Métodos Monte Carlo basados en cadenas de Markov

David Pérez Ros

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Directores del trabajo: Ana Carmen Cebrián Guajardo
y Jorge Castillo-Mateo
27 de junio de 2022

Prólogo

En el ámbito de la estadística bayesiana, los *métodos de Monte Carlo basados en cadenas de Markov* (métodos MCMC) son un conjunto de métodos computacionales cuyo principal objetivo es obtener muestras de distribuciones probabilísticas que pueden ser muy complejas y pueden no corresponder a distribuciones conocidas. Con estas muestras el principal objetivo es hacer inferencia.

Dichos métodos están basados en la integración de Monte Carlo y las cadenas de Markov. Según Metropolis [17], los precursores de los métodos de Monte Carlo fueron los físicos nucleares Stanislaw Ulam y John von Neumann, que trabajaban en 1944 en un proyecto de armas nucleares en Los Álamos, EEUU. Dado que su aplicación estaba relacionada con un proyecto nuclear secreto, el también físico nuclear Nicholas Metropolis propuso el nombre en clave *Monte Carlo* para este método, en relación a la dependencia que tenía un familiar suyo al casino de Monte Carlo. En los juegos del casino, estamos buscando la probabilidad de ganar frente al crupier. Mediante la computación estadística se pueden simular un número grande de partidas y sumando el número de partidas ganadas entre el numero total de partidas simuladas, se tiene un estimador de la probabilidad de ganar al juego. Informalmente, esta es la base de la integración de Monte Carlo.

Aunque los métodos de Monte Carlo y los métodos MCMC fueron ambos propuestos alrededor de los años 50, sólo los métodos de Mote Carlo gozaron inicialmente de popularidad y una gran aceptación en el ámbito estadístico. No fue hasta los años 90 cuando gracias a Gelfand y Smith (1990) [20], los métodos MCMC se empezaron a popularizar en el ámbito de la estadística bayesiana, aunque hubieran escrito sobre ellos previamente autores como Hastings (1970) [16] o Geman y Geman (1984) [21]. Esto fue debido a que a finales de los 90 con los grandes avances en la computación ya era posible construir algoritmos computacionales con las características de estos y además la presentación del software BUGS (Bayesian inference Using Gibbs Sampling) en 1991 que facilitaba su implementación, ayudó a su propagación.

El principal objetivo del presente trabajo es desarrollar la teoría de los métodos MCMC en el ámbito bayesiano. Es por ello que en el Capítulo 1 se comienza dando nociones sobre la estadística bayesiana y las cadenas de Markov. En el Capítulo 2 se exponen métodos de generación de variables aleatorias, y tras ver sus inconvenientes, se introducen los métodos MCMC. Finalmente, en el Capítulo 3 se muestra una aplicación de los métodos MCMC en un caso práctico del ámbito bayesiano. En la literatura hay múltiples publicaciones y libros sobre resultados teóricos y aplicaciones de estos métodos, la mayoría con diferentes notaciones. Es por ello que con el propósito de sintetizar e intentar hacer comprensible al lector los dos primeros capítulos, se ha optado por seguir principalmente el hilo narrativo de Robert y Casella [1], y Gelman et al. [2], complementando los resultados con el resto de publicaciones/libros mostrados en la bibliografía. En los anexos se pueden encontrar teoremas usados y el código de R elaborado para implementar los ejemplos a lo largo del trabajo junto a las gráficas derivadas de estos códigos, que debido a su extensión es razonable incluirlas en este apartado.

Antes de dar paso al trabajo, agradecer a mis directores de TFG, Ana Carmen Cebrián y Jorge Castillo-Mateo su implicación y disposición a lo largo del proceso. También me gustaría nombrar en estas lineas a mi familia y amigos que me han acompañado (y aguantado) a lo largo de estos cuatro años de carrera y sin los que esta etapa de mi vida no hubiera sido lo mismo, y que además por compromiso o interés acabarán leyendo este trabajo.

Summary

Bayesian inference is the branch of statistics that deals with the fact that the unknown parameter (single or multi-dimensional) is considered as a random variable. As we will see, a wide and strong statistical theory has been developed in the Bayesian field since Thomas Bayes laid the foundations of it in 1763. In this way, the main objective of this work is to use Bayesian statistical tools together with Monte Carlo Integration and Markov chain theory in order to present and develop *Monte Carlo Methods based on Markov chains* (MCMC methods). In fact this work is named after those methods. The importance given to this methods lay on the fact that they are able to generate random samples from complex probability distributions and therefore, they can be used to approximate or evaluate expressions which are mathematically difficult to evaluate. Not only did this methods become popular in the last 20 years with the development of computers, but also their usage is becoming more and more recurrent in different sectors since this methods can be used to solve a lot of problems, apart from the Bayesian ones.

To this aim, we begin in the first half of Chapter 1 giving a detailed description about Bayesian theory, and setting a comparison with the frecuentist one, based on the work made by Robert and Casella [1, Chap. 1-3] and Gelman et al. [2, Chap. 1-3] in their respective books. The purpose of this section is to review the main and most important concepts of Bayesian statistics and let the reader to fully understand the Bayesian basis of MCMC methods. Suppose we have a certain distribution with unknown vector of parameters θ. In Bayesian statistics we consider the parameter $\theta = (\theta_1, \cdots, \theta_p)$, as a random vector with a certain distribution. The *prior distribution of* θ aims to summarize all previous information about the parameters in the shape of a probabilistic distribution, before taking a sample of the data. It is denoted by $p(\theta)$. Once we have a sample $y = (y_1, \cdots, y_n)$ and the prior distribution, it can be obtained the *posterior distribution of* θ, denoted by $p(\theta|y)$ which is in fact an update of the prior distribution via the sample $y = (y_1, \cdots, y_n)$. The way to calculate this posterior distribution is given by:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}, \tag{1}$$

where $p(y|\theta)$ is called *the likelihood function*. This distribution can be used for doing inference about the parameter. We will also present some estimators for θ from the posterior distribution such as: the *maximum a posterior probability* (MAP) which is $\hat{\theta}_{\text{MAP}}(y) = \arg\max_{\theta} p(\theta \mid y)$, and the *posterior mean* $\hat{\theta}_{\text{MEAN}}(y) = \int \theta p(\theta \mid y) d\theta$. In case of θ being one-dimensional, the *credible interval for* θ *at level 1-*α is defined as $[a,b]$, with a,b satisfying that:

$$P\{a \leq \theta \leq b \mid y_1, \ldots, y_n\} \geq 1 - \alpha. \tag{2}$$

In the second half of Chapter 2, we will make a brief review about Markov chains and the theory behind them, which are going to be essential for a fair understanding of the simulation methods based on Markov chains that will be seen later. A *Markov chain* is defined as a sequence of random variables $X^{(0)}, X^{(1)}, \cdots, X^{(t)}, \cdots$ such that the probability distribution of $X^{(t)}$ given the previous values, depends only on $X^{(t-1)}$. Mathematically,

$$X^{(t+1)} \mid X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(t)} \sim K\left(X^{(t)}, X^{(t+1)}\right), \tag{3}$$

where $K(x,y)$ is the *transition kernel*.

In this part we show basic properties of Markov chains and we focus on a certain type of Markov chains, called *Ergodic chains*. The relevance of this type of chain relies on the fact that up to the limit, this chains have always a stationary distribution and it is unique. We also talk about *reversible Markov chains*, which are the ones that satisfy the *detailed balance condition*:

$$f(x)K(x,y) = f(y)K(y,x) \quad \forall x,y \tag{4}$$

Satisfying the detailed balance condition ensures that the distribution $f(x)$ is the stationary distribution of the Markov chain generated by the kernel $K(x,y)$.

In Chapter 2, we will see that MCMC methods combine both Monte Carlo integration and simulation methods based on Markov chains. Numerical integration is quite a big problem for developing computational bayesian methods because it is needed to evaluate integrals of complex functions in order to generate measures of interest for the posterior distribution, and in most cases those integrals can't be solved analytically. For solving this problem we use Monte Carlo integration which aims to estimate the expectation of a function h of θ. The estimator is the next:

$$\mathrm{E_p}(h(\theta) \mid y) = \int_\Omega h(\theta) p(\theta \mid y) d\theta \;\approx\; \frac{1}{n}\sum_{j=1}^{n} h\left(\theta^{(i)}\right), \tag{5}$$

where $\left(\theta^{(1)}, \cdots, \theta^{(n)}\right)$ is a sample from $p(\theta|y)$ obtained by a stochastic method for example. We present the *reject sampling* and the *importance sampling* as two stochastic methods. While the importance sampling gives an estimator for the expectation of a function of the variable, the reject sampling gives a sample of the posterior distribution for the parameter. As it will be shown in the chapter, when the parametric dimension goes up it is rather convenient to develop specific types of methods which could be able to work better according to higher dimensions, and those methods are the MCMC. Another inconvenient for the use of stochastic methods is that for their implementation we must evaluate $p(\theta|y)$ at some points, and unfortunately it can be complicated or take a lot of computation time in many cases.

In the second half of Chapter 2, all the battery of Bayesian and Markov chains tools explained in the previous parts of the work are going to be used for developing two of the most extended MCMC methods: *The Gibbs-Sampler* and the *Metropolis-Hastings*. One of the most appealing features of the MCMC methods is that they can be implemented even if the posterior $p(\theta|y)$ is very complicated. They work as it follows: they generate a Markov chain for θ, approximating in each iteration the value given as a sample from the posterior distribution and at stationarity the values given are from the posterior distribution. Furthermore, it will also be explained how to get an hybrid method, combining both Metropolis-Hastings and Gibbs-Sampler in order to solve a problem. Regarding the stationarity, it is needed to check that the chain has converged to the posterior distribution, and only then, sampling from the chain will be equivalent to sampling from $p(\theta|y)$, which is the main objective of this methods. To this purpose, we will see some tools to check the convergence of the chains.

Finally, in Chapter 3 we are going use all the tools developed in the whole work in order to solve a practical problem. We want to fit the daily maximum temperatures of the city of Pamplona to an autoregresive model with help of the Bayesian computation. We will need to develop a combination of a Metropolis and a Gibbs Sampler algorithm in order to solve the problem. To this aim, we will define some functions in R, which will be used in the implementation of the algorithm. Once the code is executed, and we have the output chains, we will check that all the chains have converged and that we have successfully solved the problem.

Índice general

Capítulo 1

Introducción al análisis bayesiano y a los métodos MCMC

Los métodos de Monte Carlo basados en cadenas de Markov (*métodos MCMC*) son un conjunto de algoritmos que permiten obtener muestras con una determinada distribución de probabilidad, incluso con una expresión compleja. La base de estos algoritmos se fundamenta en la construcción de una cadena de Markov cuya distribución estacionaria sea la distribución de interés.

El desarrollo de estos métodos aparece ligado al desarrollo del análisis bayesiano, puesto que dichos métodos han hecho posible la implementación de complejos modelos jerárquicos que constan de un elevado número de parámetros desconocidos y que no podrían estimarse utilizando métodos estrictamente analíticos.

Para comenzar, en este capítulo se presentan las bases para el desarrollo de este tipo de métodos. En primer lugar se introducen los conceptos básicos del análisis bayesiano y las dificultades que presenta la estimación de modelos complejos. En segundo lugar se revisa la definición y propiedades de las cadenas de Markov, que serán relevantes para el desarrollo de los métodos MCMC.

1.1. Análisis Bayesiano

La estadística bayesiana es una rama del campo de la estadística basada en una interpretación no frecuentista de la probabilidad, donde la probabilidad representa un grado de creencia sobre un cierto suceso o evento. A diferencia de la estadística frecuentista, los parámetros dejan de considerarse como constantes y se representan mediante variables aleatorias (v.a.'s).

1.1.1. Introducción y objetivos del análisis bayesiano

Esta rama de la estadística debe su nombre a Thomas Bayes, el cual formuló en 1763 un caso específico del ahora conocido como *Teorema de Bayes*, que es la base de este tipo de análisis. Posteriormente fue Pierre-Simon Laplace en 1774 y después en 1812 el que en su libro *Théorie analytique des probabilités*, formuló la la probabilidad a posteriori (actualizada) a partir de la probabilidad a priori, vía el uso de datos o evidencias. Su trabajo se puede considerar como una extensión de los resultados de Bayes (véase en [4, pág. 268]), siendo Laplace el primero en sentar las bases de esta rama de la estadística.

La estadística bayesiana ofrece una interpretación de la probabilidad que describe el grado de certidumbre sobre un evento. Si la probabilidad es 1 se está totalmente seguro de que el evento es cierto, así como se está seguro de que no lo es si la probabilidad es 0. Asimismo, como ya se ha comentado, todo el desarrollo teórico del análisis bayesiano se basa en el resultado del Teorema de Bayes, el cual se enuncia como sigue:

Teorema 1.1.1 (Teorema de Bayes)**.** *Sean* $\{A_1, A_2, \ldots, A_i, \ldots, A_n\}$ *sucesos disjuntos, que forman una partición del espacio total y tales que* $P(A_i) \neq 0 \; \forall i = 1, 2, \ldots, n$*. Sea B un suceso tal que se conocen*

$P(B \mid A_i)$, entonces la probabilidad $P(A_i \mid B)$ viene dada por la expresión:

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{P(B)}. \tag{1.1}$$

Notar que la probabilidad marginal de B se puede calcular utilizando la *ley de probabilidad total* y con esto se obtiene

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_{k=1}^{n} P(B \mid A_k)P(A_k)}. \tag{1.2}$$

Diferencias con la estadística clásica (o frecuentista)

1. La estadística clásica está basada en una interpretación frecuentista de la probabilidad. En un experimento repetible la probabilidad de un suceso es el límite de la proporción de ocurrencias del suceso en n repeticiones del experimento cuando $n \to \infty$. En la estadística bayesiana la probabilidad representa la certidumbre sobre un suceso, que no frecuencias limite.

2. Mientras que en la estadística clásica los parámetros son constantes desconocidas pero fijas y la aleatoriedad viene del muestreo, en la estadística bayesiana los parámetros son v.a.'s. Es decir, en la inferencia clásica la variabilidad proviene del muestreo, y tras caracterizar la distribución del estimador se realiza inferencia utilizando dicha distribución. Por el contrario, en la estadística bayesiana el parámetro es aleatorio y cuando se hace inferencia es utilizando la distribución del parámetro.

1.1.2. Inferencia bayesiana

Como en la estadística bayesiana los parámetros se consideran variable aleatorias, el primer objetivo es caracterizar su distribución de probabilidad. Para ello se dispone de dos fuentes de información, un posible conocimiento previo sobre la distribución del parámetro, que se especifica en la *distribución a priori del parámetro*, y la información que proporciona la muestra observada, que se incluye a través de la *verosimilitud*. El enfoque bayesiano permite unir la información de ambas fuentes mediante la *distribución a posteriori* $p(\theta \mid y)$.

Distribución a posteriori de θ

Como ya se ha comentado, la *distribución a posteriori del parámetro* $p(\theta|y)$, se define como la distribución del parámetro θ obtenida mediante una actualización de la distribución a priori $p(\theta)$, en base a la muestra de datos $y = (y_1, \ldots, y_n)$ tomada. A continuación se fija la notación usada en la memoria, antes de explicar en el teorema siguiente como se calcula la distribución a posteriori, y mostrar que esta es una aplicación directa del *Teorema de Bayes*.

- $\theta = (\theta_1, ..., \theta_p)$ es el vector de parámetros de interés. En general los parámetros se representan por letras griegas con p componentes representando cada componente un parámetro.

- $y = (y_1, ..., y_n)$ es la muestra de datos de tamaño n tomada de la v.a. Y.

- $p(\theta)$ es la *distribución a priori* del parámetro. Representa nuestro grado de creencia sobre los valores que puede tomar el parámetro θ antes de observar la muestra y. Es decir, sintetiza en forma de distribución de probabilidad la información previa que podemos tener acerca del parámetro.

- $p(y|\theta) = \prod_{i=1}^{n} p_\theta(y_i)$ es la *verosimilitud de la muestra* y. Expresa la probabilidad de la muestra y si el valor del parámetro es θ. Recordar que $p_\theta(y)$ representa la función masa de probabilidad (f.m.p.) de y si es variable discreta o la función de densidad de probabilidad (f.d.p.) de y si es variable continua, para el valor del parámetro θ.

Teorema 1.1.2. *Sea Y una v.a. , $y = (y_1, \ldots, y_n)$ una muestra de Y, $\theta = (\theta_1, \ldots, \theta_p)$ con $1 \leq p$ el vector paramétrico de dicha distribución , $p(\theta)$ la función de densidad a priori para el parámetro y $p(y|\theta)$ la verosimilitud de la muestra y . La densidad a posteriori del parámetro θ se denota como $p(\theta|y)$ y se define mediante la siguiente expresión:*

$$p(\theta \mid y) = \frac{p(y|\theta)\, p(\theta)}{\int p(y|z)\, p(z) dz} = \frac{p(y|\theta)\, p(\theta)}{p(y)}. \tag{1.3}$$

Demostración. Detallada en el Anexo A. Véase la Sección A.0.1. □

Nota 1.1.1. La expresión mostrada en el teorema corresponde a una variable aleatoria θ con distribución continua, pero el teorema también es válido para θ variable discreta. Notar que en caso de θ v.a. discreta, entonces cambiamos las integrales por sumatorios. Para unificar la notación denotaremos $p(\theta)$ indistintamente de si θ es una variable discreta o continua, representando en el primer caso la f.m.p y en el segundo la f.d.p. de la variable.

En resumen, la distribución a posteriori $p(\theta|y)$ se obtiene como una actualización de la distribución del parámetro a priori $p(\theta)$ en base a la muestra de datos observada $y = (y_1, \ldots, y_n)$, es decir, a posteriori.

Definición 1.1.1. Notar que por la expresión (1.3) se tiene que $p(\theta \mid y) = p(y \mid \theta)p(\theta)/p(y)$, y si suprimimos la constante $p(y)$ (que no depende de θ), obtenemos lo que se conoce como *densidad a posteriori no normalizada*, que es proporcional a $p(\theta|y)$ (densidad normalizada) :

$$p(\theta \mid y) \propto p(\theta)p(y \mid \theta). \tag{1.4}$$

1.1.3. Distribuciones a priori

En este punto es conveniente recordar que el objetivo del análisis bayesiano es realizar inferencia sobre el parámetro θ utilizando su distribución a posteriori. Uno de los factores básicos que determina la distribución a posteriori es el tipo de distribución a priori utilizada. En la siguiente sección se introducen algunos de los tipos de distribución a priori que se pueden utilizar y qué criterios se deben utilizar para seleccionarla.

Distribución a priori conjugada

Para obtener la distribución a posteriori $p(\theta|y)$ es necesario realizar los cálculos vistos anteriormente en el Teorema 1.1.2. En la mayoría de los casos estos cálculos llevan a una expresión compleja, especialmente en el caso en el que la dimensión de θ sea alta, que hace que la inferencia posterior a partir de la distribución sea complicada. Para facilitar el trabajo se pueden considerar distribuciones conjugadas. Es por ello que hasta la popularización de los métodos MCMC en los años 90, la estadística bayesiana se basaba principalmente en el uso de distribuciones conjugadas.

Definición 1.1.2. Sea $p(y \mid \theta)$ la verosimilitud sobre la observación y. Una clase $\mathscr{A}$ de distribuciones se dice *conjugada respecto a la verosimilitud* $p(y|\theta)$ si la distribución a posteriori $p(\theta \mid y) \in \mathscr{A} \ \forall y$, siempre que la distribución a priori $p(\theta) \in \mathscr{A}$.

Es decir, si utilizamos la distribución a priori conjugada respecto a una verosimilitud esto implica que la distribución a posteriori del parámetro va a ser de la misma familia que la a priori. Por lo tanto, el uso de distribuciones conjugadas permite obtener distribuciones a posteriori con distribución conocida sin necesidad de realizar cálculos complejos, lo que facilita la realización de inferencia basada en esta distribución.

