $p(\mathbf{y}_{obs}, \mathbf{w} \mid \theta, \alpha)$, where $\mathbf{y}_{obs}$ represents the observed data, $\mathbf{w}$ denotes the missing data, and $\theta$ is the parameter of interest. The parameter $\alpha$ is termed as the expansion parameter because it expands the scope of the model to include additional variation or structure that is not captured by $\theta$ alone.

The choice of $\alpha$ is strategic, as it allows the original model to be seamlessly embedded within the broader framework of the expanded model. This is achieved by ensuring that the marginal distribution of the observed data, when integrated over the missing data $\mathbf{w}$, remains unchanged. Mathematically, this is expressed as:

$$\int p(\mathbf{y}_{obs}, \mathbf{w} \mid \theta, \alpha)\, d\mathbf{w} = f(\mathbf{y}_{obs} \mid \theta).$$

This integral demonstrates that the observed-data model $f(\mathbf{y}_{obs} \mid \theta)$ is preserved even when the expanded model $p(\mathbf{y}_{obs}, \mathbf{w} \mid \theta, \alpha)$ is considered.

The introduction of $\alpha$ can lead to more efficient Markov Chain Monte Carlo (MCMC) algorithms by allowing for more effective exploration of the posterior distribution. It can also improve the mixing properties of the MCMC, leading to better convergence and more accurate inference.

To facilitate the explanation, we use the notation $\mathbf{w}$ to represent the missing data $\mathbf{y}_{mis}$ under the expanded model. This is done to differentiate the missing data in the expanded model from that in the original model. To apply the data augmentation algorithm to the expanded model, we must define a joint prior distribution $p(\theta, \alpha)$ for the parameters $\theta$ and $\alpha$.

### 5.4.1 Consistency of Posterior Distributions

The posterior distribution for the parameter $\theta$ should remain consistent when moving from the original model to the expanded model. This consistency is achieved if the marginal prior for $\theta$ from the joint prior $p(\theta, \alpha)$ matches the prior $f(\theta)$ of the original model. This condition can be formally stated as:

$$\int p(\theta, \alpha) d\alpha = f(\theta).$$

This integral ensures that the marginalization over $\alpha$ yields the original prior for $\theta$. Given the condition above, we can focus on specifying the conditional prior $p(\alpha|\theta)$ while keeping the marginal prior for $\theta$ unchanged at $f(\theta)$. It is important to note that the posterior distribution of $\alpha$ given the observed data $\mathbf{y}_{obs}$ and $\theta$ remains $p(\alpha|\theta)$, as $\alpha$ cannot be directly inferred from the observed data alone.

### 5.4.2 Transformation Group and Expanded Likelihood

In many practical scenarios, based on Theorem 1, the parameter $\alpha$ corresponds to an element of a transformation group, which is applied to the missing data $\mathbf{y}_{mis}$. This transformation allows for more global exploration of the missing data space, leading to a more efficient algorithm. The expanded likelihood, which incorporates the transformation indexed by $\alpha$, is given by:

$$p\left(\mathbf{y}_{obs}, \mathbf{w} \mid \alpha, \theta\right) = f\left(\mathbf{y}_{obs}, t_\alpha(\mathbf{w}) \mid \theta\right) \left|J_\alpha(\mathbf{w})\right|,$$

where $t_\alpha(\mathbf{w})$ represents the transformation applied to the missing data $\mathbf{w}$, and $J_\alpha(\mathbf{w})$ is the Jacobian of this transformation.

## The PX-DA Algorithm

The Parameter Expanded Data Augmentation (PX-DA) algorithm is an iterative procedure that incorporates the expansion parameter $\alpha$. The algorithm is outlined below:

1. Draw $\mathbf{y}_{mis} \sim f\left(\mathbf{y}_{mis} \mid \theta, \mathbf{y}_{obs}\right)$.
2. Draw $\alpha \sim p\left(\alpha \mid \mathbf{y}_{obs}, \mathbf{y}_{mis}\right) \propto f\left(\mathbf{y}_{obs}, t_\alpha\left(\mathbf{y}_{mis}\right)\right) \left|J_\alpha\left(\mathbf{y}_{mis}\right)\right| H(d\alpha)$. Compute $\mathbf{y}'_{mis} = t_\alpha\left(\mathbf{y}_{mis}\right)$.

