Métodos de Muestreo MCMC

Matias Vera

1. Estadística Bayesiana

Para comenzar, introduzcamos los ingredientes e hipótesis habituales de la inferencia bayesiana[1]:

- Sea T una variable aleatoria representativa de los parámetros y las variables no observables del modelo, con distribución a priori $p_T(\theta)$.
- La estadística bayesiana supone una relación causal $T \to \mathbf{X}$, donde $\mathbf{X}$ es cualquier conjunto aleatorio de muestras a observar.
- La relación anterior implica la independencia entre las muestras *cuando se conoce el parámetro*. Es decir que la verosimilitud de una muestra puede escribirse como $p_{\mathbf{X}|T=\theta}(\mathbf{x}) = \prod_{i=1}^{n} p_{X|T=\theta}(x_i)$. No se pierde generalidad en asumir que las variables son idénticamente distribuidas (podría haber escrito $p_{X_i|T=\theta}(x_i)$ en su lugar, pero no vale la pena).
- La *distribución a posteriori*, el corazón de la estadística bayesiana, es entonces proporcional al producto de la *prior* y la verosimilitud

$$p_{T|\mathbf{X}=\mathbf{x}}(\theta) \propto p_T(\theta) \cdot \prod_{i=1}^{n} p_{X|T=\theta}(x_i) \tag{1}$$

- Por último, se define la distribución predictiva bayesiana como:

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) = \int_{\Theta} p_{X|T=\theta}(x_{\text{test}}) p_{T|\mathbf{X}=\mathbf{x}}(\theta) d\theta = \mathbb{E}\left[p(x_{\text{test}}|T)|\mathbf{X}=\mathbf{x}\right] \tag{2}$$

  donde X_{test} es una variable aleatoria no vista en el conjunto de entrenamiento $\mathbf{X}$.

Para fijar ideas, se recomienda analizar el siguiente ejemplo.

1.1. Ejemplo de Inferencia Bayesiana

[1]Para un repaso más exhaustivo, puede verse el video `https://youtu.be/f9wqDjmR848?si=a6JUG_H3UhuHcyLz`

Ejemplo 1. *El tiempo de vida (en años) de un transistor es una variable aleatoria con distribución exponencial de parámetro θ. A priori se modela θ como una variable aleatoria con distribución $\Gamma(2,3)$. Si en 20 transistores se observó una duración total $\sum_{i=1}^{20} x_i = 7$.*

1. *Hallar la distribución a posteriori del parámetro θ.*
2. *Hallar la distribución predictiva del tiempo de vida de un transistor.*

Como primer paso en un problema bayesiano, hay que comenzar planteando la distribución *a posteriori*. En este caso evitaremos las constantes de proporcionalidad:

$$p_{T|\mathbf{X}=\mathbf{x}}(\theta) \propto p_T(\theta) \cdot \prod_{i=1}^{n} p_{X|T=\theta}(x_i) \propto \theta e^{-3\theta} \mathbf{1}\{\theta > 0\} \cdot \prod_{i=1}^{20} \theta e^{-\theta x_i} = \theta^{21} e^{-10\theta} \mathbf{1}\{\theta > 0\} \quad (3)$$

Es decir, la variable se distribuye *a posteriori* como $T|_{\mathbf{X}=\mathbf{x}} \sim \Gamma(22, 10)$. La distribución predictiva es de la forma

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) = \int_{\Theta} p_{X|T=\theta}(x_{\text{test}}) p_{T|\mathbf{X}=\mathbf{x}}(\theta) d\theta \propto \int_0^{\infty} \theta e^{-\theta x_{\text{test}}} \mathbf{1}\{x_{\text{test}} > 0\} \cdot \theta^{21} e^{-10\theta} d\theta \quad (4)$$

Reconociendo el núcleo de la integral, se puede observar que el mismo es proporcional a la densidad de una $\Gamma(\nu, \lambda)$, es decir $p(x) = \frac{\lambda^\nu}{\Gamma(\nu)} x^{\nu-1} e^{-\lambda x} \mathbf{1}\{x > 0\}$. Sabiendo que por ser densidad debe integrar 1:

$$p_{X_{\text{test}}|\mathbf{X}=\mathbf{x}}(x_{\text{test}}) \propto \int_0^{\infty} \theta^{22} e^{-\theta(10+x_{\text{test}})} d\theta \cdot \mathbf{1}\{x_{\text{test}} > 0\} \propto \frac{1}{(10 + x_{\text{test}})^{23}} \mathbf{1}\{x_{\text{test}} > 0\} \quad (5)$$

donde se utilizó $\nu = 23$ y $\lambda = 10 + x_{\text{test}}$. Es decir que $X_{\text{test}}|_{\mathbf{X}=\mathbf{x}} \sim \text{Lomax}(22, 10)$[2].