Ejemplo 1.1.1. Veamos que la familia Gamma es conjugada respecto a la verosimilitud de una muestra exponencial. En efecto, la verosimilitud de una muestra exponencial es :

$$p(y_1, ..., y_n|\alpha) = \alpha^n e^{-\alpha \Sigma y_i} \mathbf{1}_{\{y_1, ..., y_n > 0\}},$$

y si la distribución a priori del parámetro α es $\Gamma(a,p)$, es decir :

$$p(\alpha) = \frac{a^p(\alpha)^{p-1}e^{-\alpha a}}{\Gamma(p)}\mathbf{1}_{\alpha\in(0,+\infty)}.$$

En efecto, aplicando el Teorema 1.1.2 tenemos:

$$p(\alpha|y_1,\ldots,y_n) = \frac{p(y_1,\ldots,y_n|\alpha)p(\alpha)}{\int_{-\infty}^{+\infty} p(y_1,\ldots,y_n|z)p(z)dz} = \frac{\frac{a^p\alpha^{p-1}e^{-a\alpha}}{\Gamma(p)}\alpha^n e^{-\alpha\sum y_i}\mathbf{1}_{\alpha\in(0,+\infty)}}{\int_0^{+\infty} z^n e^{-z\sum y_i}\frac{a^p z^{p-1}e^{-az}}{\Gamma(p)}dz} \tag{1.5}$$

$$= C(a,p,\sum y_i)\alpha^{n+p-1}e^{-(a+\sum y_i)\alpha}\mathbf{1}_{\alpha\in(0,+\infty)} = \Gamma\left(\alpha|a+\sum y_i, n+p\right),$$

donde C queda definida por la siguiente expresión:

$$C(a,p,\sum y_i) = \frac{\frac{a^p}{\Gamma(p)}}{\int_0^{+\infty} z^n e^{-z\sum y_i}\frac{a^p z^{p-1}e^{-az}}{\Gamma(p)}dz}.$$

Dado que C es una constante que no depende de α, la expresión resultante (1.5) es proporcional a la función de densidad de la Gamma $\Gamma(a+\sum y_i, n+p)$. Por lo tanto se tiene que la familia $\Gamma(a,p)$, $\alpha\in(0,+\infty)$ es distribución conjugada respecto a la verosimilitud de una muestra proveniente de una familia exponencial $\{Exp(\alpha),\alpha>0\}$.

A priori poco informativas

En muchas situaciones reales no se dispone de información relevante para definir una distribución a priori. En dichos casos se usan *distribuciones a priori no informativas* o *poco informativas*, que son distribuciones que cubren de manera equiprobable el espacio paramétrico, de forma que aportan poca información sobre el parámetro. Una distribución a priori no informativa habitual es la distribución *Uniforme*(a,b) que asigna la misma probabilidad a todos los valores en el intervalo (a,b).

Distribuciones a priori propias e impropias

- Una *distribución a priori propia* se define como una distribución cuya función de densidad integra uno, es decir :

$$\int_\Theta p(\theta)d\theta = 1.$$

- En caso de que la distribución $p(\theta)$ cumpla:

$$\int_\Theta p(\theta)d\theta = k\in\mathbb{R}.$$

  Es decir, que su integral sea $k\neq 1$, se la conoce como *distribución a priori no normalizada*, y normalizándola se puede convertir en una distribución propia. Sea por ejemplo $p(\theta)$ con $\int_{\theta\in\Theta} p(\theta)d\theta = c$, luego simplemente normalizando $p(\theta)/c$, conseguimos definir una distribución a priori propia.

- Por el contrario, una distribución $p(\theta)$ es una *distribución a priori impropia* (véase [5]), si la función de densidad tiene medida infinita (aunque formalmente hablando en términos de probabilidad no pueda ser una densidad ya que entonces debería integrar uno), esto es:

$$\int_\Theta p(\theta)d\theta = +\infty.$$

Existen distribuciones a priori impropias que se usan a menudo como distribuciones no informativas, por ejemplo $U(-\infty,+\infty)$ ya que asigna a todos los valores la misma probabilidad. En general hay muchas distribuciones no informativas que son impropias y las podremos usar como distribuciones a priori siempre y cuando nos lleven a distribuciones a posteriori propias, como es el caso de esta $U(-\infty,+\infty)$. Véase en [2, pág. 53] y [2, cap. 5].

1.1.4. Inferencia a partir de la distribución a posteriori

Uno de los objetivos de la inferencia a partir de la distribución a posteriori es proporcionar estimadores puntuales del parámetro θ y estimadores de tipo intervalo, para ello existen distintas alternativas.

Estimador de probabilidad máxima a posteriori (MAP)

El *estimador de probabilidad máxima a posteriori* o *moda a posteriori* (MAP[1]) es el valor del parámetro θ, que maximiza la densidad a posteriori , es decir:

$$\hat{\theta}_{\text{MAP}}(y) = \arg\max_{\theta} p(\theta \mid y) = \arg\max_{\theta} \frac{p(y_1,\ldots,y_n;\theta)\,p(\theta)}{\int p(y_1,\ldots,y_n;z)\,p(z)dz}.$$

En caso de tener una expresión explícita sencilla de la distribución a posteriori, podemos obtener el MAP analíticamente. Alternativamente, lo podemos obtener mediante simulación con los métodos MCMC, como se verá posteriormente.

Media a posteriori

Otro estimador puntual para θ, es la *media a posteriori*. Se define como:

$$\hat{\theta}_{\text{MEAN}}(y) = \int_{-\infty}^{+\infty} \theta p(\theta \mid y)d\theta = \int_{-\infty}^{+\infty} \theta \frac{p(y_1,\ldots,y_n;\theta)\,p(\theta)}{\int p(y_1,\ldots,y_n;z)\,p(z)dz}d\theta.$$

Notar que si la distribución a posteriori es simétrica respecto la moda y existe la esperanza, entonces este estimador coincide con el MAP.

Intervalos de credibilidad

Un *intervalo de credibilidad para* θ (unidimensional) *a nivel* $1-\alpha$ es un intervalo de valores $[a,b]$ tal que:

$$P\{a \leq \theta \leq b \mid y_1,\ldots,y_n\} \geq 1-\alpha.$$

En consecuencia a es un cuantil α_1 de $p(\theta|y_1,\ldots y_n)$ y b es un cuantil α_2 de $(\theta|y_1,\ldots y_n)$ tal que $\alpha_1+\alpha_2=\alpha$. Un intervalo de credibilidad a nivel α para un parámetro θ determina un rango de valores al que pertenece una observación del parámetro con probabilidad $1-\alpha$.

Notar el contraste con el análisis frecuentista, aunque la idea subyacente de los intervalos de confianza es la misma, la interpretación es diferente. En el análisis frecuentista un intervalo de confianza a nivel α se interpreta como sigue: si construimos a partir de 100 muestras (independientes) sus correspondientes 100 intervalos de confianza entonces el $(1-\alpha)100\,\%$ de esos intervalos contendrán el valor real del parámetro (ya que hemos dicho que la aleatoriedad provenía del muestreo).

1.1.5. Métodos MCMC

Como se ha indicado anteriormente, el objetivo del análisis bayesiano es realizar inferencia utilizando la distribución a posteriori de θ. Si la expresión de la distribución a posteriori obtenida en (1.3) corresponde a la densidad de una distribución de probabilidad conocida (como en el Ejemplo 1.1.1) los

[1] MAP es un acrónimo del inglés: *maximum a posteriori probability* (véase en [6, cap. 6 Bayesian Learning]).

cálculos a partir de ella son más sencillos. Sin embargo en la mayor parte de los modelos la distribución resultante es una expresión compleja con la que es complicado hacer inferencia. En estos casos, es necesario recurrir a los métodos MCMC, cuyo estudio es el objetivo de este trabajo, para poder implementar análisis inferenciales.

El objetivo de los métodos MCMC es generar muestras de observaciones de una variable X a partir de una distribución de probabilidad que puede ser realmente compleja. En el marco del análisis bayesiano los métodos MCMC se utilizan para simular observaciones del parámetro de interés con la distribución a posteriori. Estos actúan mediante el siguiente esquema:

- Simular una cadena de Markov $X^{(1)}, X^{(2)}, \ldots$, que tenga una distribución estacionaria, la cual sea la distribución a posteriori.
- De la cadena anterior se extrae una muestra, sobre la que se va a realizar inferencia de la distribución a posteriori.

1.2. Cadenas de Markov

En esta sección se introducen las *cadenas de Markov*, las cuales van a ser un pilar base para el posterior desarrollo e implementación de los métodos MCMC.

Definición 1.2.1. Una *cadena de Markov* $\{X^{(t)}\}$ es una secuencia de variables dependientes aleatorias $X^{(0)}, X^{(1)}, \ldots, X^{(t)}, \ldots$ tal que la distribución condicional de $X^{(t)}$ dadas las anteriores variables depende sólo de $X^{(t-1)}$. Es decir, la distribución de probabilidad del estado en el instante t depende del estado en el instante $t-1$, y no depende de los estados por los que la cadena pasó hasta llegar al estado del tiempo $t-1$. En caso de que la distribución de probabilidad sea la misma en todos los instantes, es decir, que no dependa de t, se dice *cadena de Markov homogénea*, que son el tipo de cadenas que usaremos en este trabajo.

Definición 1.2.2. El *kernel de Markov o de transición* $K(X^{(t)}, X^{(t+1)})$ es el nombre que recibe dicha distribución de probabilidad condicionada:

$$X^{(t+1)} \mid X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(t)} \sim K\left(X^{(t)}, X^{(t+1)}\right). \tag{1.6}$$

- Sea $(X, \mathscr{A})$ un conjunto medible, con X el conjunto de posibles estados y $A \in \mathscr{A}$. La función $K(x, A)$ con $x \in X$ es un kernel de transición si cumple:
  1. Para todo $x \in X$, $K(x, \cdot)$ es una medida de probabilidad.
  2. Para todo $A \in \mathscr{A}$, $K(\cdot, A)$ es medible.
- Cuando X es discreto, el kernel de transición se puede ver como una matriz de transición con elementos:
  $$k_{xy} = K(x,y) = K(X^{(n)} = y | X^{(n-1)} = x) \quad x, y \in X$$
  Notar que esta matriz $K = (k_{ij})$ $i, j \in X$ cumple que $\sum_j k_{ij} = 1 \ \forall i$.
- Cuando X es continuo, el kernel de transición es la densidad condicionada $K(x,y)$ y cumple:
  $$K(X \in A | x) = \int_A K(x, x') \mathrm{d}x' = 1 \quad x \in X \tag{1.7}$$
  por ser densidad.
- Una *distribución estacionaria* (límite) f de una Cadena de Markov consiste en una distribución de estado estable para los estados de una cadena, que además es independiente de la distribución inicial. Es decir, que si $X^{(t)} \sim f$ entonces $X^{(t+1)} \sim f$, sin importar el $X^{(0)}$ tomado. En concreto se debe cumplir que:
  $$\int_X K(x,y) f(x) \mathrm{d}x = f(y). \tag{1.8}$$

Habitualmente se suele tomar un kernel de transición que modele procesos de interés, para posteriormente determinar bajo que condiciones hay una única distribución estacionaria y obtenerla. Para los métodos MCMC nos será útil realizar el camino inverso. Dada una distribución queremos encontrar un kernel de transición (en la práctica hay muchos) que tenga como distribución estacionaria a dicha distribución.

Para garantizar que una cadena de Markov tiene una distribución estacionaria y es la que nosotros buscamos conviene recordar primero varios conceptos:

- La cadena de Markov debe ser *irreducible*, esto es, el kernel K permite moverse libremente por el espacio de estados sin importar el estado inicial $X^{(0)}$. Esto es que la secuencia $\{X^{(t)}\}$ tiene una probabilidad positiva de alcanzar cualquier región del espacio de estados. Una condición suficiente para la irreducibilidad es que se tenga $K(x,\cdot) > 0$ para todos los puntos.

- Una cadena de Markov se dice *aperiódica* si para todo estado de la cadena, el máximo común divisor del número de pasos necesarios para volver al dicho estado i (supuesto se ha partido de él) es 1.

- Una cadena de Markov se dice *recurrente positiva* si para todo estado de la cadena de Markov, la probabilidad de volver a dicho estado en tiempo finito, suponiendo se ha partido de él, es 1.

En concreto, si una cadena de Markov es aperiódica, irreducible y recurrente-positiva entonces tiene una única distribución estacionaria (véase en [3, pág. 175-176]). A las cadenas recurrentes-positivas y aperiódicas se les llama *ergódicas*.

Una cadena de Markov con distribución límite f se dice que es *reversible en el tiempo* si su kernel de transición cumple la *ecuación de equilibrio* siguiente:

$$f(x)K(x,y) = f(y)K(y,x) \quad \forall x,y \in X \tag{1.9}$$

Teorema 1.2.1. *Si una cadena de Markov es irreducible y ergódica, entonces tiene una única distribución estacionaria a la que converge dicha cadena cuando* $t \longrightarrow \infty$.

Demostración. Véase en [3, pág. 175-176]. □

Teorema 1.2.2. *Una condición suficiente para que cierta* f *sea la distribución estacionaria de una cadena de Markov ergódica, es que el kernel de transición cumpla la ecuación de equilibrio* $f(x)K(x,y) = f(y)K(y,x) \quad \forall x,y \in X$

Demostración. Si tenemos que nuestra cadena de Markov cumple la ecuación de equilibrio, esto es $f(x)K(x,y) = f(y)K(y,x)\ \forall x,y$, entonces integrando en x en ambos lados:

$$\int_X f(x)K(x,y)\mathrm{d}x = \int_X f(y)K(y,x)\mathrm{d}x,$$

y esto es lo mismo que:

$$\int_X f(x)K(x,y)\mathrm{d}x = f(y)\int_X K(y,x)\mathrm{d}x$$

y ahora puesto que $\int_X K(y,x)\mathrm{d}x = 1$ puesto que es una densidad, entonces:

$$\int_X f(x)K(x,y)\mathrm{d}x = f(y).$$

Luego ya tenemos que f es distribución estacionaria. Para una demostración alternativa, véase en [12, pág. 238, Theorem 4].

□

Capítulo 2

Métodos Monte Carlo basados en cadenas de Markov

2.1. Introducción a la computación bayesiana

Uno de los objetivos principales de la computación bayesiana es el cálculo u obtención de una muestra de la distribución a posteriori $p(\theta|y)$ de θ, para a partir de ella hacer inferencia y calcular medidas resumen. En los casos más sencillos, por ejemplo, cuando se utilizan distribuciones a priori conjugadas, la distribución a posteriori corresponde a una distribución conocida. Sin embargo, en modelos más complejos o con un número de parámetros alto, es necesario utilizar algoritmos más elaborados que permitan obtener muestras de la distribución a posteriori con las que caracterizar dicha distribución. Con este objetivo se pueden utilizar los métodos métodos MCMC.

Estos métodos fusionan dos tipos de técnicas, los métodos de integración Monte Carlo y los métodos de simulación basados en cadenas de Markov. En efecto, un problema clave en el desarrollo de los métodos computacionales bayesianos es la integración numérica, ya que el cálculo de medidas de la distribución a posteriori de los parámetros requiere la evaluación de integrales de funciones complejas, que en la mayoría de casos no se pueden obtener analíticamente. Para solucionar este problema se utilizan métodos de integración numérica estocásticos, como el método de integración Monte Carlo. Un problema adicional es que estos métodos requieren la simulación de variables con una distribución a menudo compleja y multidimensional y no siempre existen algoritmos de simulación directos para dichas distribuciones. Los métodos de simulación basados en cadenas de Markov ofrecen una solución a este problema, ya que permiten simular observaciones con cualquier distribución.

En este capítulo se repasará la integración de Monte Carlo y varios métodos de simulación de v.a's , que en su conjunto son uno de los pilares fundamentales de los métodos MCMC. Se verá la problemática que puede conllevar el uso de métodos estocásticos y se verá como solucionar estos problemas mediante los métodos MCMC. Para ello, se describen dos de los algoritmos MCMC más extendidos en el ámbito de la computación bayesiana, el Metropolis-Hastings y el muestreo de Gibbs.

2.2. Métodos de integración Monte Carlo

Los *métodos de integración numérica* son métodos cuyo objetivo es evaluar la integral de una función continua, calculando el valor de la función en un número finito de puntos. Dentro de los métodos de integración numérica se puede distinguir los *métodos estocásticos*, o de Monte Carlo, y los *métodos deterministas*, como las reglas de cuadratura. La diferencia principal entre ambos métodos es que mientras los métodos deterministas evalúan el integrando en una cuadricula regular, los estocásticos eligen aleatoriamente los puntos en los que se evalúa el integrando.

La idea subyacente de el *método de integración Monte Carlo*, es que cualquier integral definida sobre el espacio de $x = (x_1, \ldots, x_p)$ con $p \geq 1$ se puede expresar como una esperanza de una función $h(x)$, definiendo $h(x)$ adecuadamente, es decir:

$$\mathrm{E_f}(h(x)) = \int h(x)f(x)dx. \tag{2.1}$$

Es decir, dada una muestra de n observaciones de $x = (x_1, \ldots, x_p)$ con $p \geq 1$ con distribución $f(x)$, $(x^{(1)}, \ldots, x^{(n)})$, donde cada $x^{(i)}$ es $x^{(i)} = (x_1^{(i)}, \ldots, x_p^{(i)})$, se puede estimar la integral de interés estimando la esperanza de $h(x)$ mediante la media muestral:

$$\overline{h_n} = \frac{1}{n}\sum_{i=1}^{n} h(x^{(i)}). \tag{2.2}$$

La *Ley Fuerte de Grandes Números* (véase A.0.2) para variables independientes idénticamente distribuidas (iid) permite asegurar que la media muestral converge casi seguro a la esperanza de la variable, es decir:

$$Prob\big[\lim_{n\to\infty} \overline{h_n} = \mathrm{E_f}(h(x))\big] = 1,$$

lo que garantiza que si n es suficientemente grande, la media muestral proporciona un buen estimador de la esperanza de la variable y en consecuencia una buena aproximación de la integral deseada. Notar que en el ámbito bayesiano identificamos x con el vector de parámetros θ y la función $f(x)$ es la distribución a posteriori $p(\theta \mid y)$, sobre la que estamos interesados en obtener alguna medida resumen a partir de una muestra $(\theta^{(0)}, \ldots, \theta^{(n)})$ de dicha distribución.

Como se ha comentado, la aplicación del procedimiento de integración Monte Carlo requiere disponer de una muestra de n observaciones $x^{(i)}$ con distribución $f(x)$, por lo que es necesario disponer de un procedimiento de simulación de observaciones aleatorias con esa distribución. Si la distribución a posteriori no corresponde a una distribución estándar o la dimensión del vector de parámetros es alta, la simulación de estos valores no es trivial. En estas situaciones existen distintos métodos, como el *método del muestreo por importancia*, que permite expresar la integral de interés como función de otra distribución de la que sí sea posible simular observaciones. Otra alternativa posible es el uso de algoritmos de simulación de variables que permitan obtener observaciones $x^{(i)}$, con una distribución $f(x)$ mediante el uso de distribuciones propuestas, como el *método de simulación por rechazo*, o mediante distribuciones aproximadas.

2.2.1. Simulación por importancia

Este método es un precursor del *Algoritmo de Metropolis-Hastings* y se basa en generar muestras aleatorias de distribuciones que sean aproximadas a la distribución objetivo, distribución respecto a la que estamos interesados en tomar la esperanza.