3. Draw $\theta \sim f\left(\theta \mid \mathbf{y}_{obs}, \mathbf{y}'_{mis}\right)$.

Step 2 in the PX-DA algorithm effectively adjusts the missing data. When the prior for $\alpha$ is proper, Liu and Wu (1999) demonstrated that the second step should take the form of

1. Draw $\alpha_0 \sim p(\alpha)$; compute $\mathbf{w} = t_\alpha^{-1}\left(\mathbf{y}_{mis}\right)$.
2. Draw $\alpha_1 \sim p\left(\alpha \mid \mathbf{y}_{obs}, \mathbf{w}\right) \propto f\left(\mathbf{y}_{obs}, t_\alpha(\mathbf{w})\right) \left|J_\alpha(\mathbf{w})\right| p_0(\alpha)$. Compute $\mathbf{y}'_{mis} = t_{\alpha_1}\left(t_{\alpha_0}^{-1}\left(\mathbf{y}_{mis}\right)\right)$.

The second step of the PX-DA algorithm involves an adjustment of the missing data based on the transformation indexed by $\alpha$. When a proper prior for $\alpha$ is used, this step ensures that the distribution of the adjusted missing data is consistent with the expanded likelihood.

When a Haar measure prior is used for $\alpha$, the step of sampling from the prior can be skipped, which is particularly advantageous when the Haar prior is improper. This simplification can lead to more efficient implementation of the algorithm.

### 5.4.3 Invariance of the PX-DA Algorithm

Based on Theorem 2, Step 2 of the Parameter Expanded Data Augmentation (PX-DA) algorithm is designed to leave the joint distribution of the observed and missing data, $f\left(\mathbf{y}_{obs}, \mathbf{y}_{mis}\right)$, invariant. This means that the probability of the observed data and the distribution of the missing data remain unchanged after the execution of Step 2. As a direct consequence, the entire PX-DA algorithm maintains the invariance of the posterior distribution, which is a critical property for ensuring that the algorithm produces valid posterior samples.

Liu and Wu (1999) demonstrated a particularly advantageous property of using a Haar prior for the expansion parameter $\alpha$. When the Haar prior is employed, the adjustment of the missing data, denoted as $\mathbf{y}'_{mis}$ in Step 2, is conditionally independent of the previous value of the missing data, $\mathbf{y}_{mis}$, given that both values lie on the same orbit (or fiber) of the transformation group. This conditional independence simplifies the sampling process and can lead to more efficient exploration of the posterior distribution.

The conditional independence property implies that the new value of the missing data, $\mathbf{y}'_{mis}$, is determined solely by the current state of the observed data, $\mathbf{y}_{obs}$, and the current value of the expansion parameter, $\alpha$. This decoupling of the new missing data value from its previous value reduces the correlation between successive samples and can improve the mixing of the Markov chain, leading to more accurate posterior estimates.

Given that Steps 2 and 2′ can be implemented with equal computational cost, it is generally more preferable to use Step 2 over Step 2′ due to the aforementioned benefits. The use of the Haar prior for $\alpha$ thus enhances the efficiency and effectiveness of the PX-DA algorithm, making it a more attractive option for Bayesian inference with missing data.

## 5.5 Simulation Studies: Probit Regression

Let $\mathbf{y} = (y_1, \ldots, y_n)$ be a set of independent and identically distributed (i.i.d.) binary observations drawn from a probit regression model. The $i$-th observation $y_i$ is modeled as

$$y_i \mid \theta \sim \text{Bernoulli}\left\{\Phi\left(X_i'\theta\right)\right\},$$

where $X_i \in \mathbb{R}^p$ represents the vector of covariates for the $i$-th observation, $\theta \in \mathbb{R}^p$ is the vector of unknown regression coefficients, and $\Phi$ denotes the standard Gaussian cumulative distribution function (CDF). The focus of the analysis is on the posterior distribution of the parameter vector $\theta$, which is assumed to have a flat prior, indicating a non-informative and uniform belief about its possible values before observing the data.

To facilitate the computation of the posterior distribution, a "complete-data" model is introduced, which augments the observed data with a set of latent variables, denoted as $z_1, \ldots, z_n$. The joint distribution of the latent variables and the observed binary outcomes is given by

$$[z_i \mid \theta] \sim \text{N}\left(X_i'\theta, 1\right),$$

$$y_i = \text{sgn}\left(z_i\right),$$

where $\text{sgn}(z)$ is the sign function, taking the value 1 if $z > 0$ and 0 otherwise (Albert and Chib, 1993). This formulation allows for a more straightforward application of the Data Augmentation (DA) algorithm, which is an iterative method for sampling from complex distributions.

