Métodos avanzados de muestreo MCMC

Víctor Pascual del Olmo

Proyecto de Fin de Carrera

MÉTODOS AVANZADOS DE MUESTREO MCMC

Autor

VÍCTOR PASCUAL DEL OLMO

Tutor

LUCA MARTINO

La defensa del presente Proyecto de Fin de Carrera se realizó el día 6 de Octubre de 2011, siendo evaluada por el siguiente Tribunal:

PRESIDENTE: David Luengo García

SECRETARIO: Efraín Mayhua López

VOCAL: Pedro Contreras Lallana

y habiendo obtenido la siguiente CALIFICACIÓN:

Agradecimientos

En primer lugar me gustaría agradecer a mi tutor Luca Martino por el interés mostrado y el tiempo dedicado. Se ha involucrado por segunda o tercera vez, según como se quiera ver, en esta locura llamada Proyecto de Fin de Carrera. Me dedicó más tiempo del que me hubiese imaginado, y eso que tenía que preparar su Tesis Doctoral. Me gustaría agradecer a todo aquel ingenuo que me preguntaba aquello de *¿y de qué va tu proyecto?* Esas preguntas me entrenaban y me hacían temer un poco menos al Día de la Verdad. También a mis padres y hermanos que sólo me veían cuando llegaba a casa para comer. A los amigos y compañeros que estaban conmigo en la biblioteca, y que sufrían el rato en el que hablaba de mi proyecto, de los libros de *Terry Pratchett* o blogs de ciencia, tecnología y humor que leía diariamente.

En definitiva, agradezco todo el esfuerzo a mis padres (*Los Marqueses del Priorato*), a mi hermanos David (*Deivid*) y Esther (*Estherity*), a mi cuñada Aurora (*Au*), a mis amigos Adri (*Panto*), Javi (*el hijo*), Sam (*el padre*), Elenis (*la madre*), Elia (la chica antes conocida como "*oh honey!*"), Alba (*la chica más rara del grupo*), Mon (*Monicaja o Moniquilla*, en definitiva la mujer más fuerte que jamás he conocido), Ana (*la danzarina*), Álex (*Juan Villares*), Natalia (*la sistemática*), Bowen (*la jefa de Vodafone*), Ana (*la qué más mola*), Mª José (*la vecina*), Elena (*la chica de El Jueves*), Cerrón (*Cerrasmus*), Isaac y Maica (*los cristianos con los que siempre discuto*), y a otros muchos más que por falta de apodos no he puesto. ¡GRACIAS A TODOS y que *la Fuerza os acompañe*!

Índice general

Capítulo 1

Objetivos y organización del proyecto

1.1. Objetivos

Este proyecto se propone estudiar, analizar e investigar las diferentes metodologías de generación de números aleatorios mediante técnicas avanzadas y modernas de *Monte Carlo Markov Chain* (*MCMC*). Los métodos de Monte Carlo son métodos numéricos usados para calcular, aproximar y simular expresiones o sistemas matemáticos complejos y difíciles de evaluar. Aunque estos métodos comenzaron a desarrollarse en los años cuarenta, hasta que las computadoras no se hicieron más potentes estuvieron en un segundo plano.

Los métodos MCMC se basan en el diseño de una adecuada *cadena de Markov*. Bajo ciertas condiciones estas cadenas convergen a una densidad estacionaria invariante en el tiempo. La idea fundamental de los métodos MCMC es la generación de una cadena de Markov cuya densidad estacionaria coincide con la densidad que se quiere muestrear. Las cadenas de Markov son procesos estocásticos en el que la probabilidad de que ocurra un evento depende del evento inmediatamente anterior. Por lo tanto, los métodos MCMC producen números aleatorios correlacionados entre sí.

Como veremos, las técnicas MCMC pueden ser aplicadas teóricamente (y de manera fácil e inmediata, sin estudios analíticos previos) a cualquier densidad de probabilidad. Esta característica las hace particularmente interesantes en la práctica. De

hecho, no sólo se han multiplicado las aplicaciones en las últimas décadas sino que, a través de pequeñas variaciones, se han diseñado algoritmos parecidos para problemas de *optimización estocástica* y otros campos diferentes al del muestreo.

Los pasos que se han seguido en el desarrollo de este proyecto han sido los siguientes:

1. Una primera fase de estudio bibliográfico profundo y exhaustivo, en la que se buscó y examinó una cantidad considerable de información sobre algoritmos básicos de muestreo aleatorio con el fin de asimilar el mayor conocimiento posible.

2. Análisis y estudio específico de los algoritmos MCMC tradicionales con especial énfasis en el algoritmo *Metropolis-Hastings*.

3. Amplio análisis y búsqueda bibliográfica sobre los métodos modernos MCMC (sobre todo se ha utilizado la base de [4, 38]).

4. Elección, estudio teórico y análisis comparativo de las técnicas avanzadas MCMC a nuestro juicio más interesantes.

5. Implementación y análisis comparativo de los algoritmos más interesantes.

6. Redacción de la memoria[1] y revisión del proyecto.

1.2. Organización del proyecto

El proyecto se estructura en partes y en capítulos. Las partes que consta el proyecto son las siguientes:

[1] Esta memoria ha sido redactada utilizando LyX, procesador de documentos que combina la potencia de TeX/LaTeX con la facilidad de uso de una interfaz gráfica. Es un soporte universal para la creación de contenido matemático (mediante un editor de ecuaciones totalmente integrado) y documentos estructurados como artículos académicos, tesis o libros (para más información sobre LyX).

Parte I. Métodos básicos de muestreo

- **Capítulo 2:** Breve introducción a las técnicas básicas de muestreo para la generación de números aleatorios.
- **Capítulo 3:** Descripción y análisis teórico del algoritmo Metropolis-Hastings y sus variantes básicas.

Parte II. Técnicas avanzadas de muestreo

- **Capítulo 4:** Descripción de las limitaciones del algoritmo Metropolis-Hastings tradicional y de unas posibles estrategias para solventar estos defectos.
- **Capítulo 5:** Descripción y estudio de los métodos multi-punto.
- **Capítulo 6:** Descripción y estudio de los métodos basados en población.

Parte III. Simulaciones y conclusiones

- **Capítulo 7:** Implementación de los algoritmos principales.
- **Capítulo 8:** Resumen y conclusión del proyecto.

Parte IV. Apéndices

Esta parte contiene material teórico adicional y el presupuesto estimado del proyecto.

1.3. Notación

Para una mejor y ágil comprensión del texto que seguirá a continuación, presentamos la notación que vamos a utilizar en el resto del proyecto.

Vectores, matrices, puntos e intervalos

Los escalares los denotaremos con letras regulares y minúsculas, por ejemplo x o y. Los vectores se denotarán usando letras negritas y minúsculas, por ejemplo $\mathbf{x}$. Las componentes de un vector n-dimensional se denotarán de las dos maneras siguientes $\mathbf{x} = [x_1, ..., x_n]$ o $\mathbf{x} = (x_1, ..., x_n)$ (esta última notación se utilizará más para denotar un punto geométrico contenido en el espacio n-dimensional).

La notación en las matrices será similar a la notación de los vectores pero añadiendo una barra superior, $\bar{\mathbf{x}}$, por ejemplo una matriz $d \times n$ se indicará como

$$\bar{\mathbf{x}} = \begin{bmatrix} x_{11} & . & . & . & x_{1n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ x_{d1} & . & . & . & x_{dn} \end{bmatrix} = [\mathbf{x_1}, ..., \mathbf{x_n}] .$$

La notación $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$ indicará un intervalo cerrado, $[a, b) = \{x \in \mathbb{R} : a \leq x < b\}$ semi-abierto y $(a, b) = \{x \in \mathbb{R} : a < x < b\}$ abierto.

Variables aleatorias

Las variables aleatorias (v.a.) las denotaremos con letras mayúsculas, por ejemplo X o Y, en cambio sus realizaciones (*muestras*) las denotaremos con letras minúsculas, por ejemplo x o y.

Las densidades de probabilidad de una variable aleatoria las denotaremos como $f(\cdot)$. Por ejemplo, la variable aleatoria X tiene como densidad de probabilidad $f(x)$. Si definimos otra variable aleatoria Y, podemos definir la densidad de probabilidad condicional de X dado $Y = y$ como $f(x|y)$.

La probabilidad de que se de un evento, por ejemplo $X \leq x$, se denotará como $\text{Prob}\{X \leq x\}$. Podemos escribir la función de distribución de una v.a. $F_X(x) = \text{Prob}\{X \leq x\}$.

Con la siguiente notación $\mathcal{U}([a, b])$ indicaremos una distribución uniforme en el intervalo $[a, b]$, mientras con la expresión $\mathcal{N}(\mu, \sigma^2)$ indicaremos una densidad Gaus-

siana con media μ y varianza σ^2. Con el símbolo $\sim$ indicaremos que una variable aleatoria X o una muestra x tiene una distribución o una densidad indicada, por ejemplo, $X \sim \mathcal{N}(\mu, \sigma^2)$. Específicamente, indicaremos con $\mathcal{N}(\mu, \sigma^2)$ la densidad o la distribución Gaussiana correspondiente de forma indistinta, mientras con

$$\mathcal{N}\left(x; \mu, \sigma^2\right) \propto \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{1.3.1}$$

nos referiremos a una densidad Gaussiana.

Conjuntos

Los conjuntos se denotarán mediante letras mayúsculas caligráficas, por ejemplo $\mathcal{T}$. El soporte de la variable aleatoria de interés X se denotara como $\mathcal{S} \subseteq \mathbb{R}$, siendo $\mathbb{R}$ el conjunto de números reales. En algunos casos, sin pérdida de la generalidad se puede considerar $\mathcal{S} = \mathbb{R}$ por conveniencia.

La función indicadora del conjunto $\mathcal{D}$ se escribe como $\mathbb{I}_{\mathcal{D}}(x)$. Dicha función toma el valor 1 si $x \in \mathcal{D}$ y 0 en caso contrario, es decir,

$$\mathbb{I}_{\mathcal{D}}(x) = \begin{cases} 1 & \text{si } x \in \mathcal{D} \\ 0 & \text{si } x \notin \mathcal{D} \end{cases}. \tag{1.3.2}$$

Por otro lado, definimos la función $\delta(x-\mu)$ (delta de Dirac) como

$$\delta(x-\mu) = \begin{cases} +\infty & \text{si} \quad x = \mu \\ 0 & \text{si} \quad x \neq \mu \end{cases}, \tag{1.3.3}$$

donde μ es una constante.

Nomenclatura de funciones más utilizadas

Para evitar confusiones utilizaremos la siguiente nomenclatura:

$p_o(\cdot)$ densidad objetivo o *target*. Es la densidad que queremos muestrear.

$p(\cdot)$ función proporcional a la densidad objetivo, $p(x) \propto p_o(x)$.

$\pi(\cdot|\cdot)$ densidad tentativa o *proposal*. Es la densidad que somos capaces de muestrear.

$K(\cdot|\cdot)$ núcleo o *kernel*, es la función que define la probabilidad de saltar de estados en la cadena de Markov.

$\mathcal{S} \subseteq \mathbb{R}^m$ conjunto de definición de la variable x de interés.

1.3.1. Acrónimos

Los acrónimos que utilizaremos serán los siguientes:

fdp *función de densidad de probabilidad*

GMTM *Generalization Multiple-Try Metropolis*

IS *Importance Sampling*

MCMC *Monte Carlo Markov Chain*

MH *Metropolis-Hastings*

MTM *Multiple-Try Metropolis*

MTMIS *Multiple-Trial Metropolized Independence Sampler*

OBMC *Orientation bias Monte Carlo*

PT *Parallel Tempering*

RS *Rejection Sampling*

SMH *Sample Metropolis-Hastings*

v.a. *variable aleatoria*

Parte I

Métodos básicos de muestreo

Cualquiera que considere métodos aritméticos para producir dígitos aleatorios está, por supuesto, en pecado mortal.
(John von Neumann)

Capítulo 2

Introducción al muestreo aleatorio

La generación de variables aleatorias no uniformes es un campo de investigación interdisciplinar entre las matemáticas, estadística e informática. También se considera como una sub-área del cálculo estadístico y la metodología de simulación [1, 2, 3, 4, 5, 6][1].

El objetivo de los algoritmos de muestreo es generar números aleatorios que se distribuyen como una determinada densidad $p_o(x)$, que llamaremos *densidad objetivo*. Cada algoritmo de muestreo siempre asume disponible una fuente de números aleatorios con densidad $\pi(x)$, que llamaremos *densidad tentativa*. Todos los métodos que discutiremos a continuación convierten las muestras desde $\pi(x)$ a muestras de $p_o(x)$.

Las diferentes técnicas de muestreo se pueden clasificar en tres grandes categorías:

***Métodos directos*:** estas técnicas se basan en la utilización de una transformación adecuada de las muestras obtenidas por un generador de números aleatorios, cuya densidad de probabilidad es la densidad tentativa, para obtener muestras cuya densidad de probabilidad es la densidad objetivo. En general, este método es el más rápido y genera muestras independientes (al menos igual de independientes como las obtenidas antes de la transformación). La principal desventaja de estos métodos es que son difícilmente aplicables en la práctica

[1] Nótese que las referencias están ordenadas por aparición en el texto.

por el desconocimiento de la transformación que es necesario aplicar.

***Métodos de aceptación/rechazo*:** dadas las muestras aleatorias de un generador aleatorio disponible, este algoritmo acepta o rechaza las muestras mediante un test. Igual que en los métodos directos, las muestras son independientes tanto en cuanto el generador inicial lo sea. El principal problema de estos métodos es que tienen un coste computacional alto si su ratio de aceptación es bajo. Es decir, la probabilidad de aceptar una muestra puede ser muy baja.

***MCMC (Monte Carlo Markov Chain)*:** estas técnicas están basadas en la construcción de una cadena de Markov que converge a la densidad objetivo. La principal ventaja de estos algoritmos es que se pueden aplicar en casi cualquier caso, son casi universales. Su principal desventaja es que producen muestras correladas. Por lo tanto, las estimaciones resultantes de estas muestras tienden a tener una mayor varianza que las obtenidas de muestras independientes.

Muchos libros sobre métodos de Monte Carlo añaden otra clase: las técnicas de ***Importance Sampling***. Estos métodos asignan *pesos* a las muestras generadas por la densidad tentativa de manera que se aproximan a la medida de probabilidad representada por la densidad objetivo. Las técnicas de *Importance Sampling* **no** son en si generadores de números aleatorios, por esta razón no incluimos esta categoría en la clasificación anterior. De todas formas, en el Capítulo 5 veremos como la aproximación de la densidad objetivo lograda con la técnica Importance Sampling, puede ser utilizada como generador de números aleatorios, mezclando esta técnica con los principios básicos de los algoritmos MCMC.

En este capítulo vamos a describir rápidamente diferentes aspectos y técnicas de las categorías anteriores. Antes de estudiar cada clase previamente descrita, presentamos primero un simple resultado básico del muestreo aleatorio conocido como *Teorema fundamental de la simulación.*

Importante: nótese que, por razones de sencillez, en este capítulo hemos considerado la notación de variables escalares para la $x \in \mathbb{R}$ pero la mayoría de los resultados pueden aplicarse al caso general, $\mathbf{x} \in \mathbb{R}^m$.

2.1. Teorema fundamental de la simulación

Muchas técnicas de Monte Carlo se basan en un simple resultado que enunciamos a continuación.

Theorem 1. *Muestreando una variable aleatoria unidimensional X con densidad $p_o(x) \propto p(x)$ es equivalente a muestrear uniformemente una región bidimensional definida por*

$$\mathcal{A}_0 = \left\{(x,u) \in \mathbb{R}^2 : 0 \leq u \leq p(x)\right\}. \tag{2.1.1}$$

Es decir, si (x',u') está distribuida uniformemente en $\mathcal{A}_0$, entonces x' es una muestra de $p_o(x)$[11].

Demostración. Si consideramos dos variables aleatorias (X,U) uniformemente distribuidas en la región $\mathcal{A}_0$, y sea $q(x,u)$ su densidad de probabilidad conjunta, es decir,

$$q(x,u) = \frac{1}{|\mathcal{A}_0|}\mathbb{I}_{\mathcal{A}_0}(x,u), \tag{2.1.2}$$

donde $\mathbb{I}_{\mathcal{A}_0}(x,u)$ es la función de indicador en $\mathcal{A}_0$ y $|\mathcal{A}_0|$ es el área en la región $\mathcal{A}_0$. Además, podemos escribir $q(x,u) = q(u|x)\,q(x)$. El teorema está demostrado si la densidad marginal $q(x)$ es exactamente $p_o(x)$.

Si (X,Z) está uniformemente distribuida en la región $\mathcal{A}_0$, tenemos $q(u|x) = \frac{1}{p(x)}$ con $0 \leq u \leq p(x)$, es decir,

$$q(u|x) = \frac{1}{p(x)}\mathbb{I}_{\mathcal{A}_0}(x,u). \tag{2.1.3}$$

Por lo tanto, podemos expresar lo siguiente,

$$q(x,u) = q(u|x)\,q(x) = \frac{1}{p(x)}\mathbb{I}_{\mathcal{A}_0}(x,u)\,q(x). \tag{2.1.4}$$

Dada la Ecuación (2.1.2) podemos escribir entonces,

$$\frac{1}{|\mathcal{A}_0|}\mathbb{I}_{\mathcal{A}_0}(x,u) = \frac{1}{p(x)}\mathbb{I}_{\mathcal{A}_0}(x,u)\,q(x), \tag{2.1.5}$$

y despejando $q(x)$ llegamos a

$$q(x) = \frac{1}{|\mathcal{A}_0|} p(x) = p_o(x). \tag{2.1.6}$$

□

Por lo tanto, si obtenemos uniformemente un par de muestras (x', u') en la región $\mathcal{A}_0$, la muestra x' está distribuida acorde a la marginal $p_o(x)$. Muchas técnicas de Monte Carlo (por ejemplo, slice sampling, método de la densidad inversa, métodos de aceptación/rechazo, etc.) generan un par de variables aleatorias (X, U) y consideran solo la primera muestra x', siendo la variable U una variable auxiliar. En la Figura 2.1.1 podemos observar gráficamente la idea del teorema fundamental de la simulación.

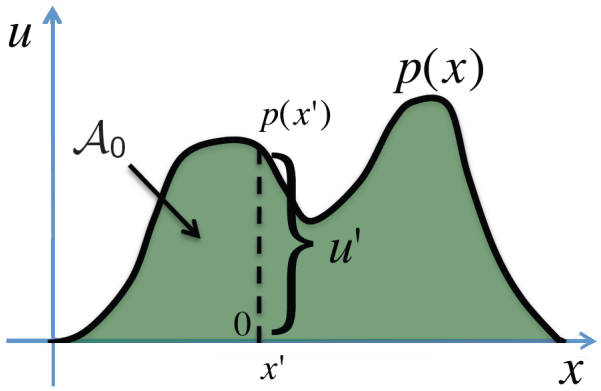

Figura 2.1.1: El área $\mathcal{A}_0$ es el área debajo de la curva $p(x) \propto p_o(x)$

En este capítulo veremos dos técnicas que muestran claramente esta idea: método de la densidad inversa y método de aceptación/rechazo.

2.2. Métodos Directos

A continuación, vamos a describir diferentes ejemplos de técnicas de muestreo basados en transformaciones que relacionan variables aleatorias.

2.2.1. Método de inversión

Dada la variable aleatoria X con función de densidad de probabilidad (fdp) $p_o(x)$ y consideramos su función de distribución acumulativa como

$$F_X(x) = \text{Prob}\{X \leq x\} = \int_{-\infty}^{x} p_o(v)\, dv, \tag{2.2.1}$$

siendo una función creciente monótona, entonces podemos definir la función inversa como

$$F_X^{-1}(y) \triangleq \text{ínf}\{x \in \mathcal{D} : F_X(x) \geq y\}. \tag{2.2.2}$$

Theorem 2. *Si $U \sim \mathcal{U}([0,1])$, entonces $Z = F_X^{-1}(U)$ tiene una densidad $p_o(x)$.*

Demostración. Por definición, tenemos

$$\left\{(u,x) \in \mathbb{R}^2 : F_X^{-1}(u) \leq x\right\} = \left\{(u,x) \in \mathbb{R}^2 : u \leq F_X(x)\right\}. \tag{2.2.3}$$

Por lo tanto, podemos escribir

$$F_Z(z) = \text{Prob}\left\{Z = F_X^{-1}(U) \leq z\right\} = \text{Prob}\{U \leq F_X(z)\} = F_X(z), \tag{2.2.4}$$

entonces X y Z tienen la misma distribución, $X \overset{d}{=} Z$, además, la misma función de densidad $p_o(x)$. □

Si conocemos analíticamente la expresión de la función inversa $F_X^{-1}(\cdot)$, entonces podemos generar inicialmente una muestra $u' \sim \mathcal{U}([0,1])$, y transformar $x' = F_X^{-1}(u')$, obteniendo una muestra x' que se distribuye acorde a $p_o(x)$.

Más en general, dada una variable aleatoria Y con una función de distribución $F_Y(y)$, la variable aleatoria definida con la transformación monótona

$$X = F_X^{-1}(F_Y(Y)), \tag{2.2.5}$$

está distribuida acorde a $p_o(x)$.

2.2.2. Transformaciones no monótonas

En este punto, asumimos que la relación entre dos variables aleatorias X e Y, con densidades de probabilidad $p_o(x)$ y $q(y)$ respectivamente, se puede expresar con una transformación no monótona $Y = \psi(X)$ [12]. Definimos la función inversa como

$$\psi^{-1}(y) = \{x \in \mathbb{R} : \psi(x) = y\}. \tag{2.2.6}$$

Ya que ψ es una función no monótona, para un valor genérico y, $\psi^{-1}(y)$ contiene más de una solución, por lo tanto, $\psi^{-1}(y) = \{x_1, ..., x_n\}$. Entonces, si somos capaces de generar una muestra y' desde $q(y)$, tendremos que elegir adecuadamente x' entre las n soluciones que obtenemos de $\psi^{-1}(y')$ para que x' se distribuya como $p_o(x)$.

Para simplificar los cálculos, supongamos el caso de $n = 2$, es decir, $\psi^{-1}(y) = \{x_1, x_2\}$ para todas los posibles valores de y. Podemos descomponer $\psi(x)$ en dos funciones monótonas e invertibles. Para cada valor de y es posible encontrar dos valores de $x_c \in \mathbb{R}$ de tal manera que $\psi_1(x) \triangleq \psi(x)$ es una función monótona en el dominio $(-\infty, x_c]$, y $\psi_2(x) \triangleq \psi(x)$ es otra función monótona en el dominio $x \in [x_c, +\infty)$. Por consiguiente, usamos la notación

$$x_1 = \psi^{-1}(y), \quad x_2 = \psi^{-1}(y), \tag{2.2.7}$$

para indicar las dos soluciones obtenidas de la ecuación $y = \psi(x)$. La densidad de probabilidad $q(y)$ de la variable aleatoria $Y = \psi(X)$ se puede expresar como

$$q(y) = p_o\left(\psi_1^{-1}(y)\right)\left|\frac{d\psi_1^{-1}}{dy}\right| + p_o\left(\psi_2^{-1}(y)\right)\left|\frac{d\psi_2^{-1}}{dy}\right|, \tag{2.2.8}$$

donde recordemos que $p_o(x)$ es la función de densidad de probabilidad de X. Entonces, generando una muestra y' desde $q(y)$, podemos obtener una muestra x' de $p_o(x)$ tomando $x_1' = \psi_1^{-1}(y')$ con probabilidad

$$
\begin{aligned}
w_1 &= \frac{p_o\left(\psi_1^{-1}(y')\right)\left|\frac{d\psi_1^{-1}(y')}{dy}\right|}{q(y')} \\
&= \frac{p_o\left(\psi_1^{-1}(y')\right)\left|\frac{d\psi_1^{-1}(y')}{dy}\right|}{p_o\left(\psi_1^{-1}(y')\right)\left|\frac{d\psi_1^{-1}(y')}{dy}\right| + p_o\left(\psi_2^{-1}(y')\right)\left|\frac{d\psi_2^{-1}(y')}{dy}\right|},
\end{aligned}
\tag{2.2.9}
$$

o eligiendo $x_2' = \psi_2^{-1}(y')$ con probabilidad $w_2 = 1 - w_1$. Además, definimos que

$$
\left|\frac{d\psi_i^{-1}(y')}{dy}\right| = \left|\frac{1}{\frac{d\psi\left(\psi_i^{-1}(y')\right)}{dx}}\right| = \left|\frac{1}{\frac{d\psi\left(x_i'\right)}{dx}}\right|, \tag{2.2.10}
$$

con $i = 1, 2$, podemos reescribir la probabilidad w_1 de la siguiente forma,

$$
\begin{aligned}
w_1 &= \frac{p_o(x_1')\left|1/\frac{d\psi\left(x_1'\right)}{dx}\right|}{p_o(x_1')\left|1/\frac{d\psi\left(x_1'\right)}{dx}\right| + p_o(x_2')\left|1/\frac{d\psi\left(x_2'\right)}{dx}\right|} \\
&= \frac{p_o(x_1')\left|\frac{d\psi\left(x_2'\right)}{dx}\right|}{p_o(x_1')\left|\frac{d\psi\left(x_2'\right)}{dx}\right| + p_o(x_2')\left|\frac{d\psi\left(x_1'\right)}{dx}\right|}.
\end{aligned}
\tag{2.2.11}
$$

2.2.3. Método de la deconvolución

Consideremos dos variables aleatorias X y Z con densidad de probabilidad conjunta $f(x, z)$, y sea $p_o(x)$ la densidad de probabilidad de X. Supongamos, además, la siguiente relación

$$
Y = \varphi(X, Z), \tag{2.2.12}
$$

donde φ es una función invertible respecto a z, y la variable Y tiene una densidad de probabilidad $q(y)$. Suponiendo que somos capaces de muestrear $q(y)$, podemos

obtener muestras que se distribuyan según $p_o(x)$. De hecho, asumiendo que $\frac{d\varphi}{dz} \neq 0$, podemos escribir que

$$p(x,y) = f\left(x, \varphi^{-1}(x,y)\right) \left|\frac{d\varphi^{-1}}{dy}\right|, \tag{2.2.13}$$

donde hemos sustituido $z = \varphi^{-1}(x,y)$. Además, $q(y)$ es la densidad marginal de $p(x,y)$, es decir, $q(y) = \int_{\mathcal{S}} p(x,y)\, dx$. Obviamente, $p_o(x)$ es la otra densidad marginal, $p_o(x) = \int_{\mathcal{C}} p(x,y)\, dy$.

Dado que podemos siempre escribir una densidad de probabilidad conjunta como $p(x,y) = h(x|y)\, q(y)$, podemos muestrear una densidad $p_o(x)$ siguiente el siguiente procedimiento:

1. Generamos y' desde $q(y)$.

2. Luego muestreamos x' desde $h(x|y')$, donde la fdp condicional es

$$\begin{aligned} h(x|y) = \frac{p(x,y)}{q(y)} &= \frac{f\left(x, \varphi^{-1}(x,y)\right) \left|\frac{d\varphi^{-1}}{dy}\right|}{q(y)} \\ &= \frac{f\left(x, \varphi^{-1}(x,y)\right) \left|\frac{d\varphi^{-1}}{dy}\right|}{\int_{\mathcal{S}} f\left(x, \varphi^{-1}(x,y)\right) \left|\frac{d\varphi^{-1}}{dy}\right| dx}. \end{aligned} \tag{2.2.14}$$

Esta técnica está conectada con otros métodos llamados *polares* y puede considerarse un caso especial de una familia de métodos de muestreo que construyen una densidad conjunta para muestrear una marginal.

2.2.4. Método de la densidad inversa para fdp monótonas

El método de la densidad inversa está también conocido como *método Khinchine* [13, 14, 15]. Dada una densidad objetivo monótona $u = p_o(x)$, indicamos con $p_o^{-1}(u)$ la correspondiente función inversa. Es importante observar que $p_o^{-1}(u)$ está también normalizada porque describe la misma área $\mathcal{A}_0$ que $p_o(x)$. Es decir, $p_o^{-1}(u)$ es también

una densidad. Por lo tanto, podemos escribir lo siguiente

$$\mathcal{A}_0 = \left\{(x,u) \in \mathbb{R}^2 : 0 \leq u \leq p_o(x)\right\},$$

o también,

$$\mathcal{A}_0 = \left\{(u,x) \in \mathbb{R}^2 : 0 \leq x \leq p_o^{-1}(u)\right\}.$$

Entonces, para generar muestras uniformes en $\mathcal{A}_0$ tenemos dos procedimientos posibles:

1. Muestrear x' desde $p_o(x)$ y generar u' uniformemente en el intervalo $[0, p_o(x')]$, es decir, $u' \sim \mathcal{U}([0, p_o(x')])$.

2. Muestrear u' desde $p_o^{-1}(u)$ y generar x' uniformemente en el intervalo $[0, p_o^{-1}(u')]$, es decir, $x' \sim \mathcal{U}([0, p_o^{-1}(u')])$.

La Figura 2.2.1 muestra los dos procedimientos.

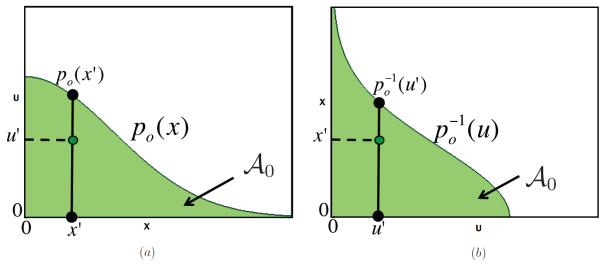

Figura 2.2.1: Dos formas de obtener un punto aleatorio (x', u') uniformemente en el área $\mathcal{A}_0$. (**a**) Podemos obtener una muestra x' de $p_o(x)$ y luego $u' \sim \mathcal{U}([0, p_o(x')])$. (**b**) También, podemos obtener una muestra u' de $p_o^{-1}(u)$ y luego $x' \sim \mathcal{U}([0, p_o^{-1}(u')])$.

Claramente, ambos procedimientos generan puntos (x', u') uniformemente distribuidos en la región $\mathcal{A}_0$. En ambos casos, la muestra x' está distribuida acorde a la densidad objetivo $p_o(x)$, mientras que la segunda muestra u' está distribuida mediante su inversa $p_o^{-1}(u)$. Entonces, si generamos muestras u' desde $p_o^{-1}(u)$, podemos usar el segundo procedimiento para generar muestras x' que se distribuyan según $p_o(x)$. Obviamente, este método puede utilizarse sólo si se puede muestrear $p_o^{-1}(u)$.

Nótese que generar una muestra x' uniformemente en el intervalo $[0, a]$, es decir, $x' \sim \mathcal{U}([0, a])$, es equivalente a generar una muestra z' uniformemente en el intervalo $[0, 1]$ y multiplicarla por a, es decir, $x' = z'a$. Entonces, esta técnica se puede expresar de la siguiente forma

$$X = Z p_o^{-1}(U), \tag{2.2.15}$$

donde X tiene la densidad $p_o(x)$, $Z \sim \mathcal{U}([0, 1])$, y U se distribuye acorde a $p_o^{-1}(u)$.