El objetivo de este método es calcular $E(h(x))$ con $x = (x_1, \ldots, x_p)$ con distribución $f(x)$ pero no tenemos la capacidad de generar muestras de $f(x)$, luego no podemos aplicar un método de integración estocástico como el Monte Carlo y evaluar la integral por la media muestral de los valores simulados. Para solucionar este problema el *muestreo por importancia* actúa como sigue: Se requiere considerar una *distribución propuesta* $g(x)$ de la que sí podamos generar muestras $\{x^{(1)}, \ldots, x^{(n)}\}$ y se utiliza la siguiente propiedad:

$$\mathrm{E}(h(x)) = \int h(x)f(x)dx = \int \frac{[h(x)f(x)]g(x)}{g(x)}dx = \mathrm{E}_g\Big[\frac{h(x)f(x)}{g(x)}\Big], \tag{2.3}$$

donde $E_g[\cdot]$ es la esperanza en la distribución $g(x)$. Dado que sí podemos simular observaciones con distribución $g(x)$, ahora sí podemos estimar $E_g[h(x)f(x)/g(x)]$ mediante el método Monte Carlo. En concreto el estimador es el siguiente:

$$\overline{\mathrm{E}(h(x))} = \frac{1}{n}\sum_{i=1}^{n} h(x^{(i)})w(x^{(i)}). \tag{2.4}$$

donde:

- $x^{(i)} = (x_1^{(i)}, \ldots, x_p^{(i)})$ con $i = 1, \ldots, n$ son los valores simulados con distribución $g(x)$.
- $w(x^{(i)}) = \frac{f(x^{(i)})}{g(x^{(i)})}$ con $i = 1, \ldots, n$ es la llamada *razón de importancia.*

El muestreo por importancia no es muy preciso si las razones de importancia varían sustancialmente, por lo que examinando la distribución de las razones de importancia podemos ver si está habiendo problemas. Destacar que la calidad de la estimación dependerá en gran medida de la elección de la $g(x)$. La idónea sería una parecida a $f(x)$ en forma pero con unas colas más pesadas, véase [9, secc. 2.4. Muestreo por importancia] o [8, secc. 9.2.].

Ejemplo 2.2.1. Sabemos que una distribución $N(0,1)$ tiene esperanza 0, luego nuestro objetivo va a ser estimar mediante el muestreo por importancia, la esperanza de la $N(0,1)$ y ver que es prácticamente 0. Suponemos que no podemos muestrear una $N(0,1)$ debido a causas ajenas a nosotros y por lo tanto no podemos aproximar la esperanza por la media muestral aplicando la integración de Monte Carlo directamente. Entonces, aplicando el muestreo por importancia, generaremos 10^5 simulaciones de una distribución aproximada, como puede ser una t_3 (*t student con 3 grados de libertad*).

Lo primero veamos que efectivamente estas dos distribuciones son aproximadas. Para ello vemos que el histograma de una muestra de 10^7 valores para una t_3 se ajusta bien a una densidad $N(0,1)$. Véase código en Anexo B.1. Ahora fijando en R como *semilla* 785248 (mi NIP), obtenemos que simulando 10^7 observaciones de una t_3 (distribución propuesta) y aplicando la Ecuación 2.4 del muestreo por importancia obtenemos que el estimador de la esperanza de la $N(0,1)$ es 6.304878e-05, el cual es muy cercano al valor real de dicha esperanza, que es 0. (Véase código en Anexo B.1).

En caso de que no sea fácil simular sobre la distribución propuesta puesto que la dimensión del vector de parámetros es elevada o bien sea muy difícil encontrar esa $g(x)$, surge la necesidad de generar otro tipo de métodos de muestreo. Más adelante veremos como aunque los métodos MCMC se adaptan mejor que los Monte Carlo a problemas multiparamétricos de alta dimensión, estos producen muestras dependientes.

2.2.2. Simulación por rechazo

El objetivo de este método de simulación es obtener una muestra aleatoria de tamaño n con la *distribución objetivo* $f(x)$. Este método fue el principal precursor de los métodos MCMC. La idea general de su funcionamiento subyace en la elección de una distribución llamada *distribución propuesta* $g(x)$ que acote absolutamente a $f(x)$ y con la cual se puedan general valores aleatorios de manera sencilla, a los que se les aplica una regla de decisión para rechazarlos o no como observaciones de f. Destacar que este método requiere que la $f(x)$ y la $g(x)$ se puedan evaluar fácilmente.

Intuitivamente lo que haremos a partir de ahora, con este método y posteriormente con los MCMC, es simular observaciones de la distribución objetivo $f(x)$ utilizando algoritmos de simulación de la distribución propuesta $g(x)$. Una vez dadas $g(x)$ y $f(x)$ los pasos del algoritmo son:

Algoritmo 1 Simulación por rechazo

1. Escoger la menor constante $1 < M < \infty$, tal que $f(x) \leq Mg(x)$.
2. Generar una observación $x^{(i)}$ a partir de $g(x)$ y también u_i a partir de una $U(0,1)$.
3. *Regla de decisión*: Solo se acepta $x^{(i)}$ como observación de $f(x)$ si $u_i < \frac{f(x^{(i)})}{Mg(x^{(i)})}$. Esto es, se acepta $x^{(i)}$ como muestra de $f(x)$ con probabilidad $\frac{f(x^{(i)})}{Mg(x^{(i)})}$.
4. Volver al paso 1) y repetir proceso hasta obtener el tamaño de muestra deseado n.

Para ello la distribución propuesta $g(x)$ debe ser una función positiva que este definida para todo x con $g(x) > 0$ y con las siguientes propiedades :

- $g(x)$ tenga una integral finita y se disponga de un algoritmo para generar observaciones con densidad proporcional a $g(x)$.
- La *razón de importancia* $\frac{f(x)}{g(x)}$ debe tener un límite conocido, es decir, $\exists\, M$ tal que $\frac{f(x)}{g(x)} \leq M \ \forall x$.

Demostración. Veamos que la función de distribución de las observaciones $x^{(i)}$ aceptadas se corresponde con la función de distribución de $f(x)$. Sea pues la función de distribución de los valores aceptados:

$$\begin{aligned} P\Big(x^{(i)} \leq x \mid u_i \leq \frac{x^{(i)}}{Mg(x^{(i)})}\Big) &= \frac{P(x^{(i)} \leq x, u_i \leq f(x^{(i)})/\{Mg(x^{(i)})\})}{P(u_i \leq f(x^{(i)})/\{Mg(x^{(i)})\})} \\ &= \frac{\int_{-\infty}^{x}\int_{0}^{f(x^{(i)})/\{Mg(x^{(i)})\}} \mathrm{d}u g(x^{(i)})\mathrm{d}x}{\int_{-\infty}^{\infty}\int_{0}^{f(x^{(i)})/\{Mg(x^{(i)})\}} \mathrm{d}u g(x^{(i)})\mathrm{d}x} \\ &= \frac{\int_{-\infty}^{x}[f(x^{(i)})/\{Mg(x^{(i)})\}]g(x^{(i)})\mathrm{d}x}{\int_{-\infty}^{\infty}[f(x^{(i)})/\{Mg(x^{(i)})\}]g(x^{(i)})\mathrm{d}x} \\ &= \frac{\int_{-\infty}^{x} f(x^{(i)})\mathrm{d}x}{\int_{-\infty}^{\infty} f(x^{(i)})\mathrm{d}x} = \int_{-\infty}^{x} f(x^{(i)})\mathrm{d}x \end{aligned}$$

En esta demostración vemos que aunque estemos simulando únicamente desde $g(x)$, la muestra finalmente obtenida tiene como función de densidad $f(x)$.

□

De acuerdo con Florescu, I. (véase [7, pág. 95. Prop. 3.3]) la probabilidad de aceptar los valores propuestos es $1/M$, luego contra más se acerque a 1, menos valores se rechazarán. Una buena densidad aproximada para este algoritmo sería utilizar $g(x)$ estrictamente proporcional a $f(x)$. Es decir, $g \propto f$. En ese caso, tomando un adecuado M podemos aceptar toda observación con probabilidad 1.

En la Figura 2.1 se puede observar que se cumple $f(x) \leq Mg(x)\ \forall x$. La línea vertical indica una observación de la densidad proporcional a g. La probabilidad de que esa observación sea aceptada para f, es el ratio entre la altura de la curva inferior y la altura de la curva superior cuando se cortan con la línea vertical.

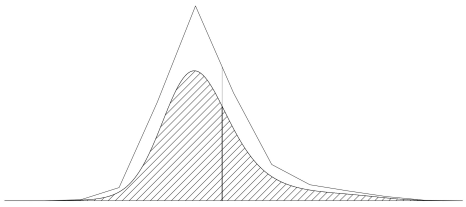

Figura 2.1: La curva superior es $M.g(x)$ mientras que la inferior es la $f(x)$ (dist. objetivo). Gráfico obtenido de [2, pág. 264].

Ejemplo 2.2.2. Se quiere simular una variable con función de densidad Beta(α,β), es decir:

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \text{ si } 0 \leq x \leq 1.$$

Se toma como distribución propuesta $U(0,1)$, que tiene densidad $g(x) = 1$ si $0 \leq x \leq 1$. Notar que si α y β son mayores que 1, la moda es $\frac{\alpha-1}{\alpha+\beta-2}$, luego definiendo la M como $M := \text{máx}_{0\leq x\leq 1} f(x) = f((\alpha-1)/(\alpha+\beta-2))$ se obtiene una cota M que cumple las condiciones necesarias para aplicar el algoritmo. En la Figura 2.2 se muestra la función de densidad Beta con $\alpha = 3$ y $\beta = 5$ y veamos que se cumple $f(x) \leq Mg(x)$. Ahora aplicamos el algoritmo de muestreo por rechazo para obtener una muestra de 1000 observaciones de una $B(3,5)$ a partir de la $U(0,1)$ y en la Figura 2.2 mostramos los resultados

para ver que efectivamente los valores aceptados por el muestreo por rechazo se ajustan muy bien a una muestra de una $B(3,5)$. Véase código en Anexo B.2.

Este método presenta varios inconvenientes ya que es necesario encontrar una función g y una cota M que cumplan las condiciones dadas. Además si la dimensión del vector de parámetros $\theta = (\theta_1, \ldots, \theta_p)$ es alta, el algoritmo se vuelve muy ineficiente.

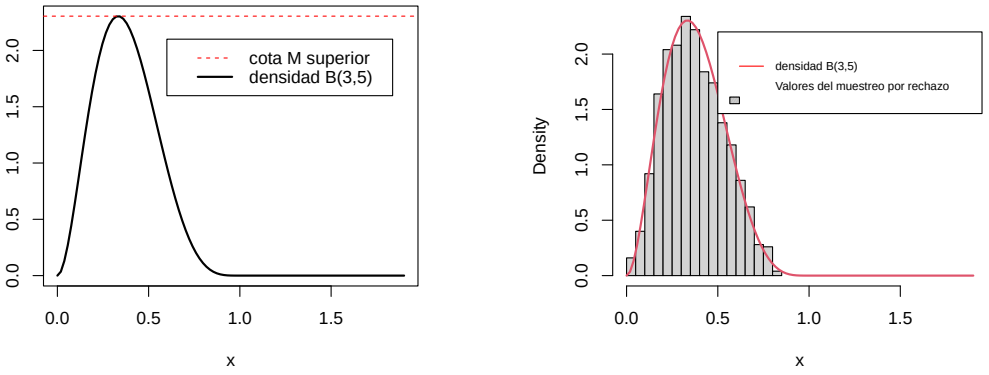

Figura 2.2: (Izquierda) Función de densidad de la $Beta(3,5)$ y la cota $Mg(x)$ obtenida en linea en discontinua, que en efecto corresponde a una cota superior. (Derecha) Histograma de frecuencias relativas de los valores aceptados por el muestreo por rechazo y la densidad teórica de la $B(3,5)$.

2.3. Simulación basada en cadenas de Markov. Métodos MCMC

Los métodos de simulación basados en cadenas de Markov son métodos generales que permiten generar observaciones de $x = (x_1, \ldots, x_p)$ con $p \geq 1$ con cualquier distribución f aunque sea compleja o multidimensional. Por ello, son útiles cuando no es posible (o no es computacionalmente eficiente) simular directamente con la distribución objetivo f. La idea subyacente es simular observaciones de $x = (x_1, \ldots, x_p)$ utilizando una distribución aproximada y luego corregirlas para que correspondan a la distribución objetivo. Esta aproximación se hace secuencialmente, con la distribución de los valores generados dependientes del anterior valor simulado, de forma que la serie de valores simulados es una cadena de Markov. La bondad del procedimiento se basa en que las distribuciones aproximadas utilizadas en cada paso mejoran y convergen finalmente a la distribución objetivo. Es por ello que para simular una serie $\{x^{(0)}, \ldots, x^{(n)}\}$ del vector aleatorio x, se elige un punto de inicio $x^{(0)} = (x_1^{(0)}, \ldots, x_p^{(0)})$ y para cada t, se simula una observación a partir del *kernel de transición* $K(x^{(t)} \mid x^{(t-1)})$. Como ya hemos visto, los kernel de transición se deben construir de forma que la cadena de Markov resultante converja a una distribución estacionaria única, que sea la función objetivo f. La aplicabilidad de los métodos basados en cadenas de Markov se basa en que se puede demostrar que para cualquier distribución objetivo f se pueden construir cadenas de Markov cuya distribución estacionaria converja a dicha f. Dado el carácter asintótico del resultado, es imprescindible que una vez que se han generado una serie de valores de x, se compruebe la convergencia de la serie simulada, es decir, que los valores generados tengan una distribución suficientemente aproximada a la distribución estacionaria de la cadena.

Por lo tanto, en esta sección se expondrán dos de los algoritmos MCMC más extendidos en el ámbito de computación bayesiana: el *Metropolis-Hastings* y el *Muestreo de Gibbs*[1]. Destacar que en este punto del trabajo ya queda justificado el acrónimo MCMC (*Markov Chain Monte Carlo*), debido a que ya se ha explicado que estos algoritmos se basan en cadenas de Markov y la intergación de Monte Carlo.

[1] En inglés se le conoce como *Gibbs-Sampler*.

2.4. Metropolis-Hastings

El término de *Metropolis-Hastings* (M-H) engloba a un conjunto de algoritmos MCMC que comparten unas características comunes como veremos a continuación. Su principal identificador dentro de los métodos MCMC, es que estos métodos M-H van actualizando el vector de parámetros de manera conjunta mediante distribuciones propuestas de igual dimensión que el vector de parámetros.

2.4.1. Algoritmo de Metropolis-Hastings

Este algoritmo fue propuesto por el estadístico Wilfred Keith Hastings en 1970 como una extensión del trabajo realizado por Metropolis (véase en [16, Monte Carlo Sampling Methods])). El objetivo es simular observaciones con una distribución objetivo f (en el ámbito Bayesiano es la distribución a posteriori para el parámetro) y para ello se debe tener una *distribución propuesta o instrumental* q (a veces llamada *distribución de salto* [2]) tal que q sea fácil de simular. Intuitivamente q es la distribución que le vamos a proponer a nuestro algoritmo, el cual la va ir modificando hasta ser capaz de muestrear la distribución objetivo. Notar que no hay necesidad de conocer explícitamente la distribución objetivo, sólo necesitamos poder evaluarla.

La idea del algoritmo es la siguiente: suponemos que tenemos la cadena de Markov en un estado $x^{(t)} = (x_1^{(t)}, \ldots, x_p^{(t)})$ y queremos calcular el valor $x^{(t+1)} = (x_1^{(t+1)}, \ldots, x_p^{(t+1)})$. La distribución propuesta $q(.|x^{(t)})$ genera un valor aleatorio propuesto Y_t, que se acepta con una determinada probabilidad. La distribución propuesta q puede depender del estado actual de la cadena. Los pasos del algoritmo son:

Algoritmo 2 Metropolis-Hastings

Dado $x^{(t)} = (x_1^{(t)}, \ldots, x_p^{(t)})$, que es el vector x en la iteración t,

1. Generar $Y_t \sim q\left(y \mid x^{(t)}\right)$.
2. Tomar $x^{(t+1)} = \begin{cases} Y_t & \text{con probabilidad} \quad \rho\left(x^{(t)}, Y_t\right) \\ x^{(t)} & \text{con probabilidad} \quad 1-\rho\left(x^{(t)}, Y_t\right) \end{cases}$ donde

$$\rho(x,y) = \text{mín}\left\{\frac{f(y)}{f(x)}\frac{q(x \mid y)}{q(y \mid x)}, 1\right\}$$

Notar que en caso de que un valor sea rechazado $x^{(t+1)} = x^{(t)}$, es decir, se debe incluir el valor repetido en la muestra. Además la dimensión de la distribución propuesta q debe ser igual al número de parámetros.

Demostración. Veamos que este algoritmo efectivamente nos produce una cadena de Markov con distribución estacionaria f.

Se trata de una cadena de Markov puesto que la distribución de estados sucesivos depende únicamente del estado anterior. Además la cadena es irreducible puesto que en cada paso hay una probabilidad positiva de alcanzar cualquier región en el soporte de la distribución propuesta. Además, de acuerdo con Gelman, véase [1, pág. 279], excepto para excepciones triviales, toda cadena generada con q distribuciones propias es irreducible y recurrente-positiva. Luego la cadena de Markov generada va a ser irreducible y aperiódica, entonces aplicando el Teorema 1.2.1 se tiene que esta convergerá a una única distribución límite. Para ver cual es la distribución estacionaria veamos que que el kernel de transición cumple la ecuación de equilibrio y por tanto, usando el Teorema 1.9, deducimos que f es la distribución estacionaria. Para comprobar la ecuación de equilibrio seguimos los siguientes pasos:

[2] En la literatura bayesiana anglosajona a esta distribución se la conoce como *Jumping distribution* (véase en [2, pág. 278]), puesto que esta es la encargada de ir simulando valores para el parámetro a lo largo del espacio paramétrico.

1. Comprobamos que la probabilidad de que $x^{(t+1)} = x^{(t)}$ es

$$\underline{\rho}\left(x^{(t)}\right) = \int \left\{1 - \rho\left(x^{(t)}, y\right)\right\} q\left(y \mid x^{(t)}\right) \mathrm{d}y.$$

2. Comprobamos que el kernel se puede expresar como

$$K\left(x^{(t)}, y\right) = \rho\left(x^{(t)}, y\right) q\left(y \mid x^{(t)}\right) + \underline{\rho}\left(x^{(t)}\right) \delta_{x^{(t)}}(y).$$

3. Comprobamos que el algoritmo cumple la ecuación de equilibrio (1.9)

1.

$$\underline{\rho}\left(x^{(t)}\right) = 1 - \int \rho\left(x^{(t)}, y\right) q\left(y \mid x^{(t)}\right) \mathrm{d}y = \int \left(1 - \rho(x^{(t)}, y)\right) q\left(y \mid x^{(t)}\right) \mathrm{d}y$$

2.

$$K\left(x^{(t)}, y\right) = \underbrace{\rho\left(x^{(t)}, y\right) q\left(y \mid x^{(t)}\right)}_{\text{probabilidad de salir de } x^{(t)}} + \underbrace{\underline{\rho}\left(x^{(t)}\right) \delta_{x^{(t)}}(y)}_{\text{prob. permanecer en } x^{(t)}}$$

3. En caso de que $x^{(t+1)} = x^{(t)}$ es claro que $f(x^{(t)})K(x^{(t+1)}|x^{(t)}) = f(x^{(t+1)})K(x^{(t)}|x^{(t+1)})$. Sea pues ahora $x^{(t+1)} \neq x^{(t)}$, entonces:

$$\begin{aligned} f(x^{(t)})K(x^{(t+1)}|x^{(t)}) &= f(x^{(t)})q(x^{(t+1)}|x^{(t)})\rho(x^{(t)}, x^{(t+1)}) \\ &= \text{mín}\left\{\frac{f(x^{(t+1)})}{f(x^{(t)})}\frac{q(x^{(t)} \mid x^{(t+1)})}{q(x^{(t+1)} \mid x^{(t)})}, 1\right\} f(x^{(t)})q(x^{(t+1)}|x^{(t)}) \\ &= \text{mín}\left\{f(x^{(t+1)})q(x^{(t)} \mid x^{(t+1)}), f(x^{(t)})q(x^{(t+1)}|x^{(t)})\right\} \end{aligned}$$

$$\begin{aligned} f(x^{(t+1)})K(x^{(t)}|x^{(t+1)}) &= f(x^{(t+1)})q(x^{(t)}|x^{(t+1)})\rho(x^{(t+1)}, x^{(t)}) \\ &= \text{mín}\left\{\frac{f(x^{(t)})q(x^{(t+1)}|x^{(t)})}{f(x^{(t+1)})q(x^{(t)}|x^{(t+1)})}, 1\right\} f(x^{(t+1)})q(x^{(t)}|x^{(t+1)}) \\ &= \text{mín}\left\{f(x^{(t)})q(x^{(t+1)} \mid x^{(t)}), f(x^{(t+1)})q(x^{(t)}|x^{(t+1)})\right\} \end{aligned}$$

Ambas expresiones son iguales luego se cumple la ecuación de equilibrio, luego usando el Teorema 1.2.2, deducimos que f es la distribución estacionaria de la cadena.