The standard DA algorithm proceeds as follows:

1. For each $i$, draw $z_i \sim \text{N}\left(X_i'\theta, 1\right)$ conditioned on the observed $y_i$. Specifically, if $y_i = 1$, $z_i$ must be drawn such that $z_i \geq 0$; if $y_i = 0$, $z_i$ must be drawn such that $z_i < 0$.

2. Draw $\theta \sim \text{N}\left(\hat{\theta}, V\right)$, where $\hat{\theta}$ is the sample mean of the latent variables weighted by the corresponding covariates, and $V$ is the sample covariance matrix of the latent variables, computed using the Sweep operator (Little and Rubin, 2019).

One limitation of this approach is the strong correlation between the scales of the latent variables $z_i$ and the parameter vector $\theta$. The center of the distribution of $\theta$ given the latent variables is a weighted average of the $z_i$, while the center of the distribution of $z_i$ given $\theta$ is $X_i'\theta$. To address this issue, a parameter-expansion approach is considered.

The parameter-expansion approach introduces an expansion parameter $\alpha$, which modifies the distribution of the latent variables to

$$[w_i \mid \theta] \sim \text{N}\left(X_i'\theta\alpha, \alpha^2\right),$$
$$y_i = \text{sgn}\left(w_i\right),$$

where $w_i$ is a transformed latent variable, and $\alpha$ represents the variance of the residuals. This transformation allows for a more flexible exploration of the parameter space.

The Parameter-Expanded Data Augmentation (PX-DA) algorithm, detailed in Section 5.4, follows the same initial step as the standard DA but modifies the subsequent steps as follows:

1. Draw $\hat{\alpha}^2 \sim \frac{\text{RSS}}{\chi^2_n}$, where RSS is the residual sum of squares calculated as $\sum_i \left(z_i - X_i'\hat{\theta}\right)^2$.

2. Draw $\theta \sim \text{N}\left(\frac{\hat{\theta}}{\hat{\alpha}}, V\right)$, where $\hat{\theta}$ and $V$ are updated to reflect the influence of the expansion parameter $\alpha$.

The PX-DA scheme can also be interpreted more abstractly as a method for sampling from the target distribution $\pi(\theta, \mathbf{z})$. This is achieved through an iterative process that includes the following steps:

- Draw $\theta$ from its conditional distribution $\pi(\theta \mid \mathbf{z})$.
- Update each latent variable $z_i$ using samples from its conditional distribution $\pi\left(z_i \mid \mathbf{z}_{[-i]}, \theta\right)$, for $i = 1, \ldots, n$.
- Draw a new value for the scaling parameter $\gamma$ from its conditional distribution $p(\gamma) \propto \gamma^{n-1}\pi(\gamma\mathbf{z})$, and adjust the latent variables according to this new scale.

This approach provides a more flexible and robust framework for Bayesian inference in probit regression models, allowing for more efficient exploration of the posterior distribution and better handling of the latent data augmentation process.

## 5.6 Implementation and Comparison of DA and PX-DA on Probit Regression Model

Consider a probit regression model: For $i = 1, \ldots, n$, given a vector of covariates $x_i \in \mathbb{R}^p$, we observe the following probability distribution for the binary outcome $Y_i$:

$$Y_i \sim \text{Bernoulli}(\Phi(x_i^T\beta)).$$

Here, $\beta \in \mathbb{R}^p$ represents the vector of coefficients, and $\Phi$ denotes the cumulative distribution function of the standard normal distribution.

### 5.6.1 DA based Sampler

Consider augmented data $Y_{aug} = ((Y_i, \phi_i))_{i=1}^n$, where $\phi_i \sim \text{N}(x_i^T\beta)$ represents a latent variable. We only observe $Y_i = 1_{[\phi>0]}$, which is an indicator function that equals 1 if $\phi_i > 0$ and 0 otherwise. A non-informative prior is imposed on $\beta$, such that $p(\beta) \propto 1$. To implement the Gibbs sampler, we need to derive the full conditional

distributions for $\beta$ and $\phi = \{\phi_{\mathbf{1}}, \ldots, \phi_{\mathbf{n}}\}$. For the full conditional distribution of $\beta$, given the noninformative prior $p(\beta) \propto 1$, we have

$$\begin{aligned} P(\beta \mid \mathbf{Y}) &\propto P(\mathbf{Y} \mid \beta) P(\beta) \\ &\propto \exp\left[-\frac{1}{2}(X\beta - \phi)^T (X\beta - \phi)\right] \\ &\propto \exp\left[-\frac{1}{2}\left(\beta^\top X^\top X\beta - \beta^\top X^\top \phi - \phi^\top \mathbf{X}\beta + \phi^\top \phi\right)\right] \\ &\propto \exp\left[-\frac{1}{2}\left(\beta^\top X^\top X\beta - 2\beta^\top X^\top \phi\right)\right], \end{aligned}$$

which is the kernel of the normal distribution. Thus, we have the full conditional distributions of $\beta$ as

$$\begin{aligned} P(\beta \mid Y) &\sim N\left(\beta^*, \Sigma\right), \\ \text{where } \beta^* &= \left(X^T X\right)^{-1} X^T \phi \\ \Sigma &= \left(X^T X\right)^{-1}. \end{aligned}$$