En la literatura podemos encontrar otra versión equivalente de este método. En esta versión alternativa se considera la variable aleatoria $W = p_o^{-1}(U)$, donde U se distribuye según $p_o^{-1}(u)$, con densidad

$$q(w) = p_o^{-1}(p_o(w)) \left| \frac{dp_o}{dw} \right| = w \left| \frac{dp_o}{dw} \right|. \tag{2.2.16}$$

Esta función es conocida como *densidad vertical* asociada a la inversa $p_o^{-1}(x)$[14, 16, 17]. Usando la variable aleatoria W, podemos expresar el método con este producto

$$X = ZW, \tag{2.2.17}$$

donde $Z \sim \mathcal{U}([0, 1])$ y $W \sim q(w)$.

2.3. Métodos de aceptación/rechazo

El método de aceptación/rechazo (en inglés, *accept/reject method*, también conocido como *rejection sampling* (*RS*)), es un método de Monte Carlo *universal* de muestreo, es decir, teóricamente puede ser utilizado para muestrear cualquier tipo de densidad objetivo. Este método fue propuesto por John von Neumann (matemático astro-húngaro que realizó grandes contribuciones en física cuántica, análisis funcional, teoría de conjuntos, ciencias de la computación, economía, análisis numérico, cibernética, hidrodinámica, estadística y otros campos de las matemáticas[18]) en 1951.

Consideramos la función $p(x) \propto p_o(x)$, siendo $p_o(x)$ la función objetivo que

queremos generar. También consideramos una fdp tentativa $\pi(x)$ fácil de muestrear. Además, elegimos una constante M tal que la curva $M\pi(x)$ está siempre por encima de la curva de $p(x)$, es decir,

$$M\pi(x) \geq p(x), \forall x \in \mathcal{S}, \tag{2.3.1}$$

siendo $\mathcal{S}$ la región en la que se distribuye $p(x)$.

En el algoritmo estándar de *RS*, primero obtenemos una muestra $x' \sim \pi(x)$, la aceptamos con probabilidad

$$p_A(x') = \frac{p(x')}{M\pi(x')} \leq 1. \tag{2.3.2}$$

En caso contrario descartamos la muestra y volvemos a muestrear $\pi(x)$. En la Figura 2.3.1, podemos ver el funcionamiento del algoritmo.

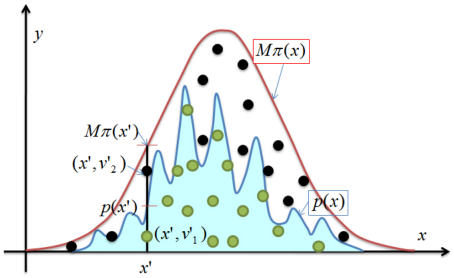

Figura 2.3.1: Los puntos que se encuentran por debajo de la curva $p(x)$, son aceptados. Los que se encuentran por debajo de la curva $M\pi(x)$ y por encima de la curva $p(x)$, son rechazados.

El algoritmo de aceptación/rechazo se puede expresar también con los siguientes pasos:

1. Genero dos muestras aleatorias, x' y u', donde $x' \sim \pi(x)$ y $u' \sim \mathcal{U}([0,1])$.
2. Si $u' \leq \frac{p_o(x')}{M\pi(x')}$, entonces aceptamos x', en caso contrario la muestra x' se rechaza y volvemos al paso 1.

La técnica de aceptación/rechazo está basada en el siguiente teorema.

Theorem 3. *Dada dos variables aleatorias X_1 y X_2 con fdp $\pi(x)$ y $p(x) \propto p_o(x)$, respectivamente, y dada una variable aleatoria uniforme U con distribución $\mathcal{U}([0,1])$. Si existe una constante $M \geq \frac{p(x)}{\pi(x)}$ $\forall x \in \mathcal{S}$, entonces*

$$Prob\left\{X_1 \leq y \middle| U \leq \frac{p(X_1)}{M\pi(X_1)}\right\} = Prob\left\{X_2 \leq y\right\}. \tag{2.3.3}$$

Demostración. Asumiendo que $\mathcal{S} = \mathbb{R}$, sin perder generalización, recordando que $U \sim \mathcal{U}([0,1])$ y X_1 está distribuida acorde a $\pi(x)$, podemos escribir lo siguiente,

$$\begin{aligned}\text{Prob}\left\{X_1 \leq y \middle| U \leq \frac{p(X_1)}{M\pi(X_1)}\right\} &= \frac{\text{Prob}\left\{X_1 \leq y, U \leq \frac{p(X_1)}{M\pi(X_1)}\right\}}{\text{Prob}\left\{U \leq \frac{p(X_1)}{M\pi(X_1)}\right\}} \\ &= \frac{\int_{-\infty}^{y}\int_{0}^{\frac{p(x)}{M\pi(x)}}\pi(x)\,dudx}{\int_{-\infty}^{+\infty}\int_{0}^{\frac{p(x)}{M\pi(x)}}\pi(x)\,dudx}.\end{aligned}$$

Entonces, si integramos primero respecto al diferencial u y realizamos un cálculos triviales, llegamos a la siguiente expresión

$$\text{Prob}\left\{X_1 \leq y \middle| U \leq \frac{p(X_1)}{L\pi(X_1)}\right\} = \frac{\int_{-\infty}^{y} p(x)\,dx}{\int_{-\infty}^{+\infty} p(x)\,dx}.$$

Además, si $p(x) \propto p_o(x)$, es decir, $p(x) = cp_o(x)$ donde x es una constante de normalización, podemos escribir la siguiente expresión

$$\begin{aligned}\text{Prob}\left\{X_1 \leq y \middle| U \leq \frac{p(X_1)}{M\pi(X_1)}\right\} &= \frac{\int_{-\infty}^{y} p(x)\,dx}{\int_{-\infty}^{+\infty} p(x)\,dx} = \frac{\int_{-\infty}^{y} cp_o(x)\,dx}{\int_{-\infty}^{+\infty} cp_o(x)\,dx} \\ &= \int_{-\infty}^{y} p_o(x)\,dx.\end{aligned}$$

Finalmente, si la variable aleatoria X_2 tiene una fdp $p_o(x)$, podemos escribir

$$\text{Prob}\left\{X_1 \leq y \middle| U \leq \frac{p(X_1)}{M\pi(X_1)}\right\} = \int_{-\infty}^{y} p_o(x)\,dx = \text{Prob}\{X_2 \leq y\} \tag{2.3.4}$$

verificando la Ecuación (2.3.3). □

2.3.1. ¿Qué ocurre si $M\pi(x) < p(x)$, $\forall x \in \mathcal{S}$?

Si no hemos elegido bien el valor de M, el algoritmo no funciona adecuadamente y la distribución que obtendremos no será la que deseábamos. En la Figura 2.3.2 podemos ver un ejemplo de este caso, en el que una parte de la fdp de la función objetivo $p(x) \propto p_o(x)$, se encuentra por encima de la curva $M\pi(x)$.

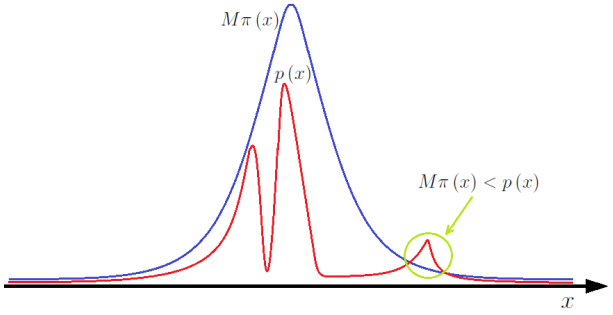

Figura 2.3.2: La parte señalada indica que la curva $p(x) \propto p_o(x)$ se encuentra por encima de la curva $M\pi(x)$, es decir, $p(x) > M\pi(x)$.

Si en un subespacio del espacio de trabajo $\mathcal{S}$, se cumple que $p(x) > M\pi(x)$, entonces todas las muestras generadas por la fdp tentativa siempre serán aceptadas. Pero, debido a la probabilidad de que aparezcan estas muestras en la densidad tentativa es menor, la función de densidad que obtendremos en el subespacio indicado, es exactamente la densidad tentativa restringida en el subespacio en el que *no* se cumple que $p(x) \leq M\pi(x)$. En definitiva, podemos escribir que en este caso la densidad muestreada sería

$$q(x) \propto \min\left[\pi(x), p(x)\right]. \tag{2.3.5}$$

En la Figura 2.3.3 podemos ver la curva correspondiente a la densidad $q(x)$ que obtendremos.

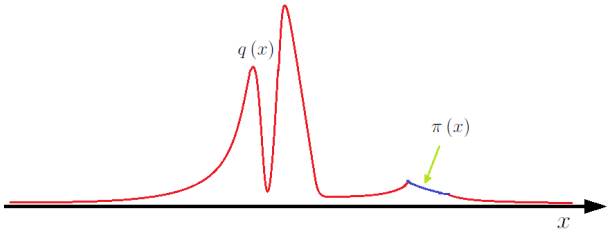

Figura 2.3.3: La curva correspondiente a la densidad $q(x) \propto \min[\pi(x), p(x)]$.

2.3.2. Squeeze principle

Si la función $p(x) \propto p_o(x)$ es muy compleja y muy costosa de evaluar se suele aplicar el conocido "squeeze principle". Supongamos conocer una función $\varphi(x)$ tal que se cumple la siguiente relación

$$\varphi(x) \leq p_o(x) \leq M\pi(x). \tag{2.3.6}$$

En este caso, la idea básica del "squeeze principle" consiste en añadir una prueba previa, evitando la evaluación de $p(x) \propto p_o(x)$. Así que el algoritmo será el siguiente:

1. Genero dos muestras aleatorias, x' y u', donde $x' \sim \pi(x)$ y $u' \sim \mathcal{U}([0,1])$.
2. Si $u' \leq \frac{\varphi(x')}{M\pi(x')}$, aceptamos x' sin evaluar la función $p_o(x)$.
3. En cambio, si $u' > \frac{\varphi(x')}{L\pi(x')}$, entonces
   - si $u' \leq \frac{p_o(x')}{M\pi(x')}$, aceptamos x',
   - en cambio, si $u' > \frac{p_o(x')}{M\pi(x')}$, rechazamos x'.

En la Figura 2.3.4 ilustramos el "squeeze principle". En el área verde (ambas tonalidades) aceptamos la muestra x'. Evidentemente, en el área roja rechazamos la muestra x'.

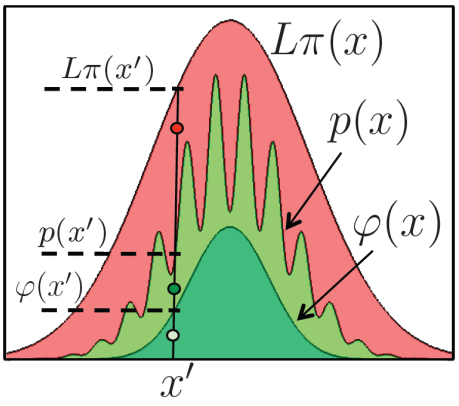

Figura 2.3.4: Con el *squeeze principle*, primero chequeamos si un punto $(x', u'L\pi(x'))$ cae dentro de la región verde oscuro, en ese caso, la muestra x' se acepta sin tener que evaluar la función $p(x)$.

2.4. MCMC

Los métodos de Markov Chain Monte Carlo (MCMC) se basan en la creación de una adecuada *cadena de Markov*. Las cadenas de Markov, bajo ciertas condiciones que veremos en el Capítulo 3, convergen a una densidad estacionaria invariante en el tiempo, que nosotros denotaremos como $p_e(x)$.

La idea fundamental de los métodos MCMC es la de diseñar una cadena de Markov cuya densidad estacionaria $p_e(x)$ coincida con la densidad objetivo $p_o(x)$. En este caso, la función tentativa $\pi(x_t|x_{t-1})$ será una probabilidad condicional dependiente del instante anterior, es decir, dependiente de la muestra generada en el paso $t-1$.

Como ya hemos mencionado, los métodos MCMC pueden ser aplicados en la casi totalidad de densidades que se encuentran en la práctica. Esta característica los hace particularmente interesantes en los problemas de optimización y en la resolución de sistemas complejos. Desgraciadamente, estos métodos tienen dos inconvenientes:

- Las muestras producidas están correlacionadas. Este hecho proviene de la utilización de la cadena de Markov. Este problema produce una pérdida de información y limita la aplicación de estos métodos. En ciertos casos, la cadena

puede incluso quedarse "atrapada" en un subconjunto del espacio de estado.

- Existencia de un transitorio. Las muestras obtenidas en las primeras iteraciones del algoritmo, no se distribuyen mediante $p_o(x)$. A este periodo se le conoce como *burn in period*. Por lo tanto, si deseamos generar N muestras que se distribuyan según $p_o(x)$ y sabemos que el algoritmo debe generar M muestras para que el algoritmo converja, debemos generar $M + N$ muestras.

Antes de describir en detalle el más famoso método MCMC (el algoritmo Metropolis-Hastings) en el siguiente capítulo, a continuación veremos una técnica que aproxima la densidad medida de probabilidad de la densidad objetivo. Este método es *Importance Sampling*.

2.5. Importance Sampling

Como ya hemos visto, si queremos generar muestras aleatorias distribuidas mediante $p_o(x)$ mediante una fuente aleatoria con densidad $\pi(x)$, $x \in \mathcal{S} \subset \mathbb{R}$, debemos "modificar" las muestras producidas por $\pi(x)$ mediante una adecuada transformación, un test apropiado o mediante la creación de una cadena de Markov. En general, hay que corregir el sesgo entre las muestras de la función tentativa y la función objetivo mediante algún procedimiento estadístico.

Otro enfoque, conocido como importance sampling [22, 23] consiste en asociar *pesos* a las muestras producidas por la densidad tentativa $\pi(x)$. El peso representa la *importancia* estadística de la muestra (si los pesos están normalizados, entonces respecto al conjunto de todas muestras generadas). En la Figura 2.5.1 podemos ver gráficamente el funcionamiento del *importance sampling*.

Como ya hemos mencionado, la idea básica del importance sampling es asignar *pesos* a cada realización x' generada desde la fdp tentativa $\pi(x)$ de forma proporcional a

$$w(x') = \frac{p_o(x')}{\pi(x')} \propto \frac{p(x')}{\pi(x')}, \tag{2.5.1}$$

donde $p(x) \propto p_o(x)$. Estos pesos nos dan información sobre la *importancia* de la

muestra x'. La *importancia* es

- directamente proporcional a $w(x) \propto p(x)$ a la fdp objectivo. Esto es lógico dado que más alto es valor $p(x')$, más masa de probabilidad estará concentrada alrededor de x' (hay que dar "valor" a esta muestra).
- inversamente proporcional a $w(x) \propto 1/\pi(x)$ a la fdp tentativa. Esto es lógico porque si el valor $\pi(x')$ es bajo, significa que la probabilidad de proponer otra muestra x' (o alguna cercana alrededor de x') es muy poco probable, entonces hay que asignarle un "valor" alto (esta muestra es difícil que se repita). Además, si el valor $\pi(x')$ es alto, con alta probabilidad se propondrán otros puntos cerca de x' así que no necesario darle un peso alto.

Estos dos "efectos" juntos generan los pesos del *importance sampling*. En general, se puede dividir tres situaciones:

1. $p(x')$ alto, $\pi(x')$ bajo: la "importancia" (el peso) $w(x')$ será grande (mirar Figura 2.5.1 izquierda(a)).
2. $p(x') \approx \pi(x')$: tendremos $w(x') \approx 1$ (mirar Figura 2.5.1 central(b)).
3. $p(x')$ bajo, $\pi(x')$ alto: la "importancia" (el peso) $w(x')$ será pequeñaa (mirar Figura 2.5.1 derecha(c)).

En general, *los pesos*

$$w(x) \propto \frac{p(x)}{\pi(x)},$$

miden cuanto de distintas son las fdp tentativa y la fdp objectivo. Es decir, en realidad la situación mejor es la segunda de la lista anterior (cuando $p(x') \approx \pi(x')$). En el caso ideal, que $\pi(x) = p_o(x) \propto p(x)$ los pesos serán todos iguales a 1, $w(x) = 1\ \forall x \in \mathcal{S}$ (porque estamos muestreando directamente la fdp objectivo). De hecho, este factor $\frac{p(x)}{\pi(x)}$ aparece en otros métodos básicos de muestreo como el aceptación/rechazo o el Metropolis-Hastings con fdp tentativa independiente.

Como resulta obvio, importance sampling ***no*** es un generador de números aleatorios, ya que las muestras obtenidas siguen una densidad *con soporte aleatorio (discreto)* $h_N(x)$ *que se aproxima la medida de probabilidad de la densidad* $p_o(x)$. De

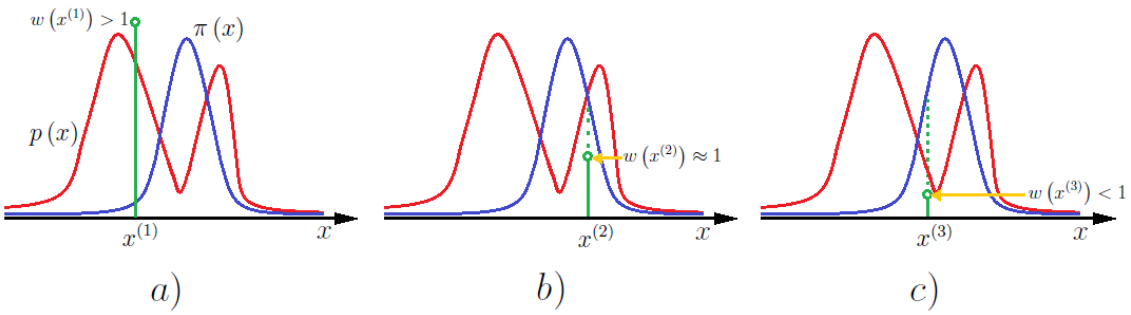

Figura 2.5.1: Funcionamiento de *importance sampling*. Cada muestra $x^{(i)}$ obtenida tiene asignado un peso en función de la densidad objetivo $p_o\left(x\right)$ y la densidad tentativa $\pi\left(x\right)$. Los pesos en la figura no se encuentran normalizados, por lo tanto los pesos $w\left(x^{(i)}\right)$ se definen como $w\left(x^{(i)}\right) = \frac{p\left(x^{(i)}\right)}{\pi\left(x^{(i)}\right)}$. *a)* El peso $w\left(x^{(1)}\right) > 1$ porque $p\left(x^{(1)}\right) > \pi\left(x^{(1)}\right)$. *b)* El peso $w\left(x^{(2)}\right) \approx 1$ porque $p\left(x^{(2)}\right) \approx \pi\left(x^{(2)}\right)$. *c)* El peso $w\left(x^{(3)}\right) < 1$ porque $p\left(x^{(3)}\right) < \pi\left(x^{(3)}\right)$.

todas formas, en los capítulos siguientes veremos varios ejemplos de como combinar este enfoque con diferentes estrategias MCMC para diseñar un generador aleatorio.

El algoritmo se compone de cuatro pasos:

1. Generar N muestras $x^{(i)}$, $i = 1, ..., N$, de la fdp tentativa $\pi\left(x\right)$.

2. Asociar pesos a cada muestra de esta forma

$$w_i = \frac{p_o\left(x^{(i)}\right)}{\pi\left(x^{(i)}\right)}, \tag{2.5.2}$$

   con $i = 1, ..., N$.

3. Entonces, podemos aproximar la medida de probabilidad de $p_o\left(x\right)$ con

$$h_N\left(x\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{p_o\left(x^{(i)}\right)}{\pi\left(x^{(i)}\right)}\delta\left(x - x^{(i)}\right) = \frac{1}{N}\sum_{i=1}^{N} w_i\delta\left(x - x^{(i)}\right). \tag{2.5.3}$$

Si no conocemos la constante de normalización de $p_o\left(x\right)$ (o de la fdp tentativa)

podemos proceder de la siguiente forma:

1. Generar N muestras $x^{(i)}$, $i = 1, ..., N$, de la fdp tentativa $\pi(x)$.

2. Asociar pesos a cada muestra de esta forma

$$w_i = \frac{p\left(x^{(i)}\right)}{\pi\left(x^{(i)}\right)}, \tag{2.5.4}$$

   con $i = 1, ..., N$, y con $p(x) \propto p_o(x)$.

3. Normalizar los pesos

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^{N} w_i}, \tag{2.5.5}$$

   con $i = 1, ..., N$.

4. Obtener $\bar{h}_N(x)$ como un tren de deltas de la siguiente forma,

$$\bar{h}_N(x) = \sum_{i=1}^{N} \bar{w}_i \delta\left(x - x^{(i)}\right). \tag{2.5.6}$$

Estas dos funciones $h_N(x)$ y $\bar{h}_N(x)$ son muy útiles por ejemplo para aproximar integrales relativas a la densidad $p_o(x)$, como puede ser el cálculo de momentos de esta densidad.

Cálculo de momentos utilizando Importance Sampling

Dada una variable aleatoria X con densidad de probabilidad $p_o(x)$ con $x \in \mathcal{S}$, la esperanza matemática $E[g(X)]$ se define de esta forma

$$I = E[g(X)] = \int_{\mathcal{S}} g(x)\, p_o(x)\, dx.$$

Si esta integral es intratable analíticamente, se puede estimar el valor de $E\left[g\left(X\right)\right]$ utilizando la aproximación $h_N\left(x\right)$ dada por el importance sampling, es decir,

$$I \approx I_N = \int_{\mathcal{S}} g\left(x\right) h_N\left(x\right) dx = \frac{1}{N} \sum_{i=1}^{N} \frac{p_o\left(x^{(i)}\right)}{\pi\left(x^{(i)}\right)} g\left(x^{(i)}\right). \tag{2.5.7}$$

Se puede demostrar que I_N *es un estimador insesgado de* I. Por otra parte, si no conocemos alguna constante de normalización de las dos fdp podemos aproximar la integral de esta forma

$$I \approx \bar{I}_N = \int_{\mathcal{S}} g\left(x\right) \bar{h}_N\left(x\right) dx = \sum_{i=1}^{N} \bar{w}_i g\left(x^{(i)}\right), \tag{2.5.8}$$

donde los $\bar{w}_i$ son los pesos normalizados. Aunque este estimador $\bar{I}_N$ está sesgado suele tener una varianza del error menor respecto al anterior I_N.

2.5.1. Resampling

El "resampling" [38] (remuestreo) es una técnica utilizada sobre todo en el muestreo secuencial de variables aleatorias junto con el *importance sampling*. Hay diferentes tipos de remuestreo, como ejemplo aquí describimos el remuestreo multinomial.

Multinomial Resampling

Sustancialmente, se trata de muestrear (suele ser N veces) la variable aleatoria discreta W con densidad de probabilidad dada a través de pesos como

$$\text{Prob}\{W = i\} = \bar{\omega}_i,$$

o se puede ver también como muestrear (N veces) una variable continua X con densidad un tren de deltas,

$$h_N(\mathbf{x}) = \sum_{i=1}^{N} \bar{\omega}_i \delta(\mathbf{x} - \mathbf{x}^{(i)}).$$

Claramente, las partículas con menor peso $\bar{\omega}_i$ tienen poca probabilidad de "*sobrevivir*" al resampling, mientras las partículas con mayor peso $\bar{\omega}_i$ serán "*replicadas/duplicadas*" con alta probabilidad después de un paso de resampling. Como ejemplo consideremos $N = 3$ partículas, $x^{(1)} = 0{,}2$, $x^{(2)} = -2$ y $x^{(3)} = 7{,}6$ con pesos $\bar{\omega}_1 = 0{,}3$, $\bar{\omega}_2 = 0{,}5$, $\bar{\omega}_3 = 0{,}2$. Si muestreamos 3 veces la variable W logramos, por ejemplo, la secuencia

$$-2, -2, 0.2...$$

es decir, las partículas después del resampling serán las siguientes

$$x_r^{(1)} \equiv x^{(2)} = -2, \quad x_r^{(2)} \equiv x^{(2)} = -2, \quad x_r^{(3)} \equiv x^{(1)} = 0{,}2.$$

La partícula $x^{(3)} = 7{,}6$ ha desaparecido mientras $x^{(2)}$ aparece 2 veces.

Capítulo 3

Algoritmo Metropolis-Hastings

3.1. Introducción Histórica

En este capítulo trataremos la historia de los comienzos de los métodos de Monte Carlo, explicaremos en que consisten las cadenas de Markov, trataremos en profundidad el algoritmo Metropolis-Hastings y también, los casos específicos de este algoritmo cuando se utiliza con diferentes funciones tentativas.

Los métodos MCMC están basados en la utilización del método de *Monte Carlo* a través de cadenas de Markov. Los métodos de Monte Carlo son métodos estadísticos usados para aproximar relaciones complejas y costosas de evaluar. Este método fue desarrollado en los años cuarenta y su nombre le viene dado por el Casino de Montecarlo, que en aquella época era "la capital del juego de azar".

El iniciador de este método fue Stanislaw Marcin Ulam (matemático polaco que participó en el proyecto Manhattan y que desarrolló herramientas matemáticas para teoría de números, teoría de conjuntos, teoría ergódica y topología algebraica) que tras haber pasado varios días jugando al solitario debido a una enfermedad, observó que era mucho más simple tener una idea del resultado general del juego haciendo pruebas múltiples y contando las proporciones de los resultados, que calcular todas las posibilidades. Esta idea la consiguió extrapolar a su trabajo en el que estaba teniendo problemas para calcular las ecuaciones integro-diferenciales que regían la

difusión de neutrones. Como el lector se habrá percatado, Ulam estaba trabajando en el Proyecto Manhattan, cuya base militar se encontraba en Los Álamos. Este proyecto científico fue dirigido por el físico estadounidense Julius Robert Oppenheimer [25] durante la Segunda Guerra Mundial para el Gobierno de los Estados Unidos y con colaboración de Reino Unido y de Canadá. El objetivo era desarrollar la primera bomba atómica antes que Alemania, *Proyecto Uranio*, y la URS, *Proyecto Borodino* [26]. En las primeras bombas atómicas desarrolladas no se tuvo en cuenta la idea de Ulam. Años más tarde, una vez finalizada la guerra, hubo tiempo de aplicar este método recién salido del cascarón en las investigaciones de física nuclear y física cuántica.

Una vez que Ulam convenció a su colega de trabajo John von Neumann del potencial de este nuevo método, ambos matemáticos trabajaron conjuntamente para el desarrollo de este. El verdadero potencial de este método lo obtuvieron Ulam, Nicholas Constantine Metropolis (matemático, físico y computador científico griego reclutado en el Proyecto Manhattan por Oppenheimer [27]) y Enrico Fermi (físico italiano que realizó contribuciones en mecánica estadística, física cuántica , nuclear y de partículas[18]) al obtener los valores estimados de la ecuación de Schrödinger que describe la evolución temporal de una partícula masiva no relativista, para la captura de neutrones a nivel nuclear [23]. Las integrales de los estimadores que se resolvieron mediante el método de Monte Carlo se pueden encontrar en cualquier libro de física cuántica y nuclear, pero recomiendo el estudio de las ecuaciones [7.12.19], [7.12.20] y [7.12.21] del libro [29].

Aunque la investigación con este método comenzó en los proyectos secretos que llevó a cabo Estados Unidos, la primera publicación donde se citó este método data de 1949 en un artículo de Metropolis y Ulam [30]. La primera simulación de Monte Carlo fue llevada a cabo por un equipo encabezado por Metropolis. Esta simulación se realizó en la computadora ENIAC (considerada la primera computadora electrónica digital) en 1948 en la Universidad de Pennsylvania. Está considerado uno de los 10 algoritmos más importantes desarrollados en el S.XX.

En la literatura existente podemos encontrar ejemplos de la utilización de los métodos de Monte Carlo en multitud de áreas diferentes, como por ejemplo en bi-

ología (Leach en 1996 [67], Karplus y Petsko en 1990 [68], Lawrence, Altschul, Bogusski, Liu, Neuwald y Wootton en 1993 [69]), química (Alder y Wainwright en 1959 [70]), ciencia de la computación (Kirkpatrick, Gelatt y Vecchi en 1983 [71]), economía y finanzas (Gouriérourx y Monfort en 1997[72]), ingeniería (Geman y Geman en 1984 [73]), ciencia de los materiales (Frenkel y Smit en 1996, [65]), física (Metropolis, Rosenbluth, Teller y Teller en 1953 [74], Goodman y Sokal en 1989 [58], Marinari y Parisi en 1992 [53]), estadística (Efron en 1979 [75], Gelfand y Smith en 1990 [76], Tanner y Wong en 1987 [59]), etc.

Importante: nótese que, por razones de sencillez, en este capítulo hemos considerado la notación de variables escalares para la $x \in \mathbb{R}$ pero todos los resultados son aplicables a $\mathbf{x} \in \mathbb{R}^m$.

3.2. Cadenas de Markov

Una cadena de Markov[1] $\{X_t\}_{t=0}^{+\infty}$, $t \in \mathbb{N}$, es un proceso estocástico discreto $\{X_0, X_1, ...\}$, con la propiedad de que la densidad de la variable aleatoria $X_t \in \mathbb{R}$ depende solamente de la variable aleatoria X_{t-1}, es decir, de la variable aleatoria anterior. Dado esto, podemos escribir la probabilidad condicional para cualquier conjunto $\mathcal{A} \subseteq \mathbb{R}$ de la siguiente forma,

$$P\left(X_t \in \mathcal{A} | X_0, X_1, ..., X_{t-1}\right) = P\left(X_t \in \mathcal{A} | X_{t-1}\right). \tag{3.2.1}$$

En esta sección vamos a tratar dos tipos diferentes de cadenas de Markov, las cadenas discretas y las cadenas continuas. Las cadenas discretas tienen un número finito de estados, en cambio, las cadenas continuas tienen un número infinito de estados. En ambos casos, el tiempo lo consideraremos discreto $t \in \mathbb{N}$.

[1] Andrei Andreevitch Márkov (1856-1922), matemático ruso conocido por sus trabajos en la teoría de los números y la teoría de probabilidades. Aunque Márkov influyó sobre diversos campos de las matemáticas, por ejemplo en sus trabajos sobre fracciones continuas, la historia le recordará principalmente por sus resultados relacionados con la teoría de la probabilidad. En 1887 completó la prueba que permitía generalizar el teorema central del límite y que ya había avanzado Chebyshov.

Cadenas discretas

Dada una variable aleatoria discreta X_t que toma valores en un conjunto finito de estados $\mathcal{S} = \{1, 2, ..., m\}$, llamaremos cadena de Markov al proceso estocástico X_t que cumpla

$$p\left(x_t|x_{t-1}, ..., x_1\right) = p\left(x_t|x_{t-1}\right),$$

como ya hemos mencionado anteriormente. Como la cadena de Markov está perfectamente identificada por la probabilidad condicional $p\left(x_t|x_{t-1}\right)$, podemos definir su *kernel* de la siguiente manera,

$$K\left(x_t|x_{t-1}\right) \triangleq p\left(x_t|x_{t-1}\right). \tag{3.2.2}$$

Para cualquier valor de t, el *kernel* tiene que verificar $\sum_{i=1}^{m} K\left(x_t = i|x_{t-1}\right) = 1$. La Figura 3.2.1 representa las probabilidades de transición de una cadena de Markov.