1.2. Estadísticos Suficientes en Inferencia Bayesiana

Un concepto muy útil a la hora de efectuar inferencia es el de **estadístico suficiente**. Un estadístico $S(\mathbf{X})$ se denomina suficiente para θ si la distribución de $\mathbf{X}|_{S(\mathbf{X})=s}$ no depende de θ. Es decir que toda la información que posee la muestra sobre θ se encuentra en el estadístico. Además, el teorema de Neyman-Fisher nos permite encontrar estadísticos suficientes de forma muy sencilla, encontrando un $S(\mathbf{X})$ que permita descomponer la verosimilitud como:

$$p_{\mathbf{X}|T=\theta}(\mathbf{x}) = g(\theta, S(\mathbf{x})) \cdot h(\mathbf{x}) \quad (6)$$

En términos bayesianos un estadístico suficiente se interpreta como una independencia condicional $\mathbf{X} \perp \theta|_{S(\mathbf{X})=s}$ (es decir que la muestra y los parámetros son independientes cuando se conoce el estadístico suficiente). Este resultado implica que la distribución *a posteriori*

[2] La Lomax(α, β) posee densidad $p(x) = \frac{\alpha\beta^\alpha}{(x+\beta)^{\alpha+1}} \mathbf{1}\{x > 0\}$.

debe cumplir $p_{T|\mathbf{X}=\mathbf{x}}(\theta) = p_{T|S(\mathbf{X})=s(\mathbf{x})}(\theta)$, y por lo tanto nos permite intercambiar el conocimiento de toda la muestra por el del estadístico suficiente. En el ejemplo anterior la distribución solo dependía de la muestra a través de la suma, estadístico suficiente para θ en una distribución exponencial.

Esto nos permite hacer equivalencias sobre los datos de las variables observadas. En nuestro ejemplo, es equivalente pensar que se cuenta con 20 muestras $\exp(\theta)$ que con una sola muestra $\Gamma(20, \theta)$[3]. Otro ejemplo clásico donde se da este fenómeno es en las variables Bernoulli, donde también la suma es estadístico suficiente: es equivalente tener n muestras $\mathrm{Ber}(p)$ que una muestra $\mathrm{Bin}(n, p)$.

2. Muestreo Monte Carlo

Efectuar el cálculo con distribuciones predictivas, evaluando la integral directamente, puede ser computacionalmente inviable ya que rara vez se cuenta con una forma cerrada o conocida para la distribución a posteriori. Para resolver este problema, se plantea una solución Monte Carlo: combinar métodos de muestreo con la ley de los grandes números.

$$\mathbb{E}\left[p(x_{\text{test}}|T)|\mathbf{X} = \mathbf{x}\right] \approx \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} p_{X|T=\theta_t}(x_{\text{test}}) \tag{7}$$

donde $\theta_1, \cdots, \theta_{t_{\max}}$ son muestras generadas a partir de la distribución *a posteriori*. El problema con este método es que difícilmente podamos generar muestras independientes. Es entonces cuando surge el Muestreo Monte Carlo por Cadenas de Markov (MCMC) como una alternativa.

2.1. Muestreo Monte Carlo por Cadenas de Markov (MCMC)

Una **cadena de Markov** es una secuencia de variables aleatorias $\{\theta_t\}_{t\in\mathbb{N}}$ que cumple la propiedad de *falta de memoria*: la distribución del próximo estado depende únicamente del estado actual, y no del pasado completo. Formalmente,

$$p(\theta_{t+1}|\theta_t, \theta_{t-1} \cdots, \theta_0) = p(\theta_{t+1}|\theta_t) = P(\theta_t \to \theta_{t+1}) \tag{8}$$

donde $P(\theta_t \to \theta_{t+1})$ es la densidad de transición de θ_t a θ_{t+1} (en la estadística bayesiana todas estas distribuciones serán computadas implícitamente *a posteriori* de observar los datos de entrenamiento). Cuando las probabilidades de transición no dependen del tiempo t, se dice que la cadena es **homogénea**. Una distribución π se dice *estacionaria* para una cadena de Markov si no varía su estadística al propagarse por la cadena;

$$\pi(\theta') = \int \pi(\theta) P(\theta \to \theta')\, d\theta \tag{9}$$

es decir, que la distribución de θ y θ' sea la misma (que la distribución de θ no varíe al efectuar un paso en la cadena). Una condición suficiente para asegurar esto es que la conjunta

[3]La suma de 20 variables $\exp(\theta)$ independientes e idénticamente distribuidas se distribuye como una $\Gamma(20, \theta)$

de ambas sea simétrica:

$$\pi(\theta)P(\theta \to \theta') = \pi(\theta')P(\theta' \to \theta) \tag{10}$$

la cual se cumple trivialmente si $\theta = \theta'$ (repetir el estado actual no afecta la condición (10) correspondiente al estado estacionario).