□

Este método está basado en un muestreo por rechazo generalizado, donde los valores aleatorios se toman de la distribución propuesta q y son corregidos de tal manera que se comporten asintóticamente como valores de la distribución objetivo. Otro punto a destacar es que las observaciones obtenidas mediante Metropolis-Hastings tienen correlación. Dado que las muestras generadas por el algoritmo M-H son correladas contienen menos información efectiva que una muestra del mismo tamaño pero con observaciones independientes. Por ello, se necesita un número más alto de simulaciones. Este concepto lo formaliza Casella, G. en *Introducing Monte Carlo Methods with R* denotándolo como *tamaño de muestra efectivo* (*effective sample size*), que lo veremos en la última parte del trabajo (véase también en [1, secc. 8.4.3]).

Relación entre la regla de salto y la eficiencia en las simulaciones

La distribución $q(.|.)$ puede ser prácticamente de cualquier tipo y la cadena de Markov del algoritmo convergerá a la distribución estacionaria $f(x)$. No obstante a la hora de elegir q, con el fin de acelerar la convergencia, se debe tener en cuenta que:

- Para cualquier x, sea fácil simular $q(y|x)$.
- Sea fácil calcular el ratio $\rho(x^{(t)}, Y_t)$.
- $q(.|x)$ debe tener suficiente dispersión para garantizar una exploración sobre todo el soporte de f.

Si el dominio de q es muy pequeño en comparación con el de f, entonces la cadena de Markov tendrá dificultad para explorar todo el dominio de f y por tanto la convergencia será muy lenta. Si la distribución propuesta q tiene muy poca varianza, la cadena puede permanecer en una región pequeña largos periodos de tiempo mientras otras regiones de la distribución objetivo quedan sin explorar. Está claro que en la práctica, la eficiencia del algoritmo dependerá en gran medida de la elección de q. El caso ideal, en términos de optimización del algoritmo, sería tomar $q = p(\theta|y)$ pero en dicho caso no tendría sentido implementar estos métodos, ya que tendríamos la capacidad de muestrear directamente de $p(\theta|y)$. Veamos por ello casos particulares del Metropolis-Hastings según la distribución propuesta que tomemos.

2.4.2. Algoritmo básico de Metropolis

Fue el algoritmo que propuso inicialmente el físico estadounidense Nicholas Metropolis (véase en [17, The beginning of the Monte Carlo Method]). Se trata también de una generalización del *muestreo por rechazo*, visto en la Sección 2.2.2.

Destacar que en este algoritmo, pero no para el Metropolis-Hastings, la distribución instrumental debe ser simétrica, es decir $q(x|y) = q(y|x)\ \forall(x,y)$. Es por ello que ahora se tiene que:

$$\rho(x,y) = \text{mín}\left\{\frac{f(y)}{f(x)}, 1\right\}. \tag{2.5}$$

El resto de pasos son iguales que en M-H. Intuitivamente, en cada iteración el algoritmo se puede ver como; si el nuevo valor incrementa la densidad a posteriori, entonces actualizamos el valor $x^{(t+1)} = Y_t$; si el valor propuesto disminuye la densidad a posteriori, entonces $x^{(t+1)} = Y_t$ con probabilidad $\rho(x^{(t)}, Y_t)$.

2.4.3. Algoritmo de Metropolis-Hastings independiente

El algoritmo de Metropolis-Hastings visto en la sección anterior, en cada iteración la distribución propuesta dependía del estado actual de la cadena. Si ahora imponemos que esta distribución sea independiente del estado actual de la cadena , esto es $q(y|x) = g(y)$, obtenemos:

Algoritmo 3 Metropolis-Hastings con independencia:

Dado $x^{(t)}$

1. Generar $Y_t \sim g(y)$.
2. Tomar

$$x^{(t+1)} = \begin{cases} Y_t & \text{con probabilidad} \quad \text{mín}\left\{\frac{f(Y_t)g\left(x^{(t)}\right)}{f\left(x^{(t)}\right)g(Y_t)}, 1\right\} \\ x^{(t)} & \text{en otro caso.} \end{cases}$$

Este método también se puede comprender como una sencilla generalización del ya visto en la Sección 2.2.2, *Muestreo por rechazo*, puesto que la distribución instrumental es la misma que en el método de rechazo. Veamos algunas similitudes/diferencias entre ellos.

- Puesto que tienen la misma distribución instrumental g, los valores propuestos para aceptar son en ambos casos los mismos. No así los aceptados, ya que ambos algoritmos tienen distintas reglas de decisión.
- El muestreo por rechazo genera, como ya se ha visto, muestras iid. Por el contrario, aunque el Metropolis-Hastings produzca las Y_t de manera independiente, el resultado no es iid puesto que la probabilidad de aceptar Y_t como valor si que depende de $x^{(t)}$, vía $\rho\left(x^{(t)}, Y_t\right)$.
- El Metropolis-Hastings puede generar valores repetidos ya que en caso de rechazar un valor Y_t, se tendrá $x^{(t+1)} = x^{(t)}$. Esto puede generar errores a la hora de comprobar vía tests como el Kolmogorov-Smirnov que la muestra pertenece a la distribución a posteriori. Esto es debido a que estos tipos de test asumen que la muestra estudiada está formada por observaciones independientes y en el Metropolis-Hastings, como se ha comentado, se generan muestras correladas.

2.4.4. Paseo aleatorio Metropolis-Hastings

Se trata de un algoritmo en el cual el valor propuesto está basado en la elección de un paseo aleatorio como distribución propuesta. Por ello, la idea general es que el valor propuesto Y_t sea $Y_t = x^{(t)} + \varepsilon_t$, donde ε_t representa una perturbación aleatoria con distribución g independiente de $x^{(t)}$. Es por ello que se toma como distribución propuesta $g(y - x^{(t)})$ y además se le suele imponer que sea simétrica para que las probabilidades de aceptación sean las mismas que en el *Algoritmo básico de Metropolis*, visto en la Sección 2.4.2 .

Algoritmo 4 Paseo aleatorio Metropolis-Hastings:

Dado $x^{(t)}$,

1. Generar $Y_t \sim g\left(y - x^{(t)}\right)$.
2. Tomar
$$x^{(t+1)} = \begin{cases} Y_t & \text{con probabilidad } \min\left\{1, f\left(Y_t\right)/f\left(x^{(t)}\right)\right\} \\ x^{(t)} & \text{en otro caso.} \end{cases}$$

Todos estos algoritmos son casos particulares del Metropolis-Hastings y su principal virtud reside en que dadas restricciones poco restrictivas y simples, en un gran número de casos es posible construir una cadena de Markov con distribución límite la a posteriori para θ.

2.5. Muestreo de Gibbs

Como se ha visto en la sección anterior, los algoritmos de Metropolis-Hastings actualizan todas las componentes del vector de parámetros $\theta = (\theta_1, \ldots, \theta_p)$ de una sola vez. Luego si tenemos p parámetros, la dimensión de de la distribución propuesta q debe ser p. En la práctica hay casos en los que el vector de parámetros tiene una dimensión muy elevada (y en los que es difícil o incluso imposible encontrar una distribución instrumental de dimensión tan alta que sea fácil de simular), por lo que puede ser conveniente actualizar componentes del vector de parámetros de uno en uno, o en bloques y ahí es donde surge la idea del muestreo de Gibbs. La relevancia del muestreo de Gibbs reside en la capacidad de simplificar un problema complejo de alta dimensión mediante su división en problemas simples de baja dimensión.

Antes de profundizar en este algoritmo, recordemos primero la notación pertinente. Sea al igual que antes $x^{(n)} = (x_1^{(n)}, x_2^{(n)}, \ldots, x_p^{(n)})$ el vector en la iteración n de la cadena de Markov. Ahora cada componente x_i puede ser unidimensional o multidimensional. En caso de que haya alguna multidimensional se dirá que para esa componente se está realizando una *actualización en bloque*. Para usar este algoritmo

debemos tener la capacidad de simular a partir de las densidades condicionadas $f_1,\ldots,f_p$, es decir poder simular

$$x_i \mid x_1,x_2,\ldots,x_{i-1},x_{i+1},\ldots,x_p \sim f_i\left(x_i \mid x_1,x_2,\ldots,x_{i-1},x_{i+1},\ldots,x_p\right) \quad \forall i=1,\ldots,p$$

Las densidades $f_1,\ldots,f_p$ se les llama *distribuciones totalmente condicionadas*[3]. La principal ventaja del muestreo de Gibbs es que solo requiere generar observaciones con una distribución unidimensional (en caso de que se actualicen de uno en uno). Los pasos del algoritmo son:

Algoritmo 5 Muestreo de Gibbs

En la iteración $t=1,2,\ldots,$, dado $x^{(t-1)}=\left(x_1^{(t-1)},\ldots,x_p^{(t-1)}\right)$:

1. $x_1^{(t)} \sim f_1\left(x_1 \mid x_2^{(t-1)},\ldots,x_p^{(t-1)}\right)$;
2. $x_2^{(t)} \sim f_2\left(x_2 \mid x_1^{(t)},x_3^{(t-1)},\ldots,x_p^{(t-1)}\right)$;

p. $x_p^{(t)} \sim f_p\left(x_p \mid x_1^{(t)},\ldots,x_{p-1}^{(t)}\right)$.

Es decir, el algoritmo en cada paso va actualizando cada componente del vector de parámetros mediante una distribución totalmente condicionada. Si estoy actualizando por ejemplo $x_j^{(t)}$, las componentes a las que se condicionan son, para $i<j$ a las de la iteración (t) puesto que ya se han actualizado, es decir, $\{x_1^{(t)},\ldots,x_{j-1}^{(t)}\}$. Para las $i>j$ a las de la iteración $(t-1)$ puesto que aun no se han actualizado, es decir, $\{x_{j+1}^{(t-1)},\ldots,x_p^{(t-1)}\}$. En el Anexo B.4 hay un ejemplo del muestreo de Gibbs.

Demostración. Veamos que este algoritmo efectivamente nos produce una cadena de Markov con distribución estacionaria f. Lo haremos en el caso $p=2$.

Sea $x=(x_1,x_2)$. El valor $x^{(t+1)}$ en la iteración $t+1$ se obtiene a partir de $x^{(t)}$ en dos pasos:

$$x_1^{(t+1)} \sim f\left(x_1 \mid x_2^{(t)}\right)$$
$$x_2^{(t+1)} \sim f\left(x_2 \mid x_1^{(t+1)}\right)$$

Por lo tanto el Kernel de transición es $K(x,x')$ con $x=(x_1,x_2)$ y $x'=(x_1',x_2')$ puede ser factorizado como:

$$K\left(x,x'\right)=K_1(x,\tilde{x})K_2\left(\tilde{x},x'\right)$$

donde $\tilde{x}=(x_1',x_2)$ es el resultado intermedio después del actualizar sólo la primera componente. De acuerdo con Michael Eichler en Markov Chain Monte Carlo (véase en [14]) es suficiente mostrar que se cumple la ecuación de equilibrio para K_1 y K_2, y entonces deducimos que f es la distribución estacionaria de la cadena. Por lo tanto se tiene que:

$$K_1(x,\tilde{x})=f\left(x_1' \mid x_2\right) \quad \text{y} \quad K_2\left(\tilde{x},x'\right)=f\left(x_2' \mid x_1'\right).$$

Notar que para cualquier x,x', se tiene $K_1\left(x,x'\right)=0$ si $x_2\neq x_2'$ y $P_2\left(x,x'\right)=0$ si $x_1\neq x_1'$. Por ello, para cualquier estado x,x' tal que $x_2=x_2'$,

$$\begin{aligned} f(x)K_1\left(x,x'\right) &= f\left(x_1,x_2\right)f\left(x_1' \mid x_2\right)=f\left(x_1 \mid x_2\right)f\left(x_1',x_2\right) \\ &= f\left(x_1 \mid x_2'\right)f\left(x_1',x_2'\right)=K_1\left(x',x\right)f\left(x'\right) \end{aligned}$$

[3]La palabra f_i *distribución totalmente condicionada* es en referencia a que estas se construyen condicionando a todos los parámetros menos el de interés, x_i.

mientras que para x,x' con $x_2 \neq x_2'$ la ecuación se cumple de manera trivial. Igualmente, para x,x' tal que $x_1 = x_1'$,

$$\begin{aligned} f(x)K_2\left(x,x'\right) &= f\left(x_1,x_2\right) f\left(x_2' \mid x_1\right) = f\left(x_2 \mid x_1\right) f\left(x_2',x_1\right) \\ &= f\left(x_2 \mid x_1'\right) f\left(x_1',x_2'\right) = K_2\left(x',x\right) f\left(x'\right), \end{aligned}$$

mientras que para x,x' con $x_1 \neq x_1'$ la ecuación se cumple de manera trivial. Todo esto en conjunto, muestra que $f(x)$ es la distribución estacionaria del muestreo de Gibbs. □

Intuitivamente se puede ver que el muestreo de Gibbs es mas simple que el Metropolis-Hastings en el caso de las distribuciones multivariantes puesto que es mucho más sencillo generar una componente θ_i cada vez que tener que generar $\theta = (\theta_1, \ldots, \theta_p)$ de manera conjunta.

2.6. Algoritmo híbrido. Metropolis-within-Gibbs

Destacar que en la práctica es habitual que no sea posible muestrear de alguna de las distribuciones totalmente condicionadas, y por lo tanto implementar el muestreo de Gibbs. Con el fin de solucionar este problema se combinan los algoritmos de Metropolis-Hastings y el muestreo Gibbs. Suponer que tenemos las densidades condicionadas $\{f_1 \ldots, f_p\}$ y hay alguna de ellas, por ejemplo f_i, de la que no es posible muestrear. La manera de proceder es usar el algoritmo del muestreo de Gibbs, y cuando haya que actualizar la componente i del vector, en vez de simular $x_i^{(t+1)} \sim f_i\left(x_i \mid x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_p^{(t)}\right)$, hacer uso de Metropolis-Hastings (M-H) para esta componente.

Algoritmo 6 Muestreo híbrido. Metropolis-within-Gibbs

En la iteración $t = 1, 2, \ldots, ,$ dado $x^{(t-1)} = \left(x_1^{(t-1)}, \ldots, x_p^{(t-1)}\right)$:

1. Para $j < i$ generar $x_j^{(t)}$ mediante la distribución totalmente condicionada.
2. Para $j = i$, generar $x_i^{(t)}$ mediante un algoritmo de tipo M-H. Para ello necesitamos una distribución propuesta q. En este caso la distribución objetivo es la densidad totalmente condicionada a i, es decir $f_i\left(x_i \mid x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, \ldots, x_p^{(t-1)}\right)$.
3. Para $j > i$ actualizar a $x_j^{(t)}$ mediante la distribución totalmente condicionada.

2.7. Análisis de la convergencia

Como ya se ha comentado, la inferencia basada en las observaciones generadas con la cadena de Markov se basa en la hipótesis de que la distribución de los valores generados con la cadena y t suficientemente alto se aproxima a la distribución objetivo. Es necesario suponer t suficientemente alto puesto que el resultado teórico en el que nos basamos establece que la distribución de la cadena converge a la distribución objetivo cuando t tiende a infinito. Por ello es necesario descartar un número m de iteraciones iniciales en concepto de que la cadena aun no ha convergido y eliminar la dependencia del valor inicial $x^{(0)}$ tomado. A este proceso se le conoce como *periodo de calentamiento*[4] o *burn-in period*. Si queremos obtener una muestra de tamaño n de la distribución objetivo, debemos generar $m+n$ valores, siendo los m primeros los que se desecharan en concepto de *burn-in period*. Es por ellos que los valores simulados por nuestra cadena a lo largo de las iteraciones son $(\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(m-1)}, \theta^{(m)}\theta^{(m+1)}, \theta^{(m+2)}, \ldots, \theta^{(m+n)})$ que se dividen en :

$$\underbrace{\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(m-1)}, \theta^{(m)}}_{\text{m valores desechados en concepto de burn-in}} \qquad \underbrace{\theta^{(m+1)}, \theta^{(m+2)}, \ldots, \theta^{(m+n)}}_{\text{n valores que vamos a tomar como muestra}}$$

[4]Proviene del inglés *warm-up* y a veces en la literatura anglosajona a este periodo se le llama *burn-in*.

Otra forma de evaluar el número de iteraciones m que se deben eliminar es mediante el gráfico de traza como veremos en la sección siguiente.

Además, la simulación de valores de una variable con la distribución objetivo utilizando los procedimientos basados en cadenas de Markov presenta principalmente dos inconvenientes o dificultades: 1) Asegurar la convergencia de la cadena. 2) Las observaciones generadas no son independientes sino que pueden estar fuertemente correladas por venir de una cadena de Markov.

En concreto, en el ámbito bayesiano, será necesario analizar la convergencia de las cadenas de todos los parámetros estimados e incluso de algunas medidas de interés. Por lo tanto, a partir de ahora, cuando hablemos de diagnósticos para las cadenas, se deberán realizar para todos parámetros o medidas de interés.

1) Convergencia

Gráficos de traza: Una forma de comprobar que la cadena de un método MCMC ha convergido es mediante el gráfico de traza para variable de interés. Estos gráficos se realizan representando para cada parámetro de interés x_i su valor frente al número de iteración t . Se puede observar lo siguiente:

- Si a la cadena le cuesta largos periodos de tiempo moverse por el espacio de la variable, entonces necesitará un número de iteraciones alto para converger.
- Si las iteraciones, para un número de tiempo alto, forman una serie estable en torno a un valor (no hay movimientos fuertes o fluctuaciones raras) entonces indican convergencia.
- Valores muy dispersos al principio del gráfico son los correspondientes al burn-in period y como se ha comentado antes deberían eliminarse.

El problema del gráfico de traza es que al simular sólo una cadena, puede parecer falsamente que hay convergencia, debido a que hay regiones del espacio que no se han explorado. Como relata George Casella en el libro *Introducing Monte Carlo Methods with R* (véase en [1, pág. 239-240]) una limitación de estos métodos es que si la cadena de Markov no ha producido suficientes iteraciones, pueden quedar grandes regiones sin explorar. Con el fin de verificar si la cadena ha convergido a la distribución objetivo, se simulan varias cadenas (al menos 2) con distintos valores de inicio. Cuando la varianza inter e intra en las cadenas sea indistinguible, podremos suponer que ha habido convergencia. Como se ha comentado al inicio de esta parte, a la hora de generar una muestra, es muy importante eliminar las iteraciones correspondientes al periodo de calentamiento.

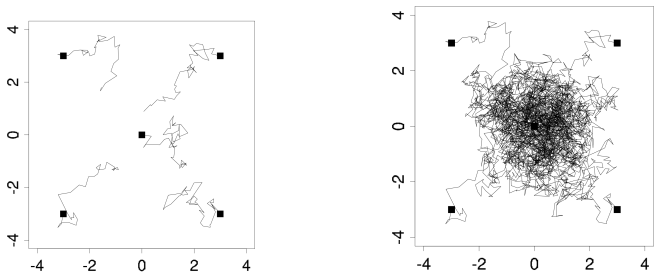

Figura 2.3: Varianza inter e intra. Gráfico obtenido de [2, pág. 276].