The full conditional distribution for $\phi_i$ is

$$P\left(\phi_i \mid \beta, Y\right) \begin{cases} \frac{N\left(x_i^\top \beta, 1\right)}{1 - \Phi\left(x_i^\top \beta\right)} & \text{if } Y_i = 1 \\ \frac{N\left(x_i^\top \beta, 1\right)}{\Phi\left(x_i^\top \beta\right)} & \text{if } Y_i = 0 \end{cases}$$

where $\Phi\left(x_i^\top \beta\right) = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}} \exp(-\frac{\left(z - x_i^\top \beta\right)^2}{2}) dz$.

So the Gibbs sampler is implemented as below.

**Step 1:** Set the initial value for $\beta^{(0)}$ and $\phi^{(0)}$ by MLE estimation from $glm()$

**Step 2:** From the step $t = 1$ to $n$ :

Simulate $\beta^{(t)}$ from the full conditional distribution (1) given $\phi^{(\mathbf{t-1})}$

Simulate $\phi$ from the full conditional distribution (2) given $\beta^{(t)}$

**Step 3:** Discard the first $n_0$ steps as the burn-in period and use the rest $\left\{\beta^{(t)}, \phi^{(\mathbf{t})}\right\}_{t=n_0+1}^{n}$ to construct the posterior distributions.

### 5.6.2 PX-DA based Sampler

Consider "parameter-expanded" augmented data $Y_{\text{aug}} = ((Y_i, s_i))_{i=1}^n$, where $s_i = x_\sigma$ and $\sigma$ is given an improper prior $p(\sigma^2)$ proportional to $\frac{1}{\sigma^2}$. Implement a Gibbs sampler to sample from the posterior distribution of each parameter, taking into account the expansion parameter $\sigma$. Given $\zeta_i = \sigma\phi_i$, by replacing $\phi_i$ with $\frac{\zeta_i}{\sigma}$, we can get the full conditional distribution of $\beta$ as

$$\begin{aligned} P(\beta \mid Y, \zeta, \sigma) &\sim N\left(\frac{\beta^*}{\sigma}, \Sigma\right), \\ \text{where } \beta^* &= \left(X^T X\right)^{-1} X^T \zeta \\ \Sigma &= \left(X^T X\right)^{-1}. \end{aligned}$$

And we can also get the full conditional distribution of $\zeta$ as

$$P\left(\zeta_i \mid \beta, \sigma, Y\right) \begin{cases} \frac{N\left(x_i^\top \sigma\beta, \sigma^2\right)}{1-\Phi\left(x_i^\top \sigma\beta, \sigma^2\right)} & \text{if } Y_i = 1 \\ \frac{N\left(x_i^\top \beta, 1\right)}{\Phi\left(x_i^\top \sigma\beta, \sigma^2\right)} & \text{if } Y_i = 0 \end{cases}$$

where $\Phi\left(x_i^\top \sigma\beta, \sigma^2\right) = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(z - x_i^\top \sigma\beta\right)^2}{2\sigma^2}\right) dz$.

Given

$$\frac{(\zeta - \mathbf{X}\beta)^T(\zeta - \mathbf{X}\beta)}{\sigma^2} \sim \chi_n^2.$$

The full conditional distribution of $\sigma^2$ is

$$P\left(\sigma^2 \mid \beta, \zeta, \mathbf{Y}\right) \sim \frac{(\zeta - \mathbf{X}\beta)^\top(\zeta - \mathbf{X}\beta)}{\chi^{\mathbf{2}}_{\mathbf{n}}}.$$

So the Gibbs sampler is implemented as below.