La idea general del MCMC es construir una cadena de Markov homogénea $\{\theta_t\}_{t\in\mathbb{N}}$ cuya distribución estacionaria sea la posteriori. Bajo ciertas condiciones, el **teorema de ergodicidad** garantiza que el promedio de los valores generados por la cadena converge a la esperanza, dando por válida (7) aunque no se trate de muestras independientes. Para una cadena de Markov, dichas condiciones se pueden resumir en:

- Irreducible: La transición $\theta \to \theta'$ debe poder ser alcanzada en una cantidad finita de pasos para todo θ y θ'.
- Áperiódica: La transición $\theta \to \theta$ (mantener el estado actual) debe tener probabilidad positiva.
- Recurrente positiva: El tiempo esperado para volver al estado actual es finito.

Los algoritmos utilizados para los modelos bayesianos fueron construidos para cumplir con estas condiciones.

2.2. Ejemplo de cadena de Markov

Ejemplo 2. *Encontrar el estado estacionario de una cadena de Markov homogénea, donde θ es una variable discreta que toma valores en $\{0, 1, 2\}$ y las probabilidades de transición se definen con la matriz* $P = \begin{pmatrix} 0{,}7 & 0{,}2 & 0{,}1 \\ 0{,}4 & 0{,}6 & 0{,}0 \\ 0{,}0 & 0{,}9 & 0{,}1 \end{pmatrix}$.

No es difícil probar que esta cadena es irreducible, aperiódica y recurrente positiva. Toda cadena de Markov irreducible en un espacio de estados finitos tiene una distribución estacionaria única. Se desea encontrar $\pi(0)$, $\pi(1)$ y $\pi(2)$ (representadas por un vector π), tal que se cumpla (9). Para variables discretas y finitas eso se puede representar como $\pi = P \cdot \pi$ con $\mathbf{1}^T \cdot \pi = 1$, donde $\mathbf{1}$ es un vector con todas sus entradas en 1. Escribiendo todo eso como un sistema de ecuaciones se obtiene

$$\begin{pmatrix} 0{,}7 & 0{,}2 & 0{,}1 \\ 0{,}4 & 0{,}6 & 0 \\ 0 & 0{,}9 & 0{,}1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \pi(0) \\ \pi(1) \\ \pi(2) \end{pmatrix} = \begin{pmatrix} \pi(0) \\ \pi(1) \\ \pi(2) \\ 1 \end{pmatrix} \tag{11}$$

La segunda ecuación $0{,}4\pi(0) + 0{,}6\pi(1) = \pi(1)$ implica que $\pi(0) = \pi(1)$. De la tercera $0{,}9\pi(1) + 0{,}1\pi(2) = \pi(2)$ se obtiene que $\pi(1) = \pi(2)$. Reemplazando podemos ver que se cumple la primera ecuación $0{,}7\pi(0) + 0{,}2\pi(1) + 0{,}1\pi(2) = \pi(0)$ y, dado que deben sumar 1 (cuarta ecuación), se obtiene $\pi(0) = \pi(1) = \pi(2) = \frac{1}{3}$.

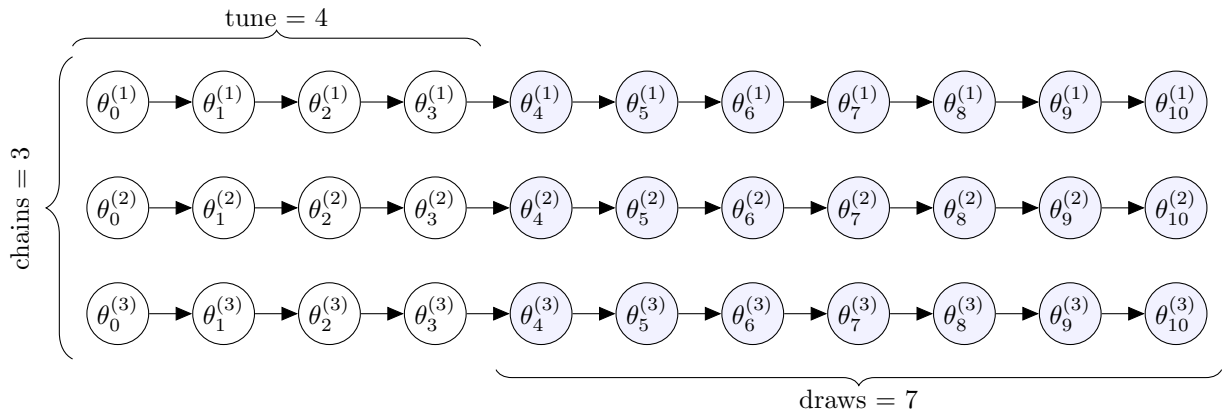

Figura 1: Ejemplo de experimento de muestreo. En este caso se simularon tres cadenas independientes (chains = 3), se esperaron cuatro pasos para considerar que se alcanzó el estado estacionario (tune = 4) y se recolectaron siete muestras efectivas en el presunto estado estacionario (draws = 7).