División de cadenas: De acuerdo con Gelman y Rubin (1992) (véase en [15]), para cada parámetro de interés, una vez simuladas m cadenas diferentes y eliminados los periodos de calentamiento (nos quedan n valores), cada cadena se divide en dos partes iguales. Podremos decir que se ha alcanzado convergencia si las cadenas antes de dividirlas se mezclaban bien, una vez divididas en dos, también lo hacen. Además si se ha alcanzado la convergencia, todas las cadenas (una vez divididas) deben converger a la misma distribución. Por lo tanto ahora tenemos el doble de cadenas, sean $2m$, y la longitud de cada una $n/2$.

Varianza intra e inter cadenas:
En la práctica, un método muy utilizado para analizar la convergencia de las cadenas para cada parámetro de interés, es el análisis de la varianza inter e intra cadenas, propuesto por Gelman y Rubin (véase [2]). En la primera gráfica de la Figura 2.3 se aprecia que las 5 cadenas simuladas no han convergido pues la varianza intra cadenas es mucho mas pequeña que la varianza inter cadenas, que es muy grande. En la segunda gráfica ya se puede ver como todas cadenas se han mezclado y las varianzas inter e intra cadena son prácticamente iguales. En estas gráficas se ve la necesidad de simular más de una cadena con puntos iniciales dispersos. El método de análisis de varianza inter e intra consta de los siguientes pasos:

1. Dispongamos de m cadenas de longitud n cada una, una vez eliminado el periodo de calentamiento. Sean estas $\{x_i^{(j)}\}$ con $1 \leq i \leq m$ indicando el numero de cadena y $1 \leq j \leq n$ la componente de la cadena.

2. Calcular la *varianza intra*, que es la media de las varianzas de cada cadena, definida como:
$$W = \frac{1}{m-1}\sum_{i=1}^{m} s_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_i^{(j)} - \overline{x_i})^2. \tag{2.6}$$

3. Calcular *varianza inter* cadenas, definida como
$$B = \frac{1}{m-1}\sum_{i=1}^{m}(\overline{x_i} - \overline{x})^2 \quad \text{con} \quad \overline{x} = \frac{1}{m}\sum_{i=1}^{m}\overline{x_i}. \tag{2.7}$$

4. Calcular la varianza estimada de x, mediante:
$$var(x) = \frac{n-1}{n}W + B. \tag{2.8}$$

5. Por último calcular el factor
$$\hat{R} = \sqrt{\frac{var(x)}{W}}. \tag{2.9}$$
Normalmente se aceptan las cadenas cuando se obtiene un $\hat{R}$ entre $\hat{R} < 1.1$

Es decir, si este factor es muy alto supondremos que no ha habido convergencia. Si este factor es cercano a 1, detendremos las cadenas suponiendo que ha habido convergencia. Para más detalles, véase Gelman [2, pág. 253- 255].

2)Muestras correladas

Debido a que la muestra obtenida para cada parámetro de interés proviene de una cadena de Markov, estas observaciones están correladas. Es por ello que está muestra nos proporciona menos información práctica que si tuviéramos una muestra de observaciones independientes. En concreto, se define el *tamaño efectivo de muestra* como el tamaño que tendría nuestra muestra si las observaciones fuesen independientes y su expresión viene dada por:
$$n_{\text{eff}} = \frac{mn}{1 + 2\sum_{t=1}^{\infty}\rho_h}. \tag{2.10}$$
con ρ_h la *correlación con retardo h* de la cadena de x_i. Esta es la autocorrelación de los valores de la serie distanciados h instantes en el tiempo. Si la cadena está convergiendo bien, para un número alto del retardo h, la correlación debería ir disminuyendo prácticamente a 0. Puede ser útil realizar un gráfico de autocorrelaciones dependiendo del retardo h, véase en Anexo B.3. Para más detalles, véase Gelman [2, pág. 286- 287].

Por último, destacar que para asegurar la convergencia es necesario aplicar varios de los métodos comentados, ya que en caso de confiar en uno sólo puede llevarnos a asegurar una falsa convergencia. Luego si hemos efectuado varios de ellos y hemos obtenido resultados satisfactorios en todos ellos, es razonable suponer que ha habido convergencia.

Capítulo 3

Aplicación de los métodos MCMC

En este capítulo vamos a aplicar los algoritmos MCMC vistos para ajustar la distribución a posteriori de los parámetros de un modelo para la temperatura máxima diaria en la ciudad de Pamplona.

3.1. Datos

Disponemos de los datos de la temperatura máxima diaria de la ciudad de Pamplona medidos en grados Celsius (°C), proporcionados por la Agencia Estatal de Meteorología (AEMET) desde 1976 hasta 2015 (40 años $\times$ 365 días) sin datos faltantes. Además se han eliminado los días 29 de febrero para facilitar el análisis. La estación que mide la temperatura está localizada a 442 metros de altitud sobre el nivel del mar, en las coordenadas latitud 42.818 y longitud -1.638. Dicha estación se encuentra en el centro de la ciudad en una zona de parque. En cuanto a la climatología de la capital navarra, esta se encuentra en la parte central-superior de la cuenca del Ebro. Su clima es de transición entre el atlántico y el mediterráneo. El objetivo de este trabajo es plantear y ajustar un modelo sencillo que recoja las principales características de la serie temporal. En la Figura 3.1 se muestran los 5 primeros y 5 últimos años para ver qué forma tiene dicha serie. Se observa una componente estacional y la temperatura de los últimos 5 años parece estar un poco por encima con respecto de la de los primeros 5 años.

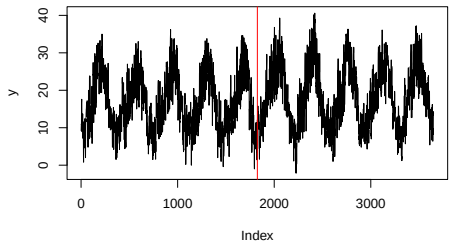

Figura 3.1: Representación de dichas temperaturas los 5 primeros años y los 5 últimos, separadas por una linea vertical.

3.2. Modelo autoregresivo

Un *Modelo Autoregresivo* (AR) representa procesos aleatorios, donde la variable de interés depende de las observaciones pasadas. Dado que las temperaturas tienen una fuerte correlación serial (las temperaturas de cada día dependen de los días anteriores) vamos a considerar un modelo AR bajo el paradigma bayesiano. En el Anexo C.1 se muestra que efectivamente no es adecuado ajustar este problema de las temperaturas máximas diarias por un modelo lineal que no tenga en cuenta dicha correlación. Volviendo al modelo AR, nuestro objetivo es un modelo que capture: la componente estacional, la tendencia a largo plazo (que por simplicidad asumimos lineal) mediante un armónico y la correlación serial (que por simplicidad asumimos de orden 1). Sea Y_t la temperatura máxima en el día t, entonces se plantea el modelo:

$$Y_t = \mu_t + \rho(Y_{t-1} - \mu_{t-1}) + \varepsilon_t \quad t = 2, \ldots, T \ , \tag{3.1}$$

donde ε_t es el término del error,

$$\mu_t = \beta_0 + \beta_1 t + \beta_2 \sin(2\pi t/365) + \beta_3 \cos(2\pi t/365) \tag{3.2}$$

y denotamos por $M_t = \mu_t + \rho(Y_{t-1} - \mu_{t-1})$, por simplificación. En un modelo AR ,suponiendo Y_1 conocido, la verosimilitud resulta $\prod_{t=2}^{T}[Y_t|Y_{t-1}, \beta_0, \beta_1, \beta_2, \beta_3, \rho, \sigma^2]$. Notar que $Y_t \sim N\left(\mu_t + \rho\left(Y_{t-1} - \mu_{t-1}\right), \sigma^2\right) = N(M_t, \sigma^2)$. De este modo μ_t captura mediante covariables la parte de la media asociada a la tendencia lineal y la componente estacional, ρ captura la correlación serial de primer orden y σ^2 representa la varianza del modelo. Ahora veamos qué distribuciones a priori les vamos a dar a los parámetros.

$$\begin{aligned} \beta_i &\sim N(a_i, b_i^2) = N(0, 10^6), && i = 0, \ldots, 3 \\ \rho &\sim Beta(a_\rho, b_\rho) = Beta(1/2, 1/2) \\ \sigma^2 &\sim \text{InvGamma}(a_\sigma, b_\sigma) = \text{InvGamma}(2, 1) \end{aligned} \tag{3.3}$$

Las a priori escogidas para β_i y σ^2 van a ser normales e inversa gamma pues son las distribuciones a priori conjugadas para nuestra verosimilitud. Por lo tanto las distribuciones completamente condicionadas resultantes se podrán simular directamente y utilizar un muestreo de Gibbs para obtener muestras de los parámetros. Para ρ vamos a tomar una beta pues hay evidencia de que la temperatura menos las covariables introducidas sigue un proceso estacionario con persistencia positiva, por lo que restringimos el valor de ρ a (0,1) a priori. Como distribución a priori tomaremos $Beta(1/2, 1/2)$. Debido a que para ρ no va a ser una distribución conjugada, emplearemos un algoritmo *Metropolis-within-Gibbs*, actualizando ρ mediante un paseo aleatorio de Metropolis.

3.2.1. A posteriori totalmente condicionada para los β_j con $j = 0, 1, 2, 3$

Veamos las distribuciones a posteriori totalmente condicionada para $\beta_0, \beta_1, \beta_2, \beta_3$. Sea en nuestro modelo $\beta_j x_{t,j}$ con $j = 0, 1, 2, 3$. P.ej. si $j = 0$ entonces $x_{t,0} = 1$ por ser el intercepto. Denotamos por μ_t^* a μ_t sin el término correspondiente a β_j. Por el Teorema 1.1.2 sabemos que la distribución totalmente condicionada a posteriori para β_j es proporcional a:

$$\begin{aligned} p(\beta_j|\cdots) &\propto \prod_{t=2}^{T} N(Y_t|\mu_t + \rho(Y_{t-1} - \mu_{t-1}), \sigma^2) \times N(\beta_j|a_j, b_j) \\ &\propto N(\beta_j|\mu_{\beta_j}, \sigma_{\beta_j}^2) \times N(\beta_j|a_j, b_j) \end{aligned}$$

Veamos que la verosimilitud se distribuye como una normal y con que parámetros lo hace. En el cálculo de la distribución correspondiente vamos quitando las partes que no dependen de β_j, que es la variable de interés. Ver dichos cálculos en Anexo C.2. Tenemos entonces que:

$$\prod_{t=2}^{T} N(Y_t|\mu_t + \rho(Y_{t-1} - \mu_{t-1}), \sigma^2) \propto N(\beta_j|\mu_{\beta_j}, \sigma_{\beta_j}^2) \quad \text{con}$$

$$\mu_{\beta_j} = \frac{\sum_{t=2}^{T}(x_{t,j} - \rho x_{t-1,j})(Y_t - \mu_t^* - \rho(Y_{t-1} - \mu_{t-1}^*))}{\sum_{t=2}^{T}(x_{t,j} - \rho x_{t-1,j})^2}, \quad \sigma_{\beta_j}^2 = \frac{\sigma^2}{\sum_{t=2}^{T}(x_{t,j} - \rho x_{t-1,j})^2}.$$

Luego tenemos que la a posteriori para β_j es un producto de normales

$$p(\beta_j|\cdots) \propto N(\beta_j|\mu_{\beta_j}, \sigma_{\beta_j}^2) \times N(\beta_j|a_j, b_j)$$

y en conclusión, que la distribución a priori es conjugada, con la siguiente distribución a posteriori totalmente condicionada:

$$p(\beta_j|\cdots) \sim N\left(\frac{\frac{\mu_{\beta_j}}{\sigma_{\beta_j}^2} + \frac{a_j}{b_j^2}}{\frac{1}{\sigma_{\beta_j}^2} + \frac{1}{b_j^2}}, \frac{1}{\frac{1}{\sigma_{\beta_j}^2} + \frac{1}{b_j^2}}\right).$$

3.2.2. A posteriori totalmente condicionada para σ^2

Sea $\sigma^2 \sim \text{InvGamma}(a_\sigma, b_\sigma)$. Por el Teorema 1.1.2 sabemos que la distribución totalmente condicionada a posteriori para σ^2 es proporcional a:

$$p\left(\sigma^2 \mid \cdots\right) \propto \prod_{t=2}^{T} N(Y_t|\mu_t + \rho(Y_{t-1} - \mu_{t-1}), \sigma^2) \times \text{Invgamma}(\sigma^2|a_\rho, b_\rho).$$

Realizando cálculos se llega a que la distribución a posteriori es también una inversa gamma. Por lo tanto se trata de una distribución conjugada. Véase cálculos en el Anexo C.2. En concreto la distribución a posteriori es la siguiente inversa gamma:

$$p(\sigma^2 \mid \cdots) \sim InvGamma\left(a_\sigma + \frac{T-1}{2}, b_\sigma + \frac{\sum_{t=2}^{T}(y_t - M_t)^2}{2}\right). \tag{3.4}$$

3.2.3. A posteriori totalmente condicionada para ρ

En este caso la a posteriori totalmente condicionada con una a priori $\rho \sim Beta(a_\rho, b_\rho)$ respecto a un muestreo normal, no es una distribución conjugada. Es por ello que para actualizar este parámetro en nuestra cadena de Markov usaremos un paseo aleatorio de M-H con distribución objetivo la totalmente condicionada, que por el Teorema 1.1.2 es proporcional a:

$$p(\rho|\cdots) \propto \prod_{t=2}^{T} N(Y_t|\mu_t + \rho(Y_{t-1} - \mu_{t-1}), \sigma^2) \times Beta(\rho|a_\rho, b_\rho) \tag{3.5}$$

Esto no es proporcional a ninguna densidad conocida en general, por lo que aplicaremos el algoritmo de Metropolis. Tener en cuenta que el soporte de esta densidad está en $(0,1)$ por lo que vamos a muestrear de una transformación de ρ en la recta real, e.g., con una transformación logit:

$$Z_\rho \equiv \text{logit}(\rho) = \log\left(\frac{\rho}{1-\rho}\right), \quad \text{con inversa} \quad \rho = \frac{1}{1+\exp(-Z_\rho)}. \tag{3.6}$$

Entonces se quiere simular Z_ρ y la distribución propuesta debe ser una densidad normal centrada en el valor del parámetro Z_ρ en la iteración anterior y con varianza tal que la proporción de aceptación sea alrededor del 15 o 40 %. La distribución objetivo vendrá dada, salvo constante, por:

$$p(Z_\rho|\cdots) \propto \prod_{t=2}^{T} N\left(Y_t|\mu_t + \frac{1}{1+\exp(-Z_\rho)}(Y_{t-1} - \mu_{t-1}), \sigma^2\right) \times Beta\left(\frac{1}{1+\exp(-Z_\rho)}|a_\rho, b_\rho\right) \tag{3.7}$$

3.2.4. Algoritmo híbrido Metropolis-within-Gibbs

Puesto que hemos encontrado que las distribuciones a posteriori totalmente condicionadas para $\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2$ son conjugadas, para estas implementaremos un muestreo de Gibbs muestreando de las distribuciones a posteriori totalmente condicionadas. Cuando tengamos que actualizar ρ aplicaremos un paseo aleatorio de M-H con función propuesta $N(Z_p, var)$, ajustando la varianza para que se acepten entre un $15-40\,\%$ de valores, de acuerdo con Gelman, Roberts y Gilks en [19]. Tras varias pruebas obtenemos que el valor idóneo para ello es $var = 0.01$. La distribución objetivo es la totalmente condicionada a posteriori para ρ, siendo esta (3.7) .

En el Anexo C.2.4 hemos desarrollado un código que ejecute este algoritmo y lo hemos incluido en una función que nos calcula dicho algoritmo Metropolis−within−Gibbs dando como entrada valores iniciales a las cadenas de parámetros. A esta función le hemos llamado `metropolis_within_gibbs`.

Algoritmo 7 Muestreo híbrido. Metropolis-within-Gibbs

En la iteración $i = 1,2,\ldots,$, dado $\theta^{(i-1)} = \left(\beta_0^{(i-1)}, \beta_1^{(i-1)}, \beta_2^{(i-1)}, \beta_3^{(i-1)}, \sigma^{2(i-1)}, \rho^{(i-1)}\right)$:

1. Generar $\beta_0^{(i)} \sim p\left(\beta_0 | y, \beta_1^{(i-1)}, \beta_2^{(i-1)}, \beta_3^{(i-1)}, \sigma^{2(i-1)}, \rho^{(i-1)}\right)$ mediante su distribución a posteriori totalmente condicionada.
2. Generar $\beta_1^{(i)} \sim p\left(\beta_1 | y, \beta_0^{(i)}, \beta_2^{(i-1)}, \beta_3^{(i-1)}, \sigma^{2(i-1),\rho^{(i-1)}}\right)$ mediante su distribución a posteriori totalmente condicionada.
3. Generar $\beta_2^{(i)} \sim p\left(\beta_2 | y, \beta_0^{(i)}, \beta_1^{(i)}, \beta_3^{(i-1)}, \sigma^{2(i-1),\rho^{(i-1)}}\right)$ mediante su distribución a posteriori totalmente condicionada.
4. Generar $\beta_3^{(i)} \sim p\left(\beta_3 | y, \beta_0^{(i)}, \beta_1^{(i)}, \beta_2^{(i)}, \sigma^{2(i-1)}, \rho^{(i-1)}\right)$ mediante su distribución a posteriori totalmente condicionada.
5. Generar $\sigma^{2(i)} \sim p\left(\sigma^2 | y, \beta_0^{(i)}, \beta_1^{(i)}, \beta_2^{(i)}, \beta_3^{(i)}, \rho^{(i-1)}\right)$ mediante su distribución a posteriori totalmente condicionada.
6. Para $\rho^{(i)}$, aplicar algoritmo paseo aleatorio de M-H. Para ello necesitamos una distribución propuesta $q = N(Z_{\rho^{(i-1)}}, sd = 0.1)$ con $Z_\rho \equiv \text{logit}(\rho)$. En este caso la distribución objetivo es la densidad totalmente condicionada a i, es decir $p\left(Z_{\rho^{(i-1)}} \mid \beta_0^{(i)}, \beta_1^{(i)}, \beta_2^{(i)}, \beta_3^{(i)}, \sigma^{2(i)}\right)$. Una vez obtenido un valor de $Z_{\rho^{(i)}}$ se realiza $\rho^{(i)} = 1/(1+\exp(-Z_{\rho^{(i)}}))$.

Ahora simulamos dos cadenas de 10000 observaciones cada una para cada parámetro, y posteriormente descartamos las 5000 primeras iteraciones en concepto de burn-in. Los valores iniciales para cada cadena de $(\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2, \rho)$ son $cadena1 = (0,0,0,0,1,0.5)$ y $cadena2 = (-4,5,6,7,4,0.8)$, dispersos entre si (con la σ^2 siempre positiva y $\rho \in (0,1)$).

```
# cadena1 con valores iniciales para beta0,beta1,beta2,beta3,sigma2,rho
cadena1<-metropolis_within_gibbs(0,0,0,0,1,0.5)
length(unique(cadena1$rho))/10000*100  #porcentaje de  aceptados para rho
# cadena2 con  valores iniciales para beta0,beta1,beta2,beta3,sigma2,rho
cadena2<-metropolis_within_gibbs(-4,5,6,7,4,0.8)
length(unique(cadena2$rho))/10000*100
#eliminamos los burn-in, la mitad, 5.000
cadena1efectiva<-cadena1[c(5001:10000), ]
cadena2efectiva<-cadena2[c(5001:10000), ]
```

Una vez eliminadas las iteraciones de calentamiento en cada cadena, mostramos en la Figura 3.2 para β_0, β_1 las dos cadenas resultantes para ver que convergen a la misma distribución. Vemos que para cada parámetro ambas dos cadenas simuladas parecen converger a la misma distribución, lo que es un buen indicador. En la Figura C.3 del Anexo C.2.4 se pueden ver para el resto de variables $\beta_2, \beta_3, \sigma^2, \rho$. Veamos ahora también los siguientes diagnósticos para cada parámetro:

- *Diagnóstico de Gelman*, del paquete `coda` que nos da el *factor de reducción de escala* para cada parámetro. Este se basa en el *Análisis de la varianza intra e inter cadenas*, visto en la Sección 2.7, y conforme más se acerca a 1 mejor convergencia a la distribución objetivo se da. La entrada de esta orden requiere de un objeto `mcmc.list()`, por lo que juntamos las dos cadenas de 5000 iteraciones que tenemos y lo incluimos como un objeto de dicho tipo.
- *Tamaño de muestra efectivo*, definido en la Sección 2.7 y calculado en R mediante la orden `coda::effectiveSize()`.