**Step 1:** Set the initial value for $\beta^{(0)}, \zeta^{(\mathbf{0})}$, and $\sigma^2$ by MLE estimation from $glm()$

**Step 2:** From the step $t = 1$ to $n$ :

Sample $\beta^{(t)}$ from the full conditional distribution (3) given $\zeta^{(\mathbf{t-1})}$

Sample $\zeta_i$ from the full conditional distribution (4) given $\beta^{(t)}$

Sample $\sigma^2$ from the full conditional distribution (5) given $\zeta^{(\mathbf{t})}$ and $\beta^{(t)}$

**Step 3:** Discard the first $n_0$ steps as the burn-in period and use the rest $\left\{\beta^{(t)}, \zeta^{(\mathbf{t})}, \sigma^{(\mathbf{t})\mathbf{2}}\right\}_{t=n_0+1}^{n}$ to construct the posterior distributions.

### 5.6.3 Comparison of the performance between two samplers

In this section, we evaluate the performance of the two sampling methods through visual and statistical diagnostics. We run Gibbs Sampler $n = 10000$ steps with the first half $n_0 = 5000$ steps as the burn-in period. Initially, we visualize the density of the posterior distribution for $\beta$. Each graph represents an individual parameter. The red curve illustrates the posterior distributions generated by DA based sampler (Sampler 1(a)), while the blue curve represents those produced by PX-DA based sampler (Sampler 1(b)). Additionally, we include the orange curve, displaying the posterior distribution of $\beta$ obtained via JAGS, where we apply Bayesian probit regression with a non-informative prior for $\beta$.

The plot shows that the posterior distribution derived from the PX-DA-based sampler closely aligns with the one obtained from JAGS, while the posterior from the DA based sampler exhibits a noticeable deviation from the other two. This discrepancy arises due to the linkage between the scale of $\beta$ and $\phi$ in the DA based sampler. The introduction of the additional parameter $\sigma$ liberates the scale of $\beta$ from dependence on the scales of $\phi$ or $\zeta$. Therefore, the posterior distributions from sampler PX-DA based sampler and JAGS exhibit a congruent alignment.

A similar pattern emerges in the summary statistics, where the 95% credit interval and the standard error from sampler 1(b) are close to JAGS but differ from sampler 1(a).

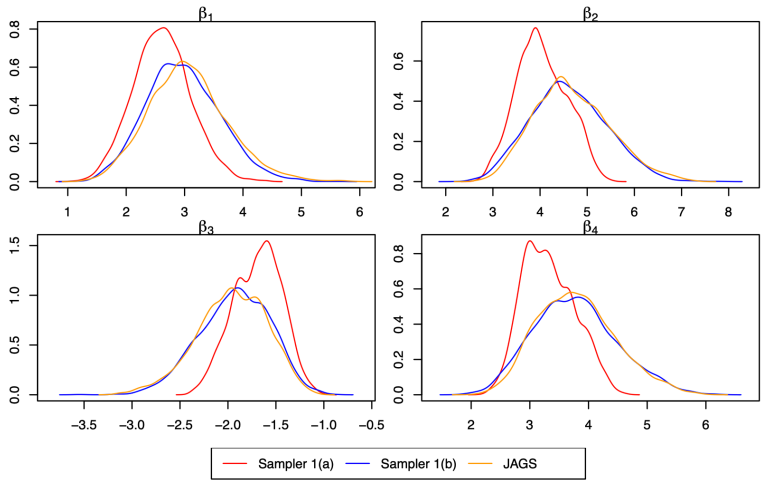


Figure 5: Comparison of distributions

| | 2.5% | 50% | 97.5% | Posterior SE |
|---|---|---|---|---|
| **Sampler 1a** | | | | |
| beta[0] | 1.672 | 2.602 | 3.619 | 0.492 |
| beta[1] | 3.048 | 3.991 | 5.061 | 0.532 |
| beta[2] | -2.192 | -1.674 | -1.241 | 0.252 |
| beta[3] | 2.575 | 3.285 | 4.231 | 0.442 |
| beta[4] | 0.535 | 0.841 | 1.182 | 0.166 |
| **Sampler 1b** | | | | |
| beta[0] | 1.811 | 2.944 | 4.292 | 0.631 |
| beta[1] | 3.133 | 4.563 | 6.234 | 0.806 |
| beta[2] | -2.674 | -1.911 | -1.285 | 0.366 |
| beta[3] | 2.513 | 3.746 | 5.249 | 0.708 |
| beta[4] | 0.567 | 0.952 | 1.406 | 0.219 |
| **JAGS** | | | | |
| beta[0] | 1.842 | 3.021 | 4.526 | 0.674 |
| beta[1] | 3.277 | 4.608 | 6.428 | 0.807 |
| beta[2] | -2.735 | -1.945 | -1.323 | 0.364 |
| beta[3] | 2.640 | 3.755 | 5.244 | 0.663 |
| beta[4] | 0.585 | 0.954 | 1.421 | 0.213 |