3. Algoritmos de Muestreo MCMC

El objetivo de los algoritmos de muestreo es generar un proceso cuya secuencia de muestras sea ergódica. Esto no siempre es sencillo, ya que puede no disponerse de toda la información necesaria sobre las distribuciones.

En la Fig. 1 puede verse un ejemplo de experimento de muestreo. Se denomina *tune* a la cantidad de muestras a descartar para considerar que se alcanzó el estado estacionario, y *draws* a la cantidad de muestras efectivas que fueron generadas. Se suelen generar varias cadenas y verificar los resultados en cada una (cantidad definida en *chains*). Una disparidad de resultados en las cadenas evidencia que no se alcanzó el mencionado estado estacionario. Dependiendo del tipo de variable aleatoria a muestrear, conviene usar una u otra estrategia. A continuación se presentarán algunos de los muestreos más habituales.

3.1. Muestreo de Gibbs

Supongamos que, debido a su complejidad, no podemos simular muestras de $\pi(x, y)$, pero que sí es posible generar muestras de las condicionales $\pi(x|y)$ y $\pi(y|x)$ (conocidas y fáciles de muestrear). El muestreo de Gibbs consiste en, a partir de un x_0, iterar alternadamente entre $y_t \sim \pi(y|x_t)$ y $x_{t+1} \sim \pi(x|y_t)$. Luego de suficientes pasos, al alcanzar el estado estacionario, los pares (x, y) estarán distribuidos por $\pi(x, y)$. En la siguiente sección se demostrará por qué este proceso cumple (10).

Esta técnica se extiende a más dimensiones muestreando de una componente a la vez. El problema con este muestreo es que es necesario conocer perfectamente todas las distribuciones condicionales, lo cual en la práctica suele ser problemático en muchos casos. Una excepción importante la constituyen las variables Bernoulli, donde se pueden computar todas las probabilidades necesarias.

3.1.1. Demostración del estado estacionario

Notar que, en esta cadena de Markov donde $\theta = (x, y)$, el proceso evoluciona como

$$(x_0, y_0) \to (x_1, y_0) \to (x_1, y_1) \to (x_2, y_1) \to (x_2, y_2) \to (x_3, y_2) \to (x_3, y_3) \to \cdots \tag{12}$$

Para corroborar que cumple la condición suficiente de estacionariedad (10) tomemos un paso de Gibbs $(x_t, y_t) \to (x_{t+1}, y_t)$:

$$\pi(x_t, y_t)\pi(x_{t+1}|y_t) = \pi(x_{t+1}, y_t)\pi(x_t|y_t) \tag{13}$$

donde todas las distribuciones son computadas *a posteriori* (la notación queda implícita). Es fácil notar que, si la medida de probabilidad es la misma, la identidad (13) representa la misma distribución conjunta de (x_t, y_t, x_{t+1}) en ambos lados de la igualdad.

3.2. Muestreo Metropolis

El muestreo Metropolis es usado típicamente en variables aleatorias discretas no binarias (como Poisson, geométrica, hipergeométrica, etc.) que toman valores en los enteros $\mathbb{Z}$, así como también en variables continuas donde no hay diferenciabilidad debido al modelo. Su característica esencial es que solamente le basta con conocer la distribución (conjunta, de todos los parámetros simultáneamente) salvo una constante de proporcionalidad. Es decir que si $\pi(\theta) = \frac{f(\theta)}{Z}$ con $Z > 0$, es suficiente con conocer $f(\theta)$ (que habitualmente se plantea como distribución *a priori* por verosimilitud).

Este tipo de muestreo propone transicionar de θ_t a θ_{t+1} con el siguiente algoritmo simétrico:

1. Se genera una $\theta' = \theta_t + \delta$, donde δ es una variable aleatoria; en el caso que θ' no esté en el soporte de la variable (por ejemplo una Poisson no puede tomar valores negativos), se repite el proceso. Para el caso discreto, típicamente se propone que $\delta \sim \mathcal{U}\{-1, 0, 1\}$ (uniforme discreta de 3 átomos) y para el caso continuo se propone $\delta \sim \mathcal{N}(0, \sigma^2)$.