- Media a posteriori e intervalo de credibilidad al 90 %.

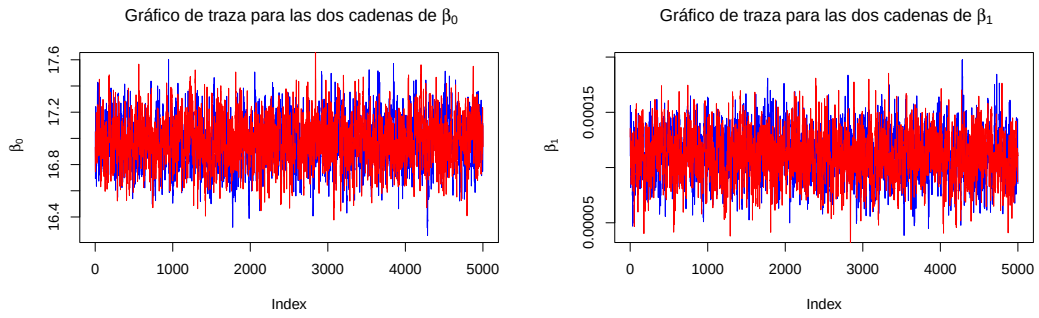

Figura 3.2: Gráficos de traza para β_0 y β_1. En azul se representa la primera cadena simulada y en rojo la segunda.

Notar que tenemos dos cadenas de 5000 observaciones cada una, es decir, 10000 observaciones en total. Observando la Tabla 3.3 vemos que los factores reductores de escala para todos parámetros son uno, y esto nos indica que parece haber convergencia, y que por lo tanto no es necesario dejar al algoritmo correr mas iteraciones. Para β_0, β_1, ρ podemos observar un tamaño efectivo de muestra muy inferior al real, lo cual es debido a la alta correlación serial que tienen las cadenas de Markov para estos parámetros. Los resultados muestran que ninguno de los intervalos de credibilidad de los parámetros contiene el valor 0, lo que indica que todos los términos incluidos en el modelo son necesarios. El parámetro β_0 indica que la media de la temperatura (sin tener en cuenta el efecto estacional) en el comienzo del periodo observado es 16.9ºC. El parámetro β_1 es positivo, luego indica que esa media va aumentando cada año. La correlación de la temperatura de un día con la del día anterior es también positiva y bastante fuerte, 0.68.

	β_0	β_1	β_2	β_3	σ^2	ρ
Factor de reducción de escala	1	1	1	1	1	1
Tamaño efectivo muestra	1421.83	1433.04	10411.14	9972.41	10000	2158.770
Media a posteriori	16.978	0.00011	-3.360	-8.600	11.666	0.686
IC al 90%	16.67 17.27	7.49e-05 1.46e-04	-3.568 -3.148	-8.811 -8.389	11.448 11.893	0.676 0.696

Tabla 3.3: Tabla con las medidas citas anteriormente.

Anexo A

Algunos resultados de teoría

Teorema A.0.1. *Sea Y una v.a. , $y = (y_1, \ldots, y_n)$ una muestra de dicha variable, $\theta = (\theta_1, \ldots, \theta_p)$ con $1 \leq p$ el vector paramétrico de dicha distribución , $p(\theta)$ la función de densidad a priori para el parámetro y $p(y|\theta)$ la verosimilitud de la muestra y . La densidad a posteriori del parámetro θ se denota como $p(\theta|y)$ y se define mediante la siguiente expresión:*

$$p(\theta \mid y_1, \ldots, y_n) = \frac{p(y_1, \ldots, y_n; \theta)\, p(\theta)}{\int p(y_1, \ldots, y_n; z)\, p(z) dz} = \frac{p(y_1, \ldots, y_n; \theta)\, p(\theta)}{p(y)} \tag{A.1}$$

Notar que en caso de θ v.a. discreta, entonces cambiamos las integrales por sumatorios.

Demostración. a)

Caso θ v.a. discreta. Luego θ tendrá f.m.p. Para la demostración nos restringimos a $Dim(\theta) = 1$. Si la f.m.p de θ toma valores $\{\theta_k\}$ su f.m.p a posteriori es:

- Supongamos primero Y v.a. discreta. En este caso $p(y)$ representara la f.m.p de la Y. Por el teorema de Bayes:

$$\begin{aligned} p(\theta = \theta_k \mid Y_1 = y_1, \ldots, Y_n = y_n) &= \frac{p(Y_1 = y_1, \ldots, Y_n = y_n \mid \theta = \theta_k)\, p(\theta = \theta_k)}{\sum_i p(Y_1 = y_1, \ldots, Y_n = y_n \mid \theta = \theta_i)\, p(\theta = \theta_i)} \\ &= \frac{p(Y_1 = y_1, \ldots, Y_n = y_n \mid \theta_k)\, p(\theta_k)}{\sum_i p(Y_1 = y_1, \ldots, Y_n = y_n \mid \theta_i)\, p(\theta_i)} \end{aligned}$$

- Supongamos ahora Y v.a. continua. En este caso p representara la f. densidad cuando se hable de las y. Por resultados de cálculo de probabilidades:

$$p(\theta = \theta_k \mid Y_1 = y_1, \ldots, Y_n = y_n) = \frac{p(y_1, \ldots, y_n \mid \theta_k)\, p(\theta = \theta_k)}{\sum_i p(y_1, \ldots, y_n \mid \theta_i)\, p(\theta = \theta_i)} = \frac{p(y_1, \ldots, y_n; \theta_k)\, p(\theta_k)}{\sum_i p(y_1, \ldots, y_n; \theta_i)\, p(\theta_i)}$$

b) Caso θ v.a. continua. Luego θ tendrá f.densidad y no función de masa de probabilidad. Luego en vez de realizar sumatorios, haremos integrales. Para la demostración nos restringimos a $Dim(\theta) = 1$.

- Supongamos primero Y v.a. discreta. En este caso $p(y)$ representara la función de densidad de la Y.

$$p(\theta \mid Y_1 = y_1, \ldots, Y_n = y_n) = \frac{p(Y_1 = y_1, \ldots, Y_n = y_n \mid \theta)\, p(\theta)}{\int p(Y_1 = y_1, \ldots, Y_n = y_n \mid z)\, p(z)\, dz}$$

- Supongamos ahora Y v.a. continua. En este caso p representara la f.densidad cuando se hable de las y. $p(\theta \mid Y_1 = y_1, \ldots, Y_n = y_n) = \frac{p(y_1, \ldots, y_n \mid \theta)\, p(\theta)}{\int p(y_1, \ldots, y_n \mid z)\, p(z)\, dz}$

□

Teorema A.0.2. *Ley Fuerte de Grandes Números*

Sea $X_1, X_2, X_3, \ldots$ *una sucesión de v.a's independientes e idénticamente distribuidas que cumple* $\mathrm{E}[X_i] < \infty$ *y con esperanza* $\mathrm{E}[X_i] = \mu$ *entonces:*

$$\mathrm{P}\left(\lim_{n\to\infty} \bar{X}_n = \mu\right) = 1$$

es decir, la media de las v.a's converge a μ *casi seguramente.*

Anexo B

Script utilizado en R

B.1. Ejemplo Muestreo por importancia

```
# Densidad objetivo
# f <- dnorm # f <- function(x) ....

yy <- rt(1000,df=3)
#hisrograma
hist(yy, freq = FALSE, breaks = "FD", ylim = c(0, 0.5),xlab = 'valores',
    main = 'Comparción' )
#densidad objetivo
curve(dnorm, col = "RED", add = TRUE,lty='longdash')
#añadimos leyenda
legend("topright", legend = c( 'densidad N(0,1)',  'muestra t_3'),
       bty = "n",
       col = c("red",NA),
       lty = c('longdash',NA),
       density=c(0,100),
       fill = c( "red","gray"),
       border = c(NA,"black"),
       x.intersp=c(2,0.5))

----------------------------------------
set.seed(785248)
nsim <-c(10^7)
y <- rt(nsim,df=3)
w <- dnorm(y)/dt(y,df=3)

est<-1/(length(y))*sum(mean(y)*w)
est

>>>est=6.304878e-05
```

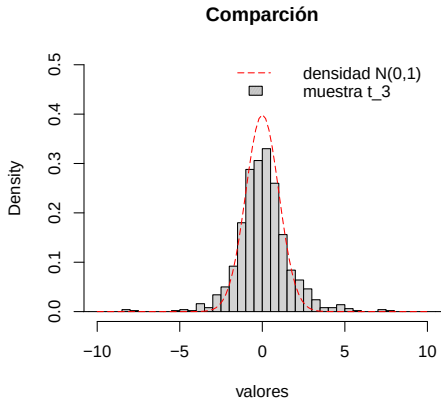

Figura B.1: Histograma de frecuencias relativas de una muestra de una t_3 y densidad de la $N(0,1)$. Notar que se ajustan muy bien los valores simulados a la densidad buscada.

B.2. Ejemplo Muestreo por Rechazo

```
shape1 <- 3
shape2 <- 5
curve(dbeta(x, shape1, shape2),  lwd = 2,xlim = c(0,1.9),ylab = '')
M <- dbeta((shape1 - 1)/(shape1 + shape2 - 2), shape1, shape2)
reescal<- M*1
abline(h = reescal, lty = 2,col='red')

legend(x = 0.6,y=2.1, legend = c( 'cota M superior',  'densidad B(3,5)'),
       col = c("red",'black'),
       lty = c(2,1),
       lwd = c(1,2),
       density=c(0,0),
       border = c(NA,NA))



---------------------------------------
nmuestra <- 0
shape1 = 3
shape2 = 5

beta2 <- function(shape1 = 3, shape2 = 5)
{        M <- dbeta((shape1 - 1)/(shape1 + shape2 - 2), shape1, shape2)
        while (TRUE) {
                U <- runif(1)
                X <- runif(1)
                nmuestra <<- nmuestra+1
                if (M*U <= (dbeta(X, shape1, shape2))/1) return(X)
        }}

beta2muestra <- function(n = 1000, shape1 = 3, shape2 = 5) {
        x <- numeric(n)
        for(i in 1:n) x[i]<-beta2(shape1, shape2)
        return(x)}

set.seed(785248)
nsim <- 1000
nmuestra <- 0
x <- beta2muestra(nsim, shape1, shape2)

hist(x, breaks = 20, freq = FALSE,xlim=c(0,1.9),main='')
curve(dbeta(x, shape1, shape2), col = 2, lwd = 2, add = TRUE)

legend(x = 0.5,y=2.2, legend = c( 'densidad B(3,5)',  'Valores del muestreo por rechazo
       col = c("red",'black'),cex=0.7,
       lty = c(1,NA),
       density=c(0,100),
       lwd = c(1,2),
       fill = c( "red","gray"),
       border = c(NA,"black") )
```

B.3. Ejemplo gráfico de traza y de correlaciones

En la primera gráfica de la Figura B.2 podemos ver un ejemplo donde se produce un buen mezclado de la cadena de Markov y por ello a lo largo de las iteraciones los valores de la cadena parecen estabilizarse en torno a una región. Por el contrario, en la segunda podemos ver una cadena con un mezclado bastante precario donde a lo largo de las iteraciones la cadena se queda estancada en determinados valores durante periodos de tiempo.

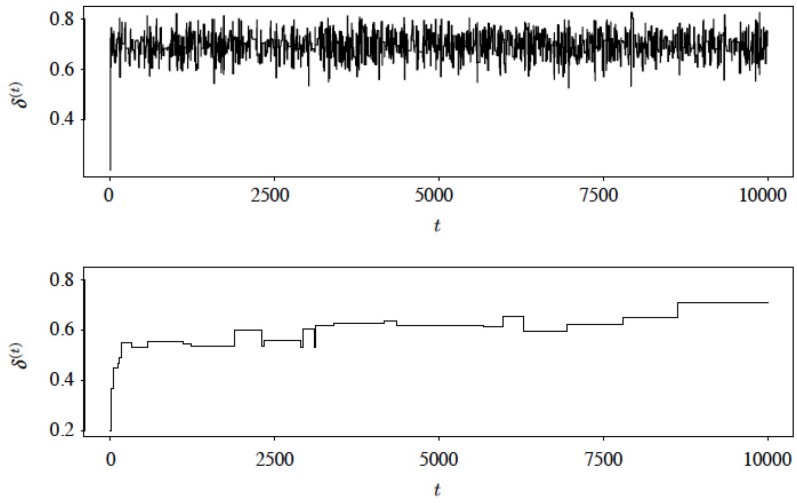

Figura B.2: Gráfico de traza que presenta buen y mal mezclado.

Por otro lado, el *gráfico de correlaciones* consiste en mostrar la correlación de $x^{(t)}$ a lo largo de las iteraciones con retardo i. La *correlación con retardo i* (proviene del inglés *lag*) se define como la correlación entre pasos que están separados por i iteraciones. Una cadena que tiene malas propiedades de mezcla exhibirá un decaimiento lento de la autocorrelación a medida que aumenta el retraso entre las iteraciones. En la estacionaridad esta correlación debería desaparecer, ya que es análogo a un muestreo iid.

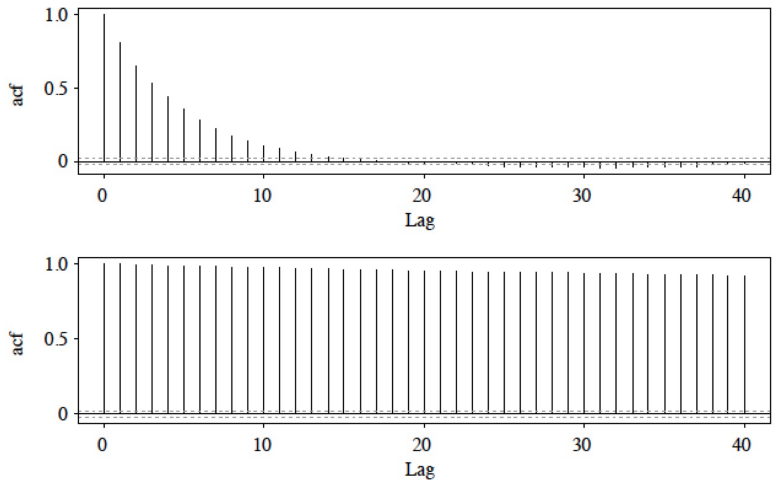

Figura B.3: Ahora en la gráfica superior podemos ver un buen ejemplo de disminución de las correlaciones conforme aumenta el *lag* entre iteraciones, lo que confirma un buen mezclado. En la gráfica inferior hay un mal mezclado debido a que esta correlación no disminuye conforme aumenta el *lag*.

B.4. Ejemplo muestra normal bivariante mediante muestreo de Gibbs

Suponer que queremos muestrear θ_1, θ_2, que siguen la distribución normal bivariante de manera que:

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right]$$

donde θ_1 y θ_2 son los parámetros del modelo y ρ es la correlación entre θ_1 y θ_2. Por las propiedades de la normal bivariante tenemos que:

$$\begin{aligned} \theta_1 \mid \theta_2, y \sim N\left(\rho\theta_2, 1-\rho^2\right) \sim \text{ estandarizando } \sim \rho\theta_2 + \sqrt{1-\rho^2}N(0,1) \\ \theta_2 \mid \theta_1, y \sim N\left(\rho\theta_1, 1-\rho^2\right) \sim \text{ estandarizando } \sim \rho\theta_1 + \sqrt{1-\rho^2}N(0,1) \end{aligned}$$

Vamos a implementar el muestreo e Gibbs bajo las siguientes condiciones iniciales:

1. Obtendremos 10000 para cada θ_i tras haber eliminado las correspondientes al burn-in.

2. Inicialmente fijaremos un periodo de burn-in de 1000 iteraciones, y luego veremos si es razonable.

3. Conocemos ρ que es 0.6

Algoritmo de Gibbs Para implementar el muestreo de gibbs tomamos como valores iniciales $(\theta_1^{(0)}, \theta_2^{(0)}) = (0,0)$. Ahora el algoritmo es el que sigue:

En la iteración $t = 2, \ldots, 11000$, dado $\theta^{(t-1)} = \left(x_1^{(t-1)}, \theta_2^{(t-1)}\right)$:

1. $\theta_1^{(t)} \sim f_1\left(\theta_1 \mid \theta_2^{(t-1)}\right) \sim 0.6\theta_2^{(t-1)} + \sqrt{1-0.6^2}N(0,1)$;

2. $\theta_2^{(t)} \sim f_2\left(\theta_2 \mid \theta_1^{(t)}\right) \sim 0.6\theta_1^{(t)} + \sqrt{1-0.6^2}N(0,1)$;

Veamos como programarlo en R:

Primero creamos una matriz para θ_1 con 3 filas correspondientes a cada una de las 3 cadenas que vamos a simular y con 11.000 columnas. Lo mismo para θ_1. Después fijamos los valores iniciales para cada cadena y programamos el Muestreo de Gibbs tal y como se ha indicado en la diapositiva anterior

```
rho<-0.6
final_iter<-10000
burn_in_iter<-1000
total<-final_iter+burn_in_iter
-------------------------------------
#Inicializamos a 0 y fijamos semilla
theta1<-matrix(0,nrow = 3,ncol = total)
theta2<-matrix(0,nrow = 3,ncol = total)
set.seed(785248)

#Gibbs sampler con 3 cadenas para cada parámetro
#con diferentes puntos de partida
fraccion<-sqrt(1-0.6^2)
for(i in 1:3){

  theta1[1,1]=0
  theta2[1,1]=0
  theta1[2,1]=0.3
  theta2[2,1]=0.6
  theta1[3,1]=0.1
  theta2[1,3]=0.87
for (j in 2:total) {
  theta1[i,j]<-0.6*theta2[i,j-1]+fraccion*rnorm(n=1)
  theta2[i,j]<-0.6*theta1[i,j]+fraccion*rnorm(n=1)
}}
```

Ahora mostramos en los siguientes graficos las trazas de las tres cadenas para θ_1 y θ_2 sin eliminar los burn-in periods. Representamos cada cadena de un color:

```
#Cargamos librerias y establecemos cadenas
library(ggplot2)
library(latex2exp)
primeracol<-c(1:11000)
cadena1<-theta1[1,]
cadena2<-theta1[2,]
cadena3<-theta1[3,]

#grafico
colores<-c("cadena1"="lightblue","cadena2"="orange","cadena3"="lightgreen")
a <- ggplot(data.frame(primeracol,cadena2,cadena3,cadena1))
grafico<-a+geom_path(aes(x=primeracol,y=cadena1,color="cadena1"))+
  geom_path(aes(x=primeracol,y=cadena2,color="cadena2"))+
  geom_path(aes(x=primeracol,y=cadena3,color="cadena3"))+
  scale_color_manual(name="Cadenas",values = colores)
grafico+theme_grey()+
  labs(
    x = "iteración t",
    y=TeX(r'($\theta _1$)'),
    title = TeX(r'(Gráfico de traza para $\theta _1$)'))+scale_y_continuous(limits=c(-5,6)
  scale_x_continuous(breaks=seq(0,11000,by=2000))
```

```
  #Lo mismo para theta2

cadenaa1<-theta2[1,]
cadenaa2<-theta2[2,]
cadenaa3<-theta2[3,]

colores<-c("cadenaa1"="lightblue","cadenaa2"="orange","cadenaa3"="lightgreen")
b <- ggplot(data.frame(primeracol,cadenaa2,cadenaa3,cadenaa1))
graficoo<-b+geom_path(aes(x=primeracol,y=cadenaa1,color="cadenaa1"))+
  geom_path(aes(x=primeracol,y=cadenaa2,color="cadenaa2"))+
  geom_path(aes(x=primeracol,y=cadenaa3,color="cadenaa3"))+
  scale_color_manual(name="Cadenas",values = colores)

graficoo+theme_grey()+
  labs(
    x = "iteración t",
    y=TeX(r'($\theta _2$)'),
    title = TeX(r'(Gráfico de traza para $\theta _2$)'))+scale_y_continuous(limits=c(-5,6)
  scale_x_continuous(breaks=seq(0,11000,by=2000))
```

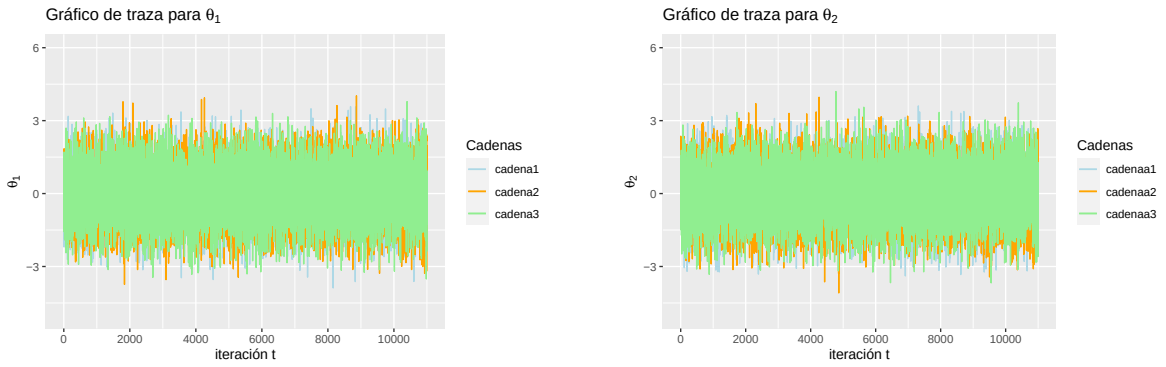

Figura B.4: En la primera gráfica vemos las tres cadenas simuladas para θ_1 y en la segunda gráfica las tres cadenas simuladas para θ_2.

