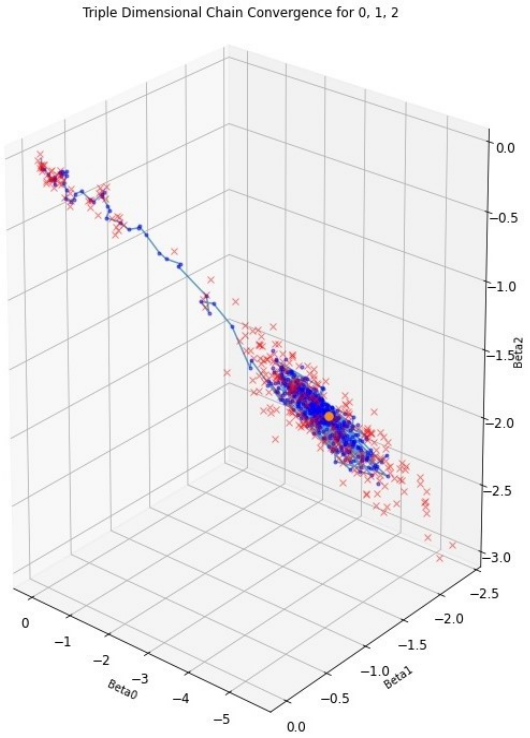
Università Bocconi
MILANO

20592 - Statistics and Probability

Computational Statistics Assignment

Master of Science in Data Science and Business Analytics



Group Members
Chiavarino Federico 3081771
Giancola Simone Maria 3074413
Liscai Dario 3080142
Marchetti Simone 3185524

January 12, 2022

Abstract

Markov Chain Monte Carlo sampling methods are efficient procedures to generate distributions sequentially. Our work proposes two ways to estimate the parameters of a Generalized Linear Model of a binary target variable linked with a probit function to the covariates. The former is a Metropolis Hastings with a Random Walk proposal algorithm. The latter method is inspired from [1], using an instrumental variable Gibbs algorithm framework. Once full conditional closed forms of the parameter are retrievable, we can sample from them. The procedure is carried out in blocks, to increase efficiency. After a theoretical introduction to the techniques, we explicitly derive characterizing features of both and provide a code to present results. Given that the distribution needs additionally to come with ergodic properties, we construct a collection of diagnostic checks to ensure this through plots. A randomly generated dataset and a real dataset [2][3] are used to graphically derive stable parameters' specifications. The two methods are then compared in terms of performance and appearance of the resulting chains.

Contents

List of Tables

List of Algorithms

List of Figures

1 Introduction

The following chapter is aimed at explaining our approach and how we drafted the rest of the document.

The task of the assignment consists in developing two algorithms to implement Bayesian techniques of estimation of a probit binary model.
We chose to present results in the most formal way possible. As a rule of thumb, if a theorem missed a proof in class or in the paper [1] we were advised to read, we tried to add it here. Any result needed was either proved or provided with sufficient reference to lectures in class or external sources as a citation (see, Theorems 16, 21, 37). Also quick and interesting theorems we encountered are present (see, Theorems 8, 31).
Easier results that are found online were skipped for the sake of simplicity (see, Theorem 12).
In addition to this, in order to obtain full points, we were required to derive a result for Theorem 37 in Section 3.2. The concept required introducing lower level ideas to provide a theory backup and elaborate an understandable proof of the statement.

For this purpose, as a first step we chose to create a common ground for notation and concepts that would help us align towards the same direction. This accounts for Section 2, where we lay basic foundations of the three concepts we will exploit: linear models, Bayesian probability, and Markov Chain Monte Carlo Methods for sampling. The three more or less add up sequentially for the purpose of the assignment.
While the first part might be useless, the aim of it is that of delivering a set of symbols and concepts that will be used throughout as to avoid confusion.

Section 3 presents an explanation of the further steps needed to implement the two methods in detail given the problem.

Section 4 outlines how the problem was translated into scripts. We used Python and Jupyter Notebooks for prototyping and plotting results.

Section 5 presents two applications, one on simulated data, and one on a real dataset, where the two algorithms are compared, offering a detailed discussion on the choice of parameters and their results.

Section 6 resumes the work done and assesses weaknesses and potential improvements.

2 Theory Background

2.1 Linear Model

We have as target variables

$$Y_1, \dots, Y_n \quad : \quad Y_i \in \{0,1\} \forall i \implies Y_i \sim \mathcal{B}(p_i)^1$$

And as covariates $X_1, \dots, X_n \ : \ X_i \in \mathbb{R}^p \forall i$.
We wish to model Y_i through the means of a probit regression model of the following form:

$$Y_i = \mathbb{E}(Y_i|X_i) + \varepsilon_i \ : \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \ \forall i \tag{1}$$

To specify the jargon used in the paper and for rigorousness, we will introduce definitions and theorems to reference across the paper.

Definition 1 (Dependent Variable Y)**.** *Another definition for the target variable Y, which is made up of independent realizations.*

$$Y = \begin{bmatrix} Y_1 \\ \dots \\ Y_n \end{bmatrix}$$

Definition 2 (Design Matrix X)**.** *To include the intercept, we construct the design matrix as the covariate matrix X stacked with a vector of ones.*

$$X = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix} \in \mathcal{M}_{n,p+1}$$

Definition 3 (Linear predictor η_i)**.** *A linear predictor is a linear combination of a covariate X_i and the β coefficient:*

$$\eta_i = X_i^T \beta = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$$

*It is often referred to as **Systematic Component** in the framework of Generalized Linear Models (GLM)*

Definition 4 (Link Function g)**.** *A link function is a differentiable and invertible function such that:*

$$g\Big(\mathbb{E}(Y_i|X_i)\Big) = g(\mu_i) = \eta_i$$

$$\implies \mu_i = g^{-1}(\eta_i)$$

Definition 5 (Probit link function)**.** *A probit link function is the inverse of the distribution of a standard normal cdf, meaning that:*

$$g(\cdot) = \Phi^{-1}(\cdot) \implies \Phi^{-1}(\mu_i) = \eta_i \implies \mu_i = \Phi(\eta_i)$$

Where:

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} exp\left(\frac{-w^2}{2}\right) dw \ \ \forall x \in \mathbb{R}$$

Observation 6. *So far, we have that as $\mu_i = p_i$ then $\forall i$:*

$$\begin{cases} \eta_i = X_i^T \beta \\ \eta_i = \Phi^{-1}(p_i) \\ \phi(x) = \dfrac{1}{\sqrt{2\pi}} exp\Big\{ -\dfrac{x^2}{2} \Big\} \ \textit{i.e. standard normal pdf} \end{cases}$$

[1] Bernoulli distribution with parameter p_i.

And we can claim that the following latent variable $Z = f(Y)$ relationship holds:

$$Y_i \ : \ Y_i = \mathbb{1}(Z_i > 0) \text{ where } Z_i \coloneqq \eta_i + \varepsilon_i = X_i^T\beta + \varepsilon_i \ : \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2 = 1) \ \forall i \tag{2}$$

Allowing us to use the following model:

$$\mathbb{P}(Y_i = 1|X_i) = \mathbb{P}(Z_i > 0|X_i) = p_i = \mathbb{P}(\varepsilon_i > -\eta_i) = \mathbb{P}(\varepsilon_i < \eta_i) = \Phi(\eta_i) \tag{3}$$

Where the parameters to estimate are $\beta = [\beta_0, \beta_1, \ldots, \beta_p] \in \mathbb{R}^{p+1}$

Definition 7 (Exponential Family of distributions $\mathcal{E}$ for W)**.** *$W \sim f(\cdot) \in \mathcal{E}$ with canonical parameter $\theta \in \mathbb{R}$ if*

$$f(w; \theta, \gamma) = exp\left\{\frac{w\theta - b(\theta)}{a(\gamma)} + c(w, \gamma)\right\}$$

For functions $a(\cdot)$, $b(\cdot)$, $c(\cdot)$ where γ is not required necessarily.

Theorem 8 (Y distribution $\in \mathcal{E}$)**.**

$$Y_i \sim \mathcal{B}(p_i) \implies Y_i \sim f(\cdot) \in \mathcal{E} \forall i$$

Proof

$$Y_i \sim \mathcal{B}(p_i) \implies f(y_i) = (p_i)^{y_i}(1 - p_i)^{1-y_i} \tag{4}$$

$$= exp\left\{y_i ln[p_i] + (1 - y_i) ln[1 - p_i]\right\} \tag{5}$$

$$= exp\left\{y_i ln[\frac{p_i}{1 - p_i}] + ln[1 - p_i]\right\} \tag{6}$$

Setting $\theta = ln[\frac{p_i}{1 - p_i}] \implies e^\theta = \frac{p}{1 - p} \implies p = \frac{e^\theta}{1 - e^\theta} \implies ln(1 - p_i) = -ln(1 - e^\theta)$

$$\implies f(y, \theta) = exp(y\theta - ln[1 - e^\theta])$$

Where:

$$\begin{cases} \theta = ln\left[\frac{p_i}{1 - p_i}\right] \\ a(\gamma) = 1 \\ b(\theta) = ln[1 - e^\theta] \\ c(y_i, \gamma) = 0 \ \forall y_i \in Y \end{cases}$$

2.2 Bayesian Probability

The approach implemented will be that of Bayesian techniques. For this reason, we quickly introduce its framework.

Definition 9 (Prior Distribution π)**.** *A prior distribution for a random variable θ is its unconditional distribution*

$$\theta \sim \pi(\cdot) \ : \ \int_\Theta \pi(\theta) d\theta = 1$$

Definition 10 (Likelihood $\mathcal{L}(\cdot, Y)$)**.** *The likelihood function for a set of observations is the joint probability distribution of the parameter of interest and the observations.*

$$\theta, Y \sim \mathcal{L}(\cdot, Y) \implies \mathcal{L}(\theta, Y = y) = \prod_{i=1}^{n} f_i(y_i|\theta) \text{ by independent samples}$$

Definition 11 (Posterior $\pi(\cdot|Y)$)**.** *The posterior distribution is the updated distribution of the parameter considering that Y was observed. Namely:*

$$\theta|Y \sim \pi(\cdot|Y)$$

Theorem 12 (Bayes Theorem implication)**.** *Given the following setting, for a parameter θ and a set of observations Y we will have the following proportionality rule when estimating the posterior distribution of a parameter of interest:*

$$\pi(\theta|Y = y) \propto \pi(\theta)\mathcal{L}(\theta, Y = y)$$

The following statements will be useful for section 3.2, and were claimed without a proof in [1]. We chose to justify them briefly for rigorousness.

Definition 13 (Conjugate Prior)**.** *A prior for a parameter as in Definition 9 is conjugate with respect to a model with a likelihood as in Definition 10 if the posterior of the parameter is of the same distribution of the prior.*
This is helpful as any Bayesian update resorts to an update of the parameters.

Definition 14 (Multivariate Normal Distribution $\mathcal{N}^d$)**.**

$$X \in \mathbb{R}^d \ : \ X \sim \mathcal{N}^d(\mu, \Sigma) \tag{7}$$

$$f(X, \mu, \Sigma) = (2\pi)^{-\frac{d}{2}}(det[\Sigma])^{-\frac{1}{2}} exp\Big\{ -\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\Big\} \tag{8}$$

Where $\mu = \mathbb{E}(X)$ and $\Sigma = V[X]$.
Oftentimes for simplicity the following notation is used: $det[\Sigma] = |\Sigma|$
And the variance matrix is expressed as its inverse, the precision matrix $\Lambda := \Sigma^{-1}$. This will be useful later.

Observation 15 (Multivariate Normal Kernel of Density)**.** *To understand the form of a posterior it is often useful to identify in the distribution the kernel of a known distribution. In the case of the multivariate normal we have that the elements directly dependent on X when $X \sim \mathcal{N}^d(\mu, \Sigma)$ are:*

$$f(X) \propto \exp\Big\{ -\frac{1}{2}(X-\mu)^T\Lambda(X-\mu)\Big\} \propto exp\Big\{ -\frac{1}{2}(X^T\Lambda X - \mu^T\Lambda X - X^T\Lambda\mu)\Big\}$$

Where by another important observation that we will remark:

$$\exists\Sigma \ symmetric \iff \Lambda = \Sigma^{-1} \ symmetric \ : \ \Lambda = \Lambda^T \tag{9}$$

Thus:

$$\implies f(X) \propto exp\Big\{ -\frac{1}{2}(X^T\Lambda X - \mu^T\Lambda^T X - X^T\Lambda\mu)\Big\} \tag{10}$$

Where the elements in red are those that would hint that the posterior is a multivariate normal with the red parameters.

Theorem 16 (Multivariate Normal model mean conjugate prior)**.** [2] *Let $i = \{1, \dots, n\}$*

$$X_i|\mu \overset{iid}{\sim} \mathcal{N}^d(\mu, \Sigma) \ : \ \Sigma \, known \ \wedge \mu \sim \mathcal{N}^d(\mu_0, \Sigma_0)$$
$$\implies \mu|X \sim \mathcal{N}^d(\widetilde{\mu}, \widetilde{\Sigma}) \ : \ \widetilde{\mu} = (n\Sigma^{-1} + \Sigma_0^{-1})^{-1}(n\Sigma^{-1}x_n + \Sigma_0^{-1}\mu_0) \ \ \widetilde{\Sigma} = (n\Sigma^{-1} + \Sigma_0^{-1})^{-1}$$

Proof *While the result holds $\forall n$, we will for simplicity prove it for $n = 1$. It is indeed easy to notice that the result would only be that the likelihood is a sum of multiple instances of the same distribution due to*

[2] This result is proposed in [4] and recalled in many other sources . Since we could not find any proof, we 9+proved it on our own using linear algebra.

the independence property, and that the X_is will sum as well as their variances Σ will multiply n times

$$X|\mu \sim \mathcal{N}^d(\mu, \Sigma) \qquad \implies f(X|\mu) = (2\pi)^{-\frac{d}{2}}(det[\Sigma])^{-\frac{1}{2}} exp\Big\{ -\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\Big\}$$

$$\mu \sim \mathcal{N}^d(\mu_0, \Sigma_0) \qquad \implies f(\mu) = (2\pi)^{-\frac{d}{2}}(det[\Sigma_0])^{-\frac{1}{2}} exp\Big\{ -\frac{1}{2}(\mu-\mu_0)^T\Sigma_0^{-1}(\mu-\mu_0)\Big\}$$

Which by Theorem 12:

$$\implies f(\mu|X) \propto (2\pi)^{-\frac{d}{2}}(2\pi)^{-\frac{d}{2}}(det[\Sigma_0])^{-\frac{1}{2}}(det[\Sigma])^{-\frac{1}{2}} exp\Big\{ -\frac{1}{2}\Big[(X-\mu)^T\Sigma^{-1}(X-\mu) + (\mu-\mu_0)^T\Sigma_0^{-1}(\mu-\mu_0)\Big]\Big\}$$