2. Se sortea una variable aleatoria Bernoulli de probabilidad $\alpha(\theta_t, \theta')$. Si dicha variable vale 1, $\theta_{t+1} = \theta'$. Caso contrario $\theta_{t+1} = \theta_t$.

donde

$$\alpha(\theta_a, \theta_b) = \text{mín}\left\{1, \frac{f(\theta_b)}{f(\theta_a)}\right\} \tag{14}$$

con $\pi(\theta) = \frac{f(\theta)}{Z}$ (no depende de la constante de normalización). A continuación se efectuará un análisis para corroborar si la distribución *a posteriori* cumple la condición de estacionariedad dada por (10).

3.2.1. Demostración del estado estacionario (caso discreto)

La probabilidad de transición del caso discreto vale

$$P(\theta_t \to \theta_{t+1}) = \begin{cases} \frac{\alpha(\theta_t, \theta_t - 1)}{3} & \theta_{t+1} = \theta_t - 1 \\ \frac{\alpha(\theta_t, \theta_t + 1)}{3} & \theta_{t+1} = \theta_t + 1 \\ 1 - \frac{\alpha(\theta_t, \theta_t - 1) + \alpha(\theta_t, \theta_t + 1)}{3} & \theta_{t+1} = \theta_t \\ 0 & \text{Otros} \end{cases} \tag{15}$$

El único caso no trivial que vale la pena analizar en (10) es que ocurre si $\theta_{t+1} = \theta_t \pm 1$; en los demás casos, la condición se cumple automáticamente. En este caso

$$\pi(\theta_t)\frac{\alpha(\theta_t, \theta_{t+1})}{3} = \pi(\theta_{t+1})\frac{\alpha(\theta_{t+1}, \theta_t)}{3} \tag{16}$$

Se puede ver que si $\pi(\theta) = \frac{f(\theta)}{Z}$, la ecuación en cuestión puede reducirse a

$$f(\theta_t)\alpha(\theta_t, \theta_{t+1}) = f(\theta_{t+1})\alpha(\theta_{t+1}, \theta_t) \tag{17}$$

Utilizando el $\alpha(\theta_a, \theta_b)$ definido en (14), se puede comprobar la identidad (17). El primer término cumple que $f(\theta_t)\alpha(\theta_t, \theta_{t+1}) = \text{mín}\{f(\theta_t), f(\theta_{t+1}\}$, y de forma análoga para el otro término $f(\theta_{t+1})\alpha(\theta_{t+1}, \theta_t) = \text{mín}\{f(\theta_{t+1}), f(\theta_t)\}$. De esta manera, queda garantizando que la distribución *a posteriori* es un estado estacionario de la cadena.

3.2.2. Demostración del estado estacionario (caso continuo)

La transición del caso continuo tiene una distribución mixta: una mezcla entre una normal y una masa puntual (delta de Dirac) en $\theta_{t+1} = \theta_t$. Dado que el único caso relevante a chequear de (10) es el caso donde $\theta_{t+1} \neq \theta_t$, basta con analizar la parte continua: $P(\theta_t \to \theta_{t+1}) = \alpha(\theta_t, \theta_{t+1}) \cdot \mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2)$ donde $\mathcal{N}(x|\mu, \sigma^2)$ hace referencia a la densidad de una normal de parámetros μ y σ^2 evaluada en x. En este caso la condición (10) puede escribirse como:

$$\pi(\theta_t) \cdot \alpha(\theta_t, \theta_{t+1}) \cdot \mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2) = \pi(\theta_{t+1}) \cdot \alpha(\theta_{t+1}, \theta_t) \cdot \mathcal{N}(\theta_t|\theta_{t+1}, \sigma^2) \tag{18}$$

La condición anterior puede reducirse a $f(\theta_t)\alpha(\theta_t, \theta_{t+1}) = f(\theta_{t+1})\alpha(\theta_{t+1}, \theta_t)$ usando que la normal es simétrica respecto a la media $\mathcal{N}(\theta_t|\theta_{t+1}, \sigma^2) = \mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2)$. De esta manera llegamos a la misma identidad que en el caso discreto y por lo tanto, podemos concluir que la distribución *a posteriori* cumple la condición de estacionariedad (10).