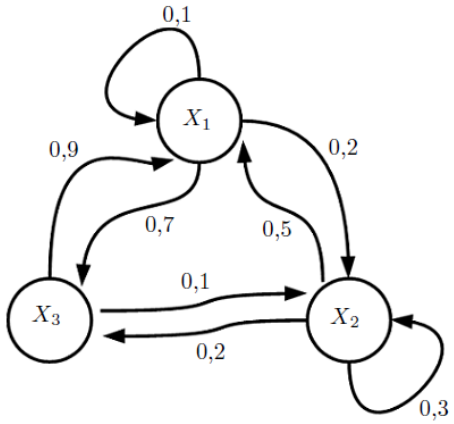

Figura 3.2.1: Como podemos observar, desde cualquier estado, podemos dirigirnos a otro estado con una determinada probabilidad de transición.

podemos definir su matriz de transición como,

$$K = \begin{bmatrix} 0{,}1 & 0{,}2 & 0{,}7 \\ 0{,}5 & 0{,}3 & 0{,}2 \\ 0{,}9 & 0{,}1 & 0 \end{bmatrix}.$$

Cada posición (i, j) de la matriz indica la probabilidad de pasar del estado i al estado j. Cada fila de la matriz suman 1.

Supongamos que el vector de probabilidades para el estado inicial $t = 0$ es $p_0 = \begin{bmatrix} 0{,}3 & 0{,}3 & 0{,}4 \end{bmatrix}$, podemos escribir el vector de probabilidades en $t = 1$ como

$$p_1 = p_0 K = \begin{bmatrix} 0{,}54 & 0{,}19 & 0{,}27 \end{bmatrix}.$$

Siguiendo el mismo proceso para calcular el vector de probabilidades en un tiempo genérico t, obtenemos

$$p_t = p_{t-1} K = (p_{t-2} K)\, K = ... = p_o K^t. \tag{3.2.3}$$

Si $\lim_{t\to\infty} p_t = p_e$, p_e es una distribución invariante o estacionaria. Esto juega un papel fundamental en los algoritmos de *MCMC*. Para cualquier estado inicial, la cadena convergerá a la distribución invariante p_e siempre y cuando la matriz de transición K sea ergódica. Para que una matriz sea ergódica tiene que cumplir dos condiciones:

1. **Irreductible.** Para cualquier estado de la cadena de Markov, existe una probabilidad positiva de visitar todos los demás estados. Esto se traduce en que la matriz de transición K no puede ser separada en matrices más pequeñas, es decir, que el grafo de transiciones es conexo.

2. **Aperiódica.** La cadena no tiene periodos, es decir, no tiene ciclos en los que los estados se repitan de una forma conocida con anterioridad.

Además, la cadena se define homogénea si $K\left(x_t | x_{t-1}\right)$ permanece invariante para todo t.

Cadenas continuas

Como ya hemos mencionado, las cadenas continuas tienen un número infinito de estados y las variables aleatorias pueden tomar valores continuos $X_t \in \mathbb{R}$. Igual que en el caso discreto, se deben cumplir las condiciones de irreductibilidad y aperiodicidad para que la cadena converja a una densidad invariante $p_e\left(x\right)$. Por definición,

una densidad invariante estacionaria para la cadena con probabilidad de transición (kernel) $K(x_t|x_{t-1})$ si se verifica esta ecuación integral,

$$p_e(x_t) = \int_{\mathcal{S}} K(x_t|x_{t-1})\, p_e(x_{t-1})\, dx_{t-1}, \tag{3.2.4}$$

donde $K(x_t|x_{t-1})$ representa la densidad de probabilidad condicional de x_t dado x_{t-1}.

Una condición necesaria para asegurar que una determinada densidad es invariante para la cadena, es la *condición de balance o reversibilidad* que se expresa como

$$p_e(x_{t-1})\, K(x_t|x_{t-1}) = p_e(x_t)\, K(x_{t-1}|x_t)\,. \tag{3.2.5}$$

De hecho, si integramos esta ecuación respecto a x_{t-1},

$$\int_{\mathcal{S}} p_e(x_{t-1})\, K(x_t|x_{t-1})\, dx_{t-1} = \int_{\mathcal{S}} p_e(x_t)\, K(x_{t-1}|x_t)\, dx_{t-1}, \tag{3.2.6}$$

dado que $p({}_e x_t)$ no depende de x_{t-1}, por lo tanto saldría fuera de la integral

$$\int_{\mathcal{S}} p_e(x_{t-1})\, K(x_t|x_{t-1})\, dx_{t-1} = p_e(x_t) \int_{\mathcal{S}} K(x_{t-1}|x_t)\, dx_{t-1}. \tag{3.2.7}$$

Además, $\int_{\mathcal{S}} K(x_{t-1}|x_t)\, d_{x-1} = 1$, por lo tanto llegamos a la siguiente ecuación,

$$\int_{\mathcal{S}} p_e(x_{t-1})\, K(x_t|x_{t-1})\, dx_{t-1} = p_e(x_t)\,, \tag{3.2.8}$$

que es la definición de probabilidad estacionaria invariante. Esto significa que si una $p_e(x)$ cumple la condición de balance, entonces, será una distribución estacionaria para la cadena de Markov.

3.3. Método Metropolis-Hastings

El algoritmo *Metropolis-Hastings* (*MH*) es el algoritmo MCMC más popular y todos los algoritmos MCMC pueden ser interpretados como casos espaciales de esta

técnica. Esta generalización permite que el MH pueda ser aplicado a múltiples dimensiones y/o sus distribuciones condicionales que no se encuentran estandarizadas.

La idea principal de este algoritmo es la creación de una cadena de Markov cuyo *kernel* de transición es $K(x_t|x_{t-1})$ y cuya distribución objetivo invariante es

$$p_e(x) \equiv p_o(x). \tag{3.3.1}$$

Supongamos que $p(x) \propto p_o(x)$, por lo tanto,

$$p(x_t) = \int_{\mathcal{S}} p(x_{t-1}) K(x_t|x_{t-1}) dx_{t-1}. \tag{3.3.2}$$

Además, esto significa que la densidad objetivo deberá cumplir la condición de balance o reversibilidad, es decir,

$$p(x_{t-1}) K(x_t|x_{t-1}) = p(x_t) K(x_{t-1}|x_t).$$

3.3.1. Algoritmo

El algoritmo MH [31] se puede expresar conceptualmente en los siguientes tres pasos:

1. Obtener una muestra x' de la densidad tentativa $\pi(x_t|x_{t-1})$, siendo x_{t-1} la última muestra validada.

2. Calcular el ratio de aceptación

$$\alpha(x_{t-1}, x') = \text{mín}\left[1, \frac{p(x')\pi(x_{t-1}|x')}{p(x_{t-1})\pi(x'|x_{t-1})}\right]. \tag{3.3.3}$$

3. Elegir $x_t = x'$ con una probabilidad $\alpha(x_{t-1}, x')$, y mantener $x_t = x_{t-1}$ con una probabilidad $1 - \alpha(x_{t-1}, x')$. Volver al paso 1) e incrementar $t = t + 1$.

Que a su vez, de manera más detallado los pasos a seguir son estos:

1. A partir de la densidad tentativa $\pi(x_t|x_{t-1})$, se genera un candidato $x' \sim \pi(x_t|x_{t-1})$ de acuerdo a dicha densidad de probabilidad.

2. Se calcula
$$\alpha(x_{t-1}, x') = \text{mín}\left[1, \frac{p(x')\,\pi(x_{t-1}|x')}{p(x_{t-1})\,\pi(x'|x_{t-1})}\right],$$
donde $p(x) \propto p_o(x)$, siendo $p_o(x)$ nuestra densidad objetivo.

3. Se genera una muestra uniforme u en el intervalo $[0,1]$, es decir, $u' \sim \mathcal{U}([0,1])$.

4. El estado siguiente de la cadena de Markov se define como
$$x_t = \begin{cases} x' & \text{si } u' \leq \alpha(x_{t-1}, x') \\ x_{t-1} & \text{si } u' > \alpha(x_{t-1}, x') \end{cases}.$$

Dado el esquema anterior, la probabilidad de transición (kernel) *inducida* por el algoritmo MH resulta ser

$$\begin{aligned} K(x_t|x_{t-1}) = \int_{\mathcal{S}} \Bigg[& \underbrace{\delta(x_t - x')\,\alpha(x_{t-1}, x')\,\pi(x'|x_{t-1})}_{(1)} \\ & + \underbrace{\delta(x_t - x_{t-1})\,(1 - \alpha(x_{t-1}, x'))\,\pi(x'|x_{t-1})}_{(2)} \Bigg] dx', \end{aligned} \tag{3.3.4}$$

donde el primer término (indicado con 1) indica la probabilidad de aceptar la muestra propuesta x', mientras que el segundo término (indicado con 2) se refiere a la probabilidad de rechazar la muestra candidata x'. Para lograr la probabilidad de transición de un x_{t-1} a paso sucesivo x_t, hay que integrar ("sumar todas las probabilidades") respecto a la variable x'.

Desarrollando un poco más la expresión, llegamos a

$$\begin{aligned} K\left(x_t|x_{t-1}\right) = &\int_{\mathcal{S}} \delta\left(x_t - x'\right) \alpha\left(x_{t-1}, x'\right) \pi\left(x'|x_{t-1}\right) dx' \\ &+ \int_{\mathcal{S}} \delta\left(x_t - x_{t-1}\right) \left(1 - \alpha\left(x_{t-1}, x'\right)\right) \pi\left(x'|x_{t-1}\right) dx', \end{aligned} \tag{3.3.5}$$

por definición de delta, la primera integral queda igual a $\alpha\left(x_{t-1}, x_t\right) \pi\left(x_t|x_{t-1}\right)$, y en el segundo término podemos sacar la delta fuera de la integral ya que esta no depende de x',

$$\begin{aligned} K\left(x_t|x_{t-1}\right) =&\alpha\left(x_{t-1}, x_t\right) \pi\left(x_t|x_{t-1}\right) \\ &+ \delta\left(x_t - x_{t-1}\right) \int_{\mathcal{S}} \left(1 - \alpha\left(x_{t-1}, x'\right)\right) \pi\left(x'|x_{t-1}\right) dx', \end{aligned} \tag{3.3.6}$$

y siguiendo con el desarrollo,

$$\begin{aligned} K\left(x_t|x_{t-1}\right) =&\alpha\left(x_{t-1}, x_t\right) \pi\left(x_t|x_{t-1}\right) \\ &+ \delta\left(x_t - x_{t-1}\right) \left[\underbrace{\int_{\mathcal{S}} \pi\left(x'|x_{t-1}\right) dx'}_{=1} - \int_{\mathcal{S}} \alpha\left(x_{t-1}, x'\right) \pi\left(x'|x_{t-1}\right) dx' \right], \end{aligned} \tag{3.3.7}$$

y finalmente llegamos a

$$\begin{aligned} K\left(x_t|x_{t-1}\right) =&\alpha\left(x_{t-1}, x_t\right) \pi\left(x_t|x_{t-1}\right) \\ &+ \delta\left(x_t - x_{t-1}\right) \left[1 - \int_{\mathcal{S}} \alpha\left(x_{t-1}, x'\right) \pi\left(x'|x_{t-1}\right) dx'\right]. \end{aligned} \tag{3.3.8}$$

Este kernel suele expresarse también de la forma siguiente

$$K\left(x_t|x_{t-1}\right) = \pi\left(x_t|x_{t-1}\right) \alpha\left(x_{t-1}, x_t\right) + \delta\left(x_t - x_{t-1}\right) \left(1 - \mathcal{A}\left(x_{t-1}\right)\right), \tag{3.3.9}$$

donde $1 - \mathcal{A}(x_{t-1})$ es la probabilidad total de descartar una muestra genérica candidata y con $\mathcal{A}(x_{t-1})$ indicamos la siguiente integral

$$\mathcal{A}(x_{t-1}) = \int_{\mathcal{S}} \alpha(x_{t-1}, x')\, \pi(x'|x_{t-1})\, dx', \tag{3.3.10}$$

que representa la probabilidad total de aceptar una muestra candidata. Finalmente, otra forma de expresar la probabilidad de transición es

$$K(x_t|x_{t-1}) = \begin{cases} \pi(x_t - x_{t-1})\, \alpha(x_{t-1}, x_t) & \text{si } x_{t-1} \neq x_t \\ \delta(x_t - x_{t-1})\, (1 - \mathcal{A}(x_{t-1})) & \text{si } x_{t-1} = x_t \end{cases} . \tag{3.3.11}$$

3.3.2. Condición de reversibilidad con el kernel del MH

Ahora, vamos a demostrar que el kernel del algoritmo MH

$$K(x_t|x_{t-1}) = \begin{cases} \pi(x_t - x_{t-1})\, \alpha(x_{t-1}, x_t) & \text{si } x_{t-1} \neq x_t \\ \delta(x_t - x_{t-1})\, (1 - \mathcal{A}(x_{t-1})) & \text{si } x_{t-1} = x_t \end{cases} ,$$

respeta la ecuación de balance,

$$p(x_{t-1})\, K(x_t|x_{t-1}) = p(x_t)\, K(x_{t-1}|x_t) .$$

Demostración. Vamos a distinguir tres casos.

1. **Caso $\mathbf{x_t = x_{t-1}}$:** Dada la Ecuación (3.3.11) el kernel queda como

   $$K(x_t|x_{t-1}) = \delta(x_t - x_{t-1})\, (1 - \mathcal{A}(x_{t-1})) .$$

   por lo tanto, se cumple la condición de balance

   $$p_o(x_{t-1})\, \delta(x_t - x_{t-1})\, (1 - \mathcal{A}(x_{t-1})) = p_o(x_t)\, \delta(x_{t-1} - x_t)\, (1 - \mathcal{A}(x_t)) ,$$

   porque $\delta(x_t - x_{t-1}) = \delta(x_{t-1} - x_t)$ por definición de función delta, $\mathcal{A}(x_{t-1}) = \mathcal{A}(x_t)$ porque $x_t = x_{t-1}$, y así mismo ocurre con $p_o(x_{t-1}) = p_o(x_t)$.

2. **Caso $\mathbf{x_t \neq x_{t-1}}$ y $\boldsymbol{\pi(x_{t-1}|x_t)\, p_o(x_t) < \pi(x_t|x_{t-1})\, p_o(x_{t-1})}$:** Dada la definición en al Ecuación (3.3.3) las probabilidades $\alpha(x_{t-1}, x_t)$ y $\alpha(x_t, x_{t-1})$ de aceptación quedan como

$$\alpha(x_{t-1}, x_t) = \frac{\pi(x_{t-1}|x_t)\, p_o(x_t)}{\pi(x_t|x_{t-1})\, p_o(x_{t-1})}$$

y

$$\alpha(x_t, x_{t-1}) = 1.$$

Como nos encontramos en el caso de $x_t \neq x_{t-1}$, el kernel es $K(x_t|x_{t-1}) = \pi(x_t|x_{t-1})\, \alpha(x_{t-1}, x_t)$, por lo tanto la condición de balance debe cumplir que,

$$\pi(x_t|x_{t-1})\, \alpha(x_{t-1}, x_t)\, p_o(x_{t-1}) = \pi(x_{t-1}|x_t)\, \alpha(x_t, x_{t-1})\, p_o(x_t),$$

sustituyendo en esta ecuación las probabilidades de aceptación de arriba, tenemos que,

$$\pi(x_t|x_{t-1}) \frac{\pi(x_{t-1}|x_t)\, p_o(x_t)}{\pi(x_t|x_{t-1})\, p_o(x_{t-1})} p_o(x_{t-1}) = \pi(x_{t-1}|x_t)\, p_o(x_t) \cdot 1,$$

simplificando obtenemos

$$\pi(x_{t-1}|x_t)\, p_o(x_t) = \pi(x_{t-1}|x_t)\, p_o(x_t).$$

Como podemos observar, el kernel del algoritmo cumple también en este caso la condición de balance.

3. **Caso $\mathbf{x_t \neq x_{t-1}}$ y $\boldsymbol{\pi(x_{t-1}|x_t)\, p_o(x_t) \geq \pi(x_t|x_{t-1})\, p_o(x_{t-1})}$:** En este caso, tenemos que las probabilidades de aceptación son

$$\alpha(x_{t-1}, x_t) = 1,$$

y

$$\alpha(x_t, x_{t-1}) = \frac{\pi(x_t|x_{t-1})\, p_o(x_{t-1})}{\pi(x_{t-1}|x_t)\, p_o(x_t)}.$$

Sustituyendo las expresiones en la condición de balance, obtenemos que,

$$\pi\left(x_t|x_{t-1}\right)\alpha\left(x_{t-1},x_t\right)p_o\left(x_{t-1}\right)=\pi\left(x_{t-1}|x_t\right)\alpha\left(x_t,x_{t-1}\right)p_o\left(x_t\right),$$

llegando a lo siguiente,

$$\pi\left(x_t|x_{t-1}\right)p_o\left(x_{t-1}\right)\cdot 1=\pi\left(x_{t-1}x_t\right)\frac{\pi\left(x_t|x_{t-1}\right)p_o\left(x_{t-1}\right)}{\pi\left(x_{t-1}|x_t\right)p_o\left(x_t\right)}p_o\left(x_t\right),$$

y simplificando obtenemos,

$$\pi\left(x_t|x_{t-1}\right)p_o\left(x_{t-1}\right)=\pi\left(x_t|x_{t-1}\right)p_o\left(x_{t-1}\right). \tag{3.3.12}$$

Como podemos observar, el kernel del algoritmo cumple la condición de balance también en este caso. Entonces, el kernel verifica siempre la condición de reversibilidad. □

En realidad para el caso de $x_t \neq x_{t-1}$, la demostración puede resumirse bastante. Esto es porque en este caso para $x_t \neq x_{t-1}$ el *kernel* del MH puede escribirse de esta forma

$$K\left(x_t|x_{t-1}\right)=\pi\left(x_t|x_{t-1}\right)\text{mín}\left[1,\frac{p\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}\right], \tag{3.3.13}$$

y multiplicando por $p\left(x_{t-1}\right)$

$$p\left(x_{t-1}\right)K\left(x_t|x_{t-1}\right)=p\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)\text{mín}\left[1,\frac{p\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}\right], \tag{3.3.14}$$

$$p\left(x_{t-1}\right)K\left(x_t|x_{t-1}\right)=\text{mín}\left[p\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right),p\left(x_t\right)\pi\left(x_{t-1}|x_t\right)\right]=p\left(x_t\right)K\left(x_{t-1}|x_t\right), \tag{3.3.15}$$

donde se ve perfectamente que podemos intercambiar las variables x_t con x_{t-1}, es decir, se cumple la condición de balance o reversibilidad.

Las prestaciones del algoritmo MH consiste en una adecuada elección de dos componentes que ponen el kernel $K\left(x_t|x_{t-1}\right)$:

1. la densidad tentativa $\pi\left(\cdot|\cdot\right)$,
2. la función de aceptación $\alpha\left(\cdot|\cdot\right)$.

La elección de una densidad tentativa adecuada puede ser crucial para el éxito del algoritmo MH. En la Sección 3.4 veremos unos casos específicos y muy tratados en la literatura existente. La función de aceptación $\alpha(\cdot|\cdot)$ es vital dado que no solo permite el cumplimiento de la condición de balance por parte de la probabilidad de transición $K(x_t|x_{t-1})$, también puede afectar a las prestaciones del algoritmo. Como veremos a continuación, la función de $\alpha(\cdot|\cdot)$ en la Ecuación (3.3.3) no es la única posible.

3.3.3. Funciones de aceptación

La función de aceptación dada por el algoritmo Metropolis-Hastings no es la única posible. De hecho, diversos autores han diseñado funciones de aceptación

$$\alpha(x_{t-1}, x_t) = \{(x_{t-1}, x_t) \in \mathcal{S} \times \mathcal{S} \to [0, 1]\} \tag{3.3.16}$$

que cumplen la condición de balance. A continuación vamos a ver a unos ejemplos.

Función de aceptación del algoritmo Metropolis-Hastings

La probabilidad de aceptación *tradicional* del algoritmo Metropolis-Hastings es la siguiente

$$\alpha_{MH}(x_{t-1}, x_t) = \text{mín}\left[1, \frac{\pi(x_{t-1}|x_t)\, p_o(x_t)}{\pi(x_t|x_{t-1})\, p_o(x_{t-1})}\right]. \tag{3.3.17}$$

El subíndice indica los autores, en este caso MH de Metropolis-Hastings.

Función de aceptación propuesta por Metropolis

El algoritmo inicial diseñado por Metropolis (que veremos más adelante) suponía una función tentativa simétrica como ya hemos indicado, por lo tanto,

$$\alpha_{M}(x_{t-1}, x_t) = \text{mín}\left[1, \frac{p_o(x_t)}{p_o(x_{t-1})}\right]. \tag{3.3.18}$$

Función de aceptación propuesta por Barker

Otra función de aceptación posible fue propuesta por Barker en 1965 [32], y tiene la forma

$$\alpha_B\left(x_{t-1}, x_t\right) = \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right) + p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}. \tag{3.3.19}$$

Si la fdp tentativa es simétrica, la función de aceptación queda como

$$\alpha_{B2}\left(x_{t-1}, x_t\right) = \frac{p_o\left(x_t\right)}{p_o\left(x_t\right) + p_o\left(x_{t-1}\right)}. \tag{3.3.20}$$

Función de aceptación propuesta por Hastings

Hastings logró una fórmula que engloba ambas probabilidades de aceptación descritas anteriormente,

$$\alpha_{H2}\left(x_{t-1}, x_t\right) = \frac{c\left(x_{t-1}, x_t\right)}{1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}}, \tag{3.3.21}$$

donde la función $c\left(x_{t-1}, x_t\right)$ tiene que ser:

- simétrica, $c\left(x_t, x_{t-1}\right) = c\left(x_{t-1}, x_t\right)$,
- no negativa, $c\left(x_{t-1}, x_t\right) \geq 0$,
- y tiene que cumplir que $c\left(x_{t-1}, x_t\right) \leq 1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}$, para que $0 \leq \alpha\left(x_{t-1}, x_t\right) \leq 1$.

La función $\alpha_{H2}\left(x_{t-1}, x_t\right)$ engloba a la original del algoritmo Metropolis-Hastings y a la propuesta por Barker. De hecho:

- Si $c\left(x_{t-1}, x_t\right) = 1$, entonces

$$\begin{aligned}\alpha_{H2}\left(x_{t-1}, x_t\right) &= \frac{1}{1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}} \\ &= \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right) + p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)} = \alpha_B\left(x_{t-1}, x_t\right),\end{aligned}$$

que es la función de aceptación de Barker.

- Si $c\left(x_{t-1}, x_t\right) = \text{mín}\left[1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}, 1 + \frac{p_o(x_t)\pi(x_{t-1}|x_t)}{p_o(x_{t-1})\pi(x_t|x_{t-1})}\right]$, tenemos

$$\alpha_{H2}\left(x_t, x_{t-1}\right) = \frac{\text{mín}\left[1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}, 1 + \frac{p_o(x_t)\pi(x_{t-1}|x_t)}{p_o(x_{t-1})\pi(x_t|x_{t-1})}\right]}{1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}},$$

que podemos desglosar en dos partes:

1. si $p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right) < p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)$, entonces

$$\alpha_{H2}\left(x_{t-1}, x_t\right) = \frac{1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}}{1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}} = 1,$$

2. mientras, si $p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right) \geq p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)$, tenemos

$$\begin{aligned}
\alpha_{H2}\left(x_{t-1}, x_t\right) &= \frac{1 + \frac{p_o(x_t)\pi(x_{t-1}|x_t)}{p_o(x_{t-1})\pi(x_t|x_{t-1})}}{1 + \frac{p_o(x_{t-1})\pi(x_t|x_{t-1})}{p_o(x_t)\pi(x_{t-1}|x_t)}} \\
&= \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)} \cdot \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right) + p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right) + p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)} \\
&= \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}.
\end{aligned}$$

Por lo tanto, podemos expresar la función $\alpha_{H2}\left(x_{t-1}, x_t\right)$ como

$$\alpha_{H2}\left(x_{t-1}, x_t\right) = \text{mín}\left[1, \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}\right] = \alpha_{MH}\left(x_{t-1}, x_t\right), \tag{3.3.22}$$

es decir, la función de aceptación del Metropolis-Hastings.

Funciones de aceptación propuestas por Charles Stein

Otra de las opciones que podemos encontrar en la literatura es

$$\alpha_{CS1}\left(x_{t-1}, x_t\right) = \frac{c\left(x_{t-1}, x_t\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}, \tag{3.3.23}$$

que fue propuesta por Charles Stein (matemático y estadístico americano autor de la paradoja Stein en teoría de la decisión y estimación [33, 34]) en una comunicación privada, y donde $c(x, y)$ es una función simétrica con las características descritas anteriormente. Stein dio también otra opción

$$\alpha_{CS2}(x_{t-1}, x_t) = \frac{p_o(x_t)\, c(x_{t-1}, x_t)}{\pi(x_t|x_{t-1})}. \tag{3.3.24}$$

Función de aceptación genérica

Podemos hallar otras funciones que cumplan las condición de reversibilidad [60]. Para ello, consideremos la función

$$R(x_{t-1}, x_t) \triangleq \frac{p_o(x_t)\, \pi(x_{t-1}|x_t)}{p_o(x_{t-1})\, \pi(x_t|x_{t-1})}, \tag{3.3.25}$$

y una transformación genérica $F(z) : [0, +\infty) \to [0, 1]$ que verifica la condición

$$F(z) = z \cdot F\left(\frac{1}{z}\right). \tag{3.3.26}$$

Así que, una función de aceptación genérica $\alpha_g(x_{t-1}, x_t)$ puede ser definida como

$$\alpha_g(x_{t-1}, x_t) \triangleq (F \circ R)(x_{t-1}, x_t) = F(R(x_{t-1}, x_t)). \tag{3.3.27}$$

Esta última ecuación vista, representa a una clase de funciones de aceptación, es decir, no todas las posibles funciones de aceptación tienen que poder expresarse de esta forma, pero engloba varias. Por ejemplo, una de las funciones de aceptación que se engloban dentro de la clase de $\alpha_g(x_{t-1}, x_t)$ es la función de aceptación que propuso Barker se puede expresar eligiendo $F(z) = \frac{z}{1+z}$, mientras que utilizando $F(z) = \text{mín}[1, z]$ calculamos la probabilidad de aceptación del algoritmo Metropolis-Hastings estándar.

Es interesante observar que para estados discretos, Peskun [35] demostró que la función de aceptación original del algoritmo MH resulta ser la elección óptima en términos de eficiencia estadística.

3.4. Casos específicos

3.4.1. Caso trivial: fdp tentativa igual a la fdp objetivo

Supongamos que, por absurdo, que podemos elegir una función tentativa igual a la densidad objetivo que queremos muestrear, es decir,

$$\pi\left(x_t|x_{t-1}\right) = p_o\left(x_t\right). \tag{3.4.1}$$

Se trata claramente de un planteamiento absurdo porque si sabemos generar números aleatorio desde la densidad objetivo, no necesitamos utilizar ninguna otra técnica. Pero queremos comprobar que el método MH es coherente. De hecho, en este caso la función de aceptación resulta ser

$$\begin{aligned} \alpha\left(x_{t-1}, x_t\right) &= \min\left[1, \frac{p_o\left(x_t\right)\pi\left(x_{t-1}|x_t\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t|x_{t-1}\right)}\right] \\ &= \min\left[1, \frac{p_o\left(x_t\right)p_o\left(x_{t-1}\right)}{p_o\left(x_{t-1}\right)p_o\left(x_t\right)}\right] = 1. \end{aligned} \tag{3.4.2}$$

Es decir: la función de aceptación en este caso es siempre igual a 1. Todas las muestras generadas serán aceptadas, como debería ser dado que provienen de la densidad objetivo $p_o\left(x\right)$. La función de aceptación del MH intenta corregir las "discrepancias" entre la fdp tentativa y la fdp objetivo, que en este caso son nulas.

3.4.2. Algoritmo de Metropolis: función tentativa simétrica

El primer algoritmo fue diseñado por Metropolis en 1953, como ya hemos mencionado. Este algoritmo está basado en fdp tentativa que cumplen la condición de simetría,

$$\pi\left(x_t|x_{t-1}\right) = \pi\left(x_{t-1}|x_t\right), \tag{3.4.3}$$

siendo x' la muestra propuesta en el estado t. La do única diferencia que existe entre este algoritmo y el algoritmo MH, es la función de aceptación, que como veremos en este algoritmo es más simple.

Como ya hemos visto la función de aceptación del MH es

$$\begin{aligned} \alpha\left(x_{t-1}, x_t\right) &= \text{mín}\left[1, \frac{\pi\left(x_{t-1}|x_t\right) p_o\left(x_t\right)}{\pi\left(x_t|x_{t-1}\right) p_o\left(x_{t-1}\right)}\right] \\ &= \text{mín}\left[1, \frac{p_o\left(x_t\right)}{p_o\left(x_{t-1}\right)}\right], \end{aligned} \tag{3.4.4}$$

donde hemos aplicado la condición de simetría de la fdp tentativa. Es importante notar que

- si $p_o\left(x_t\right) \geq p_o\left(x_{t-1}\right)$, entonces $\alpha\left(x_t, x_{t-1}\right) = 1$,
- mientras que si $p_o\left(x_t\right) < p_o\left(x_{t-1}\right)$, se tiene $\alpha\left(x_{t-1}, x_t\right) = \frac{p_o(x_t)}{p_o(x_{t-1})} < 1$.

Es decir, si el candidato x' se evalúa en la función objetivo $p_o\left(x\right)$ y tiene un valor mayor o igual que la muestra anteriormente aceptada x_{t-1} evaluada en la función objetivo, la muestra se acepta. En cambio, si $p_o\left(x_t\right) < p_o\left(x_{t-1}\right)$, la muestra x' se aceptará con probabilidad $\frac{p_o(x_t)}{p_o(x_{t-1})}$. En la Figura 3.4.1 podemos ver claramente el funcionamiento del algoritmo Metropolis.

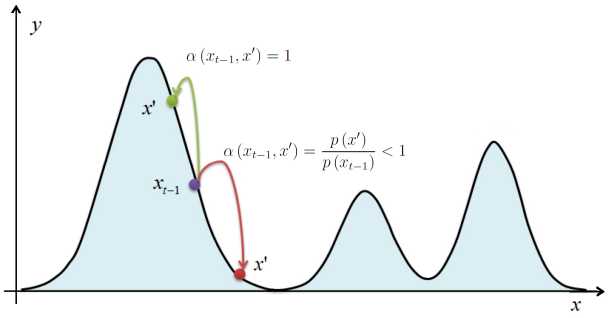

Figura 3.4.1: Algoritmo Metropolis. Las muestras generadas por la función tentativa que se dirigen hacia "arriba" en la función objetivo, siempre se aceptan. En cambio, las muestras generadas que se dirigen hacia "abajo" se aceptan con una probabilidad $\frac{p_o(x_t)}{p_o(x_{t-1})}$.