Where the elements in blue are those that pass the "requirement" of being actually linked to μ, which is the variable of interest. Next, we will iteratively cancel out all those that are not linked to μ with the same scheme to end up with the kernel of $f(\mu|X)$.

$$f(\mu|X) \propto exp\Big\{ -\frac{1}{2}\Big[X^T\Lambda X - X^T\Lambda\mu - \mu^T\Lambda X + \mu^T\Lambda\mu + \mu^T\Lambda_0\mu - \mu_0^T\Lambda_0\mu - \mu^T\Lambda_0\mu_0 + \mu_0^T\Lambda_0\mu_0\Big]\Big\}$$

Where we expanded the product and used the precision - covariance substitution.

$$f(\mu|X) \propto exp\Big\{ -\frac{1}{2}\Big[\mu^T\Lambda\mu + \mu^T\Lambda_0\mu - X^T\Lambda\mu - \mu^T\Lambda X - \mu_0^T\Lambda_0\mu - \mu^T\Lambda_0\mu_0\Big]\Big\}$$

Where all elements are linked to μ. The operation done was just reordering of the summands to better view that they can be grouped together by powers as:

$$f(\mu|X) \propto exp\Big\{ -\frac{1}{2}\Big[\mu^T(\Lambda+\Lambda_0)\mu - (X^T\Lambda + \mu_0^T\Lambda_0)\mu - \mu^T(\Lambda X + \Lambda_0\mu_0)\Big]\Big\}$$

Which, using Equation 9, namely $\Lambda = \Lambda^T$:

$$\implies f(\mu|X) \propto exp\Big\{ -\frac{1}{2}\Big[\mu^T(\Lambda+\Lambda_0)\mu - (X^T\Lambda^T + \mu_0^T\Lambda_0^T)\mu - \mu^T(\Lambda X + \Lambda_0\mu_0)\Big]\Big\} \tag{11}$$

Where the elements in red recall Equation 10 of Observation 15. We can claim that the posterior of $\mu|X$ is proportional to the kernel of a multivariate normal with updated parameters thanks to its conjugate prior structure.
The new Variance can be extracted directly from the last formula as:

$$\widetilde{\Lambda} = (\Lambda + \Lambda_0) \implies \widetilde{\Sigma} = \widetilde{\Lambda}^{-1} = (\Lambda+\Lambda_0)^{-1} = (\Sigma^{-1} + \Sigma_0^{-1})^{-1} \tag{12}$$

While the mean is the result of:

$$\widetilde{\Lambda}\widetilde{\mu} = \Lambda x + \Lambda_0\mu_0 \tag{13}$$

$$\implies \widetilde{\mu} = \widetilde{\Lambda}^{-1}(\Lambda X + \Lambda_0\mu_0) \tag{14}$$

$$\implies \widetilde{\mu} = (\Sigma^{-1} + \Sigma_0^{-1})^{-1}(\Sigma^{-1}X + \Sigma_0^{-1}\mu_0) \tag{15}$$

The usual standard is to substitute $\sum_{i=1}^n X = n\bar{X}_n$ and conclude that in a setting where there are n iid observations:

$$\mu|\{X_i\} \sim \mathcal{N}^d(\widetilde{\mu}, \widetilde{\Sigma})$$

$$\widetilde{\Sigma} = (n\Sigma^{-1} + \Sigma_0^{-1})^{-1} \quad \widetilde{\mu} = (n\Sigma^{-1} + \Sigma_0^{-1})^{-1}(n\Sigma^{-1}\bar{X}_n + \Sigma_0^{-1}\mu_0)$$

2.3 Markov Chain Monte Carlo

We will propose two approaches, which come from the same family of computational methods.

As an objective, we wish to exploit a Monte Carlo approach to obtain a stationary Markov Chain representing $\pi(\beta|y)$. The algorithm is proposed below along with some observations. Later in the sections we will explore specific cases that will be implemented and their additional features.

Definition 17 (Metropolis Hastings (MH) Algorithm)**.** *Let $\theta|x \sim \pi(\cdot|x)$ where $\theta \in \Theta$ connected*[3]*. Given a proposal distribution for candidates $\theta_*|\theta_t \sim q(\cdot|\theta_t)$: $supp\{q(\cdot|\theta)\} \supseteq \Theta$ and a starting value θ_0 we do the following:*

Algorithm 1 MH Algorithm Pseudocode

Input: T, $x \in X$, θ_0, $\pi(\cdot|\cdot)$, $q(\cdot|\cdot)$
Output: *list* MC of parameter

1: $list = []$
2: **for** $t = 0, \dots, T$ **do**
3: Draw $\theta^* \sim q(\cdot|\theta_t)$
4: Draw $u \sim \mathcal{U}(0,1)$
5: Evaluate $\alpha(\theta_*, \theta_t) = min\left\{1, \frac{\pi(\theta_*|x)q(\theta_t|\theta_*)}{\pi(\theta_t|x)q(\theta_*|\theta_t)}\right\}$
6: **if** $u \leq \alpha(\theta_*, \theta_t)$ **then** $\theta_{t+1} \leftarrow \theta_*$
7: **else** $\theta_{t+1} \leftarrow \theta_t$
8: **end if**
9: append θ_{t+1} to *list*
10: **end for**
11: **return** list

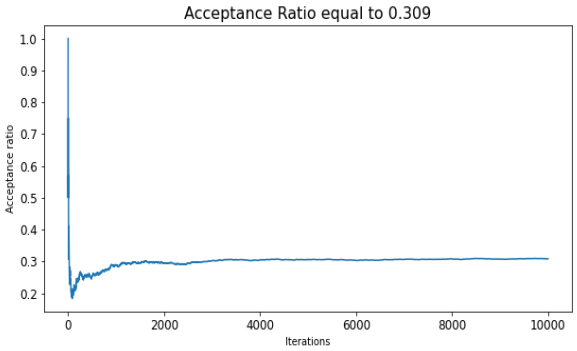


Figure 1: Example of the acceptance rate's behavior in the long run

Theorem 18 (MH Algorithm Properties)**.** *Any procedure satisfies the following:*

- $\{\theta_t\}_{T \geq 1}$ *is a Markov Chain*
- *Detailed Balance condition, given $K(\theta_1, \theta_2) = K(\theta_1 \to \theta_2)$ as transition kernel of the chain:*

$$K(\theta_1, \theta_2)\pi(\theta_1|x) = K(\theta_2, \theta_1)\pi(\theta_2|x)$$

- *If the chain is irreducible and aperiodic the ergodic theorem holds. This implies that:*

$$\forall f \ \exists t_0 \ : \ \frac{1}{T - t_0} \sum_{t=t_0}^{t=T} f(\theta_t) \overset{a.s.}{\to} \mathbb{E}[f(\theta)]$$

[3] Θ could also not be connected but we assume the easy case for simplicity

This property is useful. It implies that the time average of a sequence of Markov samples is equivalent to the average of the distribution of the parameter of interest. After checking that irreducibility and aperiodicity hold we can claim that the first moment of the sampled values after some iterations will almost surely tend to the expected value of the distribution.

In some occasions, sampling from the joint distribution of all directions is computationally demanding. Namely, doing as proposed in Algorithm 1 where the whole of θ is sampled at the same time is not always possible or advised.

Gibbs Sampling or Algorithm is one of the most famous and used methods for sampling multidimensional parameters θ using univariate conditional distributions.

Remark (Notation)**.** *In this last part of Section 2.3, throughout Section 3.2, and in any part of the document in which Gibbs techniques are used the notation will slightly change for time and position indexes meaning that:*

$$\theta_i^{(t)} \coloneqq i^{th} \text{ entry of theta at time } t \text{ where } \theta^{(t)} \in \mathbb{R}^{p+1} \forall t$$

To use a more appropriate notation.

Definition 19. *Full conditional* $\pi(\cdot|\theta_{-i})$ *Let* $\theta_{-i} \coloneqq \{\theta | j \; \forall \, j \neq i\} \; \forall i$ *then*

$$\theta_i|\theta_{-i} \sim \pi(\cdot|\theta_{-i}, X)$$

The probability distribution of a direction of θ *given knowledge about all the others and* X*.*

The following modification of Algorithm 1 makes use of the full conditionals to update parameters, and presents some nice properties.

Algorithm 2 Gibbs Sampler Algorithm

Input: T, $x \in X$, $\theta^{(0)}$, $\pi(\cdot, \theta_{-i}, x) \forall i \in \{0, \dots, p\}$
Output: *list* MC of parameter

list = []
for $t = 1, \dots, T$ **do**
  for p times (described by the series) **do**
    Draw $\theta_0^{(t)} \sim \pi(\cdot|\theta_{-0}^{(t-1)}, x)$
    Draw $\theta_1^{(t)} \sim \pi\left(\cdot \, |\{\theta_0^{(t)}, \; \theta_j^{(t-1)} \forall j > 1\}, x \right)$
    Draw $\theta_2^{(t)} \sim \pi\left(\cdot \, |\{\theta_0^{(t)}, \theta_1^{(t)} \; \theta_j^{(t-1)} \forall j > 2\}, x \right)$
    ...
    Draw $\theta_p^{(t)} \sim \pi\left(\cdot \, |\{\theta_0^{(t)}, \theta_1^{(t)}, \dots, \theta_{p-1}^{(t)}\}, x \right)$
    append $\theta^{(t)}$ to list
  end for
end for
return list

Where at each iteration we sample from an ideally easy to treat full conditional distribution conditioned on all the current updates possible and filled by the previous updates where not possible.

Theorem 20 (Gibbs Sampling Properties)**.** *The following holds:*

- *Gibbs sampling is a special case of the MH Algorithm 1*
- $\{\theta_{-i}^{(t)}|\theta_{-i}^{(t-1)}, X = x\}_{t=1}^{\infty} \sim \pi(\cdot|\theta_{-i}^{(t)}, x)$ *stationary*
- $\{\theta_i^{(t)}\}_{i=0,\dots,p}^{T \geq 1} \sim \pi_{\theta_i} = \int_{\Theta_{-i}} \pi(\theta|x) d\theta_{-i}$ *MC stationary* $\forall i$

Theorem 21 (Gibbs sampler is a perfect sampling MH Algorithm)**.** *Gibbs sampler* $\in \{MH Algorithms\ set\}$ *where the candidate at time* $t+1$*:*

$$\theta_i^* \sim \pi(\cdot|\theta_{-i}^{(t)} = (\theta_0^{(t)}, \dots, \theta_{i-1}^{(t)}, \theta_{i+1}^{(t-1)}, \dots, \theta_p^{(t-1)})$$

Is such that:

$$\forall i, t \;\; \alpha(\theta_i^*, \theta^{(t)}) = 1$$

Proof
We have that:

$$\theta_i^* \sim \pi(\cdot|\theta_{-i}^{(t)}, X = x) \;:\; q(\theta^*, \theta_{-i}^{(t)}) = \pi(\theta^*|\theta_{-i}^{(t)}, x)\delta_{\theta_{-i}^*}(\theta_{-i}^{(t)})$$

Where $\delta_{\theta_{-i}^*}(\theta_{-i}^{(t)}) \coloneqq \mathbb{1}(\theta_{-i}^* = \theta_{-i}^{(t)})$ *as all other dimensions have to be equal for each individual update.* [4] *This implies that:*

$$\implies \alpha(\theta_i^{(t)}, \theta_i^*) = min\left\{1, \frac{\pi(\theta^*|x)\pi(\theta_i^{(t)}|\theta_{-i}^*, x)}{\pi(\theta^{(t)}|x)\pi(\theta_i^*|\theta_{-i}^{(t)}, x)}\right\}$$

Where in general for any density $f(\cdot|x)$ *we have that we can decompose it as:*

$$f(\theta|x) = f(\theta_i, \theta_{-i}|x) = f(\theta_i|\theta_{-i}, x)f(\theta_{-i}|x)$$

Which applied above and below is equivalent to claiming:

$$\implies \alpha(\theta_i^{(t)}, \theta_i^*) = min\left\{1, \frac{\pi(\theta_i^*|\theta_{-i}^*, x)\pi(\theta_{-i}^*|x)\pi(\theta_i^{(t)}|\theta_{-i}^*, x)}{\pi(\theta_i^{(t)}|\theta_{-i}^{(t)}, x)\pi(\theta_{-i}^{(t)}|x)\pi(\theta_i^*|\theta_{-i}^{(t)}, x)}\right\}$$

Where by the condition of delta: $\delta_{\theta_{-i}^*}(\theta_{-i}^{(t)}) = 1 \iff (\theta_{-i}^* = \theta_{-i}^{(t)})$ *And eventually*

$$\implies \alpha(\theta_i^{(t)}, \theta_i^*) = min\left\{1, \frac{\pi(\theta_i^*|\theta_{-i}^{(t)}, x)\pi(\theta_{-i}^{(t)}|x)\pi(\theta_i^{(t)}|\theta_{-i}^{(t)}, x)}{\pi(\theta_i^{(t)}|\theta_{-i}^{(t)}, x)\pi(\theta_{-i}^{(t)}|x)\pi(\theta_i^*|\theta_{-i}^{(t)}, x)}\right\} = min\Big\{1, 1\Big\} = 1$$

So Gibbs sampling is a version of the MH Algorithm 1 where at each iteration all proposals are accepted.

Observation 22 (Practical Approach)**.** *In [5] and [1] the framework proposed simplifies sampling in that the values are taken all at once, and the chain is run for a longer time.*

Thanks to Theorems 20 and 21 once full conditionals are available and efficient to sample from the problem reduces to simple iterations over time.

This unique requirement is not always satisfied.

Definition 23 (Instrumental Variable (IV) W)**.**

$$W = \{W_i\}_{i-1}^n \;:\; f(\theta|X) = \int_{\mathcal{W}} f(\theta, w|X)dw$$

Usually, it is not the variable but its ***properties*** *that make it* ***instrumental*** *for some problem.*

Definition 24 (Data Augmentation Approach Process)**.** *In some precise cases, introduced with the data augmentation approach by [6] mentioned in [1] it may be the case that:*

- *Sampling from full conditionals of* θ *is "difficult"*
- *There is a set of instrumental variables* $\{Z_i\}_{i=1}^n$ *such that it is instead easy to devise a sampler for* $f\Big(\theta' = (\theta, z)|X\Big)$
- $\forall t$ *extract* $(\theta^{(t)}, z^{(t)})$ *with Algorithm 2 and ignore the updates of* z

[4] In this case, it is also true that we should better define the full conditional as: "all updated before i, all not updated after i"

3 Problem & Solving Methods

In our setting, we can say that $\theta = \beta = [\beta_0, \ldots, \beta_p]^T \in \mathbb{R}^{p+1}$. Thus, there will be $p+1$ parameters to identify and a prior to propose for them.