Table 2: Summary Statistics

**Convergence Diagnostics** To perform convergence diagnostics, we simulate 5 chains with different initial values. After discarding the first half as a warm-up, we have $L$ steps in each chain. Then we split each into two parts. We now have $m = 10$ chains, each with $n = L/2$ steps. Then the between and within-sequence variances

are calculated as

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\psi}_{.j} - \bar{\psi}_{..}), \quad \text{where } \bar{\psi}_{.j} = \frac{1}{n}\sum_{i=1}^{n}\psi_{ij},\ \bar{\psi}_{..} = \frac{1}{m}\sum_{j=1}^{m}\bar{\psi}_{.j},$$

$$W = \frac{1}{m}\sum_{j=1}^{m}s_j^2, \quad \text{where } s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\psi_{ij} - \bar{\psi}_{.j})^2.$$

The marginal posterior variance is calculated based on the weighted average of $B$ and $W$

$$\widehat{\text{var}}^{+}(\psi \mid y) = \frac{n-1}{n}W + \frac{1}{n}B.$$

Given this quantity overestimates the marginal posterior variance, we calculate the potential scale reduction index by

$$\hat{R} = \sqrt{\frac{\hat{\text{var}}^{+}(\psi \mid y)}{W}}.$$

We monitor this index as we increase the number of steps $n$ to determine whether the posterior sample convergences. The result is plotted as below. The potential scale reduction is on the y-axis and the number of steps is on the x-axis. Each line represents a parameter in the model. The red dashed represents the threshold value 1.2. As we increase the number of steps, both model 1(a) and model 1(b) show convergence as the potential scale reduction falls below 1.2. Furthermore, model 1(b) shows a faster convergence at the earlier step than model 1(a).

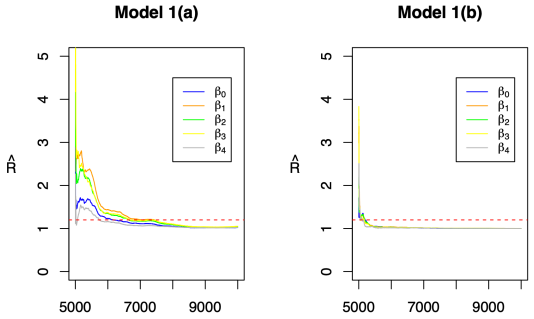


Figure 6: Convergence of posterior samples

We can also check the convergence by plotting the posterior samples from different chains. For model 1(a), each $\beta_i$ shows the convergence with the chains (in different colors) mixed up.

A similar pattern is observed in model 1(b) as each $\beta_i$ shows the convergence with the chains (in different colors) mixed up.

Since the simulated sequences have mixed, we can also compute an approximate effective number of independent simulation draws' for the estimated $\beta$. To calculate the effective sample size, we estimate the correlations by computing the variogram $V_t$ at each lag $t$:

$$V_t = \frac{1}{m(n-t)}\sum_{j=1}^{m}\sum_{i=t+1}^{n}(\psi_{i,j} - \psi_{i-t,j})^2.$$