Claramente, esta propiedad es muy interesante desde el punto de vista de la optimización, dado que, los movimientos hacia "arriba" se aceptan siempre, mientras que los movimientos hacia "abajo" se aceptan con cierta probabilidad.

3.4.3. Función tentativa independiente

Un caso particular del algoritmo MH muy interesante, es cuando la densidad tentativa es independiente del estado anterior de la cadena de Markov. Es decir, consideramos que

$$\pi\left(x_t|x_{t-1}\right) = \pi\left(x_t\right), \tag{3.4.5}$$

entonces la función de probabilidad de aceptación es la siguiente

$$\alpha\left(x_{t-1}, x_t\right) = \text{mín}\left[1, \frac{p_o\left(x_t\right)\pi\left(x_{t-1}\right)}{p_o\left(x_{t-1}\right)\pi\left(x_t\right)}\right] = \text{mín}\left[1, \frac{w\left(x_t\right)}{w\left(x_{t-1}\right)}\right], \tag{3.4.6}$$

donde hemos definido los pesos

$$w\left(x\right) \triangleq \frac{p_o\left(x\right)}{\pi\left(x\right)}.$$

Esta situación es particularmente interesante porque puede compararse con el método de *aceptación/rechazo* y con *importance sampling* vistos en el Capítulo 2. De hecho, podemos observar que la definición de pesos que hemos aportado para la función tentativa independiente, es la misma que la definición de pesos que se dieron en importance sampling. La diferencia que aquí se añade, es un test de aceptación que depende del peso actual y del anterior. En este test, siempre aceptaremos la nueva muestra si su peso es mayor que el peso de la muestra anterior. Además, todas las muestras de la cadena producida por el MH tendrán el "mismo peso", es decir, todas las muestras son igual de "buenas" y se distribuyen según $p_o\left(x\right)$, sin contar, evidentemente, con las muestras producidas en el transitorio.

Por otra parte, la gran diferencia existente entre el método de aceptación/rechazo y el algoritmo MH con función tentativa independiente, es que en el método de aceptación/rechazo es necesario conocer una constante adecuada M para que $p\left(x\right) \leq M\pi\left(x\right)$, mientras que el algoritmo MH se puede aplicar siempre. En la Figura 3.4.2 podemos observar gráficamente las diferencias entre el método de aceptación/rechazo y el algoritmo MH, respecto a la utilización de la función tentativa y la función objetivo. Además de todo esto, hay que tener en cuenta que el método de aceptación/rechazo

se descartan muchas muestras, pero estas son independientes (si la función tentativa genera muestras independientes), en cambio, en el algoritmo MH, una vez pasado el tiempo de transición, se aceptan todas las muestras generadas, pero estas muestras se encuentran correlacionadas y a veces incluso repetidas.

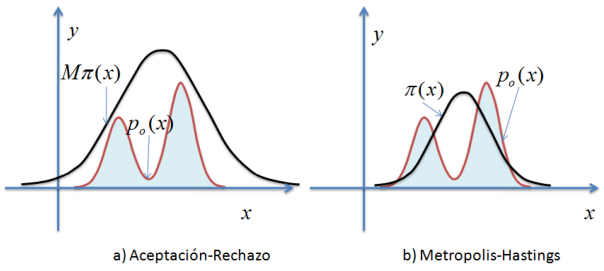

Figura 3.4.2: Comparativa entre el método de aceptación/rechazo y el algoritmo MH. **(a)** El método de aceptación/rechazo puede aplicarse solo cuando conocemos una constante M que cumpla $p(x) \leq M\pi(x)$. **(b)** El algoritmo MH no precisa que se cumpla ninguna desigualdad entre $\pi(x)$ y $p(x) \propto p_o(x)$.

3.4.4. Función tentativa como camino aleatorio

Considerando X_t como una variable aleatoria propuesta del proceso estocástico asociado al algoritmo MH en el instante t, podemos expresar que

$$X_t \sim \pi(x|x_{t-1}). \tag{3.4.7}$$

De forma equivalente podemos expresar lo siguiente,

$$X_t = X_{t-1} + E \tag{3.4.8}$$

siendo E una variable aleatoria con densidad genérica $q(\epsilon)$. La ecuación $X_t = X_{t-1} + E$ define un camino aleatorio (en inglés, *random walk*). Cuando esto ocurre, se suele denotar la densidad tentativa como

$$\pi(x_t|x_{t-1}) = \pi(x_t - x_{t-1}). \tag{3.4.9}$$

Esto se realiza para remarcar que en cada instante de tiempo la densidad tentativa no cambia en su forma sino que se traslada según la muestra anterior x_{t-1} de la cadena de Markov, como mostramos en la Figura 3.4.3. Por ejemplo, el caso típico es utilizar la $q(\epsilon)$ Gaussiana estándar y en este caso tenemos una proposal del tipo

$$\pi(x_t|x_{t-1}) = \pi(x_t - x_{t-1}) = \exp\left(-\frac{(x_t - x_{t-1})^2}{2}\right). \tag{3.4.10}$$

Nótese que el camino aleatorio es una elección especial de fdp tentativa, porque en general, en cada instante de tiempo la densidad tentativa podría también cambiar su forma (como cambiar su varianza, por ejemplo, $\pi(x_t|x_{t-1}) \propto \exp\left(-\frac{(x_t-k)^2}{2x_{t-1}}\right)$).

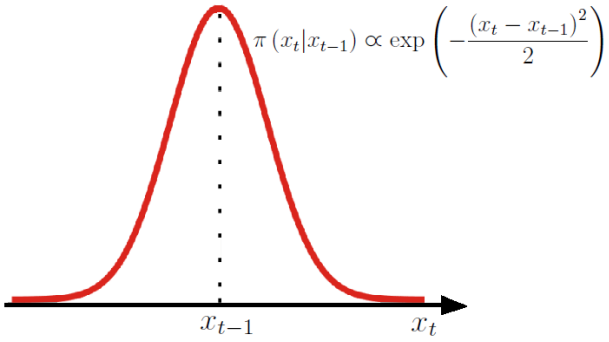

Figura 3.4.3: Función tentativa Gaussiana como camino aleatorio. Esta función tentativa Gaussiana, se encuentra centrada en x_{t-1}.

Parte II

Técnicas avanzadas de muestreo

Nuestros primeros esfuerzos son puramente instintivos,
de una imaginación vívida e indisciplinada.
(Nikola Tesla)

Capítulo 4

Introducción

4.1. Limitaciones del algoritmo MH

En teoría, el método MH puede aplicarse a cualquier distribución objetivo. Pero, en la práctica no siempre se logran resultados aceptables. Obstante, la generalización de Hastings en 1970 que permite la utilización de funciones *proposals* asimétricas, en ciertos casos la convergencia puede resultar muy lenta. En general, los problemas principales en las prestaciones del algoritmo MH son los siguientes:

1. Debida a la correlación entre las muestras, la cadena puede quedarse enganchada alrededor de una moda de la densidad objetivo. Más en general, la cadena puede resultar atrapada en un subconjunto del dominio de la variable de interés. Esto claramente, ralentiza la convergencia a la densidad objetivo. Este fenómeno resulta típico en los problemas descritos por densidades con modas "estrechas".

2. El segundo problema principal aparece cuando la densidad objetivo esta compuesta por un factor expresado con integrales analíticamente no tratables. Concretamente, el problema resulta ser la incapacidad para evaluar dicho factor. Supongamos, por ejemplo, que la densidad objetivo tiene una forma del tipo $p_o(x) \propto c(x)\,\psi(x)$, donde $c(x)$ se expresa con una integral intratable. Evidentemente, el algoritmo MH no puede ser aplicado dado que implicaría

evaluar una relación desconocida $\frac{c(x_t)}{c(x_{t-1})}$, donde x_t denota la muestra propuesta y x_{t-1} la muestra anterior. Esta dificultad se encuentra naturalmente en la interferencia bayesiana para la mayoría de los modelos estadísticos, como modelos estadísticos espaciales, generalización de modelos mixtos lineales y modelos exponenciales de grafos aleatorios [4].

4.2. Estrategias avanzadas MCMC

Para solventar estas dos dificultades principales, se han propuesto diferentes variaciones al algoritmo MH tradicional. Unos ejemplos de estas distintas estrategias son los siguientes:

- los métodos basados en *variables auxiliares* (como la muy conocida variable *temperatura*) [4, Capítulo 4],
- los métodos basados en *pesos de importancia* (siguiendo en un cierto sentido la estrategia de *importance sampling*) [4, Capítulo 6][5, Capítulo 2],
- los métodos basados en *densidad tentativa adaptativa* (estas técnicas intentan mejorar la fdp tentativa "online" aprendiendo de las muestras anteriores)[4, Capítulo 8],
- los métodos *multi-punto* [4, Capítulo 5],
- los métodos *basados en población* [4, Capítulo 5],
- etc.

Muchas técnicas utilizan estrategias de diferentes clases, así que pueden considerarse incluidas en diferentes categorías. Por ejemplo, existen métodos que utilizan variables auxiliares y al mismo tiempo estrategias de población.

4.2.1. Objetivos de este proyecto

En este trabajo intentaremos describir técnicas que intentan resolver sobre todo el primer problema descrito anteriormente ("cadenas atrapadas"). Como consecuencia, también estos métodos disminuyen la correlación de estas muestras y aceleran la convergencia de las cadenas de Markov generada por el algoritmo MH a la densidad objetivo.

Para lograrlo, de las estrategias anteriormente vistas, nos concentraremos en los métodos *multi-punto* y los *basados en población*. Estas dos categorías están conceptualmente conectadas entre ellas, y pueden considerarse como extensiones de una simple estrategia para acelerar la convergencia:

- utilizar diferentes algoritmos MH en paralelo.

A continuación, presentaremos esta simple idea.

4.3. Algoritmos MH en paralelo

La idea más simple para acelerar la convergencia es la utilización de varias cadenas de Markov independientes entre sí. La utilización de varias cadenas de Markov mejoran claramente la convergencia del algoritmo, dado que permite explorar todo el espacio de la variable de interés $\mathcal{S}$ más rápidamente.

Además, inicializando cada cadena en distintas sub-regiones del espacio $\mathcal{S}$ se evita con alta probabilidad que todas las cadenas queden atrapadas en un mismo sub-conjunto de $\mathcal{S}$. En este esquema trivial, aunque las muestras de cada cadena de Markov estén correlacionadas entre sí, las muestras entre las diferentes cadenas no lo están. En la Figura 4.3.1 podemos ver gráficamente esta idea.

Como podemos ver en la Figura 4.3.1, ejecutamos k algoritmos Metropolis-Hastings independientes entre sí, obteniendo muestras $x_t^{(1)}, x_t^{(2)}, ..., x_t^{(k)}$. Las muestras generadas por cada cadena de Markov se distribuyen conforme a la densidad objetivo, por lo tanto, en cada instante de tiempo t, logramos k muestras, $x_t^{(1)}, x_t^{(2)}, ..., x_t^{(k)}$,

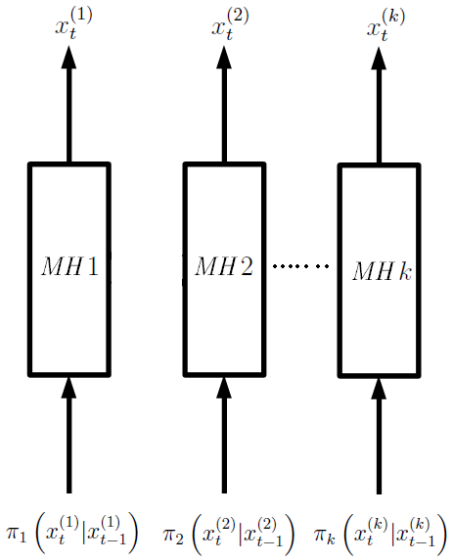

Figura 4.3.1: Ejecución de k cadenas de Markov independientes entre sí.

que se distribuyen conforme a la densidad objetivo. Claramente, cada algoritmo MH puede utilizar diferentes fdp tentativas $\pi_j\left(x_t^{(j)}|x_{t-1}^{(j)}\right)$, $j = 1, ..., k$, o la misma función tentativa.

Evidentemente, el principal problema de hacer correr k Metropolis-Hastings en paralelo, es el coste computacional. Como resulta evidente, el coste computacional es k veces el coste de un algoritmo Metropolis-Hastings. Por esta razón, se han estudiado mejoras a esta simple estrategia.

En la literatura existente, se ha planteado un esquema más complejo: la utilización de varias cadenas de Markov que se *comuniquen entre sí*. En la Figura 4.3.2 podemos ver gráficamente la idea de comunicación de información entre las cadenas de Markov.

Los métodos multi-punto y los métodos basados en población pueden considerarse dentro de este esquema (algoritmos MH en paralelo con comunicación de información). En los siguientes capítulos presentaremos diferentes ejemplos de técnicas pertenecientes a estas dos metodologías.

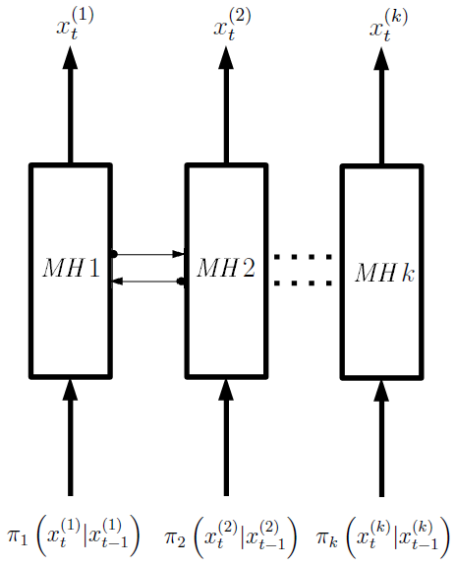

Figura 4.3.2: Ejecución de k cadenas de Markov dependientes entre sí. Estas cadenas se comunican información para mejorar la convergencia del algoritmo.

Capítulo 5

Métodos multi-puntos

Como ya hemos mencionado varias veces, el algoritmo MH puede ser aplicado para casi cualquier tipo de densidad objetivo. En la práctica, de todas formas es frecuente descubrir que encontrar una "buena" densidad tentativa es bastante difícil. Así que, la opción más utilizada resulta ser la de *random walk*. Pero esto, a parte de los problemas descritos anteriormente en el Capítulo 4, suele favorecer transiciones con "paso pequeño" mientras que los "saltos grandes" suelen tener muy baja probabilidad de aceptación media.

A continuación vamos a describir una generalización de la regla de transición MH [42], que permite al algoritmo MCMC realizar "saltos" más largos sin bajar la probabilidad de aceptación. Claramente, el coste computacional será más alto respecto al de un MH clásico, pero menor respecto a una serie de algoritmo MH independientes en paralelo.

5.1. Multiple-Try Metropolis (MTM)

En este capítulo, trataremos una metodología avanzada MCMC conocida como *algoritmo Multiple-Try Metropolis* (*MTM*)[38, 39, 42, 4], así como sus casos específicos y posibles generalizaciones [37].

Para facilitar la convergencia de la cadena de Markov generada por MH, podría

ser útil disponer de la información sobre el gradiente (o más en general sobre la "forma") de la función objetivo. En la práctica, es muy complicado o imposible obtener información analítica sobre el gradiente, por esta razón se opta también por la aproximación mediante muestras aleatorias. Una posible y muy interesante estrategia es el algoritmo MTM, que utiliza la *filosofía* del *importance sampling* dentro de un algoritmo MH. Nótese también que el significado de los pesos y la definición de los mismos es distinta que en el importance sampling.

Recordamos que siempre indicamos con $p(x)$ una función proporcional a la densidad objetivo $p_o(x)$, y con $\pi(\cdot|\cdot)$ la fdp tentativa. Además, definimos con $\lambda(x,y)$ una función no negativa y simétrica , es decir

- $\lambda(x,y) \geq 0$, $\forall x$,y,
- $\lambda(x,y) = \lambda(y,x)$, $\forall x, y$.

Dados estos elementos, podemos introducir los siguientes *pesos*

$$w(x,y) \triangleq p(x)\,\pi(y|x)\,\lambda(x,y). \tag{5.1.1}$$

Por sencillez, el lector puede considerar directamente $\lambda(x,y) = 1$ dado que esta función realmente no afecta en absoluto a la condición de balance. Pero aquí nosotros seguimos un desarrollo más genérico. Esto de todas formas no tiene que despistar ya que la función $\lambda(x,y)$ no tiene mucha importancia para la comprensión de la idea básica del algoritmo.

El método MTM consiste en generar no una sino más muestras desde la densidad tentativa $\pi(\cdot|\cdot)$. Desde estas realizaciones se generará una aproximación de la medida de probabilidad definida por la densidad objetivo, mediante *pesos* siguiendo la idea del importance sampling. Dada esta aproximación, se elige una muestra de acuerdo a los pesos calculados y se procede con otros pasos adecuados para que se cumplan la *condición de balance o reversibilidad.*

Específicamente, el algoritmo MTM consiste en los siguientes pasos:

1. Elegir aleatoriamente un punto x_0 e inicializar $t = 1$.

2. Generar k muestras aleatorias $y^{(i)}$, $i = 1, ..., k$, a partir de la densidad $\pi\left(y|x_{t-1}\right)$, obteniendo un vector $\left[y^{(1)}, ..., y^{(k)}\right]$. Estas realizaciones serán las posibles candidatas a convertirse en la siguiente muestra x_t de la cadena de Markov.

3. Calcular los pesos

$$w\left(y^{(i)}, x_{t-1}\right) = p\left(y^{(i)}\right) \pi\left(x_{t-1}|y^{(i)}\right) \lambda\left(y^{(i)}, x_{t-1}\right),$$

asociados a las muestras $y^{(i)}$, $i = 1, ..., k$.

4. Normalizar los pesos

$$\bar{w}^{(i)} \triangleq \frac{w\left(y^{(i)}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(y^{(i)}, x_{t-1}\right)},$$

con $i = 1, ..., k$.

5. Elegir aleatoriamente una muestra $y^{(j)}$, con $j \in \{1, ..., k\}$, de acuerdo a los pesos $\bar{w}^{(i)}$, $i = 1, ..., k$.

6. Generar $k-1$ muestras aleatorias $x^{(r)}$, $r = 1, ..., k-1$, a partir de la densidad $\pi\left(x|y^{(j)}\right)$, logrando el vector $\left[x^{(1)}, ..., x^{(k-1)}\right]$. A continuación, en este capítulo, a veces nos referiremos a estas realizaciones como "*muestras de conjunto de referencia*".

7. Construir el vector $\left[x^{(1)}, ..., x^{(k-1)}, x^{(k)}\right]$, donde $x^{(k)} = x_{t-1}$.

8. Calcular los pesos asociados

$$w\left(x^{(r)}, y^{(j)}\right) = p\left(x^{(r)}\right) \pi\left(y^{(j)}|x^{(r)}\right) \lambda\left(x^{(r)}, y^{(j)}\right),$$

con $r = 1, ..., k$.

9. Calcular la probabilidad de aceptación de la muestra $y^{(j)}$ mediante (recordemos que $x^{(k)} = x_{t-1}$)

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{w\left(y^{(1)}, x_{t-1}\right) + ... + w\left(y^{(k)}, x_{t-1}\right)}{w\left(x^{(1)}, y^{(j)}\right) + ... + w\left(x^{(k)}, y^{(j)}\right)}\right]. \tag{5.1.2}$$

10. Aceptar la muestra $x_t = y^{(j)}$, con probabilidad $\alpha\left(x_{t-1}, y^{(j)}\right)$ sino $x_t = x_{t-1}$.

11. Actualizar $t = t + 1$ y volver al paso 2.

Es importante notar que en el cálculo de la probabilidad de aceptación $\alpha\left(x_{t-1}, y^{(j)}\right)$ se utilizan sólo los *pesos no normalizados*.

La Figura 5.1.1 intenta resumir gráficamente los pasos principales del algoritmo MTM con $k = 3$. A partir de la función tentativa $\pi\left(\cdot|x_{t-1}\right)$, por ejemplo centrada en x_{t-1}, generamos $k = 3$ muestras $\left[y^{(1)}, y^{(2)}, y^{(3)}\right]$ y asociamos unos *pesos normalizados*, representados mediante funciones delta centradas en $\left[y^{(1)}, y^{(2)}, y^{(3)}\right]$. Después, elegimos aleatoriamente de acuerdo a los pesos normalizados una de las muestras generadas previamente. En este caso suponemos haber elegido $y^{(j)} \equiv y^{(1)}$. Ahora, se generan $k - 1 = 2$ realizaciones desde $\pi\left(\cdot|y^{(j)}\right)$, $\left[x^{(1)}, x^{(2)}\right]$ y se define $x^{(3)} \equiv x_{t-1}$. Finalmente se calculan los *pesos no normalizados* para el vector $\left[x^{(1)}, x^{(2)}, x^{(3)}\right]$. En este caso no se necesita normalizar los pesos.

Las elecciones más utilizadas en la literatura de la función $\lambda(x, y)$ son las tres siguientes:

- $\lambda(x, y) = 1$, dispone el menor coste computacional, como es obvio.
- $\lambda(x, y) = \left[\pi(x|y) + \pi(y|x)\right]^{-1}$, proporciona una ventaja mayor en convergencia respecto al anterior. En este caso los pesos quedan como

$$w(x, y) = p(x)\,\pi(y|x)\,\frac{1}{\pi(x|y) + \pi(y|x)} = p(x)\,\underbrace{\frac{\pi(y|x)}{\pi(x|y) + \pi(y|x)}}_{\geq 0 \text{ y } \leq 1}, \qquad (5.1.3)$$

  donde el segundo término de la multiplicación es un número entre 0 y 1 (se puede interpretar como una probabilidad, un peso).

- $\lambda(x, y) = \left(\pi(x|y)\,\pi(y|x)\right)^{-\beta}$, siendo β una constante que hay que fijar. Empíricamente, en la literatura existente [38], se ha visto que esta función parece dar mejores resultados que las dos anteriores. Además, nótese que cuando $\beta = -1$ tenemos

$$w(x, y) = p(x)\,\pi(y|x)\,\frac{1}{\pi(x|y)\,\pi(y|x)} = \frac{p(x)}{\pi(x|y)}, \qquad (5.1.4)$$

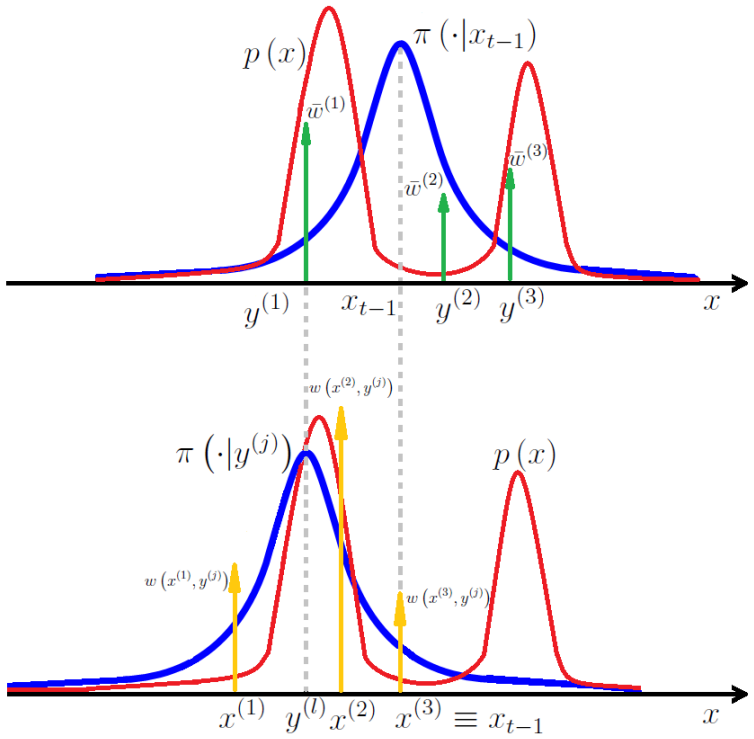

Figura 5.1.1: Ejemplo de funcionamiento del algoritmo MTM con $k = 3$.

que en este caso se parecen muchos a los *pesos* definidos por el *importance sampling*.

Otras posibles elecciones de la función $\lambda(x, y)$ pueden dar lugar a interesantes casos específicos que discutiremos más adelante.

Es importante evidenciar que la eficiencia del algoritmo MTM se basa en una buena elección de un número k adecuado de puntos según la diferencia en la forma de la fdp tentativa y la fdp objetivo. También es importante notar que este algoritmo puede interpretarse también como un conjunto de k-técnicas MH en paralelo que utilizan la misma fdp tentativa $\pi(\cdot|\cdot)$ y con intercambio de información (mediante los pesos y la probabilidad de aceptación) entre los diferentes cadenas de Markov.

A continuación, vamos a comprobar que el algoritmo MTM cumple con la condición de balance, definida en la Ecuación (3.2.5). Para ello, primero tendremos que

definir el *kernel* (probabilidad de transición de la cadena de Markov producida) del método MTM.

5.1.1. Kernel y reversibilidad

En esta sección, analizaremos y estudiaremos la probabilidad de transición (kernel) del algoritmo MTM en tres casos distintos con $k = 1$, $k = 2$ y k genérico, para facilitar la comprensión del lector.

Caso $k = 1$

Para $k = 1$ el algoritmo MTM es equivalente al algoritmo MH estándar. De hecho en este caso, el método consta de estos pasos:

1. Elegir aleatoriamente un punto x_0 e inicializar $t = 1$.
2. Generar una muestra $y^{(1)}$, a partir de la densidad $\pi\left(y|x_{t-1}\right)$.
3. Calcular el peso

$$w\left(y^{(1)}, x_{t-1}\right) = p\left(y^{(1)}\right)\pi\left(x_{t-1}|y^{(1)}\right)\lambda\left(y^{(1)}, x_{t-1}\right),$$

   asociados a las muestras $y^{(1)}$.
4. El peso "normalizado" es claramente igual a 1,

$$\bar{w}^{(1)} \triangleq \frac{w\left(y^{(1)}, x_{t-1}\right)}{w\left(y^{(1)}, x_{t-1}\right)} = 1.$$

5. Dada que $\bar{w}^{(1)} = 1$, elegimos la única muestra posible, $y^{(1)}$.
6. Generar $k-1 = 0$ muestras aleatorias $x^{(r)}$ desde la densidad $\pi\left(x|y^{(1)}\right)$, es decir, *no generamos ninguna.*
7. Construir el vector de una sola componente $(k = 1)$ $\left[x^{(1)}\right]$, donde $x^{(1)} = x_{t-1}$.

8. Calcular el peso asociado a $x^{(1)}$,

$$w\left(x^{(1)}, y^{(1)}\right) = p\left(x^{(1)}\right) \pi\left(y^{(1)}|x^{(1)}\right) \lambda\left(x^{(1)}, y^{(1)}\right).$$

9. Calcular la probabilidad de aceptación de la muestra $y^{(1)}$ mediante (recordemos que $x_{t-1} = x^{(1)}$)

$$\begin{aligned} \alpha\left(x_{t-1}, y^{(1)}\right) &= \text{mín}\left[1, \frac{w\left(y^{(1)}, x_{t-1}\right)}{w\left(x_{t-1}, y^{(1)}\right)}\right] \\ &= \text{mín}\left[1, \frac{p\left(y^{(1)}\right) \pi\left(x_{t-1}|y^{(1)}\right) \lambda\left(y^{(1)}, x_{t-1}\right)}{p\left(x_{t-1}\right) \pi\left(y^{(1)}|x_{t-1}\right) \lambda\left(x_{t-1}, y^{(1)}\right)}\right] \\ &= \text{mín}\left[1, \frac{p\left(y^{(1)}\right) \pi\left(x_{t-1}|y^{(1)}\right)}{p\left(x_{t-1}\right) \pi\left(y^{(1)}|x_{t-1}\right)}\right], \end{aligned} \tag{5.1.5}$$

   donde hemos utilizado la simetría $\lambda(x, y)$.

10. Aceptar la muestra $x_t = y^{(1)}$, con probabilidad $\alpha\left(x_{t-1}, y^{(1)}\right)$ sino $x_t = x_{t-1}$.

11. Actualizar $t = t + 1$ y volver al paso 2.

Como resulta evidente, los pasos arriba coinciden con el algoritmo MH descrito en el Capítulo 3: dado que algunos pasos resultan irrelevantes se pueden eliminar, por ejemplo los puntos 4, 5, 6 y 7, y la función de aceptación coincide con la de MH estándar.

Por lo tanto, con $k = 1$ el kernel del algoritmo MTM *coincide* con el de MH, así que el método MTM cumple la condición de balance como hemos demostrado en la Sección 3.3.2.

Caso $k = 2$

Vamos a analizar el siguiente caso más sencillo, con $k = 2$. En este caso, dado x_{t-1} vamos a generar dos muestras $\left[y^{(1)}, y^{(2)}\right]$. Antes de todo, vamos a escribir el *kernel* del algoritmo siguiendo literalmente los pasos antes definidos. Vamos a considerar la

probabilidad de transición de x_{t-1} a un genérico $x_t = y = c$, es decir, $K(y = c|x_{t-1})$. Recordamos que el método MTM nos dice que tenemos que:

1. Muestrear $y^{(1)} \sim \pi(y|x_{t-1})$.
2. Muestrear $y^{(2)} \sim \pi(y|x_{t-1})$.
3. Seleccionar un $y^{(j)} = y^{(1)}$ o $y^{(j)} = y^{(2)}$ de acuerdo a los pesos normalizados $\bar{w}^{(1)} = \frac{w\left(y^{(1)},x_{t-1}\right)}{w\left(y^{(1)},x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}$ y $\bar{w}^{(2)} = \frac{w\left(y^{(2)},x_{t-1}\right)}{w\left(y^{(1)},x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}$.
4. Muestrear $x^{(1)} \sim \pi\left(x|y^{(j)}\right)$.
5. Aceptar $x_t = y^{(j)}$ (que será igual $y^{(1)}$ o a $y^{(2)}$) con probabilidad

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{w\left(y^{(1)}, x_{t-1}\right) + w\left(y^{(2)}, x_{t-1}\right)}{w\left(x^{(1)}, y^{(j)}\right) + w\left(x_{t-1}, y^{(j)}\right)}\right]. \tag{5.1.6}$$

Hay que notar que, además, tenemos dos posibilidades de pasar de x_{t-1} a $y = c$: la primera es muestreamos $y^{(1)} = c$, elegimos $y^{(1)}$ y finalmente aceptamos $y^{(1)}$; la segunda consiste en muestrear $y^{(2)} = c$, elegir $y^{(2)}$ y finalmente aceptamos $y^{(2)}$. Todo esto en fórmula se expresa de la siguiente forma (para el caso $x_{t-1} \neq y$[1])

[1] Como en el caso del algoritmo MH clásico deberíamos añadir una delta de Dirac correspondiente al caso $x_{t-1} = y$. Pero en este caso por simplificar el desarrollo hemos preferido considerar sólo el caso más genérico. Las demostraciones para el caso $x_{t-1} = y$ son triviales.

$$
\begin{aligned}
&K\left(y=c|x_{t-1}\right)= \\
&=\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(1)}=c|x_{t-1}\right)\pi\left(y^{(2)}|x_{t-1}\right)\frac{w\left(y^{(1)}=c,x_{t-1}\right)}{w\left(y^{(1)}=c,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}\times \\
&\quad\times\pi\left(x^{(1)}|y^{(1)}=c\right)\text{mín}\left[1,\frac{w\left(y^{(1)}=c,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}{w\left(x_{t-1},y^{(1)}=c\right)+w\left(x^{(1)},y^{(1)}=c\right)}\right]dy^{(2)}dx^{(1)}+ \\
&\quad+\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(1)}|x_{t-1}\right)\pi\left(y^{(2)}=c|x_{t-1}\right)\frac{w\left(y^{(2)}=c,x_{t-1}\right)}{w\left(y^{(1)},x_{t-1}\right)+w\left(y^{(2)}=c,x_{t-1}\right)}\times \\
&\quad\times\pi\left(x^{(1)}|y^{(2)}=c\right)\text{mín}\left[1,\frac{w\left(y^{(2)}=c,x_{t-1}\right)+w\left(y^{(1)},x_{t-1}\right)}{w\left(x_{t-1},y^{(2)}=c\right)+w\left(x^{(1)},y^{(2)}=c\right)}\right]dy^{(1)}dx^{(1)},
\end{aligned}
\tag{5.1.7}
$$

donde hemos seguido los pasos anteriormente descritos e integramos respecto a todas las variables auxiliares. Es importante notar que el primer término de la suma se refiere al vector $\left[y^{(1)}=c,y^{(2)}\right]$, mientras el segundo término de la suma se refiere a $\left[y^{(1)},y^{(2)}=c\right]$.