Observation 25. *When feasible, it might be useful to assume:*

$$\beta_{start} = \beta^{(t=0)} = \beta_{MLE} \in \mathbb{R}^{p+1}$$

This allows us to start from a "reasonable" value [1]. In the event that the MLE were not feasible to calculate, one can resort to the EM algorithm to approximate or other considerations. Programming languages often do not find the exact solution but iteratively find a result.

In our setting $Y_i \sim \mathcal{B}(p_i)$ and we can claim that for a given parameter β the likelihood will be:

$$\mathcal{L}(\beta, (y_1, \ldots, y_n)) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} = \prod_{i=1}^{n} (\phi(x_i^T\beta))^{y_i}(1-\phi(x_i^T\beta))^{1-y_i} = \prod_{i=1}^{n} \phi(\eta_i)^{y_i}(1-\phi(\eta_i))^{1-y_i} \quad (16)$$

Which in maybe easier terms applying a *log* transformation is equivalent to:

$$\ell(\beta, y) \coloneqq log(\mathcal{L}(\beta, y)) = \sum_{i=1}^{n} y_i log[\phi(\eta_i)] + (1-y_i) log[1-\phi(\eta_i)] \quad (17)$$

Following a Bayesian fashion it might be interesting to sample directly from the posterior distribution which is of the form:

$$\pi(\beta|y) = \frac{\pi(\beta)\mathcal{L}(\beta, y)}{f(y)} = \frac{\pi(\beta)\mathcal{L}(\beta, y)}{\int_B \pi(\beta)\mathcal{L}(\beta, y)d\beta} = C\pi(\beta)\mathcal{L}(\beta, y)$$

Where C is a constant as $Y = y$ is a realization.
However, it can be noticed that the complete posterior is difficult to sample from, and one has to resort to lighter computation methods, which avoid C.

3.1 Random Walk MH

Given the design, different choices of the proposal lead to different features of the procedure. One possible approach is using a perturbation scheme which explores candidates at time t following a normal probability distribution.

Assumption 26. *For this chapter, we will assume that:*

$$\beta_* \sim q(\cdot, \beta_t) = \mathcal{N}^{p+1}(\beta_t, \tau V) \ : \ \tau^5 \in \mathbb{R} \ \& \ V = -\left[\frac{\partial^2}{\partial \beta_t^2}\ell(\beta_t)\right]^{-1}$$

Which is equivalent to:

$$\beta_* = \beta_t + \nu \ : \ \nu \sim \mathcal{N}^{p+1}(0, \tau V)$$

Definition 27 (Score function $\varrho(\theta|X)$)**.** *Let $X|\theta \sim f(\cdot|\theta)$ then:*

$$\varrho(\theta|X = x) \coloneqq \mathbb{E}\left[\frac{\partial}{\partial\theta} ln[f(\theta|X = x)]\right] = \mathbb{E}\left[\frac{\partial}{\partial\theta}\updownarrow(\theta, X = x)\right]$$

Definition 28 (Fisher Information $\mathcal{I}(\theta)$)**.**

$$\mathcal{I}(\theta) \coloneqq V\left[\varrho(\theta|X = x)\right]$$

[5] Preset parameter.

Theorem 29 (Fisher Matrix properties)**.**

$$\mathcal{I}(\theta) = -\mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\ell(\theta)\right]$$

Moreover, in a GLM framework in which we wish to estimate θ to infer about μ such as: $\{(Y_i, X_i)\}_{i=1}^n$: $Y_i \overset{iid}{\sim} f(\cdot, \mu_i) \in \mathcal{E}(\mu_i, \phi)\ \mu_i \in \mathbb{R}$ We will have that:

$$\implies \widehat{\mathcal{I}_n(\theta)} = X^T W X$$

Where $W \coloneqq \left\{ w_{ij} = 0\ \forall i \neq j,\ w_{ii} = \frac{1}{V(Y_i)}\left(\frac{\partial\eta_i}{\partial\mu_i}\right)^{-2}\ \forall i \right\}$

And also:

$$\left[\mathcal{I}_n(\theta)\right]^{-1} \approx V(\hat{\theta}_{MLE})$$

In order to sample, $\forall t$ there has to be an efficient way of computing the inverse of the second derivative of the likelihood of β_t. Thanks to the fact that $Y_i \sim \mathcal{E}(\mu_i, \_)\forall i$ this can be achieved by estimating the Fisher Information matrix, that will approximate the behavior of the variance of a normal distribution centered at the current update.
Thus, an explicit representation has to be found for our setting.

Claim 30 (Fisher Matrix $\mathcal{I}_n(\beta)$ explicit representation)**.** *Recollecting Observation 6, we can construct the inner partial derivative noting that in our specific case $\mu_i = p_i$, $\eta_i = X_i^T\beta$, $\mu_i = \Phi(\eta_i)$*

$$\implies \eta_i = \Phi^{-1}(\mu_i) \implies \frac{\partial\eta_i}{\partial\mu_i} = \frac{\partial\eta_i}{\partial p_i} = \frac{\partial\Phi^{-1}(p_i)}{\partial p_i} = \frac{1}{\phi(\Phi^{-1}(p_i))} = \frac{1}{\phi(\eta_i)}$$

$$\implies w_{ii} = \frac{1}{V(Y_i)}\left(\frac{\partial\eta_i}{\partial p_i}\right)^{-2} = \frac{1}{p_i(1-p_i)}\phi^2(\eta_i) = \frac{\phi(\eta_i)^2}{\Phi(\eta_i)(1-\Phi(\eta_i))}\ \forall i$$

The peculiar proposal function simplifies the iterations by direct implication.

Theorem 31 (Acceptance function $\alpha(\beta_*, \beta_t)$ form)**.** *If $\beta_* = \beta_t + \nu$ then*

$$\implies \alpha(\beta_*, \beta_t) = min\left\{1, \frac{\pi(\beta_*|y)}{\pi(\beta_t|y)}\right\}$$

Meaning that the proposal is symmetric and it cancels out.
Proof

We have that $\forall t\ \beta_* = \beta_t + \nu \implies \beta_* \sim \mathcal{N}^{p+1}(\beta_t, \tau V) \implies \beta_* \sim q(\cdot|\theta_t) = \mathcal{N}^{p+1}\left(\beta_t, -\tau\left[\frac{\partial^2}{\partial\beta_t^2}\ell(\beta_t)\right]^{-1}\right)$.
If instead we sampled from β_* the distribution would have been the same as a normal function is symmetric itself. Thus the probability of sampling either of the two from the other is the same and the proposals are symmetric.

$$\implies \alpha(\beta_*, \beta_t) = min\left\{1, \frac{\pi(\beta_*|y)q(\beta_t|\beta_*)}{\pi(\beta_t|y)q(\beta_*|\beta_t)}\right\} = min\left\{1, \frac{\pi(\beta_*|y)}{\pi(\beta_t|y)}\right\}$$

For each iteration, we only have to sample from a multivariate normal, sample form a uniform, and update confronting the two posteriors.

Observation 32 (Constant C cancels out inside α)**.** *It is worth noticing that as $Y = y$ is the common information, the normalizing factor cancels out when evaluating a fraction. Indeed:*

$$\frac{\pi(\beta_*|y)}{\pi(\beta_t|y)} = \frac{\frac{\pi(\beta_*)\mathcal{L}(\beta_*, y)}{f(y)}}{\frac{\pi(\beta_t)\mathcal{L}(\beta_t, y)}{f(y)}} = \frac{\pi(\beta_*)\mathcal{L}(\beta_*, y)}{\pi(\beta_t)\mathcal{L}(\beta_t, y)}$$

Given knowledge of the prior and the likelihood formula, we can safely modify Algorithm 1 into:

Algorithm 3 MH Algorithm with normal proposal Pseudocode

Input: T, τ, $\{y_i\}_{i=1}^n$, $\{x_i \in \mathbb{R}^p\}_{i=1}^n$ β_0, $\pi(\cdot|\cdot)$, $q(\cdot|\cdot) = \mathcal{N}(\cdot,\cdot)$
Output: *list* MC of parameter

list = []
for $t = 0, \dots, T$ **do**
  $W = diag\left\{ \frac{\phi(\eta_i)^2}{\Phi(\eta_i)(1-\Phi(\eta_i))} \; \eta_i = x_i^T \beta_t \; \forall i \right\}$
  Evaluate $\hat{V} = (X^T W X)^{-1}$
  Draw $\beta^* \sim \mathcal{N}(\beta_t, \tau V)$
  Draw $u \sim \mathcal{U}(0,1)$
  Evaluate $\alpha(\beta_*, \beta_t) = min\left\{1, \frac{\pi(\beta_*)\mathcal{L}(\beta_*, y)}{\pi(\beta_t)\mathcal{L}(\beta_t, y)}\right\}$
  if $u \leq \alpha(\beta_*, \beta_t)$ **then** $\beta_{t+1} \leftarrow \beta_*$
  else $\beta_{t+1} \leftarrow \beta_t$
  end if
  append β_{t+1} to list
end for
return list

Observation 33 (Likelihood simplified taking logs)**.** *Using the link between equations 16 and 17, we ease computations evaluating instead:*

$$\log\left(\frac{\pi(\beta_*|y)}{\pi(\beta_t|y)}\right) = \log(\pi(\beta_*))\ell(\beta_*, y) - \log(\pi(\beta_t))\ell(\beta_t, y)$$

And at every iteration we will resort to calculating those 4 elements only. By applying this transformation, the updated procedure to accept a proposal is the following:

$$\log(u) \leq \log(\pi(\beta_*))\ell(\beta_*, y) - \log(\pi(\beta_t))\ell(\beta_t, y) \; , u \sim \mathcal{U}(0,1)$$

Otherwise the proposal is rejected.

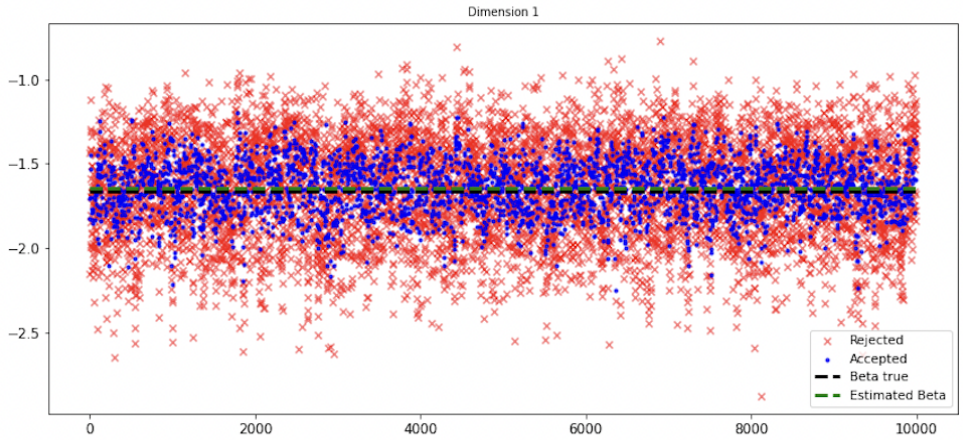


Figure 2: An example of acceptance rejection procedure

3.2 Gibbs Sampling with Instrumental Variables

This second method takes inspiration from the method described in [1].
For this purpose, we introduce the concept of instrumental variable as latent proposed in equations 2 and 3.

Assumption 34 (Instrumental Variable is Z_i)**.** *From here henceforth, we will refer to the latent variable Z_i as an instrumental variable.*
In particular, we know that:

$$Y_i = \mathbb{1}(Z_i > 0) \ : \ \{Z_i\}_{i=1}^n \ : \ \forall i \ Z_i = X_i^T\beta + \varepsilon_i \ : \ \varepsilon_i \sim \mathcal{N}(0,1)$$

Consequently, given the linear transformation of normally distributed random variable, we have::

$$Z_i \sim \mathcal{N}(X_i^T\beta, 1)$$

Moreover, as claimed by [1]

$$\beta|Z \sim \mathcal{N}^{p+1}(\cdot,\cdot) \wedge Z_i \text{ known } \implies \beta|X \sim \pi(\cdot|X) \text{ retrievable} \tag{18}$$

Due to the fact that it would be a normal linear regression model with standard normal errors.
*However, Z_is are unknown as they are **latent**.*

An intuitive approach to extract their distribution followed by formal considerations is proposed below.

Observation 35. *In such a framework, it is known that:*

$$Z_i|Y_i = y_i : \begin{cases} Z_i > 0 \iff y_i = 1 \\ Z_i < 0 \iff y_i = 0 \end{cases} \quad \text{By definition of } Z \tag{19}$$

$$\wedge \ Z_i = X_i^T\beta + \varepsilon_i \implies Z_i|\beta \sim \mathcal{N}(\cdot,\cdot) \quad \text{By } Z_i = \text{linear combination of normal independent multivariate r.v.s} \tag{20}$$

$$\implies Z_i|Y_i = y_i, \ \beta \sim \mathcal{TN}(\cdot,\cdot) \quad \text{By } Y_i\&\beta \implies \text{sign\&distribution of } Z_i \text{ is known} \tag{21}$$

Where the symbol $\mathcal{TN}$ denotes the Truncated Normal Density

Theorem 36 (Sampling from a $\mathcal{TN}$)**.** *Let $W \sim \mathcal{TN}^{a,b}(c, d^2)$ restricted to (a, b) with mean c and variance d^2 then:*

$$\implies f(W) = c + d\Phi^{-1}\left(\Phi\left(\frac{a-c}{d}\right) + U\left(\Phi\left(\frac{b-c}{d}\right) - \Phi\left(\frac{a-c}{d}\right)\right)\right) \ : \ U \sim \mathcal{U}(0,1)$$

Where Φ is the cdf of a standard normal $\mathcal{N}(0,1)$.
Thus, to sample from $\mathcal{TN}$ we have to sample from $\mathcal{U}$ a value u and multiply the values of the formula.
Proof
The proof and the statement can be achieved with the inverse transform method, or in [7].

What we observe is only $Y = (y_1, \dots y_n)$ and X_is linked to them. From these two objects some conclusions described by [1] can be taken.