Ahora vamos a eliminar el periodo del burn-in de cada cadena y vamos a representar la gráfica de autocorrelación para una cadena de cada parámetro.

```
#eliminamos burnin
theta1<-theta1[,-c(1:burn_in_iter)]
theta2<-theta2[,-c(1:burn_in_iter)]
#correlaciones
acf(theta1[1,])
acf(theta2[1,])
```

Vemos que para retardos mayores que 5, las autocorrelaciones ya están por debajo de 0.05 lo cual es aceptable. Por último dibujemos ahora las densidades estimadas de cada cadena para cada parámetro.

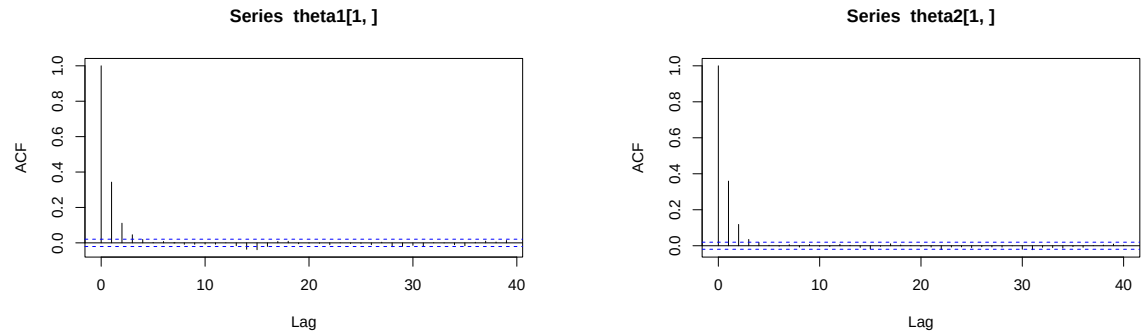

Figura B.5: En la primera gráfica vemos la autocorrelación para una cadena de θ_1 y en la segunda gráfica vemos la autocorrelación para una cadena de θ_2.

```
hist(theta1[1,],probability = TRUE)
dx<-density(theta1[1,])
dx2<-density(theta1[2,])
dx3<-density(theta1[3,])
lines(dx,col='blue')
lines(dx2,col='red')
lines(dx3,col='green')

hist(theta2[2,],probability = TRUE)
dx<-density(theta2[1,])
dx2<-density(theta2[2,])
dx3<-density(theta2[3,])
lines(dx,col='blue')
lines(dx2,col='red')
lines(dx3,col='green')
```

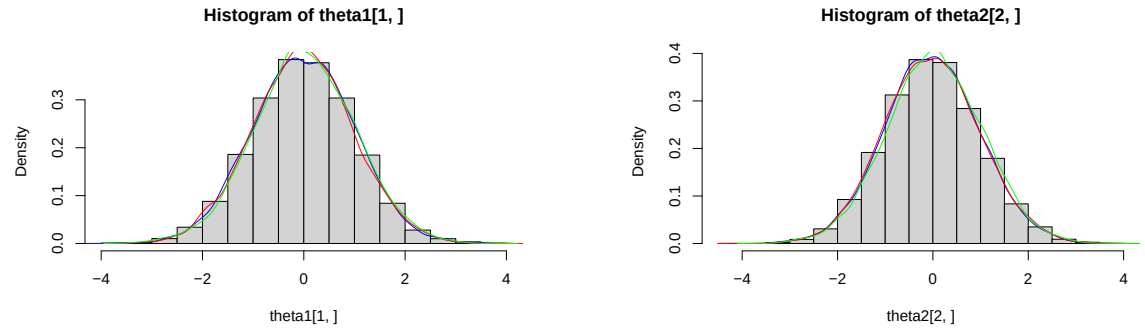

Figura B.6: Densidades de las cadenas simuladas para cada parámetro.

Anexo C

Aplicación de los métodos MCMC

C.1. Modelo RLM

Como se ha comentado en las temperaturas de Pamplona se observa una fuerte componente estacional. Podríamos introducir una tendencia lineal para comprobar si hay una tendencia de cambio climático significativa, ya que la temperatura de los últimos 5 años parece estar un poco por encima con respecto de la de los primeros 5 años. Vamos a plantear un modelo de regresión lineal múltiple bajo el paradigma frecuentista donde los parámetros son fijos y desconocidos para los cuales se ajustan unos estimadores. El número de datos es $40 \times 365 = 14600$. Luego presentamos el modelo $Y = \beta_0 + \beta_1 t + \beta_2 \sin(2\pi t/365) + \beta_3 \cos(2\pi t/365)$, con las siguientes covariables:

- la tendencia lineal t
- Un armónico: $\sin(2\pi t/365)$ y $\cos(2\pi t/365)$

```
y<-readRDS("/David/TFG/R/pamplona19762015.rds")

#grafica de 5 primeros y 5 ultimos años
 plot(y[c(1:(5 * 365), (35 * 365 + 1):(40 * 365))], ylab = "y", type = "l")
 abline(v = 5 * 365 + 0.5, col = "red")

 t<-1:T #numero de observaciones que tenemos de la temperatura
uno<-rep(1,T)
seno<-sin(2*pi*t/365)
cose<-cos(2*pi*t/365)


#modelo RLM con un UN ARMÓNICO
T<-length(y)
t <- 1:T
model1 <- lm(y ~ t + seno + cose)
```

Veamos como ajusta el modelo nuestros datos:

```
plot(rowMeans(matrix(y, nrow = 365)),
     xlab = "Day", ylab = "y", ylim = c(-3, 40),
     col = "gray", type = "l")
for (i in 1:40) {
  lines(x = 1:365, y = matrix(y, nrow = 365)[, i], type = "l", col = "gray")
}
lines(x = 1:365, y = rowMeans(matrix(y, nrow = 365)), col = "green")
lines(model1$fitted.value[1:365], col = "red")    #del año 1
lines(model1$fitted.value[1:365 + 39 * 365], col = "red")   # del ultimo año
#le sumo adelante y atras
```

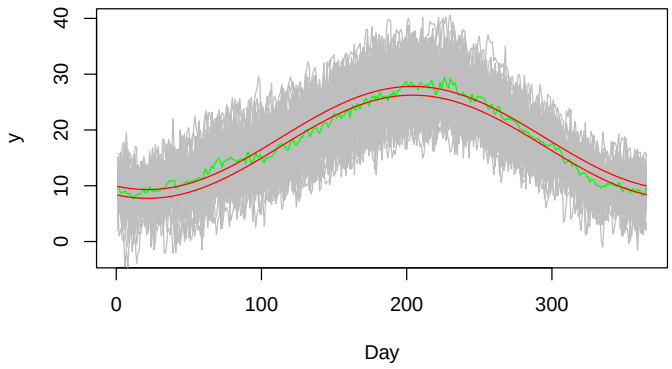

Figura C.1: Modelo con un armónico.

Notar que a la media de las temperaturas máximas de cada día para los 40 años, en color verde, no queda del todo bien ajustada por el modelo lineal con un armónico (color rojo). Veamos ahora el gráfico de residuos.

```
acf(residuals(model1))
```

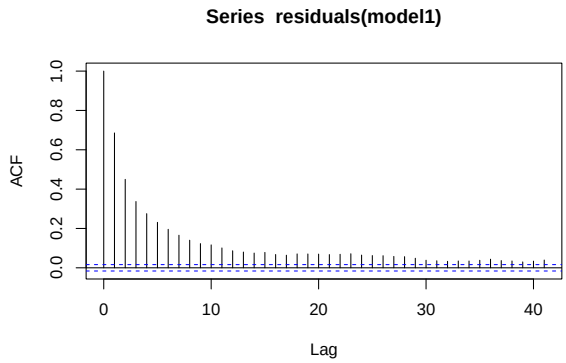

Figura C.2: Correlación de residuos.

A la hora de validar el modelo, la correlación serial en los residuos es muy alta, próxima a 0.7 para la correlación serial de orden 1. Esto es, estamos violando la hipótesis de independencia entre los residuos y por lo tanto nuestro modelo no es adecuado.

C.2. Modelo autoregresivo. Distribuciones conjugadas

En este apartado se verá como las distribuciones usadas para los parámetros $\beta_o, \beta_1, \beta_2, \beta_3, \sigma^2$, en el caso del análisis de temperaturas máximas en la ciudad de Pamplona, son distribuciones conjugadas respecto a un muestreo exponencial. Recordar que teníamos un modelo

$$Y_t = \underbrace{\mu_t + \rho(Y_{t-1} - \mu_{t-1})}_{M_t} + \varepsilon_t \quad t = 2, \ldots, T \tag{C.1}$$

Notar que $Y_t \sim N\left(\mu_t + \rho\left(Y_{t-1} - \mu_{t-1}\right), \sigma^2\right) = N(M_t, \sigma^2)$ con:

$$\mu_t = \beta_0 + \beta_1 t + \beta_2 \sin(2\pi t/365) + \beta_3 \cos(2\pi t/365) \tag{C.2}$$

Recordar que tomábamos las siguientes distribuciones a priori. El objetivo es encontrar las distribuciones a posteriori totalmente condicionadas para poder aplicar el muestreo de Gibbs.

$$\begin{aligned} \beta_i &\sim N(a_i, b_i^2) = N(0, 10^6), && i = 0, \ldots, 3 \\ \rho &\sim Beta(a_\rho, b_\rho) = Beta(1/2, 1/2) \\ \sigma^2 &\sim \text{InvGamma}(a_\sigma, b_\sigma) = \text{InvGamma}(2, 1) \end{aligned} \tag{C.3}$$

C.2.1. A posteriori totalmente condicionada para los β_j con $j = 0, 1, 2, 3$

Consideramos β_j con $j = 0, 1, 2, 3$. Sea $\beta_j x_{t,j}$, p.ej., si $j = 0$ entonces $x_{t,0} = 1$ por ser el intercepto. Denotamos por μ_t^* a μ_t sin la parte correspondiente a β_j. Por el Teorema 1.1.2 sabemos que la distribución a posteriori totalmente condicionada para σ^2 es proporcional a:

$$\begin{aligned} p(\beta_j | \cdots) &\propto \prod_{t=2}^{T} N(Y_t | \mu_t + \rho(Y_{t-1} - \mu_{t-1}), \sigma^2) \times N(\beta_j | a_j, b_j) \\ &\propto N(\beta_j | \mu_{\beta_j}, \sigma^2_{\beta_j}) \times N(\beta_j | a_j, b_j) \end{aligned}$$

Veamos que la verosimilitud se distribuye como una normal y con que parámetros lo hace. A lo largo de las cuentas vamos quitando las partes que no dependen de β_j, que es la variable de interés.

$$\prod_{t=2}^{T} N(Y_t|\mu_t+\rho(Y_{t-1}-\mu_{t-1}),\sigma^2) = \prod_{t=2}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(Y_t-\mu_t-\rho(Y_{t-1}-\mu_{t-1}))^2}{2\sigma^2}\right]$$

$$\propto \exp\left[-\sum_{t=2}^{T} \frac{\mu_t^2+\rho^2\mu_{t-1}^2-2Y_t\mu_t+2\rho Y_t\mu_{t-1}+2\rho Y_{t-1}\mu_t-2\rho\mu_t\mu_{t-1}-2\rho^2 Y_{t-1}\mu_{t-1}}{2\sigma^2}\right]$$

$$\propto \exp\left[-\sum_{t=2}^{T} \frac{(\beta_j x_{t,j}+\mu_t^*)^2+\rho^2(\beta_j x_{t-1,j}+\mu_{t-1}^*)^2-2Y_t(\beta_j x_{t,j}+\mu_t^*)+2\rho Y_t(\beta_j x_{t-1,j}+\mu_{t-1}^*)+}{2\sigma^2}\right.$$

$$\left.\frac{+2\rho Y_{t-1}(\beta_j x_{t,j}+\mu_t^*)-2\rho(\beta_j x_{t,j}+\mu_t^*)(\beta_j x_{t-1,j}+\mu_{t-1}^*)-2\rho^2 Y_{t-1}(\beta_j x_{t-1,j}+\mu_{t-1}^*)}{2\sigma^2}\right]$$

$$\propto \exp\left[-\sum_{t=2}^{T} \frac{\beta_j^2 x_{t,j}^2+2\mu_t^*\beta_j x_{t,j}+\rho^2\beta_j^2 x_{t-1,j}^2+2\rho^2\beta_j x_{t-1,j}\mu_{t-1}^*-2Y_t\beta_j x_{t,j}+2\rho Y_t\beta_j x_{t-1,j}}{2\sigma^2}\right.$$

$$\left.\frac{+2\rho Y_{t-1}\beta_j x_{t,j}-2\rho\beta_j x_{t,j}\beta_j x_{t-1,j}-2\rho\beta_j x_{t,j}\mu_{t-1}^*-2\rho\beta_j x_{t-1,j}\mu_t^*-2\rho^2 Y_{t-1}\beta_j x_{t-1,j}}{2\sigma^2}\right]$$

$$\propto \exp\left[-\sum_{t=2}^{T} \frac{\beta_j^2\left(x_{t,j}^2+\rho^2 x_{t-1,j}^2-2\rho x_{t,j}x_{t-1,j}\right)-2\beta_j(x_{t,j}-\rho x_{t-1,j})(Y_t-\mu_t^*-\rho(Y_{t-1}-\mu_{t-1}^*))}{2\sigma^2}\right]$$

$$\propto \exp\left[-\frac{1}{2}\sum_{t=2}^{T} \frac{\beta_j^2(x_{t,j}-\rho x_{t-1,j})^2-2\beta_j(x_{t,j}-\rho x_{t-1,j})(Y_t-\mu_t^*-\rho(Y_{t-1}-\mu_{t-1}^*))}{\sigma^2}\right]$$

$$\propto \exp\left[-\frac{\sum_{t=2}^{T}(x_{t,j}-\rho x_{t-1,j})^2}{2\sigma^2}\left(\beta_j-\frac{\sum_{t=2}^{T}(x_{t,j}-\rho x_{t-1,j})(Y_t-\mu_t^*-\rho(Y_{t-1}-\mu_{t-1}^*))}{\sum_{t=2}^{T}(x_{t,j}-\rho x_{t-1,j})^2}\right)^2\right]$$

$$\propto N(\mu_{\beta_j},\sigma_{\beta_j}^2)$$

con

$$\mu_{\beta_j}=\frac{\sum_{t=2}^{T}(x_{t,j}-\rho x_{t-1,j})(Y_t-\mu_t^*-\rho(Y_{t-1}-\mu_{t-1}^*))}{\sum_{t=2}^{T}(x_{t,j}-\rho x_{t-1,j})^2}, \quad \sigma_{\beta_j}^2=\frac{\sigma^2}{\sum_{t=2}^{T}(x_{t,j}-\rho x_{t-1,j})^2}.$$

Luego tenemos que la a posteriori para β_j es un producto de normales

$$p(\beta_j|\cdots)\propto N(\beta_j|\mu_{\beta_j},\sigma_{\beta_j}^2)\times N(\beta_j|a_j,b_j)$$

y en conclusión

$$p(\beta_j|\cdots)\sim N\left(\frac{\frac{\mu_{\beta_j}}{\sigma_{\beta_j}^2}+\frac{a_j}{b_j^2}}{\frac{1}{\sigma_{\beta_j}^2}+\frac{1}{b_j^2}}, \frac{1}{\frac{1}{\sigma_{\beta_j}^2}+\frac{1}{b_j^2}}\right).$$

C.2.2. A posteriori totalmente condicionada para σ^2

Sea $\sigma^2\sim \text{InvGamma}(a_\sigma,b_\sigma)$. Por el Teorema 1.1.2 sabemos que la distribución totalmente condicionada a posteriori para σ^2 es proporcional a:

$$p\left(\sigma^2\mid\cdots\right)\propto\prod_{t=2}^{T} N(Y_t|\mu_t+\rho(Y_{t-1}-\mu_{t-1}),\sigma^2)\times \text{Invgamma}(\sigma^2|a_\rho,b_\rho).$$

Hagamos las cuentas para ver que sale otra inversa gamma y por lo tanto que es conjugada.

$$\begin{aligned}
p(\sigma^2 \mid \mathbf{y}, \beta_0, \beta_1, \beta_2, \beta_3, \rho) &\propto \prod_{t=2}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_t - M_t)^2}{2\sigma^2}\right) \times \frac{b_\sigma^{a_\sigma}}{\Gamma(a_\sigma)} (\sigma^2)^{-(a_\sigma+1)} \exp\left(-\frac{b_\sigma}{\sigma^2}\right) \\
&\propto \prod_{t=2}^{T} (\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{(y_t - M_t)^2}{2\sigma^2}\right) \times (\sigma^2)^{-(a_\sigma+1)} \exp\left(-\frac{b_\sigma}{\sigma^2}\right) \\
&= \sigma^{2^{-\frac{T-1}{2}}} \exp\left(-\frac{\sum_{t=2}^{T} (y_t - M_t)^2}{2\sigma^2}\right) \times (\sigma^2)^{-(a_\sigma+1)} \exp\left(-\frac{b_\sigma}{\sigma^2}\right) \\
&= (\sigma^2)^{-\left(a_\sigma + \frac{T-1}{2} + 1\right)} \exp\left[-\left(\frac{b_\sigma}{\sigma^2} + \frac{\sum_{t=2}^{T} (y_t - M_t)^2}{2\sigma^2}\right)\right] \\
&= (\sigma^2)^{-\left(a_\sigma + \frac{T-1}{2} + 1\right)} \exp\left[-\left(\frac{2b_\sigma + 2\left(\frac{\sum_{t=2}^{T} (y_t - M_t)^2}{2}\right)}{2\sigma^2}\right)\right] \\
&= (\sigma^2)^{-\left(a_\sigma + \frac{T-1}{2} + 1\right)} \exp\left[-\left(\frac{b_\sigma + \frac{\sum_{t=2}^{T} (y_t - M_t)^2}{2}}{\sigma^2}\right)\right] \\
&= InvGamma(a_\sigma + \frac{T-1}{2}, b_\sigma + \frac{\sum_{t=2}^{T} (y_t - M_t)^2}{2})
\end{aligned} \tag{C.4}$$