Además, no es difícil notar que por simetría las dos integrales dobles (los dos términos de la suma) dan el mismo resultado, por lo tanto podemos escribir

$$
\begin{aligned}
&K\left(y=c|x_{t-1}\right)= \\
&=2\int_{\mathcal{S}}\int_{\mathcal{S}}\underbrace{\pi\left(y^{(1)}=c|x_{t-1}\right)}_{\text{paso 1}}\underbrace{\pi\left(y^{(2)}|x_{t-1}\right)}_{\text{paso 2}}\underbrace{\frac{w\left(y^{(1)}=c,x_{t-1}\right)}{w\left(y^{(1)}=c,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}}_{\text{paso 3}}\times \\
&\quad\times\underbrace{\pi\left(x^{(1)}|y^{(1)}=c\right)}_{\text{paso 4}}\underbrace{\text{mín}\left[1,\frac{w\left(y^{(1)}=c,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}{w\left(x_{t-1},y^{(1)}=c\right)+w\left(x^{(1)},y^{(1)}=c\right)}\right]}_{\text{paso 5}}dy^{(2)}dx^{(1)},
\end{aligned}
\tag{5.1.8}
$$

donde hemos también indicado la referencia a los pasos anteriormente descritos. Evidenciamos otra vez que en la Ecuación (5.1.8) aparece el kernel del MTM sólo para $x_{t-1}\neq y$. Recordamos que para $x_{t-1}=y$ deberíamos añadir una delta de Dirac

(la demostración que todo funciona adecuadamente también para $x_{t-1} = y$ es trivial).

A continuación vamos a comprobar que el kernel en la Ecuación (5.1.8) cumple la condición de reversibilidad siguiente

$$p(x) K(y|x) = p(y) K(x|y).$$

Demostración. La expresión en el primer miembro quedaría en nuestro caso

$$\begin{aligned} & p(x_{t-1}) K(y = c|x_{t-1}) = \\ & \quad = 2p(x_{t-1}) \int_{\mathcal{S}} \int_{\mathcal{S}} \pi\left(y^{(1)} = c|x_{t-1}\right) \pi\left(y^{(2)}|x_{t-1}\right) \frac{w\left(y^{(1)} = c, x_{t-1}\right)}{w\left(y^{(1)} = c, x_{t-1}\right) + w\left(y^{(2)}, x_{t-1}\right)} \times \\ & \qquad \times \pi\left(x^{(1)}|y^{(1)} = c\right) \text{mín}\left[1, \frac{w\left(y^{(1)} = c, x_{t-1}\right) + w\left(y^{(2)}, x_{t-1}\right)}{w\left(x_{t-1}, y^{(1)} = c\right) + w\left(x^{(1)}, y^{(1)} = c\right)}\right] dy^{(2)} dx^{(1)}. \end{aligned} \tag{5.1.9}$$

A continuación desarrollaremos con cálculos triviales la expresión de arriba y, para simplificar la notación, vamos a escribir solamente y en lugar de $y^{(1)} = c$ y de $y = c$. Como primer paso vamos a sustituir a la Ecuación (5.1.9) la expresión de los pesos de la Ecuación (5.1.1)

$$\begin{aligned} & p(x_{t-1}) K(y|x_{t-1}) = \\ & \quad = 2p(x_{t-1}) \int_{\mathcal{S}} \int_{\mathcal{S}} \pi(y|x_{t-1}) \pi\left(y^{(2)}|x_{t-1}\right) \frac{w(y, x_{t-1})}{w(y, x_{t-1}) + w\left(y^{(2)}, x_{t-1}\right)} \times \\ & \qquad \times \pi\left(x^{(1)}|y\right) \text{mín}\left[1, \frac{w(y, x_{t-1}) + w\left(y^{(2)}, x_{t-1}\right)}{w(x_{t-1}, y) + w\left(x^{(1)}, y\right)}\right] dy^{(2)} dx^{(1)}. \end{aligned} \tag{5.1.10}$$

Ahora sacamos fuera de las integrales las funciones que no dependen de las variables

diferenciales y intercambiamos la posición de unos factores, llegando a

$$\begin{aligned} p(x_{t-1})K\left(y|x_{t-1}\right) = \\ =2p(x_{t-1})\pi\left(y|x_{t-1}\right)w\left(y,x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right)\times \\ \times\frac{1}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}\text{mín}\left[1,\frac{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}{w\left(x_{t-1},y\right)+w\left(x^{(1)},y\right)}\right]dy^{(2)}dx^{(1)}. \end{aligned} \tag{5.1.11}$$

Ahora, introducimos el factor que multiplica a la función mín $[\cdot|\cdot]$ dentro de esta

$$\begin{aligned} p(x_{t-1})K\left(y|x_{t-1}\right) = \\ =2p(x_{t-1})\pi\left(y|x_{t-1}\right)w\left(y,x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right)\times \\ \times\text{mín}\left[\frac{1}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)},\frac{1}{w\left(x_{t-1},y\right)+w\left(x^{(1)},y\right)}\right]dy^{(2)}dx^{(1)}. \end{aligned} \tag{5.1.12}$$

Además, dado que

$$w\left(x_{t-1},y\right)=p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)\lambda\left(x_{t-1},y\right)\rightarrow p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)=\frac{w\left(x_{t-1},y\right)}{\lambda\left(x_{t-1},y\right)},$$

podemos finalmente escribir

$$\begin{aligned} p\left(x_{t-1}\right)K\left(y|x_{t-1}\right) = \\ =2\frac{w\left(x_{t-1},y\right)}{\lambda\left(x_{t-1},y\right)}w\left(y,x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right)\times \\ \times\text{mín}\left[\frac{1}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)},\frac{1}{w\left(x_{t-1},y\right)+w\left(x^{(1)},y\right)}\right]dy^{(2)}dx^{(1)}. \end{aligned} \tag{5.1.13}$$

Dado que $\lambda\left(x_{t-1},y\right)=\lambda\left(y,x_{t-1}\right)$ por definición, esta última expresión (5.1.13) es *simétrica* respecto a x_{t-1} y a y. Es decir, podemos intercambiar la variable x_{t-1} con

al variable y, que en fórmula se escribe

$$p\left(x_{t-1}\right) K\left(y|x_{t-1}\right) = p\left(y\right) K\left(x_{t-1}|y\right),$$

que es exactamente la *condición de balance o reversibilidad.* □

Demostración para k genérico

Vamos ahora a analizar el kernel del algoritmo para k genérico. Recordar que $y^{(1)} = y$ en nuestra notación, la probabilidad de transición de x_{t-1} a $x_t = y$ con el método MTM es la siguiente

$$\begin{aligned} K\left(y|x_{t-1}\right) =& k \int_{\mathcal{S}} \cdots \int_{\mathcal{S}} \left[\pi\left(y|x_{t-1}\right) \prod_{i=2}^{k} \pi\left(y^{(i)}|x_{t-1}\right)\right] \frac{w\left(y, x_{t-1}\right)}{w\left(y, x_{t-1}\right) + \sum_{i=2}^{k} w\left(y^{(i)}, x_{t-1}\right)} \times \\ & \times \left[\prod_{i=1}^{k-1} \pi\left(x^{(i)}|y\right)\right] \alpha\left(x_{t-1}, y\right) dy^{(2)} \cdots dy^{(k)} dx^{(1)} \cdots dx^{(k-1)} \end{aligned} \tag{5.1.14}$$

siendo $\alpha\left(x_{t-1}, y\right)$ expresado en la Ecuación (5.1.2), es decir,

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{w\left(y^{(1)}, x_{t-1}\right) + ... + w\left(y^{(k)}, x_{t-1}\right)}{w\left(x^{(1)}, y^{(j)}\right) + ... + w\left(x^{(k)}, y^{(j)}\right)}\right].$$

Siguiendo exactamente los mismos pasos de la demostración para $k = 2$, es fácil de demostrar que el kernel de la Ecuación (5.1.14) cumple la condición de balance.

5.2. Casos específicos

A continuación vamos a describir unos casos particulares interesantes.

5.2.1. Algoritmo Metropolis-Hastings

Para $k = 1$, ya hemos visto en la Sección 5.1.1 que el método MTM coincide con el algoritmo MH tradicional. De hecho, la probabilidad de aceptación en este caso queda como

$$\begin{aligned} \alpha(x, y) &= \text{mín}\left[1, \frac{w(y, x)}{w(x, y)}\right] \\ &= \text{mín}\left[1, \frac{p(y)\,\pi(x|y)\,\lambda(y, x)}{p(x)\,\pi(y|x)\,\lambda(x, y)}\right] \\ &= \text{mín}\left[1, \frac{p(y)\,\pi(x|y)}{p(x)\,\pi(y|x)}\right], \end{aligned} \tag{5.2.1}$$

que es exactamente la función de aceptación del MH estándar.

5.2.2. Caso ideal

Supongamos que nuestra función tentativa $\pi(\cdot|\cdot)$ es exactamente el *kernel* que buscamos, es decir, que la $\pi(\cdot|\cdot)$ cumple la condición de reversibilidad

$$p(x)\,\pi(y|x) = p(y)\,\pi(x|y), \tag{5.2.2}$$

donde recordamos que $p(x) \propto p_o(x)$. En este caso, podemos elegir

$$\lambda(x, y) = \frac{1}{p(x)\,\pi(y|x)}, \tag{5.2.3}$$

dado que en esta situación será simétrica dado que ya se cumple la ecuación de balance. Los pesos del MTM en este caso quedan como

$$w(x, y) = p(x)\,\pi(y|x)\,\lambda(x, y) = p(x)\,\pi(y|x)\,\frac{1}{p(x)\,\pi(y|x)} = 1, \tag{5.2.4}$$

es decir, *todos los pesos serán iguales*. La función de aceptación α quedará como

$$\alpha\left(x_{t-1}, y\right) = \text{mín}\left[1, \frac{\overbrace{1+...+1}^{k}}{\underbrace{1+...+1}_{k}}\right] = \text{mín}\left[1, \frac{k}{k}\right] = 1, \tag{5.2.5}$$

es decir, *aceptamos siempre las nuevas muestras propuestas por la fdp tentativa* $\pi\left(\cdot|\cdot\right)$ dado que esta ya cumple la ecuación de balance y no necesitamos ninguna corrección. Finalmente, se puede ver este caso específico del MTM como un *Gibbs sampler* (mírese la Sección B.2) que opera con las dos fdp condicionales $\pi\left(x|y\right)$ y $\pi\left(y|x\right)$ (en este caso, las dos condicionales tienen la misma forma analítica).

Nótese también la importancia de una buena elección de la función $\lambda\left(x, y\right)$. De hecho, incluso en este caso ideal otra elección de la función $\lambda\left(x, y\right)$ no proporcionaría una probabilidad de aceptación siempre igual a 1.

5.2.3. Fdp tentativa independiente y $\lambda\left(x, y\right) = 1$

Consideremos el caso en el que la función tentativa no depende del estado anterior,

$$\pi\left(y|x\right) = \pi\left(y\right), \tag{5.2.6}$$

$$\pi\left(x|y\right) = \pi\left(x\right), \tag{5.2.7}$$

es decir,

$$\pi\left(\cdot|x\right) = \pi\left(\cdot|y\right) = \pi\left(\cdot\right), \tag{5.2.8}$$

las dos condicionales y las dos marginales son todas la misma fdp, y además elegimos

$$\lambda\left(x, y\right) = 1. \tag{5.2.9}$$

En este caso, los pesos quedan como

$$w\left(x, y\right) = p\left(x\right)\pi\left(y\right). \tag{5.2.10}$$

Para este caso especial vamos a comentar sólo dos aspectos que nos parecen interesantes. Primero, que la función de aceptación 5.1.2 queda como

$$\begin{aligned}\alpha\left(x_{t-1}, y^{(j)}\right) &= \text{mín}\left[1, \frac{p\left(y^{(1)}\right)\pi\left(x_{t-1}\right) + ... + p\left(y^{(k)}\right)\pi\left(x_{t-1}\right)}{p\left(x^{(1)}\right)\pi\left(y^{(j)}\right) + ... + p\left(x^{(k)}\right)\pi\left(y^{(j)}\right)}\right] \\ &= \text{mín}\left[1, \frac{\pi\left(x_{t-1}\right)}{\pi\left(y^{(j)}\right)} \cdot \left(\frac{p\left(y^{(1)}\right) + ... + p\left(y^{(k)}\right)}{p\left(x^{(1)}\right) + ... + p\left(x^{(k)}\right)}\right)\right],\end{aligned} \tag{5.2.11}$$

además, recordando que $x^{(k)} = x_{t-1}$, podemos escribir

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{\pi\left(x_{t-1}\right)}{\pi\left(y^{(j)}\right)}\left(\frac{p\left(y^{(1)}\right) + ... + p\left(y^{(k)}\right)}{p\left(x^{(1)}\right) + ... + p\left(x_{t-1}\right)}\right)\right]. \tag{5.2.12}$$

La segunda observación interesante es que dado que las dos fdp condicionales tentativas son iguales, realmente no necesitamos muestrear nuevas muestras $\left[x^{(1)}, ..., x^{(k-1)}\right]$. De hecho, podríamos reciclar la $k-1$ muestras desde el vector $\left[y^{(1)}, ..., y^{(k)}\right]$ eliminando sólo la muestra previamente seleccionada $y^{(j)}$. De este modo, la función α quedaría como

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{\pi\left(x_{t-1}\right)}{\pi\left(y^{(j)}\right)}\left(\frac{p\left(y^{(1)}\right) + ... + p\left(y^{(j-1)}\right) + p\left(y^{(j)}\right) + p\left(y^{(j+1)}\right) + ... + p\left(y^{(k)}\right)}{p\left(y^{(1)}\right) + ... + p\left(y^{(j-1)}\right) + p\left(y^{(j+1)}\right) + ... + p\left(y^{(k)}\right) + p\left(x_{t-1}\right)}\right)\right].$$

5.2.4. Orientation Bias Monte Carlo (OBMC)

Consideremos el caso en el que la fdp tentativa sea simétrica, es decir,

$$\pi\left(y|x\right) = \pi\left(x|y\right). \tag{5.2.13}$$

En este caso, podemos definir la función $\lambda\left(x, y\right)$ de esta forma

$$\lambda\left(x, y\right) = \frac{1}{\pi\left(x|y\right)}, \tag{5.2.14}$$

dado que $\pi(\cdot|\cdot)$ es simétrica, lo será también $\lambda(x,y)$. Con estas dos condiciones (5.2.13) y (5.2.14), los pesos del algoritmo MTM quedan como

$$w(x,y) = p(x), \tag{5.2.15}$$

es decir, sólo evaluamos la densidad objetivo $p(x) \propto p_o(x)$. Es importante notar que los pesos en este caso sólo dependen de la primer variable de $w(\cdot,\cdot)$.

Este caso específico es denominado en la literatura *Orientation Bias Monte Carlo* (*OBMC*) [38]. Los pasos del algoritmo son los siguientes:

1. Dada una muestra x_{t-1}, se generan k muestras $\left[y^{(1)},...,y^{(k)}\right]$ desde la función tentativa $\pi(y|x_{t-1})$.

2. Se calculan los pesos $w\left(y^{(i)},x_{t-1}\right) = p\left(y^{(i)}\right)$, $i = 1,...,k$.

3. Se normalizan los pesos
$$\bar{w}^{(i)} = \frac{p\left(y^{(i)}\right)}{\sum_{i=1}^{k} p\left(y^{(i)}\right)},$$
con $i = 1,...,k$.

4. Elegimos una componente $y^{(j)}$, $j \in \{1,...,k\}$, de acuerdo a los pesos con una probabilidad proporcional a $\bar{w}^{(i)}$, $i = 1,...,k$.

5. Obtenemos $k-1$ muestras $\left[x^{(1)},...,x^{(k-1)}\right]$ a partir de $\pi\left(x|y^{(j)}\right)$.

6. Construimos el vector $\left[x^{(1)},...,x^{(k-1)},x^{(k)}\right]$, siendo $x^{(k)} = x_{t-1}$.

7. Aceptamos la muestra $x_t = y^{(j)}$ con una probabilidad (recordemos que $x^{(k)} = x_{t-1}$)

$$\begin{aligned}\alpha\left(x_{t-1},y^{(j)}\right) &= \text{mín}\left[1,\frac{p\left(y^{(1)}\right)+...+p\left(y^{(k)}\right)}{p\left(x^{(1)}\right)+...+p\left(x^{(k)}\right)}\right] \\ &= \text{mín}\left[1,\frac{p\left(y^{(1)}\right)+...+p\left(y^{(k-1)}\right)+p\left(y^{(k)}\right)}{p\left(x^{(1)}\right)+...+p\left(x^{(k-1)}\right)+p\left(x_{t-1}\right)}\right],\end{aligned} \tag{5.2.16}$$

8. Si no, con probabilidad $1-\alpha\left(x_{t-1}, y^{(j)}\right)$, definimos $x_t = x_{t-1}$.

Es interesante observar que este algoritmo puede verse como la versión multi-punto del algoritmo MH estándar con fdp tentativa simétrica propuesta en la Sección 3.4.2. Pero, respecto al método de la Sección 3.4.2, este algoritmo no tiene una interpretación tan intuitiva. Mientras con el MH estándar con fdp tentativa simétrica se aceptaban automáticamente todos los puntos con $p\left(y^{(j)}\right) \geq p\left(x_{t-1}\right)$, en este caso la función de aceptación resulta ser bastante más compleja. La intuición sugiere que en esta situación multi-punto, se están comparando "regiones" más que valores específicos de la densidad objetivo. Sin embargo, es muy importante subrayar que los valores correspondientes a las muestras $y^{(1)}, y^{(2)}, ..., y^{(k)}$ generadas desde $\pi\left(\cdot|x_{t-1}\right)$ se encuentra en el numerador (mientras el valor $p\left(x_{t-1}\right)$ está en el denominador), y los valores correspondientes a las muestras $x^{(1)}, ..., x^{(k-1)}$ generadas desde $\pi\left(\cdot|y^{(j)}\right)$ se encuentran en el denominador (mientras el valor $p\left(y^{(j)}\right)$ esta en el numerador).

Con una reflexión más profunda se puede concluir que, por ejemplo, este método favorece la transiciones a zonas de alta probabilidad concentradas en pequeñas áreas (regiones con modas muy "estrechas"). En este caso, por ejemplo, suponiendo que $y^{(j)}$ esté muy cerca de esta moda "estrecha" los valores $p\left(x^{(1)}\right), ..., p\left(x^{(k-1)}\right)$ con alta probabilidad podrían ser pequeños, ayudando la transición. La Figura 5.2.1 quiere representar esta situación.

5.2.5. MTM "inverso"

A continuación vamos a presentar un caso específico de MTM cuyos pesos se "parecen" mucho a los pesos definidos con el *importance sampling*. Esto ocurre al definir la función $\lambda(x, y)$ de esta forma [37],

$$\lambda(x, y) = \frac{1}{\pi(x|y)\,\pi(y|x)}. \tag{5.2.17}$$

Entonces, los pesos quedan como

$$w(x, y) = p(x)\,\pi(y|x)\,\frac{1}{\pi(x|y)\,\pi(y|x)} = \frac{p(x)}{\pi(x|y)}, \tag{5.2.18}$$

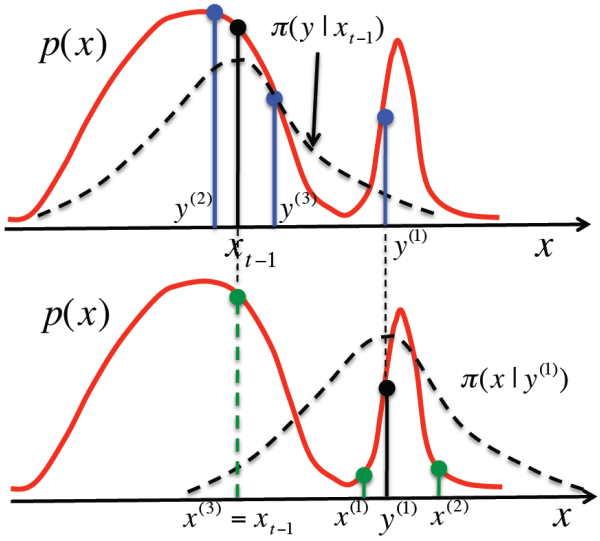

Figura 5.2.1: Observación sobre el algoritmo OBMC. En la figura consideramos $k = 3$ y suponemos que se ha seleccionado la muestra $y^{(1)}$ $(j = 1)$, que se encuentra cerca de una moda muy "estrecha". Utilizando el MH estándar la muestra $y^{(1)}$ sería aceptada con probabilidad $\frac{p\left(y^{(1)}\right)}{p(x_{t-1})} \leq 1$, dado que el MH interpreta que nos alejamos de una zona de alta probabilidad; el OBMC sin embargo, consigue analizar la región del espacio de estado en la que se encuentra la nueva muestra y favorece la transición. De hecho, en este caso, los puntos alrededor de la muestra seleccionada $y^{(j)}$ $(j = 1)$ tienen valores de probabilidad muy bajos, aumentando la probabilidad de aceptación. Esto es porque los valores en "verde" se suman al denominador, mientras que los valores en "azul" se suman al numerador.

donde tenemos la densidad objetivo $p(x) \propto p_o(x)$ en el numerador y una fdp condicional tentativa en el denominador, como en el importance sampling.

Un caso de MTM "inverso" aún más especial y particularmente interesante desde el punto de vista práctico y teórico, es el siguiente.

Multiple-trial Metropolized independence sampler (MTMIS)

El algoritmo *Multiple-trial Metropolized independence sampler* [38] es un caso especial del MTM "inverso" particularmente interesante, donde $\lambda(x, y) = \frac{1}{\pi(x|y)\pi(y|x)}$

y además la función tentativa no depende del estado anterior, es decir

$$\pi\left(y|x\right)=\pi\left(y\right), \tag{5.2.19}$$

$$\pi\left(x|y\right)=\pi\left(x\right), \tag{5.2.20}$$

es decir,

$$\pi\left(\cdot|x\right)=\pi\left(\cdot|y\right)=\pi\left(\cdot\right), \tag{5.2.21}$$

las dos condicionales y las dos marginales son todas la misma fdp.

En estos casos los pesos

$$w\left(x,y\right)=p\left(x\right)\pi\left(y\right)\frac{1}{\pi\left(x\right)\pi\left(y\right)}=\frac{p\left(x\right)}{\pi\left(x\right)},$$

que son exactamente los pesos del *importance sampling*, que aquí por comodidad denotaremos como

$$\gamma\left(x\right)\triangleq\frac{p\left(x\right)}{\pi\left(x\right)}=w\left(x,y\right). \tag{5.2.22}$$

Entonces, este algoritmo es una versión multi-punto del caso específico del algoritmo MH estándar tratado en la Sección 3.4.3.

Intuitivamente se podría pensar que debido a que las muestras se generan de forma independiente, en este algoritmo no es necesario generar las *muestras del conjunto de referencia* (el vector $\left[x^{(1)},...,x^{(k-1)}\right]$). Efectivamente, es así. Pero en realidad esto deriva de una ulterior reflexión como mostramos en el punto 5 de la lista que sigue:

1. Generar k muestras independientes $y^{(1)},...,y^{(k)}$ desde la fdp tentativa, $\pi\left(\cdot|x_{t-1}\right)=\pi\left(\cdot\right)$.

2. Calcular los pesos asociados a las muestras $y^{(1)},...,y^{(k)}$, $\gamma\left(y^{(i)}\right)=\frac{p\left(y^{(i)}\right)}{\pi\left(y^{(i)}\right)}$, $i=1,...,k$.

3. Normalizar los pesos, es decir

$$\bar{\gamma}^{(i)}=\frac{\gamma\left(y^{(i)}\right)}{\sum_{i=1}^{k}\gamma\left(y^{(i)}\right)}, \tag{5.2.23}$$

con $i = 1, ..., k$.

4. Obtener una muestra $y^{(j)} \in \left\{y^{(1)}, ..., y^{(k)}\right\}$ de acuerdo a los los pesos $\bar{\gamma}^{(i)}$, $i = 1, ..., k$.

5. *Deberíamos ahora generar el vector* $\left[x^{(1)}, ..., x^{(k-1)}\right]$ desde $\pi\left(\cdot|y^{(j)}\right) = \pi\left(\cdot\right)$. **¡Pero ya tenemos muestras desde** $\pi\left(\cdot\right)$ (mirar el punto 1)**!** Así que, podemos coger las $k-1$ muestras del vector $\left[y^{(1)}, ..., y^{(k)}\right]$ eliminando la muestra $y^{(j)}$.

6. Finalmente, aceptar la muestra $x_t = y^{(j)}$ con una probabilidad

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \\ = \text{mín}\left[1, \frac{\gamma\left(y^{(1)}\right) + \gamma\left(y^{(2)}\right) + ... + \gamma\left(y^{(j)}\right) + ... + \gamma\left(y^{(k-1)}\right) + \gamma\left(y^{(k)}\right)}{\gamma\left(y^{(1)}\right) + ... + \gamma\left(y^{(j-1)}\right) + \gamma\left(y^{(j+1)}\right) + ... + \gamma\left(y^{(k)}\right) + \gamma\left(x_{t-1}\right)}\right], \tag{5.2.24}$$

que definiendo $W = \sum_{i=1}^{k} \gamma\left(y^{(i)}\right)$ se puede expresar como

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{W}{W - w\left(y^{(l)}\right) + w\left(x_{t-1}\right)}\right], \tag{5.2.25}$$

y rechazamos con la probabilidad $1 - \alpha\left(x_{t-1}, y^{(j)}\right)$, con $x_t = x_{t-1}$.

Está claro que el paso 5 resulta ser realmente irrelevante y puede ser eliminado de la lista. A continuación, vamos a analizar los pasos importantes.

Nótese, por ejemplo, que los pasos 1, 2 *y* 3 *coinciden* con el *importance sampling* (véase la Sección 2.5). El paso 4 es resampling generando solo una muestra (véase la Sección 2.5.1). Finalmente el paso 6 (nos saltamos el paso 5 porque realmente *no existe*) es un paso de aceptación típico de un algoritmo MCMC (como el método MH). Así que, podemos afirmar la siguiente importante proposición.

Observación: Este algoritmo MTMIS puede verse como una técnica para convertir muestras "pesadas" según el importance sampling en una muestra aleatoria que se distribuye *exactamente* de acuerdo a la densidad objetivo $p\left(x\right) \propto p_o\left(x\right)$. Dicho de

otra forma, es una manera de convertir el *importance sampling* en un *generador de números aleatorios*. De hecho, hemos visto que con el importance sampling podemos aproximar la medida de probabilidad asociada a la densidad objetivo $p_o(x)$, pero no generamos muestras aleatorias desde $p_o(x)$. Sin embargo, añadiendo los pasos de selección 4 y de aceptación 6, conseguimos sacar una muestra proveniente *exactamente* desde $p_o(x)$.

Entonces, concluimos que el algoritmo MTMIS aplica la filosofía del importance sampling en cada instante de tiempo para luego generar una nueva muestra de la cadena de Markov, como muestra la Figura 5.2.2.

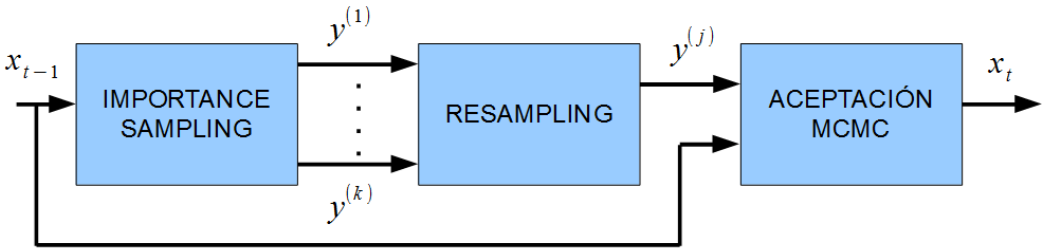

Figura 5.2.2: Esquema del algoritmo MTMIS como un sistema en cascada de importance sampling, resampling (selección de una muestra de acuerdo a los pesos $\bar{\gamma}^{(i)}$, $i = 1, ..., k$) y un paso de aceptación típico de los algoritmos MCMC.

5.2.6. MTM "degenerado"[2]

Durante el desarrollo de este proyecto, hemos casualmente "chocado" con un caso especial MTM particularmente "extraño". Consideremos otra vez una función tentativa no dependiente del estado anterior, es decir

$$\pi(y|x) = \pi(y), \tag{5.2.26}$$

[2]Hemos encontrado este caso especial de forma casual (por equivocación) durante la redacción de este proyecto. El desarrollo nos pareció tan interesante que finalmente hemos decidido incluir la sección dentro del proyecto. De todas formas, nos disculpamos con el lector de los posibles fallos que se puedan encontrar, dado que este desarrollo no está referenciado (no hemos encontrado una bibliografía adecuada).

$$\pi\left(x|y\right)=\pi\left(x\right), \tag{5.2.27}$$

o sea,

$$\pi\left(\cdot|x\right)=\pi\left(\cdot|y\right)=\pi\left(\cdot\right), \tag{5.2.28}$$

las dos condicionales y las dos marginales son todas la misma fdp. Además, definimos

$$\lambda\left(x,y\right)\triangleq\frac{1}{p\left(x\right)p\left(y\right)}. \tag{5.2.29}$$

Los pesos del MTM en este caso quedan como

$$\begin{aligned} w\left(x,y\right)&=p\left(x\right)\pi\left(y|x\right)\lambda\left(x,y\right)=p\left(x\right)\pi\left(y\right)\frac{1}{p\left(x\right)p\left(y\right)}\\ &=\frac{\pi\left(y\right)}{p\left(y\right)}=\frac{1}{\gamma\left(y\right)}, \end{aligned} \tag{5.2.30}$$

$$w\left(x,y\right)=p\left(x\right)\pi\left(y|x\right)\lambda\left(x,y\right)=p\left(x\right)\pi\left(y\right)\frac{1}{p\left(x\right)p\left(y\right)}=\frac{\pi\left(y\right)}{p\left(y\right)}=\frac{1}{\gamma\left(y\right)}, \tag{5.2.31}$$

es decir, *los pesos dependen sólo de la segunda variable* y además, son exactamente ***inverso*** respecto a los pesos $\gamma\left(y\right)$ definidos por el *importance sampling*. La no dependencia de la primera variable hace que las dos "familias" de pesos que hay que calcular para el algoritmo MTM resultan constantes. A continuación, vamos a ver en detalle como se modifican los pasos del MTM general:

1. Generar k muestras independientes $y^{(1)},...,y^{(k)}$ desde la fdp tentativa, $\pi\left(\cdot\right)$.

2. Calcular los pesos

$$w\left(y^{(i)},x_{t-1}\right)=\frac{\pi\left(x_{t-1}\right)}{p\left(x_{t-1}\right)},$$

que en este caso son *todos iguales* $\forall i\in\{1,...,k\}$. Así que los pesos normalizados resultan ser

$$\bar{w}^{(i)}=\frac{1}{k},\ i=1,...,k. \tag{5.2.32}$$

3. Elegir una $y^{(j)}\in\left\{y^{(1)},...,y^{(k)}\right\}$ de acuerdo a los pesos $\bar{w}^{(i)}$, $i=1,...,k$, que en

este caso significa de manera ***equiprobable*** (no aprovechamos ninguna información).