Theorem 37 (Full conditionals of β and Z)**.** *Given the current framework, let $\beta \sim \pi(\cdot)$ as in Definition 9 and $Z : Z_i$ iid be an instrumental latent variable as in Definition 23 with the Properties outlined in equations 2, 3, 19, 20 then it holds that:*

$$\pi(\beta|Z, y) = C\pi(\beta)\prod_{i=1}^{n}\left(\phi(Z_i; x_i^T\beta, 1)\right) \ : \quad \phi(Z_i; x_i^T\beta, 1) = pdf\left(\mathcal{N}(\mu = X_i^T\beta, \sigma^2 = 1)\right)\forall i \tag{22}$$

And if β uninformative[6] *(equally likely $\forall \beta \in B$)*

$$\implies \beta|Z,y \sim \mathcal{N}^{p+1}\left(\hat{\beta}_Z, (X^TX)^{-1}\right) \ : \ \hat{\beta}_Z = (X^TX)^{-1}(X^TZ) \tag{23}$$

While if $\beta \sim \mathcal{N}(\beta_{prior}, V_{prior})$

$$\implies \beta|Z,y \sim \mathcal{N}^{p+1}(\widetilde{\beta}, \widetilde{V}) \ : \ \begin{cases} \widetilde{\beta} = (V_{prior}^{-1} + X^TX)^{-1}(V_{prior}^{-1}\beta_{prior} + X^TZ) \\ \widetilde{V} = (V_{prior}^{-1} + X^TX)^{-1} \end{cases} \tag{24}$$

And:

$$y_i = 1 \implies Z_i|y,\beta \sim \mathcal{TN}^{+}(X_i^T\beta, 1) \tag{25}$$

$$y_i = 0 \implies Z_i|y,\beta \sim \mathcal{TN}^{-}(X_i^T\beta, 1) \tag{26}$$

Where the $+$ and $-$ signs at the apex of $\mathcal{TN}$ identify a truncated normal with only positive or only negative values.

Preliminary: *the statements will be proved in order and will be divided in two, with a common ground at the beginning.*

As observed in Observation 35 at equations 19, 20, Z is a linear combination of normal variables in its latent form. Observing $y_i \forall i$, the dependence between the two is perfectly stored in the following joint distribution:

$$\pi(\beta, Z|y) = C\pi(\beta)\prod_{i=1}^{n}\Big(\mathbb{1}(Z_i > 0)\mathbb{1}(y_i = 1) + \mathbb{1}(Z_i < 0)\mathbb{1}(y_i = 0)\Big)\phi(Z_i, X_i^T\beta, 1)$$

Where β comes from its prior, and Z_i for each sample is normally distributed as in its latent form centered at $X_i^T\beta$ and with variance 1 determined by its ε_i variance value, linked to the observed y_i by the double indicator function inside the parenthesis.

In other words, β "appears" and determines Z_is' centers which in turn are observed only through y_i, thus seeing only if the latent Z_i was positive or negative.

Proof of equation 22:

Using standard results $\forall f$ density functions $f(W,V|Q) = f(W|V,Q)f(V|Q) \implies f(W|V,Q) = \frac{f(W,V|Q)}{f(V|Q)}$ provided that the event at the denominator is not null. In this setting:

$$\pi(\beta|Z,y) = \frac{\pi(\beta,Z|y)}{\pi(Z|y)} \ : \ \pi(Z|y) = \prod_{i=1}^{n}\Big(\mathbb{1}(Z_i > 0)\mathbb{1}(y_i = 1) + \mathbb{1}(Z_i < 0)\mathbb{1}(y_i = 0)\Big) \neq 0 \ by \ definition$$

$$\implies \pi(\beta|Z,y) = C\pi(\beta)\prod_{i=1}^{n}\phi(Z_i, X_i^T\beta, 1)$$

Which is in line with our assumptions and identifies a distribution

Proof of equation 23:

Using equation 2 : $Z = X\beta + \varepsilon$: $\varepsilon \sim \mathcal{N}(0,1)$ solving for $\hat{\beta}$ using the classic linear results [7] *leads to the following:*

$$\beta|y,Z \sim \mathcal{N}^{p+1}\left(\hat{\beta_Z}, \hat{V_Z}\right)$$

Where $\hat{\beta_Z} = (X^TX)^{-1}(X^TZ)$ and $\hat{V_Z} = (X^TX)^{-1}$ are the standard results of an OLS. Namely, before having knowledge about β nothing is known, after observing information the value concentrates around the most likely one with a normally shaped distribution.

[6] In [1] it is called *diffuse*

[7] Namely, orthogonality of residuals (verified by assumption), Normal distribution of $\mathbb{E}(Z)$ in the limit, etc. Ultimately, solving for beta

Proof of equation 24
We have that the prior is a multivariate normal, and the model is a multivariate normal with known variance. By Theorem 16 the result is implied with the claimed form.
Proof of equation 25
Upon knowledge of y_i, Z_i is necessarily either positive or negative. Nevertheless, by assumption, it is also normally distributed. This implies that Z_is distribution will have a proportionality constant that sums to one all values of an either positive or negative normal distribution. This is the equivalent of claiming that $Z_i|y_i, \beta \sim \mathcal{TN}^{\pm}(X_i^T\beta, 1)$ depending on y_i's value (one positive, zero negative).

Remark (Sampling from $\mathcal{TN}^{\pm}$)**.** *Using Theorem 36 in the setting of an either positive or negative Truncated normal is equivalent to:*

$$Z_i \sim \mathcal{TN}^{+}(X_i^T\beta, 1) \implies (a = 0,\ b = \infty) \implies \Phi\left(\frac{a-c}{d}\right) = \Phi(-X_i^T\beta),\ \Phi\left(\frac{b-c}{d}\right) = \Phi(\infty) = 1$$
$$\implies f(Z_i) = X_i^T\beta + \Phi^{-1}\Big(\Phi(-X_i^T\beta) + U(1 - \Phi(-X_i^T\beta))\Big)\ :\ U \sim \mathcal{U}(0,1)$$

$$Z_i \sim \mathcal{TN}^{-}(X_i^T\beta, 1) \implies (a = -\infty,\ b = 0) \implies \Phi\left(\frac{a-c}{d}\right) = \Phi(-\infty) = 0,\ \Phi\left(\frac{b-c}{d}\right) = \Phi(-X_i^T\beta)$$
$$\implies f(Z_i) = X_i^T\beta + \Phi^{-1}\Big(U(\Phi(-X_i^T\beta))\Big)\ :\ U \sim \mathcal{U}(0,1)$$

Having in hand both full conditionals and methods to sample them we can rewrite our customized Gibbs sampler taking as a template that of Algorithm 2 with $\theta' = (\beta, Z)$ and the approach described in Definition 24. In a real application, [1] claim that it is convenient to sample with an uninformative prior of beta, that of Equation 23, our Algorithm proposal is in accordance with both options.

Algorithm 4 Instrumental Gibbs Sampler Algorithm

Input: T, $x \in X$, $y \in Y$, $\beta^{(0)}$, $\pi(\cdot)$ prior
Output: *list* MC of parameter
1: Select an appropriate posterior $\pi(\cdot|Z, y, x)$ wrt $\pi(\cdot)$ and Theorem 37
2: $list = []$
3: **for** $t = 1, \ldots, T$ **do**
4: **for** $i = 1, \ldots, n$ **do**
5: Draw $u_i \sim \mathcal{U}(0,1)$
6: **if** $y_i == 1$ **then**
7: $Z_i^{(t)} \leftarrow X_i^T\beta^{(t-1)} + \Phi^{-1}\Big(\Phi(-X_i^T\beta^{(t-1)}) + u_i(1 - \Phi(-X_i^T\beta^{(t-1)}))\Big)$
8: **else**
9: $Z_i^{(t)} \leftarrow X_i^T\beta^{(t-1)} + \Phi^{-1}\Big(u_i(\Phi(-X_i^T\beta^{(t-1)}))\Big)$
10: **end if**
11: **end for**
12: Draw $\beta^{(t)} \sim \pi(\cdot|Z^{(t)}, y, x)$
13: append $\beta^{(t)}$ to *list*
14: **end for**
15: **return** list

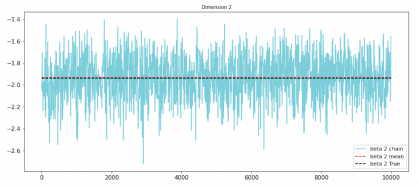
3.3 Desirable Results

The returned objects of Algorithms 3 and 4 are lists of accepted values of the parameter. Having obtained such an output, we are not done. What is missing is an application dependent analysis of the parameters

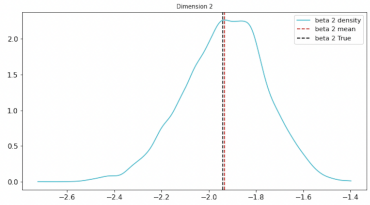
to understand if the requirements of the second point of Theorem 18 are satisfied. We wish that the chain is irreducible and aperiodic after some iterations.
To check the configurations in which these hold we decided to implement a graphical approach to lighten the analysis. Being that it depends on the data we are using, we will introduce in the next section code related observations.

Ideally, a satisfactory result would present the following:

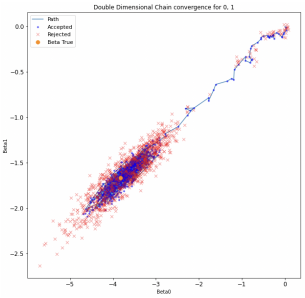
- Non serially correlated exploration of the values and sufficient exploration of the region of interest:



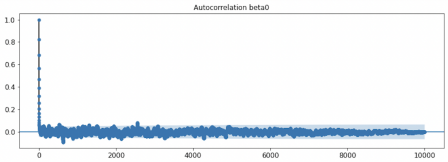
- Symmetric density of the parameter:



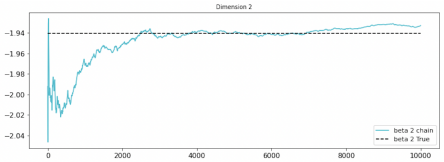
- Decay of the chain towards a region of concentration:



- Null long term autocorrelation:



- Stable cumulative mean as time goes on:



- Stability across simulations (i.e. few variability across different samples)

Given that we will proceed with estimations, we indeed expect some imprecision. Due to time and length constraints, we chose to analyze with plots all but the last. We recognize that there are also notable procedures to balance the trial and error. One interesting method we found for the last point is *Nested* $\widehat{R}$[8].

4 Code Implementation

In the following section we briefly explain how we tackled the application of such methods with code. It is mostly intended to explain the reasoning behind some functions and give credit to the sources that helped us, especially in the plotting part.

Along with this document comes a Python script. We chose to arrange it on Google Colab. **TODO add openin colab button to github**
The aim of the script is to explore applications of the theory just introduced. At first, we create a toy dataset with gaussian noise to use as leading example. After having created the whole system of functions and algorithms, we also added a synthetic[8] dataset for machine failure prediction [2][3]. We believe this is a good starting point to assess the efficacy of such methods. Other studies have also attempted to implement a bayesian probit for machine failure[9].
Intuitively, we believe that examining such faults is both useful and reasonable. For example mechanical wear and tear and wrong maintenance could be seen as failure causes after a damage threshold is reached. This resembles the probit model in reality, where the Z latent variable would in this case be an indicator of damage, which if positive, whatever the condition, implies that the machine fails.

To check what was introduced in Section 3.3 as qualitative ways to assess if Theorem 18 point 2 holds we created a set of plotting functions. Below a table explains their aims.

Function	**Description**	**Source**
plot trace dimension	trace: samples vs iterations for a given dimension	Lectures
plot autocorrelation	autocorrelation vs iterations	Lectures
plot cumulative mean	current mean of parameter list vs iterations	Basic
plot alpha	current acceptance proportion vs iterations	Basic
plot density dimension	density of a given dimension	[10]
plot accepted rejected	trace with accepted and rejected values for a given dimension	[11]
plot double dim accepted rejected	accepted and rejected values path in $\mathbb{R}^2$	[11]
plot triple dim accepted rejected	accepted and rejected values path in $\mathbb{R}^3$	Adaptation of [11]

Table 1: Graphical tools for Output checking

To compare the impact of the parameters changing, we also crafted some custom functions that explore different settings.

Function Parameters
Priors $\pi(\cdot)$
Tau τ *of proposal kernel in MH*
Stretch of the normal prior
Starting beta configuration

Table 2: Graphical tools for Output comparison

[8]The authors explicitly state that: "*Since real predictive maintenance datasets are generally difficult to obtain and in particular difficult to publish, we present and provide a synthetic dataset that reflects real predictive maintenance encountered in industry to the best of our knowledge.*"

In addition to this, the code comes itself with longer explanations of how the functions behave and what they eventually plot. For simplicity, we will only check that the cumulative mean gets closer to the true value. This is because we would also need to check with the functions of Table 1, but it would imply having lots of images to present. Instead, we will report only the plots of a stable satisfactory example.

5 Application & Results

In this section we explore how different settings for Algorithms 3 and 4 converge to a common solution or diverge to different estimates. This procedure is mainly done through the graphs introduced in Table 2.

5.1 MH and Auxiliary Gibbs Burn-in lengths

An important parameter to consider is the burn-in length, corresponding to the amount of initial iterations before the actual values of the parameter of interest are recorded. In fact, during this phase the algorithms discard every value of the parameter and start to build the sample once the burn-in iterations are over. The purpose of this initial step is to make the chain move towards its ergodic distribution, so that the sample obtained after this step is representative of the true stationary distribution. What we can do to identify the number of iterations needed before the chain is indeed a representative one is to look at trace plots for both the algorithms and check at which point the chains show approximately a stationary behavior. For this purpose we ran 5000 iterations and zoomed on the trace plots of the first 500 for the Random Walk MH and the first 3000 for the Auxiliary Gibbs Sampling.