3.3. NUTS (No-U-Turn Sampler)

El algoritmo NUTS (No-U-Turn Sampler) es un método de muestreo para variables aleatorias continuas con distribución a posteriori es diferenciable. La versión completa del algoritmo puede verse en la Fig. 2; a continuación se presentarán las ideas generales. El algoritmo introduce una variable auxiliar $r \in \mathbb{R}^d$ cuyo objetivo es actuar como dirección de exploración de la distribución *a posteriori*. Esta variable se genera en cada iteración a partir de una distribución normal estándar multivariada. A continuación, se define la función de energía, como:

$$H(\theta, r) = -\log \pi(\theta) + \frac{1}{2}\|r\|^2 \tag{19}$$

Notar que una constante de proporcionalidad en $\pi(\theta)$ se transformaría en una constante sumando y podría omitirse sin afectar el concepto. Esta función de energía combina la log-posterior de θ con una penalización cuadrática sobre r y tiene un papel central en la determinación de la calidad de las propuestas. En cada iteración, partiendo del estado actual (θ_t, r), el algoritmo genera una secuencia de nuevas propuestas (θ, r) mediante un método

Algorithm 6 No-U-Turn Sampler with Dual Averaging

Given θ^0, δ, $\mathcal{L}, M, M^{\text{adapt}}$:
Set $\epsilon_0 = \text{FindReasonableEpsilon}(\theta), \mu = \log(10\epsilon_0), \bar{\epsilon}_0 = 1, \bar{H}_0 = 0, \gamma = 0.05, t_0 = 10, \kappa = 0.75$.
for $m = 1$ to M **do**
  Sample $r^0 \sim \mathcal{N}(0, I)$.
  Resample $u \sim \text{Uniform}([0, \exp\{\mathcal{L}(\theta^{m-1} - \frac{1}{2} r^0 \cdot r^0\}])$
  Initialize $\theta^- = \theta^{m-1}$, $\theta^+ = \theta^{m-1}$, $r^- = r^0, r^+ = r^0, j = 0, \theta^m = \theta^{m-1}, n = 1, s = 1$.
  while $s = 1$ **do**
    Choose a direction $v_j \sim \text{Uniform}(\{-1, 1\})$.
    if $v_j = -1$ **then**
      $\theta^-, r^-, -, -, \theta', n', s', \alpha, n_\alpha \leftarrow \text{BuildTree}(\theta^-, r^-, u, v_j, j, \epsilon_{m-1}\theta^{m-1}, r^0)$.
    else
      $-, -, \theta^+, r^+, \theta', n', s', \alpha, n_\alpha \leftarrow \text{BuildTree}(\theta^+, r^+, u, v_j, j, \epsilon_{m-1}, \theta^{m-1}, r^0)$.
    end if
    if $s' = 1$ **then**
      With probability $\min\{1, \frac{n'}{n}\}$, set $\theta^m \leftarrow \theta'$.
    end if
    $n \leftarrow n + n'$.
    $s \leftarrow s' \mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0] \mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$.
    $j \leftarrow j + 1$.
  end while
  if $m \leq M^{\text{adapt}}$ **then**
    Set $\bar{H}_m = \left(1 - \frac{1}{m + t_0}\right) \bar{H}_{m-1} + \frac{1}{m+t_0}(\delta - \frac{\alpha}{n_\alpha})$.
    Set $\log \epsilon_m = \mu - \frac{\sqrt{m}}{\gamma} \bar{H}_m, \log \bar{\epsilon}_m = m^{-\kappa} \log \epsilon_m + (1 - m^{-\kappa}) \log \bar{\epsilon}_{m-1}$.
  else
    Set $\epsilon_m = \bar{\epsilon}_{M^{\text{adapt}}}$.
  end if
end for

function $\text{BuildTree}(\theta, r, u, v, j, \epsilon, \theta^0, r^0)$
if $j = 0$ **then**
  Base case—take one leapfrog step in the direction v.
  $\theta', r' \leftarrow \text{Leapfrog}(\theta, r, v\epsilon)$.
  $n' \leftarrow \mathbb{I}[u \leq \exp\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\}]$.
  $s' \leftarrow \mathbb{I}[u < \exp\{\Delta_{\max} + \mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\}]$.
  return $\theta', r', \theta', r', \theta', n', s', \min\{1, \exp\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r' - \mathcal{L}(\theta^0) + \frac{1}{2} r^0 \cdot r^0\}\}, 1$.
else
  Recursion—implicitly build the left and right subtrees.
  $\theta^-, r^-, \theta^+, r^+, \theta', n', s', \alpha', n'_\alpha \leftarrow \text{BuildTree}(\theta, r, u, v, j - 1, \epsilon, \theta^0, r^0)$.
  if $s' = 1$ **then**
    if $v = -1$ **then**
      $\theta^-, r^-, -, -, \theta'', n'', s'', \alpha'', n''_\alpha \leftarrow \text{BuildTree}(\theta^-, r^-, u, v, j - 1, \epsilon, \theta^0, r^0)$.
    else
      $-, -, \theta^+, r^+, \theta'', n'', s'', \alpha'', n''_\alpha \leftarrow \text{BuildTree}(\theta^+, r^+, u, v, j - 1, \epsilon, \theta^0, r^0)$.
    end if
    With probability $\frac{n''}{n' + n''}$, set $\theta' \leftarrow \theta''$.
    Set $\alpha' \leftarrow \alpha' + \alpha''$, $n'_\alpha \leftarrow n'_\alpha + n''_\alpha$.
    $s' \leftarrow s'' \mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0] \mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$
    $n' \leftarrow n' + n''$
  end if
  return $\theta^-, r^-, \theta^+, r^+, \theta', n', s', \alpha', n'_\alpha$.
end if