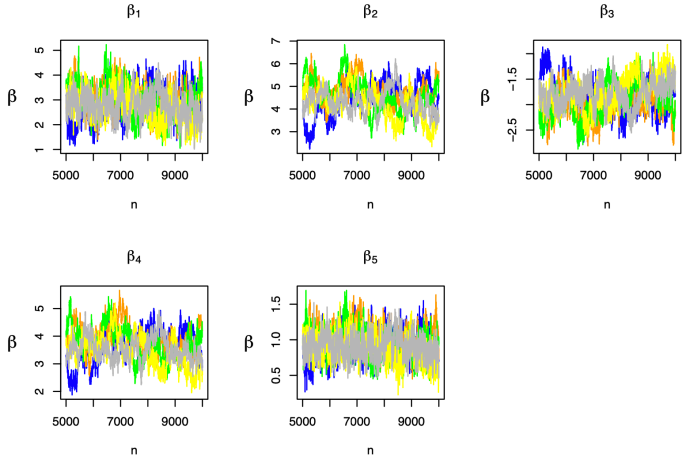


Figure 7: Posterior samples from different chains

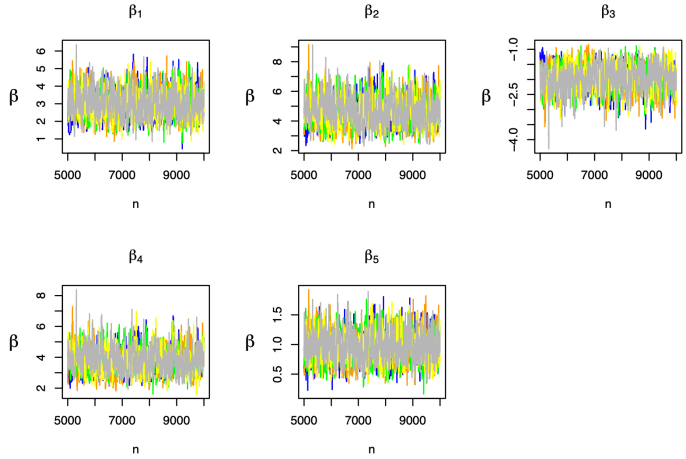


Figure 8: Posterior samples from mixed up chains

Then the correlation can be estimated by

$$\hat{\rho}_t = 1 - \frac{V_t}{2\widehat{\text{var}}^+},$$

where $\widehat{\text{var}}^+$ is the marginal posterior variance calculated in the previous section. Then the effective sample size is calculated by

$$\hat{n}_{eff} = \frac{mn}{1 + 2\sum_{t=1}^{T} \hat{\rho}_t},$$

where $T$ is the odd postive integer for which $\rho^T + 1 + \rho^T + 1$ is negative. For each interest parameter $\beta$, the effective sample size is calculated as below

We can also plot the ACF of the $\beta$ posterior samples.

From the ACF plot, we do not observe a fairly steep decline as the lag increases. This suggests that samples are likely to be autocorrelated instead of being random draws from a posterior distribution. Thus, we would like to add a thin rate in selecting the posterior samples. Based on the effective sample size, for model 1(a), we select every 20th value from the original posterior distribution. For model 1(b),

| | beta[0] | beta[1] | beta[2] | beta[3] | beta[4] |
|---|---|---|---|---|---|
| Model 1a | 49 | 28 | 32 | 30 | 64 |
| Model 1b | 876 | 721 | 759 | 736 | 990 |

Table 3: Effective Sample Size

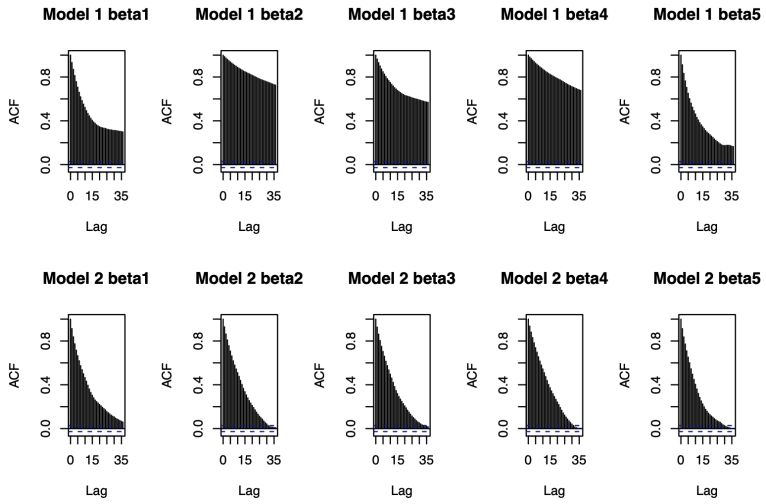


Figure 9: ACF of the $\beta$ posterior samples

we select every 5th value from the original posterior distribution. We plot the ACF again as below.

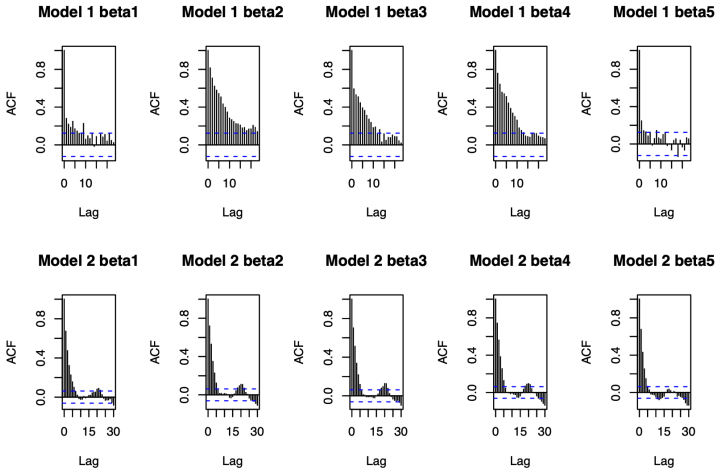


Figure 10: ACF of the $\beta$ posterior samples

In this case, we observe a fairly steep decline as lag increases, which means that samples are likely to be random draws from the posterior distribution. We update the result in Table 4.