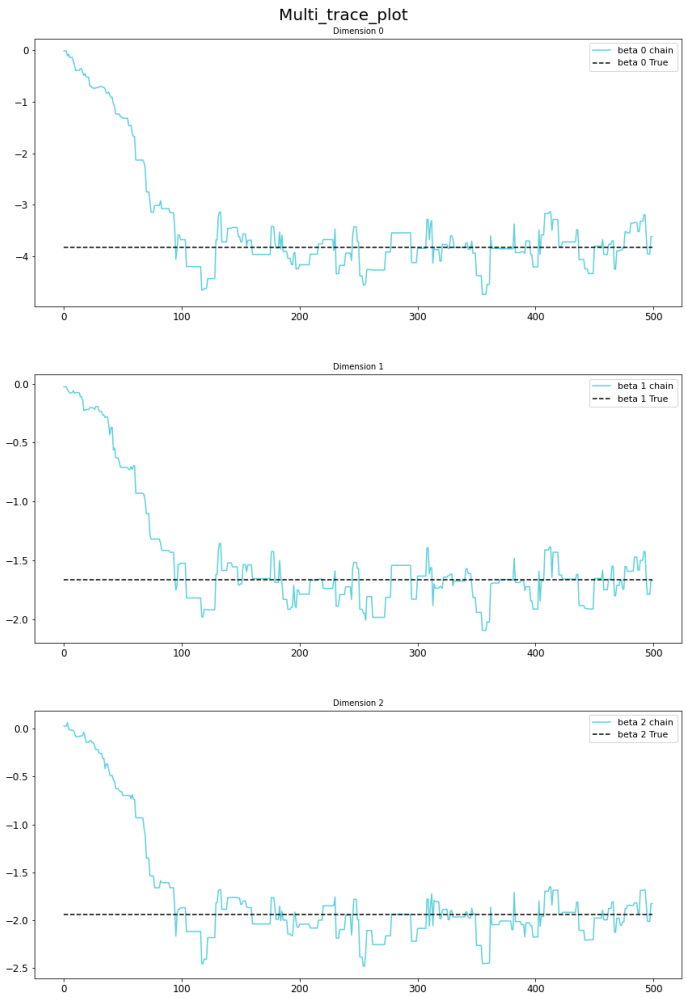


Figure 3: Trace plot Metropolis Algorithm. Burn-in=0, $\tau = 2$

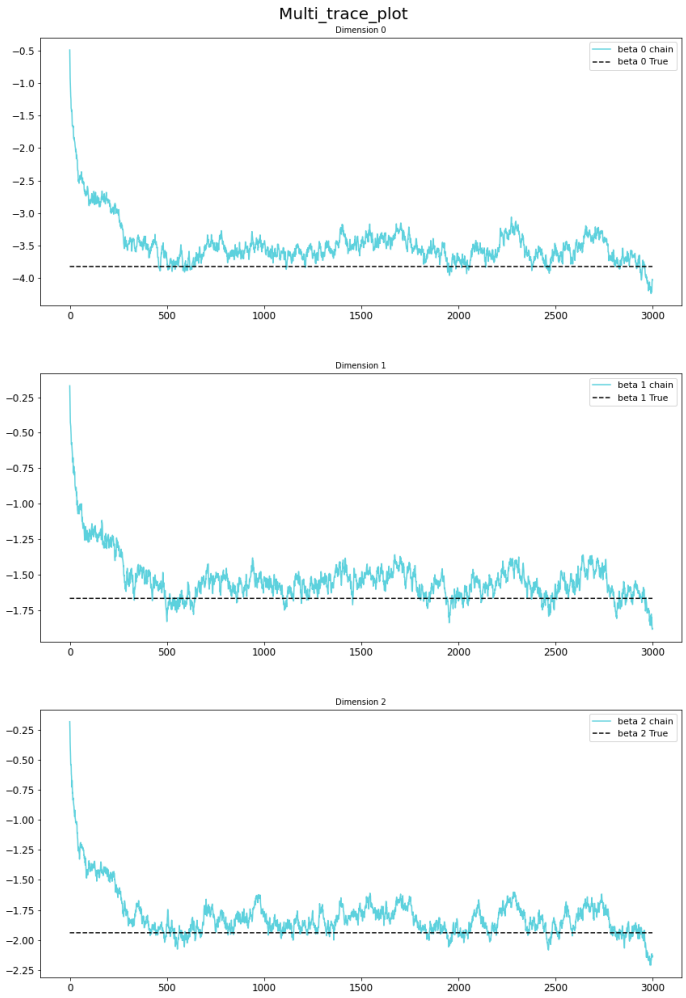


Figure 4: Trace plot Auxiliary Gibbs. Burn-in=0

Both algorithms have a starting value of $(0, 0, 0)$. By setting the burn-in parameter to 0, we can see the chain starting from the very beginning. From the plots above, it is clear how the Random Walk MH (with $\tau = 2$) reaches the stationary distribution earlier than the Auxiliary Gibbs. The former starts to move in the neighborhood of the true value of β after around 100 iterations. Given this observation, one could think to set the burn-in iterations around 150. For the Auxiliary Gibbs, we should set it between 500 and 1000, which is when the chain approximately assumes a more homogeneous path. However, the optimal burn-in parameter depends also on the starting point because the farther the starting point the more iterations the algorithms will need to reach the ergodic distribution.

Another interesting aspect is how different values of τ affect how long the chain takes to become stationary in the Random Walk MH. We know that the higher the τ, the higher the variance of the proposal and the more the chain explores farther from the previous value (at the cost of getting a lower acceptance rate due to more extreme values). This is confirmed by the fact that by increasing the τ to 3, the optimal number of burn-in iterations still stands around 150, while by setting it to 1 (hence, less variation in the proposal, determining the chain to move more slowly), stationarity is delayed to approximately 200-250 iterations.

5.2 MH-specific Parameters

Being that Algorithm 3 does not sample from full conditionals, more thoughtful choices have to be made for MH to be stable and efficient, as there are many elements interacting.

5.2.1 Different taus

One of the hypermarameter we can study is τ, which has an impact over the variance of the proposal in the Random Walk MH:

$$\beta_* \sim q(\cdot, \beta_t) = \mathcal{N}^{p+1}(\beta_t, \tau V) \ : \ \tau \in \mathbb{R}$$

Considering an application on a simulated dataset with $p + 1 = 3$ covariates, we observe a change in the acceptance rate as τ varies. The idea is that a lower τ corresponds to a lower variance and as a consequence the algorithm explores less the space of the chain. This means that there is a higher probability that the chain remains stuck in a region. On the contrary, a higher tau implies higher variance of the proposal. The chain widely explores the parameter space, inducing a lower acceptance rate. The following figure shows insights about this concept:

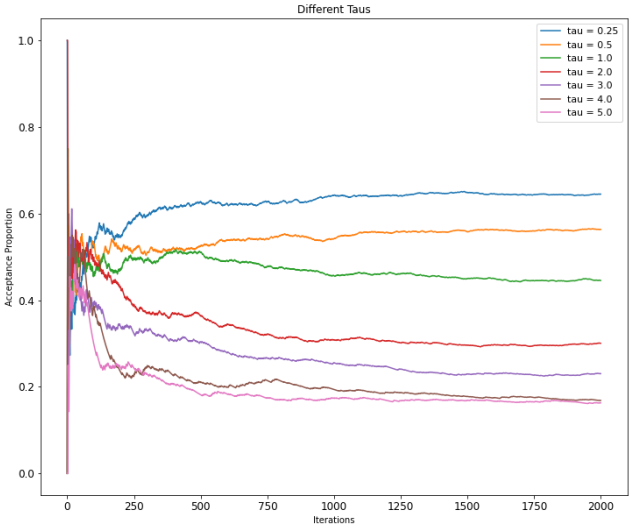


Figure 5: Acceptance rate at different taus

As previously said, it can be observed how the acceptance rate decreases as tau increases.
Regarding the convergence of the estimated parameters towards the chain, there are no strong differences to point out.
In [12] the authors advise to set $\tau = 2$, while in [13], our textbook, it is advised to set $\tau : \alpha(\cdot, \cdot) \approx 50\%$. We notice that the two approaches can be implemented as to agree with suggestions with good empirical results. As we will later see, in our setting we believe that $\tau = 2$ is the best choice for this parameter as it provides a good trade-off between the acceptance rate and mixing of the chain.

5.2.2 Different priors

Here we explore how starting from different priors affects the results of the algorithm. We considered four specifications:

- Multivariate normal distribution centered around the MLE
- Multivariate standard normal distribution
- Multivariate Student t distribution
- Uninformative prior

Where the last one assumes that each beta is equally likely. In the Gibbs Algorithm we propose all but the t student, for which we reroute the reader to [1].

For each of these, we studied the path of the cumulative mean along the iterations of the algorithm, as shown in the plot.

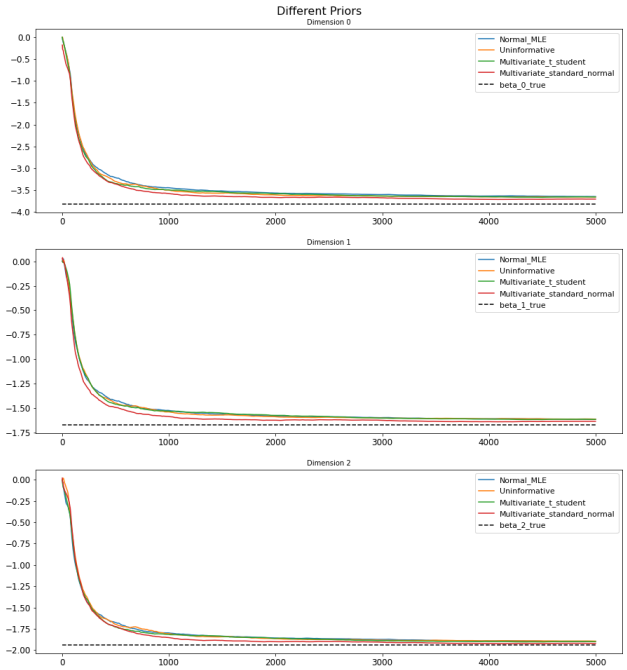


Figure 6: Cumulative mean along iterations for different priors

The trajectories look different, especially at the beginning of the chain, probably due also to the randomized nature of the algorithm. As the number of iterations increases and the algorithm enters the post burn-in phase, the cumulative means trends tend to align. This is true even for the dimensions where the estimated means are not very precise and cause some error.

Since we do not dispose of valuable information that suggests us a specific distribution for $\beta \in B$, we were particularly interested in the uninformative prior.
From this moment onwards, we will explore further its implications.

5.2.3 Different stretches

In the case of the Random Walk MH with a normal prior specification, another hyperparameter that could have an impact over our results is the stretch $s \in \mathbb{R}$ of the variance-covariance matrix[9]:

$$\pi(\beta) = \mathcal{N}^{p+1}(0, sI)$$

[9]It is possible to choose another mean, as for example the MLE estimate of β.

The idea in this setting is that a higher stretch corresponds to an higher variance of our distribution, that is we are considering a less[10] informative prior. Noninformative or weakly informative priors can be used as conventional default choices, playing a role as reference, even in presence of strong prior beliefs. In addition to this, for those who start with little knowledge, or when it is difficult to subjectively determine a reasonable prior distribution, they are the only choice
We tried different values of the stretch with $p + 1 = 3$ parameters and tau $= 2$:

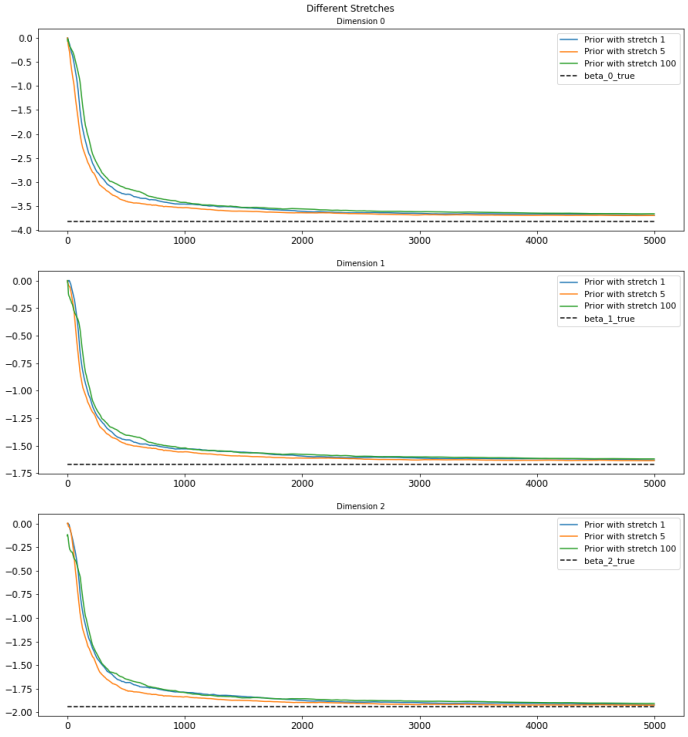


Figure 7: Comparison of beta estimation with different stretches

The graph (and also numerical results) suggests there are no relevant differences between the stretches. We can think about the choice of the stretch as "how much we are confident about the prior of beta", that is the more you are confident, the lower is the stretch (because you are reducing the variance).
In our setting this parameter does not have a real impact. Also considering other means and var-cov matrix we obtain similar results. This is line with the fact that changing the prior does not significantly affect the results. Nonetheless, it is a parameter that should be taken into account and carefully analyzed in other settings, especially when there are fewer observations available.

5.3 Gibbs Sampling Analysis of the priors

In the Gibbs sampling approach, as previously introduced, an important role is played by the prior distribution of β because this has an impact over the form of the full conditionals.
We provide a graph showing the differences among a diffuse prior and two informative priors $\mathcal{N}^{p+1}(\beta_0, V_0)$, where in one case we consider a normal centered at the MLE estimate of β and covariance matrix of the MLE, and a standard normal with a stretch of 10 (following the reasoning of section 5.2.3).

[10] Also *weakly* sometimes

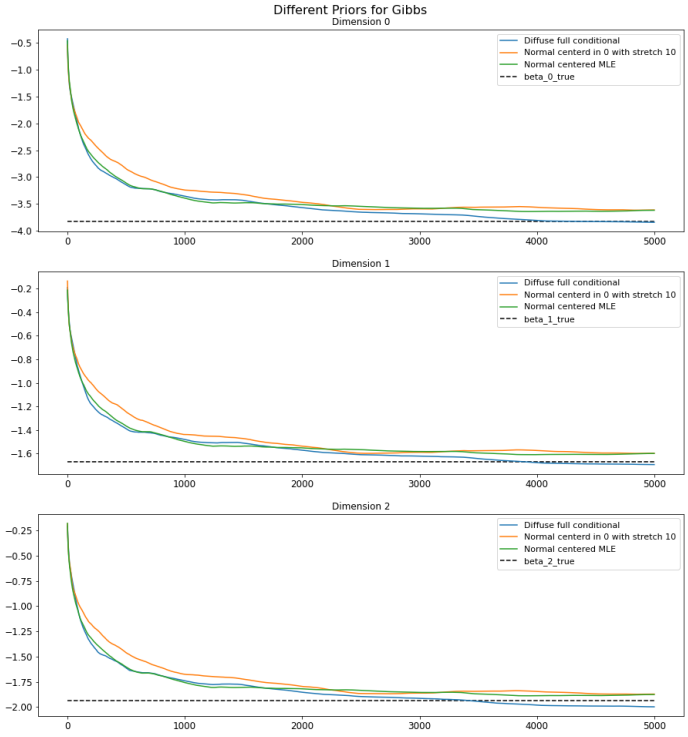


Figure 8: Comparison of beta estimation with different priors in Auxiliary Gibbs Sampling

The starting point was $(0, 0, 0)$. In this case we do not provide point estimates and MSE because running many times the algorithm we observed it is highly unstable. However, we can observe that the estimation is good independently from the choice of the prior. In particular the uninformative prior behaves well and this is good for the frameworks where we do not have any confidence about the prior distribution of our parameter.

5.4 Different starting points Random Walk and Gibbs

In order to verify the robustness of both Metropolis Hastings and Auxiliary Gibbs Sampling algorithms, we want to check their sensibility to starting points. Ideally, we would like to observe that both algorithms converge towards the true mean regardless of the starting point from which the chains start.
This might not be always true, depending on how far the starting points are from the true mean of interest.
From now on we will call the starting points of the chains β_0. We show an example of several chains starting from different β_0 by plotting how their cumulative means change across iterations.