Figura 2: Algoritmo NUTS presentado por Hoffman en "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo" `https://jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf`.

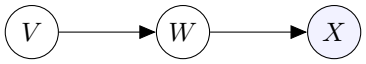

Figura 3: Ejemplo de modelado de 3 variables $V \to W \to X$, donde la única variable observable es X.

numérico que utiliza la información del gradiente de la energía. Este procedimiento se conoce como integración tipo **leapfrog**, y consiste en aplicar transformaciones que mantengan aproximadamente constante el valor de $H(\theta, r)$. Así como el gradiente descendente desplaza los parámetros hacia la dirección de decrecimiento de la función objetivo, la integración leapfrog desplaza los parámetros sobre una curva de nivel de $H(\theta, r)$.

Una característica distintiva de NUTS es que, en lugar de requerir un número fijo de pasos de integración (como en otros algoritmos relacionados), a partir de θ_t, construye dinámicamente un árbol de posibles (θ, r), expandiéndose hacia adelante y hacia atrás, hasta que detecta una "vuelta" en la trayectoria. Esta condición de detención se conoce como la "no-vuelta" (No-U-Turn), y se verifica cuando el desplazamiento entre dos nodos de la trayectoria es tal que continuar explorando implicaría retroceder en la dirección original de exploración. Una vez elegido un candidato (θ', r'), se acepta el cambio $\theta_{t+1} = \theta'$ con probabilidad $1 - e^{-(H(\theta', r') - H(\theta_t, r_t))}$ o, por el contrario, se decide por $\theta_{t+1} = \theta_t$.

Una vez finalizado este período, el algoritmo congela sus parámetros de control y comienza a generar las muestras válidas. En el contexto de modelos con parámetros continuos, NUTS es especialmente eficiente en espacios de alta dimensión, ya que genera propuestas informadas por la geometría local de la distribución, evitando caminatas aleatorias ineficientes y generando muestras con menor autocorrelación. Esto lo convierte en el algoritmo por defecto en muchas librerías bayesianas modernas como `PyMC`.

3.4. Ejemplo de modelo completo

En la Fig. 3 se muestra un ejemplo de modelado con tres variables $V \to W \to X$, donde la única variable observable es $X = x$. Supongamos que a priori $V \sim \exp(2)$, que $W|_{V=v} \sim \text{Poi}(v)$ y que $X|_{W=w} \sim \chi^2(w + 1)$. En este caso, las cadenas se generan con el siguiente procedimiento.

1. Se inicializa v_0, usualmente utilizando una muestra de la distribución a priori exp(2).

2. Se inicializa w_0, a partir de su distribución latente $\text{Poi}(v_0)$ o eventualmente con Poi(0,5) (usando la media de V en lugar de su valor).

3. Se actualiza V, definiendo v_1. Como V es continua derivable, habitualmente se utilizará NUTS aplicando sobre la distribución *a posteriori* $\pi(v, w_0) \propto p(x|w_0)P(w_0|v)p(v) \propto \text{Poi}(w_0|v) \cdot \exp(v|2)$, donde $\text{Poi}(\cdot|\mu)$ es la función de probabilidad de una Poisson de media μ y $\exp(\cdot|\lambda)$ es la función de densidad de una exponencial de intensidad λ. En este caso se absorbió la verosimilitud como constante de proporcionalidad por no depender v.

4. Se actualiza W, definiendo w_1. Al tratarse de una variable discreta, se recomienda utilizar el muestreo Metrópolis. Para el muestreo se utilizará la distribución *a posteriori* $\pi(v_1, w) \propto p(x|w)P(w|v_1)p(v_1) = \chi^2(x|w+1) \cdot \text{Poi}(w|v_1)$ donde $\chi^2(\cdot|\nu)$ es la función de densidad de una *chi-cuadrado* de ν grados de libertad. En este caso se absorbió la distribución a priori $p(v_1)$ como constante de proporcionalidad por no depender w.