4. Deberíamos calcular el conjunto de referencia $\left[x^{(1)}, ..., x^{(k-1)}\right]$ desde $\pi(\cdot)$, que en este caso podrían "reciclarse" $y^{(1)}, ..., y^{(k)}$, como en la Sección 5.2.5. Pero en el siguiente paso veremos que es absolutamente ***inútil*** porque no afectan en absoluto a los pesos.

5. Calcular los pesos
$$w\left(x^{(r)}, y^{(j)}\right) = \frac{\pi\left(y^{(j)}\right)}{p\left(y^{(j)}\right)},$$
con $r = 1, ..., k$ (donde $x^{(k)} = x_{t-1}$). Como se puede ver, los pesos no varían con r, también en este caso son *todos iguales* $\forall r \in \{1, ..., k\}$.

6. Aceptar $x_t = y^{(j)}$ con probabilidad
$$\begin{aligned}\alpha\left(x_{t-1}, y^{(j)}\right) &= \text{mín}\left[1, \frac{\frac{\pi(x_{t-1})}{p(x_{t-1})} + ... + \frac{\pi(x_{t-1})}{p(x_{t-1})}}{\frac{\pi\left(y^{(j)}\right)}{p\left(y^{(j)}\right)} + ... + \frac{\pi\left(y^{(j)}\right)}{p\left(y^{(j)}\right)}}\right] = \text{mín}\left[1, \frac{k\frac{\pi(x_{t-1})}{p(x_{t-1})}}{k\frac{\pi\left(y^{(j)}\right)}{p\left(y^{(j)}\right)}}\right] \\ &= \text{mín}\left[1, \frac{\pi\left(x_{t-1}\right) p\left(y^{(j)}\right)}{\pi\left(y^{(j)}\right) p\left(x_{t-1}\right)}\right] = \text{mín}\left[1, \frac{\gamma\left(y^{(j)}\right)}{\gamma\left(x_{t-1}\right)}\right],\end{aligned} \tag{5.2.33}$$
donde con $\gamma(\cdot)$ indicamos los pesos del importance sampling.

Es decir, finalmente todo este algoritmo se reduce ***exactamente*** al método MH estándar con función tentativa independiente de la Sección 3.4.3, como podemos notar por ejemplo observando la función de aceptación $\alpha(\cdot, \cdot)$ que es idéntica. Pero en este caso, ¡¡¡*el coste computacional* es aproximadamente k veces ***superior*** al MH estándar con fdp tentativa independiente!!!

Todo esto nos hace reflexionar sobre la importancia de elegir adecuadamente la función $\lambda(x, y)$. Por ejemplo, en esta sección hemos conocido un claro ejemplo de mala elección de función $\lambda(x, y)$ que afecta drásticamente a las prestaciones del algoritmo. Por último, estas consideraciones podrían llevar a una mayor compresión

del significado de la función $\lambda(x,y)$ (y más en general de los pesos $w(x,y)$) aún desconocida en literatura, según lo que hemos podido observar.

5.3. Extensiones de MTM

En la literatura se puede encontrar diferentes y numerosas aplicaciones y generalizaciones del algoritmo MTM. Aquí proponemos las más conocidas e interesantes a nuestro juicio.

5.3.1. Densidades tentativas distintas

Aunque esta generalización está incluida en la extensión que trataremos en la siguiente subsección, hemos preferido tratar este caso a parte para ayudar la comprensión del lector. En este caso vamos a suponer de muestrear k puntos de diferentes fdp tentativas $\pi_i(\cdot|\cdot)$, $i=1,...,k$. En este caso, el algoritmo MTM clásico puede modificarse de la siguiente forma (respetando siempre la condición de reversibilidad):

1. Elegir aleatoriamente un punto x_0 e inicializar $t=1$.

2. Generar k muestras aleatorias $y^{(i)}$, desde las densidades $\pi_i(y|x_{t-1})$, $i=1,...,k$, obteniendo un vector $\left[y^{(1)},...,y^{(k)}\right]$.

3. Calcular los pesos

$$w_i\left(y^{(i)},x_{t-1}\right)=p\left(y^{(i)}\right)\pi_i\left(x_{t-1}|y^{(i)}\right)\lambda_i\left(y^{(i)},x_{t-1}\right),$$

   asociados a las muestras $y^{(i)}$, $i=1,...,k$.

4. Normalizar los pesos

$$\bar{w}^{(i)}\triangleq\frac{w_i\left(y^{(i)},x_{t-1}\right)}{\sum_{r=1}^{k}w_r\left(y^{(r)},x_{t-1}\right)},$$

   con $i=1,...,k$.

5. Elegir aleatoriamente una muestra $y^{(j)}$, con $j \in \{1, ..., k\}$, de acuerdo a los pesos normalizados $\bar{w}^{(i)}$, $i = 1, ..., k$.

6. Generar $k-1$ muestras aleatorias $x^{(r)}$, desde las densidades $\pi_r\left(x|y^{(j)}\right)$ con $r = 1, .., j-1, j+1, ..., k$ (es decir, $r \neq j$).

7. Definimos $x^{(j)} = x_{t-1}$ y construimos el vector $\left[x^{(1)}, ..., x^{(j)} = x_{t-1}, ..., x^{(k)}\right]$. Notar que, a diferencia del MTM clásico, en este caso no insertamos la muestra x_{t-1} al final, sino en la posición j.

8. Calcular los pesos asociados

$$w_r\left(x^{(r)}, y^{(j)}\right) = p\left(x^{(r)}\right)\pi_r\left(y^{(j)}|x^{(r)}\right)\lambda_r\left(x^{(r)}, y^{(j)}\right),$$

   con $r = 1, ..., k$.

9. Calcular la probabilidad de aceptación de la muestra $y^{(j)}$ mediante (recordemos que $x^{(j)} = x_{t-1}$, en la posición j)

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{w\left(y^{(1)}, x_{t-1}\right) + ... + w\left(y^{(k)}, x_{t-1}\right)}{w\left(x^{(1)}, y^{(j)}\right) + ... + w\left(x^{(k)}, y^{(j)}\right)}\right]. \tag{5.3.1}$$

10. Aceptar la muestra $x_t = y^{(j)}$, con probabilidad $\alpha\left(x_{t-1}, y^{(j)}\right)$ sino $x_t = x_{t-1}$.

11. Actualizar $t = t+1$ y volver al paso 2.

A continuación, vamos a escribir el kernel (para $x_{t-1} \neq y$) y la demostración de la condición de balance para el caso $k = 2$. En este caso, podemos escribir (estamos considerando que la muestra escogida es la primera, es decir, $j = 1$)

$$p\left(x_{t-1}\right)K\left(y|x_{t-1}\right) = 2p\left(x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi_1\left(y|x_{t-1}\right)\pi_2\left(y^{(2)}|x_{t-1}\right)\frac{w_1\left(y, x_{t-1}\right)}{w_1\left(y, x_{t-1}\right) + w_2\left(y^{(2)}, x_{t-1}\right)}\times$$

$$\times\pi_2\left(x^{(2)}|y\right)\text{mín}\left[1, \frac{w_1\left(y, x_{t-1}\right) + w_2\left(y^{(2)}, x_{t-1}\right)}{w_1\left(x_{t-1}, y\right) + w_2\left(x^{(2)}, y\right)}\right]dy^{(2)}dx^{(2)},$$

donde hemos seguido los pasos del algoritmo descrito anteriormente. Ahora, con simples cálculos podemos escribir

$$p\left(x_{t-1}\right)K\left(y|x_{t-1}\right)=2\overbrace{p\left(x_{t-1}\right)\pi_1\left(y|x_{t-1}\right)}^{w_1(x_{t-1},y)/\lambda_1(x_{t-1},y)}w_1\left(y,x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi_2\left(y^{(2)}|x_{t-1}\right)\pi_2\left(x^{(2)}|y\right)\times$$

$$\times\,\text{mín}\left[\frac{1}{w_1\left(y,x_{t-1}\right)+w_2\left(y^{(2)},x_{t-1}\right)},\frac{1}{w_1\left(x_{t-1},y\right)+w_2\left(x^{(2)},y\right)}\right]dy^{(2)}dx^{(2)},$$

donde como se puede ver se puede intercambiar las variables x_{t-1} e y, es decir, podemos escribir

$$p\left(x_{t-1}\right)K\left(y|x_{t-1}\right)=p\left(y\right)K\left(x_{t-1}|y\right).$$

5.3.2. Muestras correlacionadas

En el MTM estándar los k puntos se generan de la misma densidad tentativa de forma independiente entre ellos. En el la subsección anterior hemos visto como generalizar este esquema utilizando diferentes densidades tentativas para cada muestra generada. Pero los k puntos generados siguen siendo independientes entre ellos.

En [38, Capítulo 5], se propone también una estrategia donde las k muestras se generan de forma dependiente entre sí. Esto puede ser interesante si pensamos en un futuro esquema "adaptativo" donde mejoramos la fdp tentativa a medida que generamos las k muestras.

Dado el estado de la cadena de Markov x_{t-1}, vamos a generar k muestras de diferentes fdp tentativas de la siguiente manera:

$$y^{(1)}\sim\pi_1\left(y|x_{t-1}\right),$$

$$y^{(2)}\sim\pi_2\left(y|x_{t-1},y^{(1)}\right),$$

$$y^{(3)}\sim\pi_3\left(y|x_{t-1},y^{(1)},y^{(2)}\right),$$

y así hasta generar k puntos, es decir

$$y^{(j)} \sim \pi_j \left(y|x_{t-1}, y^{(1)}, ..., y^{(j-1)}\right), \; j = 2, ..., k.$$

Por comodidad, definimos los vectores

$$\mathbf{y_{1:j}} \triangleq [y_1, ..., y_j], \tag{5.3.2}$$

donde $j \in \{1, ..., k\}$, y también indicamos con

$$q_j \left(\mathbf{y_{1:j}}|x_{t-1}\right) = \pi_1 \left(y_1|x_{t-1}\right) \cdots \pi_j \left(y_j|x_{t-1}, \mathbf{y_{1:j-1}}\right), \tag{5.3.3}$$

la densidad conjunta del vector $\mathbf{y_{1:j}}$ dado x_{t-1}. De la misma forma, podemos definir la función de peso

$$w_j \left(x_{t-1}, \mathbf{y_{1:j}}\right) = p\left(x_{t-1}\right) q_j \left(\mathbf{y_{1:j}}|x_{t-1}\right) \lambda_j \left(x_{t-1}, \mathbf{y_{1:j}}\right), \tag{5.3.4}$$

donde $\lambda_j(\cdot)$ es una función ***simétrica secuencial***, es decir,

$$\lambda_j \left(z_1, ..., z_{j+1}\right) = \lambda_j \left(z_{j+1}, ..., z_1\right).$$

Con todo esto, el algoritmo MTM clásico se modifica de la siguiente forma:

1. Muestrear k puntos desde

   $$y^{(j)} \sim \pi_j \left(y|x_{t-1}, y^{(1)}, ..., y^{(j-1)}\right), \; j = 1, ..., k,$$

   logrando el vector $\left[y^{(1)}, ..., y^{(k)}\right]$.

2. Calcular los pesos

   $$w \left(x_{t-1}, \mathbf{y_{1:j}}\right) = p\left(x_{t-1}\right) q_j \left(\mathbf{y_{1:j}}|x_{t-1}\right) \lambda_j \left(x_{t-1}, \mathbf{y_{1:j}}\right),$$

   donde $q_j \left(\mathbf{y_{1:j}}|x_{t-1}\right) = \pi_1 \left(y_1|x_{t-1}\right) \cdots \pi_j \left(y_j|x_{t-1}, \mathbf{y_{1:j-1}}\right)$, con $j = 1, ..., k$.

3. Seleccionar una muestra $y^{(j)} \in \left\{y^{(1)}, ..., y^{(k)}\right\}$ con probabilidad proporcional a los pesos $w\left(\mathbf{y^{(i:1)}}, x_{t-1}\right)$, con $i = 1, ..., k$. Es decir, los pesos que consideramos son $w\left(y^{(1)}, x_{t-1}\right)$, después $w\left(y^{(2)}, y^{(1)}, x_{t-1}\right)$, luego $w\left(y^{(3)}, y^{(2)}, y^{(1)}, x_{t-1}\right)$... hasta llegar a $w\left(\mathbf{y^{(k:1)}}, x_{t-1}\right)$.

4. Crear un vector $\left[x^{(1)} = y^{(j-1)}, x^{(2)} = y^{(j-2)}, ..., x^{(j-1)} = y^{(1)}, x^{(j)} = x_{t-1}\right]$, donde j es índice seleccionado en el paso previo (es decir, la posición de la muestra $y^{(j)}$).

5. Generar
$$x^{(r)} \sim \pi_r\left(\cdot|y^{(j)}, \mathbf{x^{(1:r-1)}}\right),$$
con $r = j + 1, ..., k$.

6. Aceptar la nueva muestra, $x_t = y^{(j)}$, con probabilidad (recordemos que $x^{(j)} = x_{t-1}$, como en la subsección anterior)
$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{\sum_{i=1}^{k} w\left(\mathbf{y^{(i:1)}}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(\mathbf{x^{(i:1)}}, y^{(j)}\right)}\right],$$
sino $x_t = x_{t-1}$.

7. Actualizar $t = t + 1$ y volver al paso 1.

5.3.3. Generalización MTM (GMTM)

Una posible generalización del algoritmo MTM, es la conocida como *Generalization Multiple-Try Metropolis* (*GMTM*) [37]. Esta este método es una extensión del clásico MTM en el sentido que en este algoritmo *no se asume una forma analítica específica para los pesos* $w(x, y)$. Además, veremos que con la elección particular de los pesos
$$w(x, y) = p(x)\,\pi(y|x)\,\lambda(x, y),$$
con $\lambda(x, y) = \lambda(y, x)$, el GMTM [37] se reduce al clásico MTM [38].

Entonces, a continuación consideraremos los pesos como una función arbitraria de dos variables pero asume valores positivos, es decir,

$$w(x, y) > 0, \tag{5.3.5}$$

para cualquier $[x, y] \in \mathbb{R}^2$.[3] Así que el algoritmo GMTM se puede resumir en los siguientes pasos:

1. Elegir aleatoriamente un punto x_0 e inicializar $t = 1$.

2. Dada la muestra x_{t-1}, generar k muestras de la densidad tentativa $\pi(y|x_{t-1})$, obteniendo $y^{(1)}, ..., y^{(k)}$.

3. Calcular los pesos normalizados

$$\bar{w}_y^{(i)} = \frac{w\left(y^{(i)}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(y^{(i)}, x_{t-1}\right)}, \tag{5.3.6}$$

   con $i = 1, ..., k$, asociados a $y^{(1)}, ..., y^{(k)}$.

4. Seleccionar una muestra $y^{(j)} \in \left\{y^{(1)}, ..., y^{(k)}\right\}$, de acuerdo a los pesos $\bar{w}_y^{(i)}$, con $i = 1, ..., k$.

5. Muestrear las muestras $x^{(1)}, ..., x^{(k-1)}$ desde $\pi\left(x|y^{(j)}\right)$.

6. Construir el vector $\left[x^{(1)}, ..., x^{(k-1)}, x^{(k)}\right]$, donde $x^{(k)} = x_{t-1}$.

7. Calcular el peso normalizado (*sólo uno*, no se necesita más) correspondiente a la muestra $x^{(k)} = x_{t-1}$, es decir,

$$\bar{w}_x^{(k)} = \frac{w\left(x^{(k)}, y^{(j)}\right)}{\sum_{i=1}^{k} w\left(x^{(i)}, y^{(j)}\right)}.$$

[3] Aprovechamos la ocasión para recordar que, aunque hemos considerado variables escalares también a lo largo de este capítulo, en realidad todos los algoritmos descritos funcionan perfectamente con variables multidimensionales.

8. Calcular la probabilidad de aceptación (recordamos que $x_{t-1} = x^{(k)}$)

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)\bar{w}_x^{(k)}}{p\left(x_{t-1}\right)\pi\left(y^{(j)}|x_{t-1}\right)\bar{w}_y^{(j)}}\right]. \tag{5.3.7}$$

9. Aceptamos la muestra, $x_t = y^{(j)}$, con probabilidad $\alpha\left(x_{t-1}, y^{(j)}\right)$, sino $x_t = x_{t-1}$.

10. Actualizar $t = t + 1$ y volver al paso 2.

Es importante notar que la función de aceptación α en la Ecuación (5.3.7) interviene sólo dos pesos normalizados: $\bar{w}_y^{(j)}$que se refiere a la muestra $y^{(j)}$ seleccionada, y $\bar{w}_x^{(k)}$que se refiere a la muestra anterior de la cadena, es decir, $x^{(k)} = x_{t-1}$.

Además, es interesante reescribir la Ecuación (5.3.7) de esta forma (recordando que $x^{(k)} = x_{t-1}$)

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)w\left(x_{t-1}, y^{(j)}\right)}{p\left(x_{t-1}\right)\pi\left(y^{(j)}|x_{t-1}\right)w\left(y^{(j)}, x_{t-1}\right)}\frac{\sum_{i=1}^{k} w\left(y^{(i)}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)}, y^{(j)}\right)}\right]. \tag{5.3.8}$$

Es posible demostrar que el *kernel* del GMTM cumple la condición de balance o reversibilidad, siguiendo a los pasos similares a los que seguimos para MTM clásico.

Kernel y ecuación de balance del GMTM

A continuación vamos a escribir el kernel del GMTM para $x_{t-1} \neq y$ (para $x_{t-1} = y$ habría que añadir una delta de Dirac) y demostrar que cumple la condición de balance. Lo haremos por simplicidad para $k = 2$. Para este caso podemos escribir

$$\begin{aligned} K\left(y|x_{t-1}\right) =& 2\int_{\mathcal{S}}\int_{\mathcal{S}} \pi\left(y|x_{t-1}\right)\pi\left(y^{(2)}|x_{t-1}\right)\frac{w\left(y, x_{t-1}\right)}{w\left(y, x_{t-1}\right) + w\left(y^{(2)}, x_{t-1}\right)} \\ & \times \pi\left(x^{(1)}|y\right)\alpha\left(x_{t-1}, y\right)dy^{(2)}dx^{(1)}, \end{aligned}$$

donde hemos seguido los pasos del algoritmo. Podemos reescribirlo de esta forma

$$K\left(y|x_{t-1}\right) = 2\int_{\mathcal{S}}\int_{\mathcal{S}} \pi\left(y|x_{t-1}\right)\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right)\frac{w\left(y,x_{t-1}\right)}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)} \times \text{mín}\left[1,\frac{p\left(y\right)\pi\left(x_{t-1}|y\right)}{p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)}\frac{w\left(x_{t-1},y\right)}{w\left(y,x_{t-1}\right)}\frac{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)}{w\left(x^{(1)},y\right)+w\left(x_{t-1},y\right)}\right]dy^{(2)}dx^{(1)},$$

y seguimos simplificando

$$\begin{aligned} &K\left(y|x_{t-1}\right) = \\ &= 2\pi\left(y|x_{t-1}\right)w\left(y,x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right) \\ &\times \text{mín}\left[\frac{1}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)},\frac{p\left(y\right)\pi\left(x_{t-1}|y\right)}{p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)}\frac{w\left(x_{t-1},y\right)}{w\left(y,x_{t-1}\right)}\frac{1}{w\left(x^{(1)},y\right)+w\left(x_{t-1},y\right)}\right]dy^{(2)}dx^{(1)}. \end{aligned}$$

Ahora vamos a ver si el kernel (para $x_{t-1} \neq y$) cumple la ecuación de balance, para esto multiplicamos $p\left(x_{t-1}\right)$,

$$\begin{aligned} &p\left(x_{t-1}\right)K\left(y|x_{t-1}\right) = \\ &=2p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)w\left(y,x_{t-1}\right)\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right) \\ &\times \text{mín}\left[\frac{1}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)},\frac{p\left(y\right)\pi\left(x_{t-1}|y\right)}{p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)}\frac{w\left(x_{t-1},y\right)}{w\left(y,x_{t-1}\right)}\frac{1}{w\left(x^{(1)},y\right)+w\left(x_{t-1},y\right)}\right]dy^{(2)}dx^{(1)}, \end{aligned}$$

y llevando dentro los factores fuera de la integral

$$\begin{aligned} &p\left(x_{t-1}\right)K\left(y|x_{t-1}\right) = \\ &=2\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right) \\ &\times \text{mín}\left[\frac{p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)w\left(y,x_{t-1}\right)}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)},\frac{p\left(y\right)\pi\left(x_{t-1}|y\right)}{1}\frac{w\left(x_{t-1},y\right)}{1}\frac{1}{w\left(x^{(1)},y\right)+w\left(x_{t-1},y\right)}\right]dy^{(2)}dx^{(1)}, \end{aligned}$$

es decir, podemos reescribir como

$$\begin{aligned} &p\left(x_{t-1}\right)K\left(y|x_{t-1}\right) = \\ &2\int_{\mathcal{S}}\int_{\mathcal{S}}\pi\left(y^{(2)}|x_{t-1}\right)\pi\left(x^{(1)}|y\right) \\ &\times \text{mín}\left[\frac{p\left(x_{t-1}\right)\pi\left(y|x_{t-1}\right)w\left(y,x_{t-1}\right)}{w\left(y,x_{t-1}\right)+w\left(y^{(2)},x_{t-1}\right)},\frac{p\left(y\right)\pi\left(x_{t-1}|y\right)w\left(x_{t-1},y\right)}{w\left(x^{(1)},y\right)+w\left(x_{t-1},y\right)}\right]dy^{(2)}dx^{(1)}, \end{aligned}$$

donde se puede ver perfectamente que podemos *intercambiar* las variables x_{t-1} e y, así que podemos escribir

$$p\left(x_{t-1}\right) K\left(y|x_{t-1}\right) = p\left(y\right) K\left(x_{t-1}|y\right), \tag{5.3.9}$$

que es la ecuación de balance o reversibilidad.

Caso específico del GMTM

Nos parecen interesantes dos casos particulares del GMTM, que a nuestro juicio realmente le dan valor a esta generalización.

- *MTM como caso especial del GMTM*: la única condición que el GMTM impone a los pesos $w\left(x,y\right)$ es que sean positivos. Asumimos ahora que los pesos tienen la forma

$$w\left(x,y\right) = p\left(x\right)\pi\left(y|x\right)\lambda\left(x,y\right), \tag{5.3.10}$$

con $\lambda\left(x,y\right)$ función simétrica. Vamos a sustituir dentro de la Ecuación (5.3.8), es decir,

$$\alpha\left(x_{t-1},y^{(j)}\right) = \text{mín}\left[1, \frac{p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)w\left(x_{t-1},y^{(j)}\right)}{p\left(x_{t-1}\right)\pi\left(y^{(j)}|x_{t-1}\right)w\left(y^{(j)},x_{t-1}\right)}\frac{\sum_{i=1}^{k} w\left(y^{(i)},x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)},y^{(j)}\right)}\right],$$

los dos pesos $w\left(x_{t-1},y^{(j)}\right)$ y $w\left(y^{(j)},x_{t-1}\right)$ de la forma (5.3.10),

$$\alpha\left(x_{t-1},y^{(j)}\right) = \text{mín}\left[1, \underbrace{\frac{p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)p\left(x_{t-1}\right)\pi\left(y^{(j)}|x_{t-1}\right)\lambda\left(x_{t-1},y^{(j)}\right)}{p\left(x_{t-1}\right)\pi\left(y^{(j)}|x_{t-1}\right)p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)\lambda\left(y^{(j)},x_{t-1}\right)}}_{=1}\cdot\frac{\sum_{i=1}^{k} w\left(y^{(i)},x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)},y^{(j)}\right)}\right],$$

quedando finalmente

$$\alpha\left(x_{t-1},y^{(j)}\right) = \text{mín}\left[1, \frac{\sum_{i=1}^{k} w\left(y^{(i)},x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)},y^{(j)}\right)}\right], \tag{5.3.11}$$

que es exactamente la función de aceptación de MTM estándar.

- *MTM "inverso" como caso especial del GMTM*: esto ocurre con la siguiente elección particular de los pesos

$$w(x,y) = \frac{p(x)}{\pi(x|y)}. \tag{5.3.12}$$

De hecho, sustituyendo dos pesos dentro de

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)w\left(x_{t-1}, y^{(j)}\right)}{p(x_{t-1})\,\pi\left(y^{(j)}|x_{t-1}\right)w\left(y^{(j)}, x_{t-1}\right)}\frac{\sum_{i=1}^{k} w\left(y^{(i)}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)}, y^{(j)}\right)}\right],$$

logramos

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \underbrace{\frac{p\left(y^{(j)}\right)\pi\left(x_{t-1}|y^{(j)}\right)\frac{p(x_{t-1})}{\pi\left(x_{t-1}|y^{(j)}\right)}}{p(x_{t-1})\,\pi\left(y^{(j)}|x_{t-1}\right)\frac{p\left(y^{(j)}\right)}{\pi\left(y^{(j)}|x_{t-1}\right)}}}_{=1}\cdot\frac{\sum_{i=1}^{k} w\left(y^{(i)}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)}, y^{(j)}\right)}\right],$$

volviendo otra vez

$$\alpha\left(x_{t-1}, y^{(j)}\right) = \text{mín}\left[1, \frac{\sum_{i=1}^{k} w\left(y^{(i)}, x_{t-1}\right)}{\sum_{i=1}^{k} w\left(x^{(i)}, y^{(j)}\right)}\right],$$

que es otra vez la probabilidad de aceptación del MTM estándar. Entonces, claramente, si sustituimos ahora todos los pesos de la forma anteriormente elegida logramos exactamente el MTM "inverso".

5.3.4. Posible mejoras y ulteriores estudios

Durante el desarrollo de este trabajo hemos reflexionado sobre futuras posibles mejoras y análisis del MTM. A continuación, mostramos una pequeña lista de estas posibles estudios:

- Una posible mejora computacional (posiblemente ya existente en la literatu-

ra, pero nosotros no la hemos encontrado) consiste en reciclar $k-1$ puntos generados en el paso 6 del MTM. De hecho, recordamos que las muestras $\left[y^{(1)},...,y^{(k)}\right]$ se generan desde $\pi\left(\cdot|x_{t-1}\right)$, mientras que las muestras de referencia $\left[x^{(1)},...,x^{(k-1)}\right]$ desde $\pi\left(\cdot|y^{(j)}\right)$ (mirar paso 6 del MTM). Si la muestra $y^{(j)}$ se acepta, es decir $x_t = y^{(j)}$, volviendo al paso 2 cuando $t = t+1$, vamos a tener $x_{t-1} = y^{(j)}$. Así que $\pi\left(\cdot|x_{t-1}\right) = \pi\left(\cdot|y^{(j)}\right)$, y podríamos reciclar las $k-1$ muestras $\left[y^{(1)} = x^{(1)},...,y^{(k-1)} = x^{(k-1)}\right]$ del paso anterior. Sólo necesitaríamos muestrear otro punto $y^{(k)}$.

- La elección del número k es crucial para el buen funcionamiento del algoritmo. Se podría diseñar un algoritmo donde el número k se elija y cambie de forma adaptativa.

- De forma parecida a lo que ocurre con las muestras correlacionadas se podría intentar mejorar adaptativamente la fdp tentativa utilizando las muestras generadas previamente.

Capítulo 6

Métodos basados en población

Como ya hemos mencionado, los métodos basados en *población* (término acuñado en [63])son otros de los métodos existentes para solventar unos problemas de las cadenas generadas por el algoritmo MH como, por ejemplo, que se queden enganchadas en una moda de la densidad objetivo. En estas técnicas intentan ayudarse en la exploración de todo el espacio de estado ayudando a la convergencia de la cadena.

La idea básica es siempre ejecutar en paralelo una población (un grupo) de cadenas de Markov. *Puede ser* que cada una de ellas tenga una distribución invariante *diferente*, pero una de ellas será siempre $p_o(x)$ (nuestra densidad objetivo). El intercambio de información entre las diferentes cadenas se puede proporcionar de formas distintas: utilizando pesos entre las muestras, mediante intercambio de parámetros o variables auxiliares, etc.

Matemáticamente, podemos considerar que estos métodos generen realizaciones aleatorias desde la siguiente densidad conjunta,

$$f(x_1, ..., x_N) = \prod_{i=1}^{N} q_i(x_i), \tag{6.0.1}$$

donde *por lo menos* por un $j \in \{1, ..., N\}$ tenemos

$$q_j(x_j) = p_o(x). \tag{6.0.2}$$

Es importante evidenciar que estos algoritmos muestrean en cada paso de forma *independiente* N muestras $\left[x_1^{(1)}, ..., x_N^{(N)}\right]$ desde la correspondiente $q_i(x_i)$, $i = 1, ..., N$, y luego en el intercambio de información se hace en un paso siguiente (antes de otro paso de muestreo).

A continuación, nos referiremos con el término *población* al conjunto de muestras como

$$\mathcal{P} = \left\{x_1^{(1)}, ..., x_N^{(N)}\right\}, \tag{6.0.3}$$

siendo N el tamaño de la población.

En este capítulo, hemos seleccionado dentro de la literatura existente, dos algoritmos que nos han parecido más interesantes y fáciles de comprender:

- El algoritmo *Sample Metropolis-Hastings* donde el comportamiento de un candidato es comparado con las otras muestras mediante pesos (siguiendo la filosofía del importance sampling).
- El algoritmo *Parallel Tempering* en el que adjudicamos una variable que conoceremos como *temperatura* a las cadenas de Markov. De esta forma, conseguimos dos configuraciones, a alta y a bajas temperaturas (alta y baja *variabilidad*).

Otros métodos de población conocidos son: *Adaptative Direction Sampling* [40, 41], *Conjugate Gradient Monte Carlo* [38], *Evolutionary Monte Carlo* [56], etc.