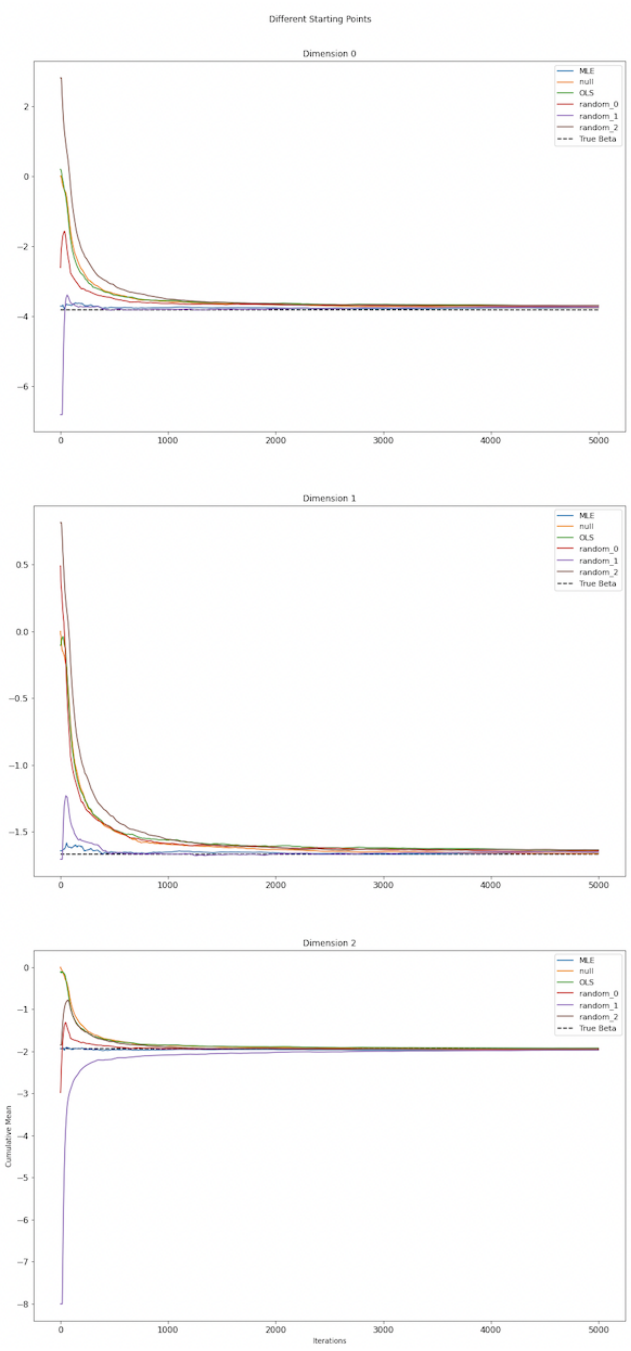


Figure 9: Comparison of beta estimation with different starting points β_0 in Metropolis Hastings

The starting points tested were the MLE approximation, the OLS approximation, a vector of β_0 equal to $(0, 0, 0)$ (null) and three random vectors. As we can see from the graphs above, the algorithm behaves well with all kinds of starting points presented.

The same testing procedure was performed using the Auxiliary Gibbs Sampling algorithm. As in the Random Walk MH, ideally we would like to observe that the chain converges to the true mean when starting from different β_0. We will once again look at how the cumulative mean changes with respect to where the chain was initialized.

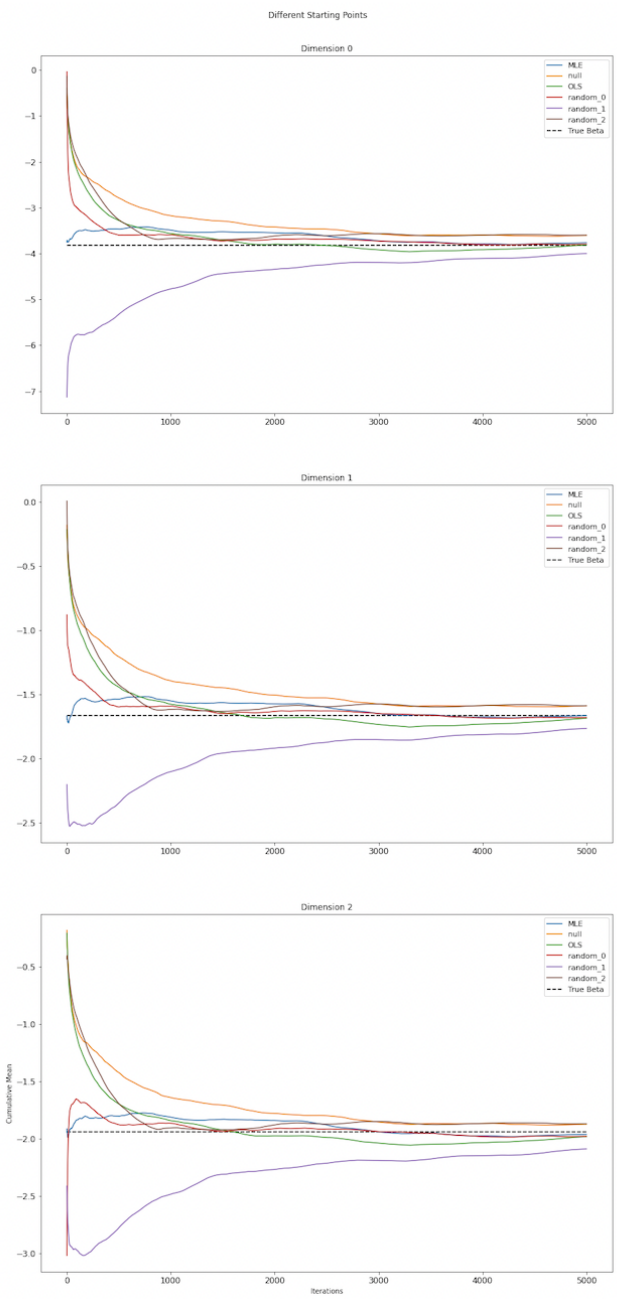


Figure 10: Comparison of beta estimation with different starting points β_0 in Auxiliary Gibbs Sampling

From the graph above we can claim that also the Auxiliary Gibbs Sampling algorithm works well in all cases. Differently from the Metropolis, the starting points seem to have a bigger influence on the speed of convergence. In fact, when the chain started from farther β_0, the chain moved more slowly towards its true value. Hence, in these cases it is recommended to raise the number of iterations to guarantee an accurate convergence.
Despite not being perfect, when provided with a starting point which is sufficiently close to the true mean, the Gibbs Sampling algorithm achieves convergence regardless of the starting β_0.

To conclude, both algorithms present pass sufficiently diagnostics independently of where they start. This is an advisable feature as it allows to have the guarantee that convergence is reached.

5.5 Random Walk Metropolis Hastings vs Auxiliary Gibbs Sampling: general comparison

After having analyzed how different selections of parameters affect the functioning of the two algorithms, we now run a comparison between them. In order to do so, a synthetic dataset of 1000 observations and 2 covariates (plus the constant) will be used.
First of all, let us look at the final hyper-parameters used to run this comparison. The Metropolis Hastings Random Walk is set with the following hyper-parameters: the prior used is the uninformative, as it is the most commonly accepted when no prior information is available (like in this case) and as we have previously seen different priors did not result in significant variations; τ equal to 2, as we have seen that it well balances the trade-off between exploration and acceptance rate (plus motivation coming from the book).
For the Auxiliary Gibbs Sampling algorithm, we decided to use an uninformative prior mainly because of the same reasons outlined for the Random Walk MH.
The other hyper-parameters left are the choice of burn-in iterations and starting points: since we wanted to make the comparison as fair as possible, we decided to use equal values for both algorithms: we set β_0 equal to $(0, 0, 0)$ and 1000 burn-in iterations, to ensure that both chains have reached their stationary distribution. In fact, we have seen before that both algorithms converge with this configuration of starting point and they both reach an ergodic distributions after 1000 burn-in iterations. Now we take a look at how both algorithms behave.
First we observe how the two algorithms explore the parameter space after the burn-in. They are both run for 10000 iterations after the burn-in. In this way we obtain a stationary distribution that has enough "time" to move around the parameter space. Below the trace plots of the two algorithms.

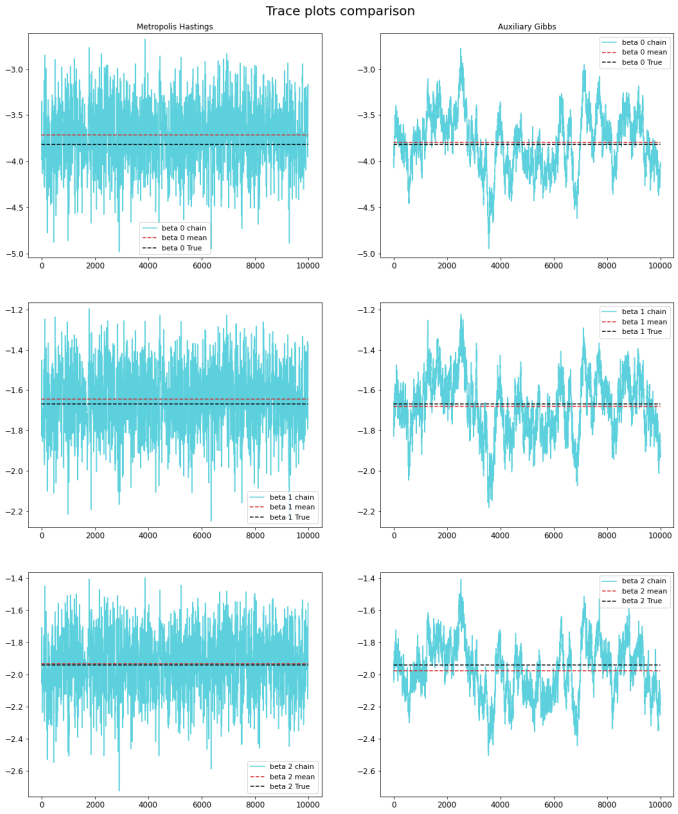


Figure 11: Comparison of trace plots between Random Walk MH and Auxiliary Gibbs Sampling

As we can see from the graph above both algorithms have reached the stationary distribution and are exploring the parameter space around the value of the true β. However, the two chains explore the state space in different ways: the Random Walk MH seems better at exploring the parameter space, moving quite far from the true mean but in a compact manner. Instead, Auxiliary Gibbs Sampling is more concentrated around the mean and it seems to explore different regions of the state space at different times. Let us now look at how these ways of exploring the state space affect the autocorrelation across iterations.
By inspecting the autocorrelation plots we can see whether the chains hide a type of periodicity, something that might invalidate the convergence towards the ergodic distribution.

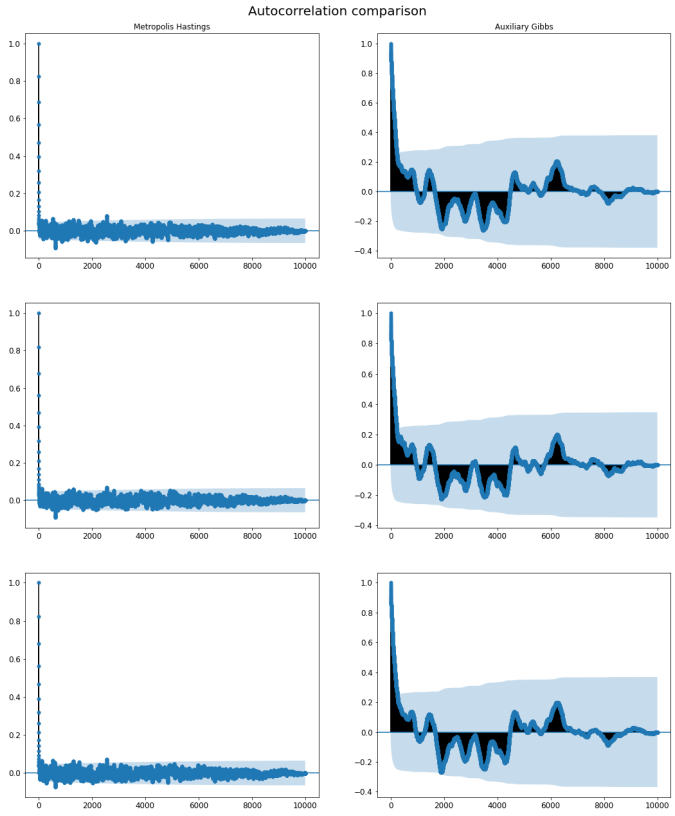


Figure 12: Comparison of autocorrelation between Random Walk MH and Auxiliary Gibbs Sampling

It seems that both algorithms do not display any type of auto correlation after the first iterations (as shown by the confidence intervals), which makes us safely say that both chains are ergodic. However, the values of the chain generated by the Random Walk MH seems to be less correlated among themselves, and this can also be seen by looking at the trace plots. Instead the Gibbs Sampling algorithm initially has a higher correlation which slowly converges towards 0.
Most importantly, we are interested in the convergence of the mean. Below, a graph showing the change in the cumulative mean after burn-in iterations.