5. Se repite el paso (**3**) definiendo v_2 y se continúa iterando la cantidad de pasos que sea necesario.

4. Calidad de las muestras

Para evaluar la calidad de las muestras de un experimento de MCMC, suelen considerarse dos propiedades: la ergodicidad y la estacionariedad. Gracias al teorema de ergodicidad, podemos aproximar esperanzas a partir de promedios sin pretender que las muestras sean independientes. El problema radica en que la velocidad de convergencia y la varianza de dicho promedio no son las mismas que en el caso de variables independientes. En MCMC, se denomina tamaño de muestra efectiva (Effective Sample Size, ESS) a la cantidad de datos independientes necesarios para alcanzar la misma varianza que posee el promedio de las muestras. Se define como

$$\text{ESS} = \frac{t_{\max}}{1 + 2\sum_{t=1}^{k_{\max}} \rho_t} \tag{20}$$

donde ρ_t es la autocorrelación de la cadena y $k_{\max}$ es el valor a partir del cual las autocorrelaciones se vuelven pequeñas o negativas. Esta ESS se conoce como **bulk** y se utiliza para cálculos predictivos. Para intervalos de confianza suele usarse otro ESS denominado **tail**.

Otra característica importante, además de la ergodicidad, es verificar si las muestras fueron generadas una vez alcanzado el estado estacionario. Supongamos que se cuenta con una simulación con varias cadenas independientes. Si todas convergieron a la misma distribución, entonces la varianza entre cadenas debería ser similar a la varianza dentro de cada cadena. Se denomina R-hat (o $\hat{R}$, también conocido como diagnóstico Gelman–Rubin) al cociente entre estas varianzas. Si las cadenas aún no convergieron, habrá más variabilidad entre cadenas, y $\hat{R}$ será mayor que 1. En la práctica suele considerarse $\hat{R} > 1{,}01$ una señal de alerta y un valor $\hat{R} > 1{,}1$ suele considerarse un problema a resolver.

4.1. Demostración del cálculo de ESS

En cálculo predictivo numérico, la hipótesis de trabajo es aproximar una esperanza con el promedio de las densidades $p_{X|T=\theta_i}(x)$ en lugar de los parámetros pero, en la práctica, se suele analizar la varianza de los parámetros para estandarizar resultados. No sería particularmente complejo trabajar con las verosimilitudes evaluadas en los parámetros, pero no hay grandes diferencias. En última instancia, bajo ciertas condiciones de regularidad, deberían ser comparables ambas ESS.

Sea $\theta_1, \cdots, \theta_{t_{\max}}$ un conjunto de muestras idénticamente distribuidas y sea $\bar{\theta}$ el promedio de las mismas; es simple ver que la varianza si fueran independientes sería de $\frac{\sigma^2}{\text{ESS}}$, donde

$\sigma^2 = \text{var}(\theta_i)$. En el caso de existir un $\rho_t = \frac{\text{cov}(\theta_i, \theta_{i+t})}{\sigma^2}$, la varianza se puede calcular como:

$$\text{var}(\bar{\theta}) = \text{var}\left(\frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} \theta_t\right) = \frac{1}{t_{\max}^2} \sum_{i=1}^{t_{\max}} \sum_{j=1}^{t_{\max}} \text{cov}\left(\theta_i, \theta_j\right) \tag{21}$$

$$= \frac{1}{t_{\max}^2} \left(\sigma^2 \cdot t_{\max} + 2\sigma^2 \sum_{i=1}^{t_{\max}-1} \sum_{j=i+1}^{t_{\max}} \rho_{j-i}\right) \tag{22}$$

$$= \frac{\sigma^2}{t_{\max}} \left(1 + 2 \sum_{t=1}^{t_{\max}-1} \left(1 - \frac{t}{t_{\max}}\right) \cdot \rho_t\right) \tag{23}$$

donde se aplicó el cambio de variables $t = j - i$. Bajo las hipótesis usuales en modelos MCMC, es razonable que tanto ρ_t como $1 - \frac{t}{t_{\max}}$ se vayan achicando a medida que se avanza en la cadena. Los algoritmos suelen truncar la suma, en un $k_{\max}$, cuando las autocorrelaciones se vuelven pequeñas o negativas. Con este procedimiento suele ocurrir que $k_{\max} \ll t_{\max}$, por lo que despreciando $\frac{t}{t_{\max}}$ se puede aproximar:

$$\text{var}(\bar{\theta}) \approx \frac{\sigma^2}{t_{\max}} \left(1 + 2 \sum_{t=1}^{k_{\max}} \rho_t\right) \tag{24}$$

Igualando este resultado con la expresión de la varianza para variables independientes $\frac{\sigma^2}{\text{ESS}}$, y despejando, se obtiene (20) finalizando la demostración.