6.1. Sample Metropolis-Hastings Algorithm (SMH)

En este algoritmo [67] en su *versión más básica*, todas las densidad $q_i(x_i)$, $i = 1, ..., N$, en la Ecuación (6.0.1) son iguales a la densidad objetivo $p_o(x)$. En definitiva, todas las realizaciones dentro de la población

$$\mathcal{P}_t = \left\{x_{1,t}^{(1)}, ..., x_{N,t}^{(N)}\right\},$$

son N muestras de $p_o(\cdot)$. Entonces, por simplicidad de notación (siendo $q_i(x_i) = p_o(x)$, $i = 1, ..., N$) vamos a indicar la población de la siguiente forma

$$\mathcal{P}_t = \left\{ x_t^{(1)}, ..., x_t^{(N)} \right\}, \tag{6.1.1}$$

Por comodidad, definimos los pesos ***"inversos"*** respecto a la filosofía del importance sampling

$$w(x) \triangleq \frac{1}{\gamma(x)} = \frac{\pi(x)}{p(x)}, \tag{6.1.2}$$

donde $\pi(x)$ es nuestra fdp tentativa, $p(x) \propto p_o(x)$ es una función proporcional a nuestra densidad objetivo $p_o(x)$, y $\gamma(x)$ es el peso definido según importance sampling. Estos pesos $w(x)$ indican cuanto ***"mala"*** es la muestra x.

Una versión *simplificada* del algoritmo consta de estos pasos:

1. Consideramos $t = 0$, y elegimos aleatoriamente N puntos de la población inicial

$$\mathcal{P}_0 = \left\{ x_0^{(1)}, ..., x_0^{(N)} \right\}. \tag{6.1.3}$$

2. Muestreamos la fdp tentativa $\pi(x)$ obteniendo un candidato x', y además, indicamos por comodidad $x_t^{(0)} = x'$. Nótese que ahora tenemos $x_t^{(0)} = x', x_t^{(1)}, ..., x_t^{(N)}$.

3. Calculamos el probabilidad de aceptación

$$\begin{aligned} \alpha_t^{(0)} &= \frac{\sum_{i=1}^{N} \frac{\pi\left(x_t^{(i)}\right)}{p\left(x_t^{(i)}\right)}}{\sum_{i=0}^{N} \frac{\pi\left(x_t^{(i)}\right)}{p\left(x_t^{(i)}\right)} - \text{mín}\left[\frac{\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)}, ..., \frac{\pi\left(x_t^{(N)}\right)}{p\left(x_t^{(N)}\right)}\right]} \\ &= \frac{\sum_{i=1}^{N} w(x_t^{(i)})}{\sum_{i=0}^{N} w(x_t^{(i)}) - \text{mín}\left[w(x_t^{(0)}), ..., w(x_t^{(N)})\right]}, \end{aligned} \tag{6.1.4}$$

donde es muy importante notar que al numerador *no* se tiene en cuenta $x_t^{(0)} = x'$, mientras que en el dominador sí. Además, con simples consideraciones se

puede enseñar que

$$0 \leq \alpha_t^{(0)} \leq 1.$$

4. Elegimos un índice $j \in \{1, ..., N\}$ de acuerdo a los pesos normalizados

$$\bar{w}^{(i)} = \frac{w(x^{(i)})}{\sum_{i=1}^{N} w(x^{(i)})}, \tag{6.1.5}$$

con $i = 1, ..., N$. Es importante notar que estamos eligiendo aleatoriamente una muestra de *mala calidad* estadística.

5. Aceptamos la *nueva población*

$$\mathcal{P}_{t+1} = \left\{ x_t^{(i)}, ..., x_t^{(j-1)}, x_t^{(0)}, x_t^{(j+1)}, ...x_t^{(N)} \right\}, \tag{6.1.6}$$

con probabilidad $\alpha_t^{(0)}$. Nótese que hemos sustituido $x_t^{(j)}$ con $x_t^{(0)} = x'$. Mientras con probabilidad $1 - \alpha_t^{(0)}$ consideraremos

$$\mathcal{P}_{t-1} = \mathcal{P}_t. \tag{6.1.7}$$

Nótese que $\mathcal{P}_{t-1}$ difiere como mucho en un elemento con $\mathcal{P}_t$.

6. Actualizamos $t = t + 1$ y volvemos al paso 2.

En este algoritmo generamos primero una posible nueva muestra $x_t^{(0)} = x'$ fdp tentativa. Luego calculamos una probabilidad de aceptación $\alpha_t^{(0)}$de la siguiente forma:

- En el numerador sumamos todos los pesos $w(x)$ relativos a las muestras anteriores en el conjunto $\mathcal{P}_t$. Esta suma es mayor si las muestras anteriores tienen pesos $w(x)$ más altos, es decir, si las muestras son "malas". De la misma forma, el numerador será menor si las muestras son "importantes" estadísticamente.

- En el denominador sumamos todos los pesos de las muestras anteriores más el peso asociado a la nueva muestra $x_t^{(0)} = x'$ y le restamos el peso de la muestra más "importante" (que en este caso corresponde a la muestra con menor peso

$w(x)$) entre todas la que componen la suma anterior. El denominador será más grande si las componentes de este nuevo conjunto son "malas" (de poca importancia estadística), mientras que el denominador sera más pequeño si el nuevo conjunto está compuesto por muestras "importantes".

Lo dicho anteriormente significa que la probabilidad de aceptación $\alpha_t^{(0)}$ realmente compara la ***calidad*** de las muestras en los dos conjuntos siguientes

$$\mathcal{P}_t = \left\{ x_t^{(1)}, ..., x_t^{(N)} \right\},$$

y otro conjunto

$$\mathcal{P}_t^* = \left\{ x_t^{(1)}, ..., \underbrace{x_t^{(0)}}_{k}, ..., x_t^{(N)} \right\},$$

donde

$$k = \underset{i=0,...,N}{\operatorname{argmin}} \frac{\pi\left(x_t^{(i)}\right)}{p\left(x_t^{(i)}\right)} \tag{6.1.8}$$

es la posición de la muestra con mínimo peso (más importancia). Si $k = 0$ realmente no sustituimos ninguna muestra $\mathcal{P}_t^* = \mathcal{P}_t$. Este es el caso que el peso $w\left(x_t^{(0)}\right)$ es el *menor* de todos, es decir, la nueva muestra generada es la de más importancia. Es importante notar que en este caso,$\mathcal{P}_t^* = \mathcal{P}_t$, la probabilidad de aceptación $\alpha_t^{(0)}$ es exactamente igual a 1, y la nueva muestra $x_t^{(0)}$ será seguramente incorporada al nuevo conjunto $\mathcal{P}_{t+1}$, sustituyendo con alta probabilidad una muestra con un peso $w(x)$ elevado, es decir una muestra de poca importancia. Entonces, es fácil concluir que el nuevo conjunto $\mathcal{P}_{t+1}$ contendrá muestras más "importantes" estadísticamente según la filosofía de importance sampling, respecto al conjunto anterior $\mathcal{P}_t$. Además, siempre es posible demostrar que la nueva muestra $x_t^{(0)}$ se distribuye como la densidad objetivo $p_o(x)$ (el lector interesado puede mirar [67]).

Observación importante: a diferencia de otras generalizaciones y otros métodos de población, en la versión básica presentada aquí, todas las componentes de los conjuntos $\mathcal{P}_t$ son muestras de la nuestra densidad objetivo $p_o(x)$.

Existen diferentes extensiones de esta versión básica presentada aquí, por ejemplo

con distintas $q_i(x_i)$, con $i = 1, ..., N$. Pero estas distintas extensiones serán objetos de otros trabajos futuros.

Es fácil de ver que en el caso de $N = 1$, SMH se reduce al algoritmo tradicional MH con fdp tentativas independientes.

6.1.1. Ejemplo caso específico: algoritmo Metropolis-Hastings

Cuando $N = 1$, la función de aceptación queda como

$$\alpha_t^{(0)} = \frac{\frac{\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)}}{\frac{\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)} + \frac{\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)} - \text{mín}\left[\frac{\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)}, \frac{\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)}\right]}. \tag{6.1.9}$$

Ahora multiplicamos numerador y denominador $\frac{p\left(x_t^{(1)}\right)}{\pi\left(x_t^{(1)}\right)}$, obteniendo

$$\begin{aligned}\alpha_t^{(0)} &= \frac{1}{\frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{\pi\left(x_t^{(1)}\right)p\left(x_t^{(0)}\right)} + \frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(1)}\right)}{\pi\left(x_t^{(1)}\right)p\left(x_t^{(1)}\right)} - \frac{p\left(x_t^{(1)}\right)}{\pi\left(x_t^{(1)}\right)}\,\text{mín}\left[\frac{\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)}, \frac{\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)}\right]} \\ &= \frac{1}{\frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{\pi\left(x_t^{(1)}\right)p\left(x_t^{(0)}\right)} + 1 - \frac{p\left(x_t^{(1)}\right)}{\pi\left(x_t^{(1)}\right)}\,\text{mín}\left[\frac{\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)}, \frac{\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)}\right]},\end{aligned}$$

ahora podemos introducir el factor $\frac{p\left(x_t^{(1)}\right)}{\pi\left(x_t^{(1)}\right)}$ que multiplica a la función mín $[\cdot, \cdot]$ dentro de esta

$$\begin{aligned}\alpha_t^{(0)} &= \frac{1}{\frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{\pi\left(x_t^{(1)}\right)p\left(x_t^{(0)}\right)} + 1 - \text{mín}\left[\frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)\pi\left(x_t^{(1)}\right)}, \frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)\pi\left(x_t^{(1)}\right)}\right]} \\ &= \frac{1}{\frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{\pi\left(x_t^{(1)}\right)p\left(x_t^{(0)}\right)} + 1 - \text{mín}\left[\frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)\pi\left(x_t^{(1)}\right)}, 1\right]}.\end{aligned}$$

Si definimos

$$\frac{1}{a} = \frac{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}{p\left(x_t^{(0)}\right)\pi\left(x_t^{(1)}\right)}, \tag{6.1.10}$$

entonces podemos reescribir la ecuación de arriba como

$$\alpha_t^{(0)} = \frac{1}{\frac{1}{a} + 1 - \text{mín}\left[1, \frac{1}{a}\right]}. \tag{6.1.11}$$

Ahora tenemos dos opciones, $\frac{1}{a} > 1$ ó $\frac{1}{a} \leq 1$. Vamos a analizar los dos casos:

- si $\frac{1}{a} > 1 \Rightarrow \alpha_t^{(0)} = \frac{1}{\frac{1}{a}+1-1} = a = \frac{p\left(x_t^{(0)}\right)\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}$,
- si $\frac{1}{a} \leq 1 \Rightarrow \alpha_t^{(0)} = \frac{1}{\frac{1}{a}+1-\frac{1}{a}} = 1$.

Por lo tanto podemos reescribir la función de aceptación para $N = 1$ como

$$\alpha\left(x_t^{(0)}, x_t^{(1)}\right) = \text{mín}\left[1, \frac{p\left(x_t^{(0)}\right)\pi\left(x_t^{(1)}\right)}{p\left(x_t^{(1)}\right)\pi\left(x_t^{(0)}\right)}\right],$$

que es exactamente la función de aceptación del algoritmo Metropolis-Hastings estándar.

6.2. Parallel Tempering

El *Parallel Tempering*, también conocido como *replica exchange MCMC sampling*, es otra técnica avanzada MCMC donde se utilizan una variable auxiliar, denominada *temperatura*. Este método es "hijo" de la técnica *simulated tempering* muy utilizada en optimización estocástica.

Las principales contribuciones sobre este método se pueden encontrar en [43, 44, 45, 46]. Se desarrolló una versión molecular dinámica de este método de la mano de Sugita y Okamoto [47].

Antes de describir el algoritmo parallel tempering, vamos a resumir rápidamente dos metodologías de muestreo y de optimización muy similares, *simulate annealing* y *simulate tempering*.

Simulated Annealing

Este algoritmo recibe su nombre de la analogía con el fenómenos físico de recocido ("annealing") del metal. Mientras que a muy altas temperaturas las moléculas del metal en estado líquido se mueven libremente, según cae la temperatura la movilidad de las moléculas decae hasta convertirse en un cristal puro que se corresponde con el estado de mínima energía posible. Si la temperatura baja demasiado rápido, el cristal generado no es un cristal puro sino policristalino, produciendo una menor entropía en el material que le confiere una mayor energía comparándola con el cristal puro.

Esta técnica puede utilizarse:

1. para muestreo (generar números aleatorios desde una fdp objetivo),
2. o para optimización,

pero su fama se debe seguramente a esta segunda aplicación.

La función de energía $E(x)$ correspondiente al fenómeno físico representa el papel de la función de coste $f(x)$ en los problemas de optimización estocástica, y el estado del cristal puro se corresponde con el mínimo global óptimo. En cambio, el estado policristalino correspondería con un mínimo local pero no global de la función de coste. Igual que en parallel tempering, la *temperatura* es un parámetro del algoritmo que nos permitirá recorrer en menor o en mayor medida el espacio de estado para así poder encontrar mínimos o máximos globales.

La idea fundamental consiste en una *variación del algoritmo MH clásico*.

1. Consideremos ahora el caso de aplicar el simulated annealing para muestreo: generamos candidatos aleatorios desde una fdp tentativa y aceptamos el movimiento "hacia arriba" con probabilidad 1, mientras que los movimientos "hacia abajo" se aceptarán con una determinada probabilidad dependiente de la variable temperatura T. Si el algoritmo de muestreo está bien diseñado, en cada paso

esta variable T debe disminuir de forma decreciente hasta llegar al valor $T = 1$ y la correspondiente probabilidad de aceptación de los movimientos "hacia abajo" debe resultar igual a la que daría un algoritmo MH clásico. Desde el punto de vista del muestreo este algoritmo considera diferentes densidades objetivos, usualmente con mayor varianza, hasta que $T = 1$ donde la densidad objetivo debería corresponder a $p_o(x)$.

2. En el caso de la optimización, la variable temperatura T (que juega un papel similar a la varianza) tiene que tender a 0, $T \to 0$, para que la cadena generada por el algoritmo se quede enganchada alrededor de las modas de la densidad correspondiente a la función de coste que se quiera analizar.

La variación de la temperatura se realiza mediante algún tipo de función determinista ϕ.

Simulated Tempering

Esta técnica fue propuesta en [53], elimina la dependencia de una función determinista específica ϕ para actualizar la temperatura T_t, utilizando una cadena de Markov también para variar la temperatura. Esta última se actualiza, entonces muestreando una densidad condicional

$$T_t \sim g(T_t|T_{t-1}), \tag{6.2.1}$$

que tiene que cumplir determinadas condiciones [53].

6.2.1. Algoritmo

Hemos dicho anteriormente que con el método de población generamos realizaciones aleatorias desde la siguiente densidad conjunta

$$f(x_1, ..., x_N) = \prod_{i=1}^{N} q_i(x_i). \tag{6.2.2}$$

En este caso específico, vamos a definir una función energía asociada a la densidad objetivo $p_o(x)$ como

$$H(x) \triangleq -\log p_o(x), \tag{6.2.3}$$

así que nuestra densidad podrá escribirse como

$$p(x) \propto p(x) = \exp\{-H(x)\}. \tag{6.2.4}$$

Además, vamos a definir

$$q_i(x_i) \propto \exp\left(-\frac{H(x)}{T_i}\right), \tag{6.2.5}$$

para $i = 1, ..., N$, y los parámetros *fijos* temperatura (elegidos por nosotros) están sujetos a estas desigualdades

$$T_1 > T_2 > ... > T_{n-1} > T_n \equiv 1. \tag{6.2.6}$$

Es muy importante evidenciar que la $q_N(x_N)$ coincide exactamente con la fdp objetivo, es decir

$$q_N(x_N) \equiv p_o(x). \tag{6.2.7}$$

Por último vamos a indicar la población en el tiempo t de esta forma

$$\mathcal{P}_t = \left\{x_{1,t}^{(1)}, ..., x_{N,t}^{(N)}\right\}, \tag{6.2.8}$$

donde el primer subíndice indica la fdp considerada "objetivo" para esta muestra, $q_i(x_i)$, con $i = 1, ..., N$.

Dadas estas definiciones, el algoritmo consta de los siguientes pasos:

1. Consideramos $t = 0$, y elegimos aleatoriamente N puntos de la población inicial

$$\mathcal{P}_0 = \left\{x_{1,0}^{(1)}, ..., x_{N,0}^{(N)}\right\}. \tag{6.2.9}$$

2. Para cada componente del conjunto $\mathcal{P}_t$, generamos N distintas cadenas de Markov utilizando cada una un algoritmo MH clásico con densidad objetivo

$q_i(x_i)$, con $i = 1, ..., N$. Las densidades tentativas

$$\pi_i(x_{i,t}|x_{i,t-1}), \tag{6.2.10}$$

con $i = 1, ..., N$, puede ser distinta o iguales. Podemos resumir este paso diciendo que pasamos desde $x_{i,t-1}^{(i)}$ a $x_{i,t}^{(i)}$, $i = 1, ..., N$, usando un algoritmo MH con densidad estacionaria $q_i(x_i)$. Así que, generamos en este paso otro conjunto

$$\mathcal{P}_{t+1}^* = \left\{x_{1,t+1}^{(1)}, ..., x_{N,t+1}^{(N)}\right\}. \tag{6.2.11}$$

3. En este paso intentamos cambiar la posición de cada elemento $x_{i,t+1}^{(i)}$, $i = 1, ..., N$, contenido en $\mathcal{P}_{t+1}^*$, con sus componentes adyacentes. Para cada $j = 1, ..., N$, vamos a repetir los siguientes pasos:

   a) Si $j = 2, ..., N-1$, intercambiamos $x_{j,t+1}^{(j)}$ con la muestra anterior $x_{j-1,t+1}^{(j-1)}$ con probabilidad 0,5 (y definimos $k = j-1$) o con la muestra siguiente $x_{j+1,t+1}^{(j+1)}$ con probabilidad 0,5 (y definimos $k = j+1$).

   b) Si $j = 1$, intercambiamos $x_{1,t+1}^{(1)}$ con la muestra siguiente $x_{2,t+1}^{(2)}$ con probabilidad 1 (y definimos $k = 2$).

   c) Si $j = N$, intercambiamos $x_{N,t+1}^{(N)}$ con la muestra anterior $x_{N-1,t+1}^{(N-1)}$ con probabilidad 1 (y definimos $k = N-1$).

   d) ***Cada intercambio tiene que aceptarse con probabilidad***

$$\begin{aligned}\alpha &= \text{mín}\left[1, \frac{q_j\left(x_{j,t+1}^{(j)}\right)}{q_k\left(x_{k,t+1}^{(k)}\right)}\right] \\ &= \text{mín}\left[1, \exp\left\{\left[H\left(x_{j,t+1}^{(j)}\right) - H\left(x_{k,t+1}^{(k)}\right)\right]\left[\frac{1}{T_j} - \frac{1}{T_k}\right]\right\}\right].\end{aligned} \tag{6.2.12}$$

   Al final, creamos un conjunto $\mathcal{P}_{t+1}$ según los intercambios previamente aceptados.

4. Actualizamos $t = t+1$ y volvemos al paso 2.

Observación importante: es crucial entender que con este algoritmo *sólo* las muestras $x_{N,t}^{(N)}$ se distribuyen como la nuestra densidad objetivo $p_o(x)$. Las demás muestras del conjunto $\mathcal{P}_t$ sirven para aportar información sobre otras regiones del espacio de estado y ayudar a "desatascar" la cadena de otras regiones donde puede haberse quedado atrapada. Otra reflexión importante es que las temperaturas actúan como varianzas y en general, siendo mayor que 1, favorecen la exploración de distintas regiones del espacio de estado.

En la práctica, para obtener buenas tasas de aceptación, las temperaturas T_i, $i = 1, ..., N$, deben ser cuidadosamente seleccionadas. No se conocen esquemas claros de elección de las temperaturas T_i. Pero es evidente que si interpretamos las diferentes cadenas como "exploradores" las temperaturas influyen y deciden en las velocidades de salto y los tiempos de estancia en una determinada región. A temperaturas más altas corresponden "partículas" más "volátiles", y temperaturas más bajas corresponden "partículas" que podríamos definir "más observadoras", cuyo tiempo de estancia en una región es más alto y aportan más información detallada sobre esta porción del espacio de estado.

La técnica de parallel tempering ha sido aplicada con gran éxito en la simulación de sistemas complejos, tales como vidrio de espín [48, 49][1] y las simulaciones de polímeros [51, 52].

[1]Un vidrio de espín (en inglés, *spin glass*) es un sistema magnético en el que el acoplamiento entre los momentos magnéticos de los distintos átomos es aleatorio, tanto ferromagnético como antiferromagnético y presenta un fuerte grado de frustración [50].

Parte III

Simulaciones y conclusiones

Cuando estás solucionando un problema, «no te preocupes».
Ahora, «después» de que has resuelto el problema «es el momento de preocuparse».
(Richard Feynman)

Capítulo 7

Comparación de algoritmos

7.1. Introducción

En este capítulo, vamos a comparar el algoritmo Metropolis-Hastings clásico visto en el Capítulo 3 con el algoritmo Multiple-Try Metropolis visto en el Capítulo 5, utilizando diferentes valores de los parámetros asociados. No vamos a comparar los métodos introducidos en el Capítulo 6 por falta de tiempo y recursos. Así mismo, no hemos podido recorrer todas las posibles combinaciones de variables del algoritmo MTM. Además, para un análisis completo necesitaríamos comparar todos las técnicas también con un conjunto de algoritmos MH independientes en paralelo. Así que estos serán objetos de unos trabajos futuros.

Claramente, tanto el algoritmo MH clásico como las técnicas MTM (con distintos parámetros y pesos), después de un transitorio, generan muestras distribuidas de acuerdo a nuestra densidad objetivo. Por ejemplo, en la Figura 7.1.1 podemos observar el histograma obtenido con $1,000,000$ realizaciones de una cadena de Markov generada con el algoritmo MH. Podemos notar como el histograma se ajusta exactamente a la curva definida por la densidad objetivo.

Además, en la Figura 7.1.2, se ilustra la serie temporal asociada a las $5,000$ primeras muestras generadas anteriormente, es decir, las correspondientes a la Figura 7.1.1. Debido a la correlación existente, podemos notar como la cadena se queda

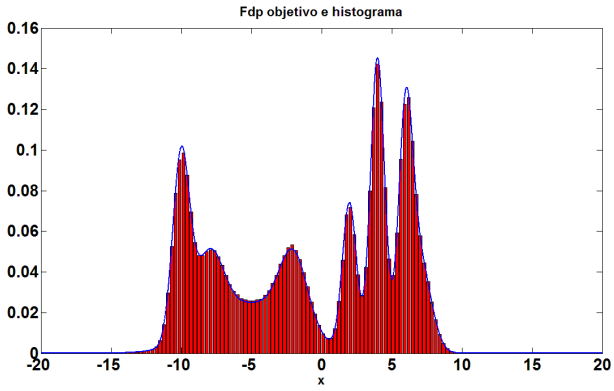

Figura 7.1.1: La curva *azul* es la densidad objetivo que deseamos generar. El histograma de las muestras obtenidas mediante el algoritmo MH se encuentra en *rojo*.

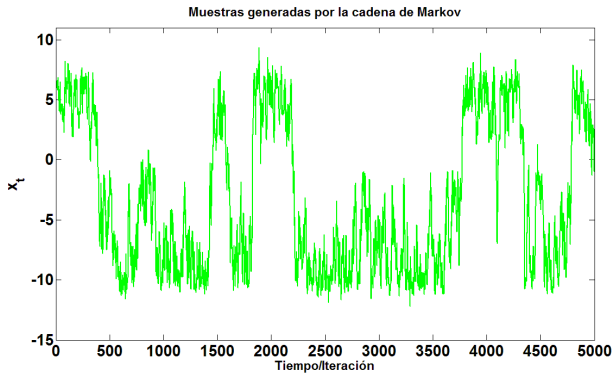

Figura 7.1.2: Camino recorrido de una cadena de Markov en un algoritmo MH. Podemos observar como la cadena produce *"saltos"* entre modas.

cerca de una moda por un cierto tiempo, produciendo luego ***"saltos"*** entre regiones distintas de alta probabilidad.

A continuación, vamos a describir las componentes principales que componen nuestras simulaciones.

7.1.1. Función objetivo y función tentativa

La función objetivo $p_o(x)$, de donde queremos sacar muestras, será

$$\begin{aligned} p_o(x) =& \frac{3}{22}\mathcal{N}(x;-13,1) + \frac{1}{11}\mathcal{N}(x;-7,0.25) + \frac{2}{11}\mathcal{N}(x;-4,0.01) + \frac{1}{11}\mathcal{N}(x;-2,0.09) \\ &+ \frac{1}{11}\mathcal{N}(x;2,0.09) + \frac{2}{11}\mathcal{N}(x;4,0.01) + \frac{1}{11}\mathcal{N}(x;7,0.25) + \frac{3}{22}\mathcal{N}(x;13,1), \end{aligned} \tag{7.1.1}$$

donde con $\mathcal{N}(x;\mu,\sigma^2)$ indicamos una fdp Gaussiana de media μ y varianza σ^2. Por comodidad, hemos definido la densidad objetivo ***simétrica***, así que sabemos que su esperanza matemática (su media) es exactamente 0, sin necesidad de calcular de aproximar o calcular analíticamente integrales.

Observación: es importante notar que para generar muestras de una mezcla de Gaussianas no es necesario utilizar un método MCMC. Aquí sólo queremos comparar las prestaciones de los diferentes casos de MTM.

La función tentativa $\pi(\cdot|\cdot)$ con la que trabajaremos se comportará como un camino aleatorio o *random walk* (subsección 3.4.4), es decir, su media dependerá de la muestra anteriormente generada. En nuestras simulaciones, $\sigma = 2$, por lo tanto generaremos la distribución Gaussiana

$$\pi(x_t|x_{t-1}) = \mathcal{N}(x_t;x_{t-1},4) \propto \exp\left(-\frac{(x_t - x_{t-1})^2}{8}\right). \tag{7.1.2}$$

Más en general, consideramos

$$\pi(x_t|x_{t-1}) = \mathcal{N}\left(x_t;x_{t-1},\sigma^2\right) \propto \exp\left(-\frac{(x_t - x_{t-1})^2}{2\sigma^2}\right), \tag{7.1.3}$$

cuando sea necesario modificar la varianza la fdp tentativa. En la Figura 7.1.3 podemos observar la función objetivo y la función tentativa.

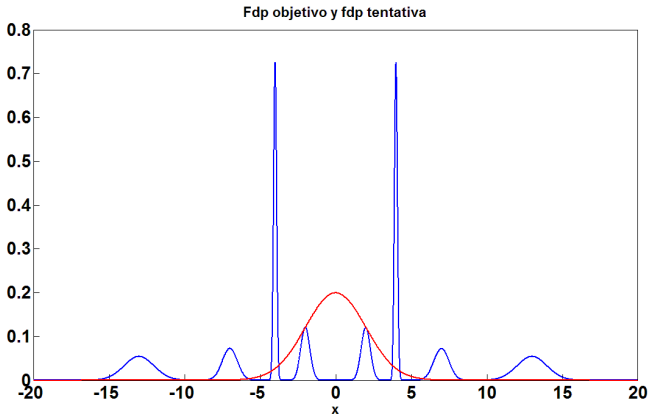

Figura 7.1.3: En *azul* podemos observar la función objetivo descrita en la Ecuación (7.1.1). En *rojo* podemos observar la función tentativa descrita en la Ecuación (7.1.2).

7.2. Simulaciones

Consideramos a continuación 4 diferentes tipos de simulaciones:

1. Fijando todos los parámetros, medimos las prestaciones del algoritmo MTM (y del MH como caso especial) con diferentes funciones $\lambda(x, y)$ (recordamos que afectan a los pesos).

2. Fijando todo el resto de funciones y parámetros, medimos las prestaciones con diferentes valores de k puntos.

3. Analizaremos específicamente el *transitorio* de diferentes algoritmos MTM.

4. Estudiaremos las prestaciones de los algoritmos con diferentes varianzas de la función tentativa, para ver como afecta a la convergencia y a la capacidad de recorrer el espacio de estado.

Para comparar los datos obtenidos, vamos a medir dos valores:

- estimación de la *esperanza matemática* (media) de los datos obtenidos (el valor real es 0),

- y número de *saltos* entre modas de la cadena de Markov.

El valor real de la esperanza de los datos será la media de la densidad objetivo, Ecuación (7.1.1), que como ya hemos mencionado es 0 dado que la fdp objetivo es simétrica,

$$p_o(x) = p_o(-x). \tag{7.2.1}$$

En cambio, el número de saltos producidos en la cadena será mejor cuanto más alto sea, disminuyendo de esta forma la correlación de los datos.

Observación: Para dar mayor fiabilidad a los resultados, la primera muestra generada en cada lanzamiento del algoritmo, proviene de $\mathcal{N}(x;\mu,4)$, siendo μ una muestra aleatoria proveniente de una distribución uniforme entre los valores -10 y 10, es decir, $\mu \sim \mathcal{U}([-10,10])$.

7.2.1. Diferentes funciones $\lambda(x,y)$

En la Tabla 7.1 presentaremos la media de los datos obtenidos sobre $10,000$ ejecuciones del algoritmo MTM clásico (con 5 puntos propuestos en cada iteración, cadena de $3,000$ muestras y función tentativa $\pi(x|\mu) \sim \mathcal{N}(x;\mu,4)$) con diferentes funciones $\lambda(x,y)$.

Función $\lambda(x,y)$	Promedio de nº de saltos	Error absoluto medio	Error cuadrático medio
$\lambda(x,y) = 1$	791.37	3.36	17.92
$\lambda(x,y) = \frac{1}{\pi(x\|y)+\pi(y\|x)}$	791.48	3.33	17.69
$\lambda(x,y) = \frac{1}{\pi(x\|y)\cdot\pi(y\|x)}$	789.87	3.37	18.01

Cuadro 7.1: Comparativa del funcionamiento del algoritmos MTM con diferentes funciones $\lambda(x,y)$.

Como podemos observar, los datos obtenidos entre $\lambda = 1$ y $\lambda = \frac{1}{\pi(x|y)+\pi(y|x)}$ son muy similares, pero $\lambda = \frac{1}{\pi(x|y)+\pi(y|x)}$ da mejor resultado. Para $\lambda = \frac{1}{\pi(x|y)\cdot\pi(y|x)}$ los resultados no son tan buenos, pero no se aleja mucho de los otros dos.

7.2.2. Diferentes número k de puntos

A continuación, en la Tabla 7.2 presentaremos la media de los datos obtenidos sobre $5,000$ ejecuciones[1] del algoritmo MTM clásico (con función $\lambda = 1$, cadena de $3,000$ muestras y función tentativa $\pi(x|\mu) \sim \mathcal{N}(x; \mu, 4)$) con diferentes número de puntos propuestos en cada iteración.