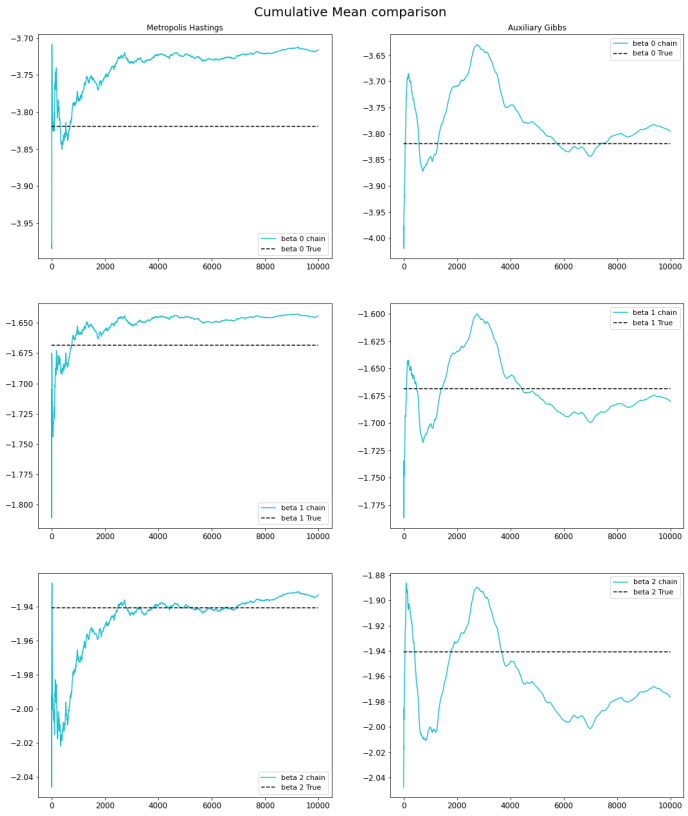


Figure 13: Comparison of cumulative mean between Random Walk MH and Auxiliary Gibbs Sampling

Both algorithms converge close to the true β (considering the graph shows a zoom on the cumulative mean). A big difference arises between the two algorithms: while the Random Walk MH stabilizes after a number of iterations, the Gibbs Sampling oscillates around the mean. This is probably due the fact that the latter tended to explore different parts of the parameter space at different times. However, even though the Gibbs Sampling shows a more fluctuating path, its approximation is actually more accurate at the end of the iterations: the Mean Square Error of the Random Walk MH is equal to 0.0038, while for the Gibbs Sampling it is equal to 0.0011. Below a table summarizing the final results that we obtained:

Algorithm	β_0	β_1	β_2
Random Walk	-3.71562272	-1.64425997	-1.93303917
Gibbs Sampling	-3.82536573	-1.68697015	-1.99398603
True Value	-3.81943841	-1.66849063	-1.94069479

Table 3: Estimations comparison on synthetic dataset

A note on the computational cost required by each algorithm. The Random Walk MH takes between 50 and 60 seconds to run 11000 iterations (1000 of burn-in + 10000 after burn-in), while the Auxiliary Gibbs Sampling takes on average 12 seconds to run the same iterations. Therefore the Gibbs Sampling is generally 5 times faster than the Metropolis, and this difference is mainly due to the fact that in the Random Walk MH the Fisher Information matrix needs to be computed at each iteration.
To conclude, in terms of efficiency, the Auxiliary Gibbs Sampling algorithm seems to perform better than the Random Walk MH: a more accurate approximation in less time. However, we cannot consider the efficiency of each algorithm only, but we must examine also their behaviour. The Random Walk MH

seemed to work "better" by looking at the trace and autocorrelation plots: a wider exploration of the state space with less correlation between values of the chain. Its biggest weak point is the computational cost: as we will see in a real world application of these two algorithms, by increasing the number of observations the difference in the time needed to conclude the same number of iterations widens enormously.

5.6 Machine Failure application

In this section, we apply the two algorithms over a real dataset, with machine failure data and numerical covariates. Its original source is [3] but it can be found on Kaggle [2]. The dataset consists of 10 000 data points stored as rows with 14 features in columns:

- UID: unique identifier ranging from 1 to 10000 productID: consisting of a letter L, M, or H for low (50% of all products), medium (30%), and high (20%) as product quality variants and a variant-specific serial number.
- Air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K.
- Process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K.
- Rotational speed [rpm]: calculated from powepower of 2860 W, overlaid with a normally distributed noise.
- Torque [Nm]: torque values are normally distributed around 40 Nm with an $\ddot{\text{I}}\text{f} = 10$ Nm and no negative values.
- Tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process.
- Target : Failure or Not.
- Failure Type : Type of Failure.

We decided to study whether there is a failure or not (*Target* variable) based on five attributes: Air temperature (β_1), Process temperature (β_2), Rotational speed (β_3), Torque (β_4), Tool wear (β_5). Then, in the estimation of the parameters there will be also β_0, which indicates the intercept.

As already argued in Section 4, we believe this is a good scenario to test our algorithms and analyze real information. An example of previous Bayesian - MCMC work applied to a similar problem is [9].

Previously, discussing different prior specification options, we pointed out how a non-informative approach can be used with lack of information. Given that we have no information about the distribution of parameters, we chose to use uninformative priors for both algorithms.
We select as a starting point the null vector. This is because in the previous analyses we showed how it could be a good starting point for both algorithms and, again, as we do not have any suggestion about the possible structure of our parameters of interest.
We run the algorithms using 1000 iterations of burn-in and 5000 after burn-in. Regarding the other hyperparameters, for the random walk approach we used $\tau = 2$ (we showed previously this may be a good suggestion). Indeed, using non-informative priors, these are the only additional parameters to establish in advance.
The results, together with the estimates obtained with Maximum Likelihood, are the following:

Algorithm	β_0	β_1	β_2	β_3	β_4	β_5
Random Walk	-17.388	0.385	-0.370	0.00538	0.131	0.00645
Gibbs Sampling	-17.455	0.381	-0.366	0.00536	0.131	0.00643
MLE	- 17.449	0.384	-0.369	0.00536	0.131	0.00643

Table 4: Estimations comparison on machine failure's dataset

It can be seen how the point estimates of the parameters are not much different. Indeed, we also notice that the intercept coefficient has a higher magnitude than the others.
At this point we propose a plot to understand whether in both algorithms the chains are mixing well:

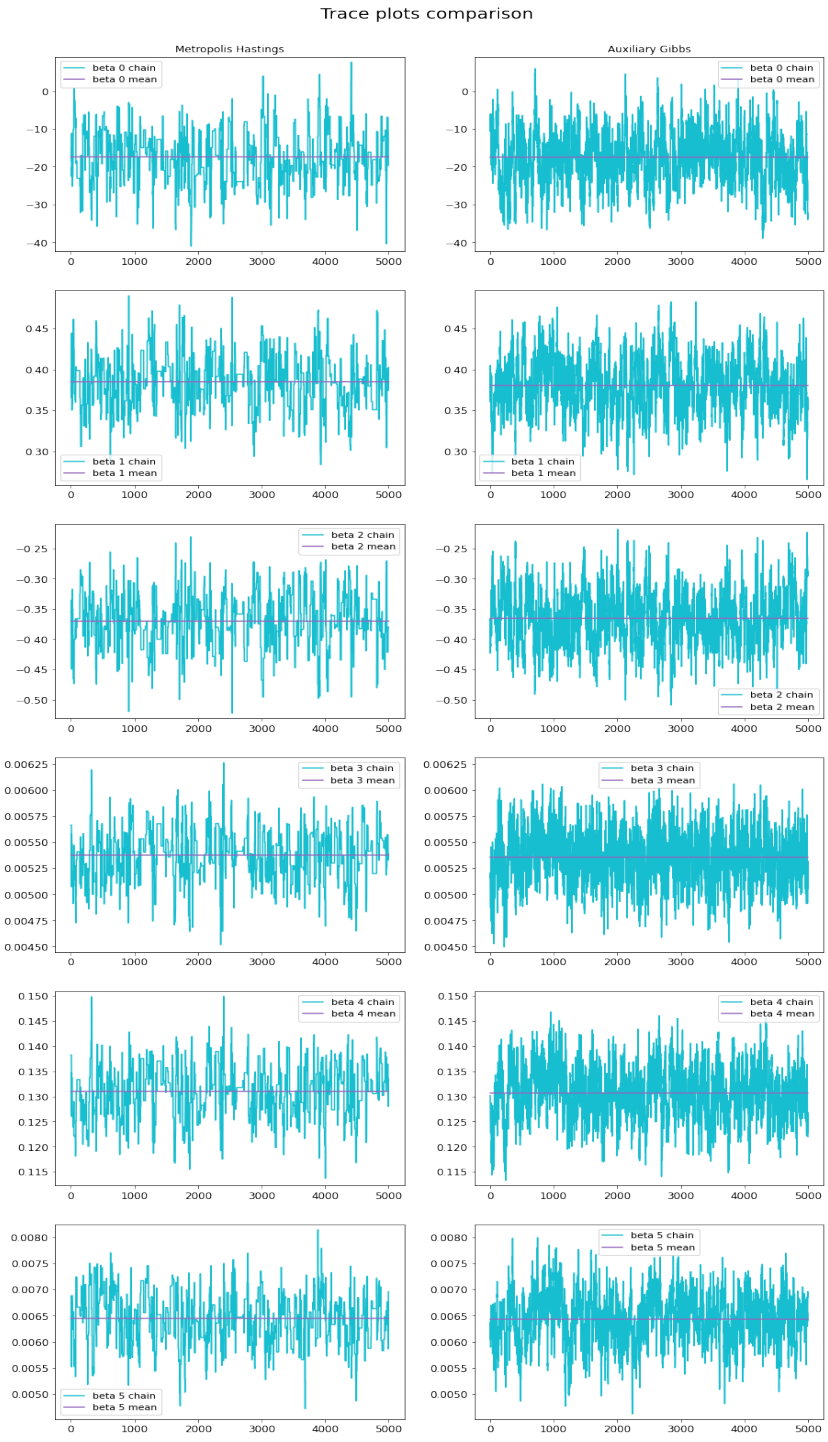


Figure 14: Comparison of the chains

This is indeed the case. However, the impression is that in the Gibbs Sampling *more* stationary chains are obtained.
In order to investigate this further, we can also take a look at the autocorrelation plots for each dimension:

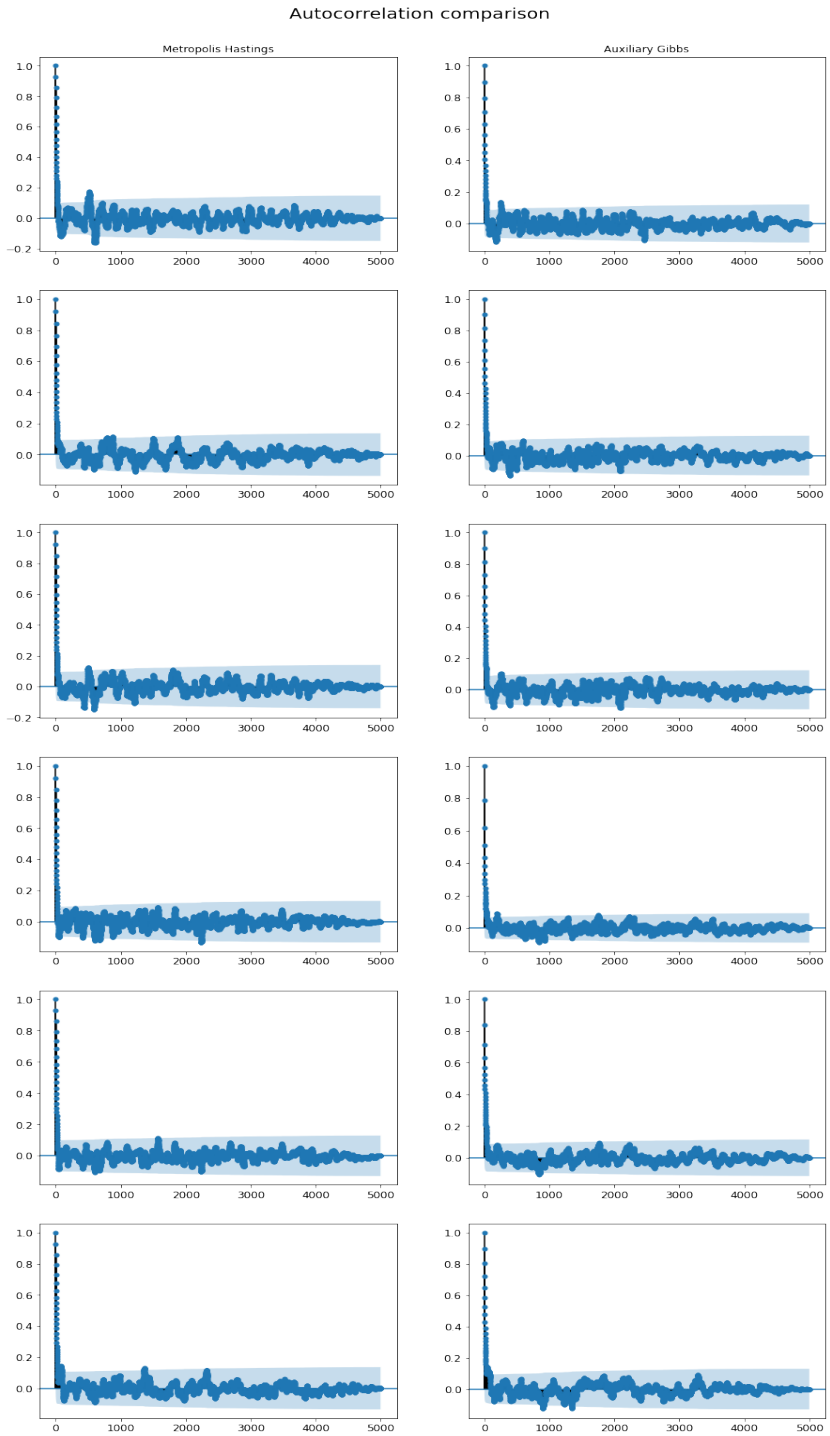


Figure 15: Autocorrelation plots comparisons

As we can see, there are weak correlations in both algorithms. This suggests, together with the previous observation, that the obtained chains are stationary.Up to this point, both algorithms behave similarly.

Despite this, there is a strong difference worth pointing out: the execution time. Considering our dataset with 10000 observations, 5 covariates (plus the intercept), 1000 iterations of burn-in and 5000 after burn-in iterations, the Gibbs sampling converges in less than one minute, while the Random Walk takes around

30 minutes. As pointed out in section 5.5, the difference in computational cost arises with the number of observations.

We can also observe differences in the distribution of the sampled beta:

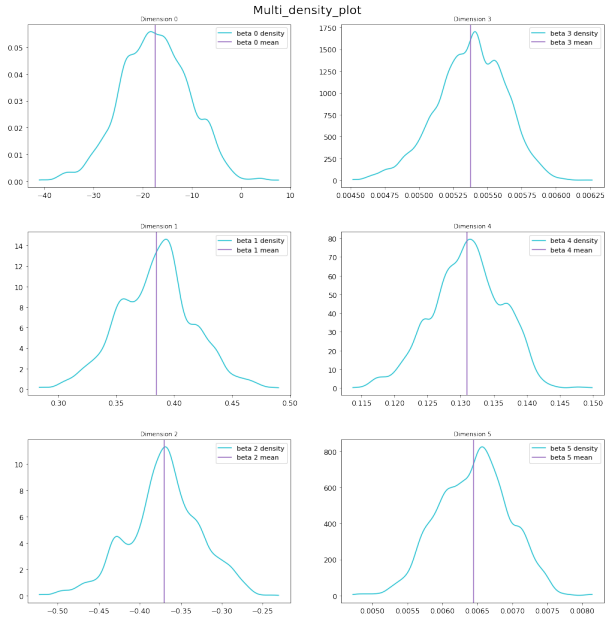


Figure 16: Density plot of Random Walk MH sample

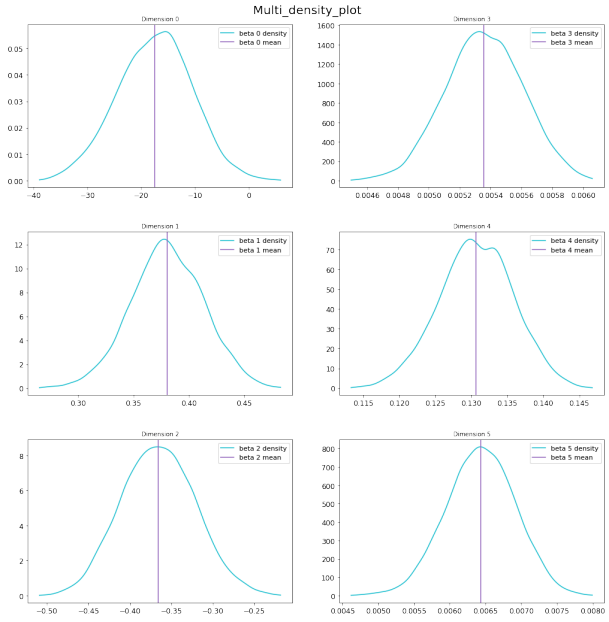


Figure 17: Density plot of Gibbs sample

All the parameters in the Gibbs Sampling algorithm present a bell shaped distribution. This is not the case for the Random Walk result, indicating a worse mixing of the chain in this scenario.