| | 2.5% | 50% | 97.5% | Posterior SE |
|---|---|---|---|---|
| **Sampler 1a** | | | | |
| beta[0] | 1.656 | 2.605 | 3.562 | 0.500 |
| beta[1] | 3.074 | 3.993 | 4.996 | 0.529 |
| beta[2] | -2.169 | -1.666 | -1.212 | 0.252 |
| beta[3] | 2.626 | 3.281 | 4.198 | 0.447 |
| beta[4] | 0.531 | 0.856 | 1.164 | 0.162 |
| **Sampler 1b** | | | | |
| beta[0] | 1.791 | 2.933 | 4.272 | 0.634 |
| beta[1] | 3.131 | 4.582 | 6.271 | 0.814 |
| beta[2] | -2.665 | -1.916 | -1.297 | 0.368 |
| beta[3] | 2.520 | 3.755 | 5.266 | 0.716 |
| beta[4] | 0.542 | 0.952 | 1.389 | 0.224 |

Table 4: Summary Statistics

# 6 Conclusion

This report presents a comprehensive exploration into the augmentation of Markov Chain Monte Carlo (MCMC) methods through Data Augmentation (DA) and Parameter Expansion (PE) techniques. Our investigation is grounded in the challenges faced when applying Bayesian inference to complex, high-dimensional models, particularly those involving missing data. We have introduced the Parameter Expansion Data Augmentation (PX-DA) algorithm, leveraging the mathematical framework of left-(invariant) Haar measures on locally compact groups, to enhance the efficiency and convergence properties of the traditional DA method.

Our theoretical analysis, supported by extensive simulation studies, demonstrates the superior performance of the PX-DA algorithm. The improved mixing and faster convergence rates of the PX-DA algorithm are attributed to the strategic introduction of auxiliary parameters, which enrich the parameter space and facilitate more effective exploration of the posterior distribution. This approach has significant implications for the Bayesian analysis of complex models, offering a robust framework for handling missing data and improving the overall reliability of inferences.

The application of the DA algorithm to Bayesian curve fitting and logistic regression models exemplifies its versatility and potential for practical implementation. Through simulation studies, we have illustrated the algorithm's ability to outperform traditional methods under various conditions, showcasing its utility in advancing Bayesian computational techniques.

The PX-DA algorithm's application to Probit regression has been shown to significantly accelerate the convergence of the DA method. This enhancement is attributed to the algorithm's adeptness at efficiently navigating complex posterior landscapes, leading to faster and more reliable Bayesian inferences. Collectively, these findings underscore the PX-DA algorithm's potential to transform Bayesian statistical analysis.

The findings of this report contribute to the broader understanding of MCMC enhancements and offer a solid foundation for future research. Potential directions for future work include the extension of the PX-DA algorithm to other Bayesian models, further exploration of its performance in different settings, and the development of more sophisticated transformation groups to tailor the algorithm to specific applications.

In conclusion, our evaluation of the DA and PE techniques has demonstrated their pivotal role in advancing Bayesian statistical methods for complex model analysis. The findings suggest that these approaches could lead to substantial improvements in Bayesian inference efficiency and accuracy, with broad applications in science and industry.

# References


J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Y. Fan, J.-L. Dortet-Bernadet, and S. Sisson. On Bayesian curve fitting via auxiliary variables. *Journal of Computational and Graphical Statistics*, 19(3):626–644, 2010.

A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

R. J. Little and D. B. Rubin. *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons, 2019.

J. S. Liu and J. S. Liu. *Monte Carlo Strategies in Scientific Computing*, volume 10. New York: springer, 2001.

J. S. Liu and C. Sabatti. Generalised Gibbs sampler and multigrid monte carlo for Bayesian computation. *Biometrika*, 87(2):353–369, 2000.

J. S. Liu and Y. N. Wu. Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448):1264–1274, 1999.

N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. doi: 10.1080/01621459.1949.10483310. URL `https://www.tandfonline.com/doi/abs/10.1080/01621459.1949.10483310`. PMID: 18139350.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987. doi: 10.1080/01621459.1987.10478458. URL `https://www.tandfonline.com/doi/abs/10.1080/01621459.1987.10478458`.

D. A. Van Dyk and X.-L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.