C.2.3. A posteriori totalmente condicionada para ρ

En este caso la a posteriori totalmente condicionada con una a priori $\rho \sim Beta(a_\rho, b_\rho)$ respecto a un muestreo normal, no es una distribución conjugada. Es por ello que para actualizar este parámetro en nuestra cadena de Markov usaremos un paseo aleatorio de M-H con distribución objetivo la totalmente condicionada, que por el Teorema 1.1.2 es proporcional a:

$$p(\rho|\cdots) \propto \prod_{t=2}^{T} N(Y_t|\mu_t + \rho(Y_{t-1} - \mu_{t-1}), \sigma^2) \times Beta(\rho|a_\rho, b_\rho) \tag{C.5}$$

Esto no es proporcional a ninguna densidad conocida en general, por lo que aplicaremos el algoritmo de Metropolis. Tener en cuenta que el soporte de esta densidad está en $(0,1)$ por lo que vamos a muestrear de una transformación de ρ en la recta real, e.g., con una transformación logit.

$$Z_\rho \equiv \text{logit}(\rho) = \log\left(\frac{\rho}{1-\rho}\right),$$

cuya inversa es

$$\rho = \frac{1}{1+\exp(-Z_\rho)}. \tag{C.6}$$

Entonces se quiere simular Z_ρ y la distribución propuesta debe ser una densidad normal centrada en el valor del parámetro Z_ρ en la iteración anterior y con varianza tal que la proporción de aceptación sea alrededor del 15 o 40 %. La distribución objetivo vendrá dada, salvo constante, por:

$$p(Z_\rho|\cdots) \propto \prod_{t=2}^{T} N\left(Y_t|\mu_t + \frac{1}{1+\exp(-Z_\rho)}(Y_{t-1} - \mu_{t-1}), \sigma^2\right) \times Beta\left(\frac{1}{1+\exp(-Z_\rho)}|a_\rho, b_\rho\right)$$

C.2.4. Código de Metropolis-within-Gibbs

```
#cargamos datos
y<-readRDS("/TFG/R/Aplicacion_Temperaturas/pamplona19762015.rds")
library(latex2exp)
nsim<-10000
T<-length(y)
#Inicializamos vectores de parametros
  beta0<-rep(NA,nsim)
  beta1<-rep(NA,nsim)
  beta2<-rep(NA,nsim)
  beta3<-rep(NA,nsim)
  rho<-rep(NA,nsim)
  sigma2<-rep(NA,nsim)

  t<-1:T #numero de observaciones que tenemos de la temperatura
  uno<-rep(1,T)
  seno<-sin(2*pi*t/365)
  cose<-cos(2*pi*t/365)

# Parámetros de las distribuciones a priori
  a_0<-0
  b2_0<-10^6
  a_1<-0
  b_1<- 10^6
  a_2<-0
  b2_2<-10^6
  a_3<-0
  b2_3<- 10^6
  a_rho<-1/2
  b_rho<-1/2
  a_sigma<-2
  b_sigma<-1

#Creamos una funcion que nos haga el random walk de metropolis
  #la dist. a posteriori para rho es proporcional a esta expresión
  posteriori<-function(y,zp,sigma2,mut,a_rhoo,b_rhoo)
  { a<-dnorm(x=y[-1],mean = mut[-1]+(y[-T]-mut[-T])/(1+exp(-zp)),
        sd=(sqrt(sigma2)),log = TRUE)
    b<-dbeta(x=1/(1+exp(-zp)),shape1 =a_rhoo,shape2=b_rhoo,log = TRUE)
    return(sum(a)+b) }

  metropolis_rho<-function(y,T,beta0,beta1,beta2,beta3,sigma2,rho,a_rhoo,b_rhoo )
    {
      #cambiologit
        zp1<-log(rho/(1-rho))
        mut<-beta0*uno+beta1*t+beta2*seno+beta3*cose
        #random walk
        viejo<-zp1
        propuesto<-rnorm(1,zp1,0.1)
        #ver si se acepta el valor
        probaceptar<-exp(posteriori(y,propuesto,sigma2,mut,a_rhoo,b_rhoo)-
                posteriori(y,viejo,sigma2,mut,a_rhoo,b_rhoo))
        if(probaceptar > runif(1))
          {1/(1+exp(-propuesto))}else{rho}    }
```

```r
 #Funcion que nos calcula la a posteriori totalmente condicionada para las betas
      betas <- function(x, mut, rho, sigma2) {
        num<-sum((x[-1]-rho*x[-T])*(y[-1]-mut[-1]-rho*(y[-T]-mut[-T])))
        den<-sum((x[-1]-rho*x[-T])^2)
        muMu<- num/den
        sigmaMu <-sigma2 / den
        varianza <-1/(1/sigmaMu + 10^-6)
        media <-muMu /sigmaMu * varianza
      return(rnorm(1, media, sqrt(varianza)))}
```

```r
#METROPOLIS-WITHIN-GIBBS

      aceptados<-0
metropolis_within_gibbs<-function(beta0,beta1,beta2,beta3,sigma2,rho)
{
  beta0[1]<-beta0
  beta1[1]<-beta1
  beta2[1]<-beta2
  beta3[1]<-beta3
  sigma2[1]<-sigma2
  rho[1]<-rho
  for (i in 2:nsim) {
  #actualizamos  los BETA en orden

  beta0[i]<-betas(uno ,beta1[i-1]*t+beta2[i-1]*seno+beta3[i-1]*cose, rho[i-1],
                  sigma2[i-1])
  beta1[i]<-betas(t   , beta0[i]+beta2[i-1]*seno+beta3[i-1]*cose, rho[i-1],
                  sigma2[i-1])
  beta2[i]<-betas(seno, beta0[i]+beta1[i  ]*t+ beta3[i-1]*cose, rho[i-1],
                  sigma2[i-1])
  beta3[i]<-betas(cose, beta0[i]+beta1[i  ]*t+beta2[i  ]*seno , rho[i-1],
                  sigma2[i-1])

#actualizamos SIGMA2
    mut<-beta0[i]+beta1[i]*t+beta2[i]*seno+beta3[i]*cose
    Mt<-mut[-1]+rho[i-1]*(y[-T]-mut[-T])
    b<-2+sum((y[-1]-Mt)^2)/2
    a<-2+(T-1)/2
    sigma2[i]<-1/rgamma(1,shape =a ,rate = b)


  #finalmente actualizamos RHO con el M- random walk
  rho[i]<-metropolis_rho(y,T,beta0[i],beta1[i],beta2[i],beta3[i],sigma2[i],
           rho[i-1],a_rho,b_rho )

}
  df <- data.frame(beta0,beta1,beta2,beta3,sigma2,rho)
  colnames(df) <- c("beta0","beta1","beta2","beta3","sigma2","rho")
  return(df)
}
```

Ahora simulamos dos cadenas de 10000 observaciones cada una para cada parámetro, y posteriormente descartamos las 5000 primeras iteraciones en concepto de burn-in. Los valores iniciales para cada cadena de $(\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2, \rho)$ son $cadena1 = (0,0,0,0,1,0.5)$ y $cadena2 = (-4,5,6,7,4,0.8)$, dispersos entre si (con la σ^2 siempre positiva y $\rho \in (0,1)$).

```
# cadena1 con valores iniciales para beta0,beta1,beta2,beta3,sigma2,rho
cadena1<-metropolis_within_gibbs(0,0,0,0,1,0.5)
length(unique(cadena1$rho))/10000*100  #porcentaje de  aceptados para rho

# cadena2 con  valores iniciales para beta0,beta1,beta2,beta3,sigma2,rho
cadena2<-metropolis_within_gibbs(-4,5,6,7,4,0.8)
length(unique(cadena2$rho))/10000*100

#eliminamos los burn-in, la mitad, 5.000
cadena1efectiva<-cadena1[c(5001:10000), ]
cadena2efectiva<-cadena2[c(5001:10000), ]
```

Una vez eliminadas las iteraciones de calentamiento en cada cadena, dibujamos para cada parámetro las dos cadenas resultantes para ver que convergen a la misma distribución.

```
#trace plots para cada parámetro
plot(cadena1efectiva$beta0,col='blue',main=TeX(r'({Gráfico de traza para las dos
    cadenas de}$\beta_0$)'),type='l' ,ylab = TeX(r'( $\beta_0$)'))
lines(cadena2efectiva$beta0,col='red',type='l')

plot(cadena1efectiva$beta1,col='blue',main=TeX(r'({Gráfico de traza para las dos
    cadenas de }$\beta_1$)'),type='l' ,ylab = TeX(r'( $\beta_1$)'))
lines(cadena2efectiva$beta1,col='red',type='l')

plot(cadena1efectiva$beta2,col='blue',main=TeX(r'({Gráfico de traza para las dos
    cadenas de }$\beta_2$)'),type='l' ,ylab = TeX(r'( $\beta_2$)'))
lines(cadena2efectiva$beta2,col='red',type='l')

plot(cadena1efectiva$beta3,col='blue',main=TeX(r'({Gráfico de traza para las dos
    cadenas de }$\beta_3$)'),type='l' ,ylab = TeX(r'( $\beta_3$)'))
lines(cadena2efectiva$beta3,col='red',type='l')


plot(cadena1efectiva$sigma2,col='blue',main=TeX(r'({Gráfico de traza para las dos
    cadenas de}$\sigma^2$)'),type='l' ,ylab = TeX(r'( $\sigma^2$)'))
lines(cadena2efectiva$sigma2,col='red',type='l')


plot(cadena1efectiva$rho,col='blue',main=TeX(r'({Gráfico de traza para las dos
    cadenas de }$\rho$)'),type='l' ,ylab = TeX(r'( $\rho$)'))
lines(cadena2efectiva$rho,col='red',type='l')
```

Vemos que para cada parámetro ambas dos cadenas parecen converger a la misma distribución, lo que es un buen indicador. Veamos ahora también los siguientes diagnósticos para cada parámetro:

- *Diagnóstico de Gelman*, del paquete `coda` que nos da el *factor de reducción de escala* para cada parámetro. Este se basa en el *Análisis de la varianza intra e inter cadenas*, visto en la Sección 2.7, y conforme más se acerca a 1 mejor convergencia a la distribución objetivo se da. La entrada de esta orden requiere de un objeto `mcmc.list()`, por lo que juntamos las dos cadenas de 5000 iteraciones que tenemos y lo incluimos como un objeto de dicho tipo.
- *Tamaño de muestra efectivo*, definido como en la Sección2.7 y calculado con la orden `coda::effectiveSize()`.
- Media a posteriori e intervalo de credibilidad al 90 %.

Veamos todos estos datos en la siguiente tabla.

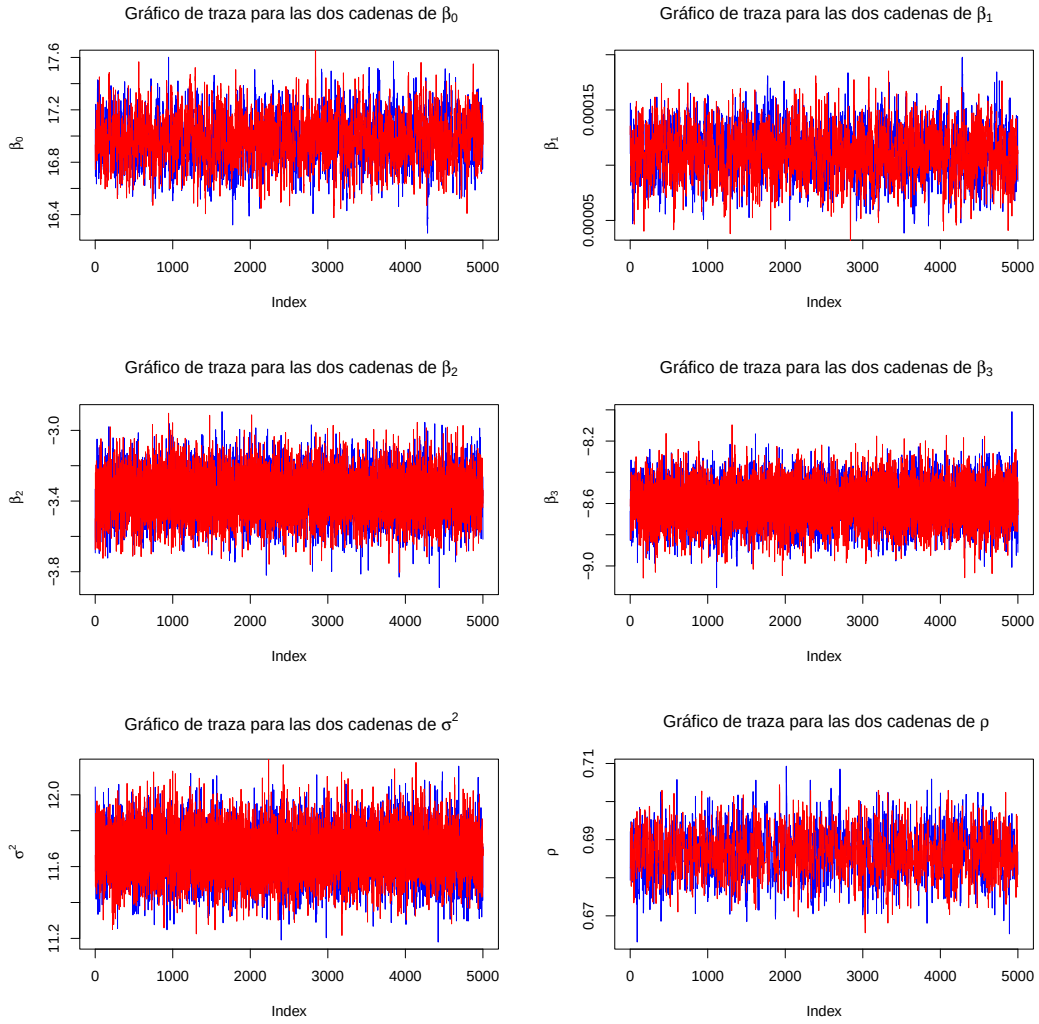

Figura C.3: Gráficos de traza para $\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2, \rho$. En azul se representa la primera cadena y en rojo la segunda.

```
#gelman.diag y tamaño efectivo
cadenasjuntas<-mcmc.list(mcmc(cadena1efectiva),mcmc(cadena2efectiva))
gelman.diag(cadenasjuntas)
effectiveSize(cadenasjuntas)
#medias a posteriori
mean(c(cadena1efectiva$beta0,cadena2efectiva$beta0))
mean(c(cadena1efectiva$beta1,cadena2efectiva$beta1))
mean(c(cadena1efectiva$beta2,cadena2efectiva$beta2))
mean(c(cadena1efectiva$beta3,cadena2efectiva$beta3))
mean(c(cadena1efectiva$sigma2,cadena2efectiva$sigma2))
mean(c(cadena1efectiva$rho,cadena2efectiva$rho))

#ic
quantile(c(cadena1efectiva$beta0,cadena2efectiva$beta0),probs=c(0.05,0.95))
quantile(c(cadena1efectiva$beta1,cadena2efectiva$beta1),probs=c(0.05,0.95))
quantile(c(cadena1efectiva$beta2,cadena2efectiva$beta2),probs=c(0.05,0.95))
quantile(c(cadena1efectiva$beta3,cadena2efectiva$beta3),probs=c(0.05,0.95))
quantile(c(cadena1efectiva$sigma2,cadena2efectiva$sigma2),probs=c(0.05,0.95))
quantile(c(cadena1efectiva$rho,cadena2efectiva$rho),probs=c(0.05,0.95))
```

Bibliografía

[1] ROBERT, C.P. y CASELLA, G. , *Introducing Monte Carlo Methods with R, 1st Edition.* Springer, (2009).

[2] GELMAN, A., CARLIN, J.B., STERN, H., DUNSON, D., VEHTARI, A. y RUBIN, D. *Bayesian Data Analysis, 3rd Edition.* CRC Press (2013).

[3] ROSS, S. M., *Stochastic Processes, 2nd Edition.* Wiley, (1995). pp, 186-187.

[4] DASTON,L., *Classical Probability in the Enlightenment.* Princeton Univ Press, (1988).

[5] HARTIGAN, J. A., *pringer Series in Statistics: Bayes Theory.* Springer, (1983).

[6] MITCHELL ,T.M. , *Machine Learning.* McGraw-Hil, (1997). Chap.6. Bayesian Learning.

[7] FLORESCU, I. , *Probability and Stochastic Processes.* Wiley, (2015). Chap. 3. Generating Random Variables.

[8] FERNÁNDEZ-CASAL R. y CAO R., *Simulación Estadística.* Github, (2022). URL: https://rubenfcasal.github.io/simbook/

[9] JIMÉNEZ, J., *Métodos Monte Carlo basados en cadenas de Markov.* Trabajo Fin de Grado, Universidad de Sevilla (2015). URL: https://idus.us.es/bitstream/handle/11441/40818/Jiménez

[10] AMARAL, M.A. , PAULINO, C.D.M. y MÜLLER, P., *Computational Bayesian Statistics: An Introduction.* Cambridge (2019).

[11] *Markov Chain Monte Carlo Method and Its Application. Journal of the Royal Statistical Society. Series D (The Statistician). Vol. 47, No. 1 (1998), pp. 69-100.*

[12] GRIMMETT, G. y STIRZAKER, D., *Probability and Random Processes.* The American Statistician. Oxford University Press, (2001).

[13] CASELLA, G. y GEORGE, E. I., *Explaining the Gibbs Sampler.* The American Statistician, Vol. 46, No. 3. (Aug., 1992), pp. 167-174.

[14] EICHLER, M. *Markov Cahin Monte Carlo.* Statistics 24600 (2004). URL: http://galton.uchicago.edu/~eichler/stat24600/

[15] GELMAN, A. y RUBIN, D.B. *Inference from Iterative Simulation Using Multiple Sequences.* (November 1992) Statist. Sci. 7 (4) 457 - 472.

[16] HASTINGS, W.K. , *Monte Carlo Sampling Methods Using Markov Chains and Their Applications (1970).*

[17] METROPOLIS, N. , *The beginning of the Monte Carlo Method. Los Alamos Science Special Issue, Vol. 15.(1987)*, 125-130.

[18] ROBERT,C. y CASELLA,G., *A History of Markov Chain Monte Carlo–Subjective Recollections from Incomplete Data.* (2011).

[19] GELMAN, A., ROBERTS, G.O., y GILKS, W.R. , *Efficient Metropolis jumping rules*. In: Bernardo JM, Berger JO, Dawid AP, Smith AFM (eds) Bayesian Statistics 5, Oxford University Press, (1996). pp 599–607

[20] GELFAND, A.E. y SMITH, A.F.M. *Sampling-based approaches to calculating marginal densities*. Journal of the American Statistical Association, (1990). pp. 398-409

[21] GEMAN, S. y GEMAN, D. *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images,* (1984). IEEE Trans. Pattern Anal. Mach. Intell. 6 721–741

[22] WICKHAM, H. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics* (2016). URL: https://cran.r-project.org/web/packages/ggplot2/

[23] PLUMMER, M. *CODA: Output Analysis and Diagnostics for MCMC* (2020) URL: https://cran.r-project.org/web/packages/coda/