Finally, we run again the algorithms over a subset of our starting dataset, that is considering only 1000 observations. This made us discover that the Gibbs sampling method behaves better when using a large

amount of data. Indeed, the chain does not mix well anymore. Both trace and autocorrelation seem to show serial trends:

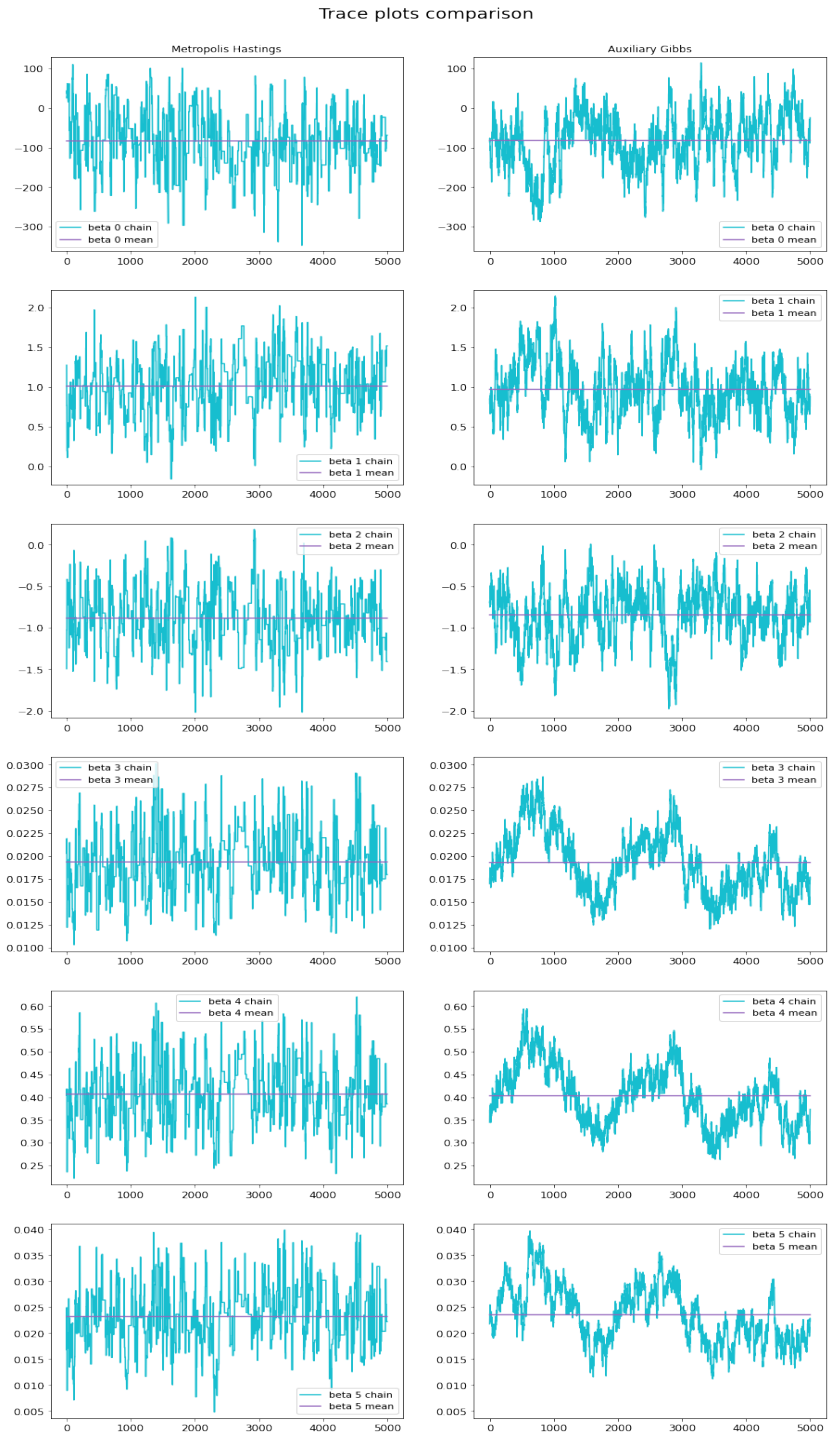


Figure 18: Trace plots comparisons 1000 observations

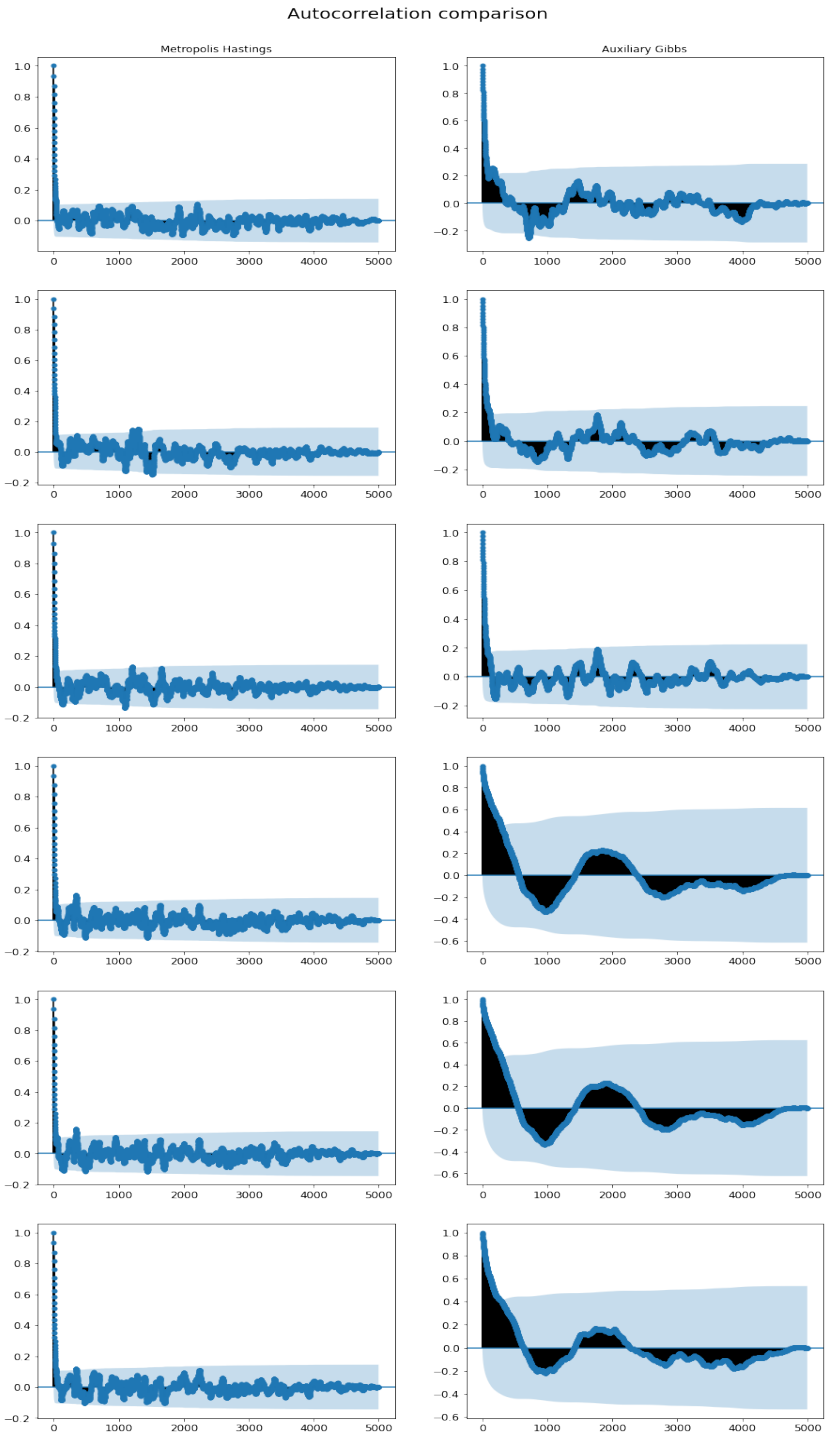


Figure 19: Autocorrelation plots comparisons 1000 observations

This results might indicate why we observed a better exploration of the Random Walk MH with the synthetic dataset used before, as in that case only 1000 observations were used. We believe this might be one of the reasons that make Random Walk MH a better option when working with small samples and instead opt for the Gibbs Sampling algorithm with larger datasets. However, further investigation should be needed to test whether our hypothesis can be confirmed, since this question is out of the scope of this paper.

6 Limitations & Conclusion

After a general theoretical introduction to the topic of the Linear Model, Bayesian Probability, and MCMC methods, we propose two algorithmic approaches to iteratively identify the coefficients of a Probit model with different choices of the agents involved, where that of Algorithm 4 is inspired from the ideas proposed in our main paper of reference [1]. The results are extracted from a careful procedure of derivation of properties, with a sequence of definitions and theorems that can serve as a basis for understanding what is required for the two methods.

Using software we design a system to estimate in a simulated scenario and a real scenario. We observed some differences based on the kind of application the algorithms were used for (the Random Walk MH showed a better exploration of the parameter space with the synthetic dataset, while the opposite happened for the real-case scenario), nonetheless the results show that both algorithms, once fine-tuned, return the desired results. To identify possible variations in the convergence, we implement a graphical analysis approach and comment the plots obtained. This pipeline is indeed adaptive to any similar problem, with a useful Python script for other potentially interesting problems.

One of the biggest weaknesses of the analysis, where by weakness we mean unexplored direction of study, is error analysis. We are aware that throughout the procedure we never attempted to provide estimation errors for the MCMC procedures. Hopefully, we will have time to study and apply methods including this branch in future projects.

We also recognize that the implementation of block sampling in Algorithm 4, argued by [1], [5] could have been improved with a sampling method for the individual parameters. This was not done in our main paper of reference and we chose to avoid it.

Other constraints we noticed were the computational power and adaptability of the exploration part to different problems. Indeed, the former is a limitation in terms of available computational resources, which allowed us to run the chain for a *not so big* number of iterations. The latter could have been designed differently.
The way in which the script is designed requires the user to autonomously find suitable parameters by analyzing plots' results. This could in some cases have been avoided by implementing methods such as the one described for the burn-in length choice in [8]. Due to the time constraint of this project, we chose to avoid this, opting for a case by case approach.

Lastly, another prior for Algorithm 4 could have been extracted as proposed by [1]. This would have required deriving another full conditional distribution. In these terms, we also observe that instead in the case of Algorithm 3 we only attempted to analyze the results with a prior t student with $dof = 3$, while we could have considered different specifications.

In conclusion, we are confident that the methods proposed present some strengths, while there is lots of ideas to improve them. The subject appears to be full of opportunities for study and applications, making this project a small but well-posed exploration of options.

References

[1] James H. Albert and Siddhartha Chib. "Bayesian Analysis of Binary and Polychotomous Response Data". In: *Journal of the American Statistical Association* 88.422 (June 1993), pp. 669–679. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.1993.10476321. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1993.10476321 (visited on 01/03/2022).

[2] *Machine Predictive Maintenance Classification*. URL: https://kaggle.com/shivamb/machine-predictive-maintenance-classification (visited on 01/03/2022).

[3] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[4] Xi He et al. "1 Multivariate Normal Distribution". In: (), p. 5.

[5] Claus Skaanning Jensen and Augustine Kong. "Blocking Gibbs Sampling for Linkage Analysis in Large Pedigrees with Many Loops". In: *The American Journal of Human Genetics* 65.3 (Sept. 1, 1999), pp. 885–901. ISSN: 0002-9297. DOI: 10.1086/302524. URL: https://www.sciencedirect.com/science/article/pii/S0002929707623398 (visited on 01/05/2022).

[6] Martin A. Tanner and Wing Hung Wong. "The Calculation of Posterior Distributions by Data Augmentation". In: *Journal of the American Statistical Association* 82.398 (June 1987), pp. 528–540. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.1987.10478458. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1987.10478458 (visited on 01/04/2022).

[7] Luc Devroye. *Non-uniform random variate generation*. New York Heidelberg: Springer, 1986. 843 pp. ISBN: 978-0-387-96305-1 978-3-540-96305-9 978-1-4613-8645-2.

[8] Charles C. Margossian, Matthew D. Hoffman, and Pavel Sountsov. "Nested $\hat R$: Assessing Convergence for Markov chain Monte Carlo when using many short chains". In: *arXiv:2110.13017 [stat]* (Oct. 25, 2021). arXiv: 2110.13017. URL: http://arxiv.org/abs/2110.13017 (visited on 01/03/2022).

[9] B. Pavlyshenko. "Machine Learning, Linear and Bayesian Models for Logistic Regression in Failure Detection Problems". In: *arXiv:1612.05740 [cs, stat]* (Dec. 17, 2016). arXiv: 1612.05740. URL: http://arxiv.org/abs/1612.05740 (visited on 01/03/2022).

[10] Sebastian Funk et al. *mcmc_diagnostics.utf8.md*. URL: http://sbfnk.github.io/mfiidd/mcmc_diagnostics.html (visited on 01/03/2022).

[11] Joseph Moukarzel. *Joseph94m/MCMC*. original-date: 2018-11-09T20:04:03Z. Jan. 1, 2022. URL: https://github.com/Joseph94m/MCMC/blob/2cc46cd7f1562c7a38bf8a4eba2148d05e5099c6/MCMC.ipynb (visited on 01/03/2022).

[12] Adam Johansen and Ludger Evers. *Monte Carlo Methods*. University of Warwick. URL: https://warwick.ac.uk/fac/sci/statistics/staff/academic-research/johansen/teaching/mcm-2007.pdf.

[13] Larry Wasserman. "Simulation Methods". In: *All of Statistics: A Concise Course in Statistical Inference*. Ed. by Larry Wasserman. Springer Texts in Statistics. New York, NY: Springer, 2004, pp. 403–433. ISBN: 978-0-387-21736-9. DOI: 10.1007/978-0-387-21736-9_24. URL: https://doi.org/10.1007/978-0-387-21736-9_24 (visited on 01/06/2022).