Nº de puntos propuestos	Promedio de nº de saltos	Error absoluto medio	Error cuadrático medio
$k = 1$	680.69	2.77	11.66
$k = 2$	719.54	2.87	12.85
$k = 5$	794.17	3.31	17.29
$k = 10$	847.19	3.54	19.66
$k = 20$	882.96	3.67	21.11

Cuadro 7.2: Comparativa del funcionamiento del algoritmos MTM con diferentes puntos generados. Recordemos que para $k = 1$ el algoritmo MTM se reduce a Metropolis-Hastings clásico.

Como observamos, el número de saltos que se producen en la cadena aumenta cuanto mayor sea el número k. Pero hay que evidenciar que, de manera sorprendente, el error en la estimación de la esperanza matemática aumenta al aumentar el número k.

7.2.3. Análisis del transitorio

En un penúltimo análisis, vamos a presentar en la Tabla 7.3 los datos de unas simulaciones MTM con sólo 500 muestras. Con estas simulaciones queremos ver el transitorio del algoritmo MTM. Para ello, vamos a disponer del análisis de los datos con $k = 2$, $k = 5$, $\lambda = 1$ y $\lambda = \frac{1}{\pi(x|y)+\pi(y|x)}$ y las combinaciones posibles entre estos (la varianza de la función tentativa es de 4, es decir, $\pi(x|\mu) \sim \mathcal{N}(x; \mu, 4)$). Los datos están promediados respecto a $5,000$ ejecuciones.

Tenemos que recordar que el número de saltos se ha medido sobre 500 muestras y, por lo tanto, estos resultados no son comparables con los resultados anteriormente

[1] Hemos reducido de $10,000$ a $5,000$ ejecuciones por problemas de sobrecalentamiento del equipo.

Caso MTM	Promedio de nº de saltos	Error absoluto medio	Error cuadrático medio
$k = 2$ y $\lambda = 1$	122.85	4.40	30.39
$k = 2$ y $\lambda = \frac{1}{\pi(x\|y)+\pi(y\|x)}$	124.19	4.37	29.88
$k = 5$ y $\lambda = 1$	140.62	4.26	30.05
$k = 5$ y $\lambda = \frac{1}{\pi(x\|y)+\pi(y\|x)}$	140.75	4.39	31.45

Cuadro 7.3: Comparativa del transitorio del algoritmo MTM.

dados. Como ya hemos destacado anteriormente, el número de saltos aumenta con el número k.

7.2.4. Diferentes varianzas de la función tentativa

Para finalizar, vamos a presentar los resultados obtenidos del algoritmo MTM (con $k = 5$, cadena de $3,000$ muestras y $\lambda = \frac{1}{\pi(x|y)+\pi(y|x)}$) cambiando el valor de valor de sigma en la función tentativa $\pi(x|\mu) \sim \mathcal{N}(x; \mu, \sigma^2)$. Los datos están promediados sobre $5,000$ ejecuciones. Podemos ver estos resultados en la Tabla 7.4.

Valor de σ	Promedio de nº de saltos	Error absoluto medio	Error cuadrático medio
$\sigma = 3$	944.37	1.31	2.69
$\sigma = 2$	791.48	3.33	17.69
$\sigma = 1$	707.74	5.21	34.14
$\sigma = 0.5$	401.59	5.53	40.04

Cuadro 7.4: Comparativa del funcionamiento del algoritmo MTM con diferente σ de la función tentativa.

Podemos ver que cuanto mayor sea la varianza, mayor será el número de saltos y menor será el error absoluto medio, como era fácil prever después de haber analizado la fdp objetivo.

7.3. Análisis de los resultados

Debido al poco tiempo disponible, la falta de recursos y al exceso de tiempo dedicado en el análisis teórico, no hemos realizado las simulaciones suficientes para haber recorrido todo el abanico de posibilidades. Para completar el estudio, es necesario realizar simulaciones en 2 o más dimensiones, con funciones objetivo aleatorias, con diferentes funciones $\lambda(x, y)$, con número k variable en la ejecución de la misma simulación MTM y con una función tentativa adaptativa.

7.3.1. Resultados esperados

Los resultado esperados antes de realizar las simulaciones son los siguientes:

- Aumento de saltos cuando se aumenta el valor de k.
- Disminución del error en estimación cuando se aumenta el valor de k.
- Convergencia más rápida cuando se aumenta el valor de k.
- Resultados significativamente diferentes cuando se utiliza diferentes funciones $\lambda(x, y)$, dado que realmente estamos utilizando diferentes pesos dentro del MTM (según su definición, los pesos pueden tener más o menos "sentido estadístico").

Como veremos a continuación, los resultados obtenidos distan significativamente de los resultados esperados.

7.3.2. Resultados obtenidos

Los resultados obtenidos en las simulaciones nos confirman que:

- El número de saltos aumenta según aumenta el valor de k, esto confirma el resultado esperado.
- El error aumenta según aumenta el valor de k, esto contradice el resultado esperado. Habrá que profundizar el estudio.

- La rapidez de convergencia del algoritmo no mejora sensiblemente según aumenta k.
- La función $\lambda(x, y)$ no produce cambios muy significativos en los resultados de los algoritmos, y esto contradice el resultado esperado sobre la importancia de esta función. Esto a nuestro juicio es el resultado más sorprendente.

Recordamos que para poder confirmar o desmentir estos resultados con mayor seguridad, precisamos de más simulaciones y de un análisis estadístico más profundo.

7.3.3. Conclusiones

Para confirmar o desmentir los resultados obtenidos, precisamos realizar más simulaciones con diferentes densidades objetivo, utilización de espacios de estado de 2 o más dimensiones, controlar el código y sacar a la luz lo fallos no detectados que estén provocando que los resultados obtenidos no coincidan con los resultados esperados.

Además de todo esto, necesitaríamos comparar el MTM con los métodos introducidos en el Capítulo 6 y con un conjunto de algoritmos MH independientes en paralelo. Finalmente, habría que profundizar un poco más en el estudio teórico para comprender mejor el funcionamiento de las técnicas MTM. Todo esto será objeto de unos trabajos futuros.

Capítulo 8

Resumen y conclusiones

8.1. Conclusiones

En este proyecto hemos estudiado diferentes metodologías de generación de números aleatorios. Después de un estudio previo de la bibliografía sobre métodos de muestreo aleatorio, nos hemos concentrado sobre la que actualmente son las técnicas más potentes y más utilizadas en la práctica: *los algoritmos MCMC*.

Los métodos MCMC se basan en el diseño de una adecuada *cadena de Markov*. Bajo ciertas condiciones, estas cadenas convergen a una densidad estacionaria invariante en el tiempo. La idea fundamental de los métodos MCMC es la generación de una cadena de Markov cuya densidad estacionaria coincida con la densidad que se quiere muestrear. Las cadenas de Markov son procesos estocásticos en el que la probabilidad de que ocurra un evento depende del evento inmediatamente anterior. Por lo tanto, los métodos MCMC producen números aleatorios correlacionados entre sí. Las técnicas MCMC pueden ser aplicadas teóricamente (y de manera fácil e inmediata, sin estudios analíticos previos) a cualquier densidad de probabilidad. Esta característica los hace particularmente interesantes en la práctica. De hecho, no sólo se han multiplicado las aplicaciones en las últimas décadas sino que, a través de pequeñas variaciones, se han diseñado algoritmos parecidos para problemas de *optimización estocástica* y otros campos diferentes al muestreo.

En este trabajo hemos analizado en profundidad el algoritmo MCMC más famoso: *el algoritmo Metropolis-Hastings*. Aunque sea un método muy potente, tiene una serie de debilidades:

1. Las muestras están correlacionadas y en algunos casos repetidas (es decir, la correlación puede ser muy alta). En general, para la totalidad de las aplicaciones posibles es preferible que las muestras tengan una correlación muy baja entre ellas y si es posible que sean *independientes*. Muchos pequeños "trucos" se ha propuesto en la literatura como permutaciones, o considerar solo un subconjunto de muestras de la cadena de Markov generada, etc. En este trabajo hemos tratado soluciones más sofisticadas.

2. Desde que se hace lanza el algoritmo hasta que las muestras generadas se distribuyen mediante la densidad objetivo, hay que esperar un tiempo transitorio en el que, evidentemente, las muestras generadas no siguen la distribución deseada. Por lo tanto, estas muestras se deben descartar.

3. Debida a la correlación entre las muestras, la cadena puede quedarse enganchada alrededor de una moda de la densidad objetivo. Más en general, la cadena puede resultar atrapada en un sub-región del dominio de la variable de interés. Esto claramente, ralentiza la convergencia a la densidad objetivo. Este fenómeno resulta típico en los problemas descritos por densidades con modas "estrechas".

4. Otro problema aparece cuando la densidad objetivo esta compuesta por un factor expresado con integrales analíticamente no tratables. En este tipo de problema es imposible evaluar la función objetivo, lo que impide la aplicación de la función de aceptación en el algoritmo MH tradicional. En este proyecto en realidad no hemos tratado este problema.

Para solventar estas dificultades, diferentes autores propusieron las siguientes variaciones en el algoritmo MH tradicional:

- los métodos basados en *variables auxiliares* (como la muy conocida variable *temperatura*) [4, Capítulo 4],

- los métodos basados en *pesos de importancia* (siguiendo en un cierto sentido la estrategia de *importance sampling*) [4, Capítulo 6][5, Capítulo 2],
- los métodos basados en *densidad tentativa adaptativa* (estas técnicas intentan mejorar la fdp tentativa "online" aprendiendo de las muestras anteriores)[4, Capítulo 8],
- los métodos *multi-punto* [4, Capítulo 5],
- los métodos *basados en población* [4, Capítulo 5],
- mezclas de las estrategias anteriores,
- etc.

Aunque todas la variantes expuestas resultan muy interesantes, decidimos centrarnos en los métodos *multi-punto* y los métodos *basados en población*. Estas dos categorías nos parecieron interesantes por diferentes razones:

1. Son generalizaciones del algoritmo Metropolis-Hastings (MH) clásico, y este último puede verse como caso específico.
2. Pueden representarse ambos como una serie de algoritmos MH en paralelo con intercambio de información.
3. Algunos técnicas en estas categorías resultan ser mezclas de algoritmos MCMC con otras estrategias de muestreo de Monte Carlo, como el *importance sampling*.
4. Nos parecen muy *"atractivos"* para una posible aplicación en optimización. En realidad, no hemos investigado mucho esta posibilidad (estamos seguros que existen ya algoritmos en literatura) pero la estructura con múltiples partículas e intercambio de información, los hace particularmente adecuados para la optimización estocástica. Claramente, habrá que diseñar pequeñas variaciones para esta finalidad.
5. Son técnicas muy potentes y muy utilizadas en la práctica. De hecho, se puede comprobar que hay una muy amplia literatura sobre estos tipos de algoritmos.

Dentro de estas categorías hemos analizado realmente sólo tres algoritmos:

- *Multiple-Try Metropolis Hastings*,
- *Sample Metropolis Hastings*,
- y *Parallel Tempering*.

Estas técnicas fueron elegidas para no extender en exceso este escrito y poder explicar de la mejor manera posible los algoritmos. Elegimos estas dos metodologías porque resultaban muy interesantes desde el punto de vista conceptual y nos parecieron fáciles de comprender.

8.1.1. Simulaciones numéricas

Las simulaciones han evidenciado claramente como la utilización de un número $k > 1$ de puntos ayuda la cadena de Markov a explorar más fácilmente el espacio de estado. Esto es puesto en evidencia con el aumento de número de saltos de una moda a otra, de una región de alta probabilidad a otra, como demuestran ampliamente los resultados numéricos. Esta propiedad, no sólo es importante desde el punto de vista del muestreo (evita que la cadena de Markov se quede atrapada en una región), sino también desde el punto de vista de la optimización estocástica.

Una vez realizadas las simulaciones correspondientes al algoritmo MH y las técnicas MTM, nos ha sorprendido que no todos los resultados esperados de MTM coinciden con los resultados obtenidos de las simulaciones. Este hecho nos lleva a plantearnos :

1. si las simulaciones realizadas tienen fallos que no se han detectado,
2. si hemos utilizado la elección adecuado de la función $\lambda(x, y)$,
3. si el problema reside en esta específica fdp objetivo,
4. o finalmente si las técnicas MTM tienen alguna problemática de la que no somos conscientes.

Nuestro asombro no proviene de que las técnicas MTM no mejoren el error de estimación respecto al algoritmo MH clásico, sino que incluso el error *empeora*. Esto nos hace pensar que con alta probabilidad hemos cometido un fallo o que no hemos considerado algún aspecto importante.

8.2. Trabajos futuros

Los trabajos futuros más interesantes para continuar con el estudio que hemos llevado a cabo son los siguientes:

- Profundizar en los resultados obtenidos y realizar nuevas simulaciones para comparar los datos que no coinciden con los resultados esperados.
- Profundizar en el análisis teórico del algoritmo MTM y sus reales posibilidades mediante diferentes simulaciones.
- Aplicación de estos algoritmos en problemas prácticos de interés científico (por ejemplo, desde el doblamiento de proteínas a problemas clasificación, *clustering*, seguimiento y inferencia Bayesiana, etc.).
- Estudio más profundo del significado de los pesos del método *Multiple-Try Metropolis* (MTM) y de la función simétrica $\lambda(x, y)$ para mejorar la convergencia del algoritmo. Comprobación del comportamiento del algoritmo modificando la función $\lambda(x, y)$ como hemos visto en el Capítulo 5.
- Estudio de la elección adecuada del parámetro k (numero de puntos) en el algoritmo MTM y estudio de la posibilidad de utilización de un valor k adaptativo (que vaya aumentando o disminuyendo según la necesidad y la complejidad de la fdp objetivo).
- Construcción adaptativa de la fdp tentativa utilizada dentro del método MTM utilizando las muestras y pesos (si es posible) generados anteriormente (pero en la misma iteración).

- Estudio más profundo del algoritmo *Sample Metropolis-Hastings* (SMH) y comparación de sus prestaciones con el algoritmo MTM.
- Análisis de la relación entre MTM y SMH. Estudio de una posible estrategia conjunta entre ambos métodos.
- Posible combinación de la técnica adaptativa de aceptación/rechazo con las técnicas avanzadas MCMC tratadas.

Parte IV

Apéndices

Si la gente no piensa que las matemáticas son simples,
es sólo porque no se dan cuenta de lo complicada que es la vida.
(John von Neumann)

Apéndice A

Planificación y presupuesto

En este Capítulo se presentan justificados los costes globales de la realización de este Proyecto Fin de Carrera, así como la planificación en el tiempo del mismo. Debido a circunstancias variadas, no todos los días se trabajó el mismo número de horas en el proyecto y en el mismo día se trabajaba en diferentes capítulos del proyecto. Así mismo, la configuración del proyecto ha cambiado a lo largo del tiempo, resultado casi imposible saber con exactitud el número de horas trabajadas en cada sección. Por esa razón, presentaremos una tabla y no un diagrama de Gant (que presentaría un estudio más exacto). El Cuadro A.1 presenta el desglose de las horas dedicadas a cada capítulo de forma aproximada. Aproximadamente un 5 % del tiempo del proyecto han sido compartidas con el tutor. Durante los primeros meses, la mayoría de las comunicaciones se realizaron mediante email, comunicándonos una vez al mes. Como es de lógica, en este último mes dedicábamos conjuntamente unas 18 horas a la semana.

En el Cuadro A.2 se recoge el coste total del proyecto desglosado en gastos de material y de personal. En el Cuadro A.3 podemos ver la factura detallada. Como podemos ver en el Cuadro A.2, el volumen de los gastos son pertenecientes a gastos de personal. Se han contabilizado las horas dedicadas por el tutor y por el autor del proyecto. Las retribuciones asignadas a las del autor del proyecto se ajustan a las retribuciones de un becario estándar en la Universidad Carlos III de Madrid, es decir, cuatro euros y medio por hora de trabajo. Las retribuciones asignadas a las del tutor

del proyecto se ajustan a las retribuciones del sueldo de un investigador contratado en la Universidad Carlos III de Madrid, es decir, 9.5 euros por hora de trabajo.

Fase	***Descripción***	***Nº horas***
1	**Documentación**	**256**
2	**Redacción proyecto**	**476**
2.1	Capítulo 1	42
2.2	Capítulo 2	67
2.3	Capítulo 3	72
2.4	Capítulo 4	33
2.5	Capítulo 5	118
2.6	Capítulo 6	117
2.7	Capítulo 7	13
2.8	Capítulo 8	8
2.9	Capítulo 9	6
3	**Desarrollo software**	**20**
4	**Revisión proyecto**	**60**
5	**Redacción presentación**	**15**
6	**Documentación no utilizada**	**35**
7	**Redacción no presentada**	**65**
TOTAL		**927**

Cuadro A.1: Desglose horas de trabajo.

Descripción	*Coste*
Costes materiales	**1100 €**
Ordenador gama media	900 €
Material de oficina	200 €
Gastos de personal	**5324 €**
Salario becario (927h x 4.5€/h)	4171.5 €
Salario investigador (95h x 9.5€/h)	902.5 €
Desplazamientos	250 €
TOTAL	**6424 €**

Cuadro A.2: Costes imputables del proyecto.

PRESUPUESTO DE GASTOS 22/09/11

Nombre de la compañía	Universidad Carlos III de Madrid
Departamento	Departamento de Teoría de la Señal y Comunicaciones
Autor	Víctor Pascual del Olmo

Gastos de Material	**Coste**
Ordenador gama media	900,00 €
Material de oficina	200,00 €

Gastos de Personal	**Coste**
Salario becario	4.171,50 €
Salario investigador	902,50 €
Desplazamientos	250,00 €

Total	**Coste**
Gastos de Material	1.100,00 €
Gastos de Personal	5.074,00 €
Gasto total sin I.V.A.	**6.424,00 €**
I.V.A. (18%)	1.156,32 €
Gasto total	**7.580,32 €**

Cuadro A.3: Factura detallada del Proyecto de Fin de Carrera.

Apéndice B

Relaciones y observaciones interesantes

Todas las técnicas descritas en este proyecto hacen un amplio uso de los conceptos de densidades conjutas, condicionales y marginales. Por esta razón, nos parece interesante proponer a continuación unas relaciones básicas, pero conceptualmente muy importantes, entre estas funciones.

B.1. Información estadística

Dada dos variables aleatorias X e Y, toda la información estadística está claramente contenida en la *densidad conjunta* $f(x,y)$. Es decir, conocida la fdp conjunta somos capaces de calcular las densidades marginales

$$p(x) = \int_{\mathcal{C}} f(x,y)\, dy, \tag{B.1.1}$$

$$q(y) = \int_{\mathcal{S}} f(x,y)\, dx, \tag{B.1.2}$$

y las dos densidades condicionales

$$h_1(y|x) = \frac{f(x,y)}{p(x)} = \frac{f(x,y)}{\int_{\mathcal{C}} f(x,y)\,dy}, \tag{B.1.3}$$

$$h_2(x|y) = \frac{f(x,y)}{q(y)} = \frac{f(x,y)}{\int_{\mathcal{S}} f(x,y)\,dx}. \tag{B.1.4}$$

Lo mismo no podemos decir si conocemos una fdp marginal, o incluso las dos densidades marginales. De hecho, en este caso nos faltaría toda la información sobre la correlación entre las dos variables X e Y (si asumimos independencia, las marginales aportarían toda la información necesaria).

Una sola fdp condicional tampoco contiene toda la información estadística. Mientras, si conociéramos una fdp condicional y la marginal *correspondiente* tendríamos toda la información necesaria, dado que en este caso podemos hallar fácilmente la densidad conjunta, a través de un simple producto,

$$f(x,y) = h_1(y|x)\,p(x) = h_2(x|y)\,q(y). \tag{B.1.5}$$

Si conocemos ambas densidades condicionales, ¿tenemos toda la información sobre X e Y ? La respuesta es *sí*. De hecho, dado $h_1(y|x)$ y $h_2(x|y)$ y sabiendo que

$$h_1(y|x)\,p(x) = h_2(x|y)\,q(y) = f(x,y) \rightarrow \frac{h_1(y|x)}{h_2(x|y)} = \frac{q(y)}{p(x)}, \tag{B.1.6}$$

así que integramos ambos miembros respecto a y teniendo

$$\int_{\mathcal{C}} \frac{h_1(y|x)}{h_2(x|y)}dy = \int_{\mathcal{C}} \frac{q(y)}{p(x)}dy = \frac{\int_{\mathcal{C}} q(y)\,dy}{p(x)} = \frac{1}{p(x)}. \tag{B.1.7}$$

Así que dadas las dos densidades condicionales podemos hallar las marginales de esta forma

$$p(x) = \frac{1}{\int_{\mathcal{C}} \frac{h_1(y|x)}{h_2(x|y)}dy}, \tag{B.1.8}$$

$$q(y) = \frac{1}{\int_{\mathcal{S}} \frac{h_2(x|y)}{h_1(y|x)} dx}, \tag{B.1.9}$$

y como consecuencia, podemos calcular la fdp conjunta

$$f(x,y) = \frac{h_1(y|x)}{\int_{\mathcal{C}} \frac{h_1(y|x)}{h_2(x|y)} dy} = \frac{h_2(x|y)}{\int_{\mathcal{S}} \frac{h_2(x|y)}{h_1(y|x)} dx}. \tag{B.1.10}$$

B.2. Observaciones y Gibbs sampling

Supongamos que se pueda muestrear la densidad conjunta $f(x,y)$. En este caso, es interesante observar que es trivial sacar muestras desde las dos marginales.

De hecho, dado un vector $(x', y') \sim f(x,y)$, la primera componente x' se distribuye como $p(x)$ y mientras y' es una muestra de $q(y)$. Es decir, al muestrear una conjunta ya tenemos muestras de las dos densidades marginales (nótese que esto no es trivial mirando la relación integral entre la fdp conjunta y las dos marginales).

Como última observación, es posible generar muestras desde la conjunta conociendo una densidad marginal y la *correspondiente* densidad condicional. Por ejemplo, si podemos muestrear $q(y)$ generando y' y luego somos capaces de producir la muestra x' desde $h_2(x|y')$ pues el punto (x', y') se distribuye como $f(x,y)$.

Por último, si somos capaces de muestrear las dos densidades condicionales también podemos generar números aleatorios desde la conjunta. Esto es exactamente el principio del *"Gibbs sampling"*. Un *"Gibbs sampler"* funciona de esta forma

1. Se genera $y^{(i+1)}$ desde $h_2\left(y|x^{(i)}\right)$,
2. y luego $x^{(i+1)}$ desde $h_1\left(x|y^{(i+1)}\right)$.
3. Actualizamos $i = i+1$, y volvemos al paso 1) hasta $i \leq N$.

Los puntos $\left(x^{(i)}, y^{(i)}\right)$, $i = 1, ..., N$, se distribuyen según $f(x,y)$. Hay que recordar que las muestras de cada coordenada se distribuirán como las correspondientes densidades marginales, aunque esto parezca sorprendente dado que las muestras han sido generadas desde las condicionales. Pero es importante observar que en cada paso las densidades condicionales que se muestrean van *cambiando*.

Por ejemplo, se considera la muestra i-ésima $x^{(i)}$. Esta muestra claramente se distribuye como $h_1\left(x|y^{(i)}\right)$, pero al mismo tiempo es una muestra de la marginal $p\left(x\right)$. Más en general, todas las $x^{(i)}$, $i = 1, ..., N$, son muestras de $p\left(x\right)$ pero por ejemplo $x^{(i+1)}$ no es una muestra de $h_1\left(x|y^{(i)}\right)$ sino de otra densidad condicional $h_1\left(x|y^{(i+1)}\right)$.

Conclusión: construyendo un histograma bidimensional a partir de los puntos aleatorios $\left(x^{(i)}, y^{(i)}\right)$, $i = 1, ..., N$, aproximamos la densidad conjunta $f\left(x, y\right)$. Mientras con todas las muestras $x^{(i)}$, $i = 1, ..., N$, aproximamos la fdp $p\left(x\right)$, y con todas las $y^{(i)}$, $i = 1, ..., N$, aproximamos la fdp $q\left(y\right)$.

Apéndice C

Variables auxiliares

Diferentes técnicas de muestreo aleatorio se basan en el uso de variables auxiliares. Dada la densidad objetivo $p_o(x)$, esto es equivalente a crear una densidad conjunta $f(x,y)$ con una fdp marginal exactamente $p_o(x)$. Es decir,

$$p_o(x) = \int_{\mathcal{C}} f(x,y)\, dy = \int_{\mathcal{C}} h_2(x|y)\, q(y)\, dy. \tag{C.0.1}$$

Si es posible generar una muestra y' desde $q(y)$ y luego producir x' desde $h_2(x|y')$. En este caso $x' \sim p_o(x)$. Muchos métodos se pueden incluir en esta categoría: teorema fundamental de la simulación, método de la densidad inversa, método de la densidad vertical, *slice sampling*, etc.

En general, cualquier método que incluya una transformación de una variable aleatoria puede considerarse como un caso especial. De hecho, dada la relación $X = g(Y)$, con $X \sim p_o(x)$ e $Y \sim q(y)$, se puede escribir

$$p_o(x) = \int_{\mathcal{C}} \delta(x - g(y))\, q(y)\, dy, \tag{C.0.2}$$

donde

$$h_2(x|y) = \delta(x - g(y)),$$

es decir, elegimos aleatoriamente y' desde $q(y)$ y luego $x' = g(y')$. Ahora, hay que

recordar dos propiedades [64] de la función delta de Dirac

$$\delta(a-x)=\delta(x-a), \tag{C.0.3}$$

$$\int f(x)\,\delta(h(x)-a)\,dx=\sum_{i=1}^{n}\frac{f(x_i)}{|h'(x_i)|}, \tag{C.0.4}$$

donde x_i son las n soluciones de la ecuación $a=h(x)$ y hemos indicado con h' la derivada primera de la función h.

Volviendo a la Ecuación (C.0.2) y asumiendo $g(x)$ monótona, podemos escribir la *única* solución de la ecuación $x=g(y)$ como $y=g^{-1}(x)$,

$$\begin{aligned}\int_{\mathcal{C}}\delta(x-g(y))\,q(y)\,dy&=\int_{\mathcal{C}}\delta(g(y)-x)\,q(y)\,dy\\&=\frac{q(g^{-1}(x))}{|g'(g^{-1}(x))|}\\&=q\left(g^{-1}(x)\right)\left|\frac{dg^{-1}}{dx}\right|,\end{aligned} \tag{C.0.5}$$

que es exactamente la bien conocida fórmula de la densidad de una transformación de una variable aleatoria

$$p_o(x)=q\left(g^{-1}(x)\right)\left|\frac{dg^{-1}}{dx}\right|. \tag{C.0.6}$$

También el algoritmo MH puede incluirse en esta categoría de métodos. De hecho, el algoritmo MH construye una densidad conjunta con *dos marginales iguales a la densidad objetivo*, es decir, también $q(y)=p_o(y)$. La densidad conjunta generada por el MH es

$$f(x,y)=K(y|x)\,p_o(x)=K(x|y)\,p_o(y), \tag{C.0.7}$$

donde $K(y|x)$ es la probabilidad de transición de la cadena de Markov (*kernel*). Nótese que la Ecuación (C.0.7) es exactamente la *ecuación de balance o reversibilidad*. Además tenemos que las dos densidades condicionales tienen la misma *forma*

analítica, es decir,

$$h_1(y|x) = K(y|x), \tag{C.0.8}$$

$$h_2(x|y) = K(x|y). \tag{C.0.9}$$

Pero hay que tener cuidado porque esto ***no*** significa que el kernel (las probabilidades condicionales) sea simétrico, es decir, en general $h_1(y|x) \neq h_2(x|y)$.

La importancia del método MH es que proporciona dos probabilidades condicionales adecuadas ($h_1(y|x) = K(y|x)$ y $h_2(x|y) = K(x|y)$) para que se cumpla la Ecuación (C.0.7). Hemos visto que un kernel adecuado para crear una densidad conjunta con dos densidades marginales iguales a $p_o(\cdot)$ es

$$K(y|x) = \pi(y|x) \min\left[1, \frac{\pi(x|y)\, p_o(y)}{\pi(y|x)\, p_o(x)}\right] + \delta(y-x)\left(1 - \mathcal{A}(x)\right), \tag{C.0.10}$$

donde $\pi(y|x)$ es una fdp genérica tentativa y $\mathcal{A}(x)$ es la probabilidad total de aceptar una nueva muestra dado x.

El resto del algoritmo MH consiste en generar muestras de la densidad conjunta $f(x,y)$ en (C.0.5) utilizando la misma idea proporcionada por el "Gibbs sampler": primero se muestre una condicional $y' \sim K(y|x)$ y luego la otra $x' \sim K(x|y')$.

Entonces, el vector (x', y') se distribuye como $f(x,y)$ (véase "Gibbs sampling" en [61]) y dado que las dos marginales son iguales a la densidad objetivo $p_o(\cdot)$, entonces ambas muestras x' e y' se distribuyen como $p_o(\cdot)$.

Índice de figuras

Bibliografía

[1] Gilks, W.R., Richardson, S. and Spiegelhalter, D.: *Markov Chain Monte Carlo in Practice.* Interdisciplinary Statistics. Taylor & Francis, Inc., UK (1995)

[2] Gentle, J. E.: *Random Number Generation and Monte Carlo Methods.* Springer (2004)

[3] Hormann, W., Leydold, J. and Der inger, G.: *Automatic nonuniform random variate generation.* Springer (2003)

[4] Liang, F., Liu, C. and Carroll, R.: *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples.* Wiley Series in Computational Statistics, England (2010)

[5] Liu, J. S.: *Monte Carlo Strategies in Scientic Computing.* Springer (2004)

[6] Troutt, M. D., Pang, W. K. and Hou, S. H.: *Vertical density representation and its applications.* World Scientic (2004)

[7] Berzuini, C., Best, N.G., Gilks, W. and Larizza, C.: *Dynamic conditional independence models and Markov chain Monte Carlo methods.* Journal of the American Statistical Association Vol. 92, No. 440, (December 1997), pp. 1403-1412

[8] Gilks, W.R., Richardson, S. and Spiegelhalter, D.: *Markov Chain Monte Carlo in Practice.* Interdisciplinary Statistics. Taylor & Francis, Inc., UK (1995)

[9] Jing, L. and Vadakkepat, P.: *Interacting MCMC particle filter for tracking maneuvering target.* Digital Signal Processing Vol. 20, Issue 2, (March 2010), pp. 561-574

[10] Berzuini, C. and Gilks, W.: *Resample-move filtering with cross-model jumps.* In A. Doucet, N. de Freitas, and N. Gordon, editors, Sequential Monte Carlo Methods in Practice, chapter 6. Springer (2001)

[11] Rappaport, T.S.: *Wireless Communications: Principles and Practice* (2nd edition). Prentice-Hall, Upper Saddle River, NJ (USA), 2001.

[12] Michael, J.R., Schucany, W.R. and Haas, R.W.: *Generating random variates using transformations with multiple roots.* The American Statistician Vol. 30, No. 2 (May, 1976), pp. 88-90

[13] Devroye, L.: *Non-Uniform Random Variate Generation.* Springer (1986)

[14] Jones, M.C.: *On khintchine's theorem and its place in random variate generation.* The American Statistician Vol. 56, No. 4, (November 2002)

[15] Khintchine, A.Y.: *On unimodal distributions.* Izvestiya NauchnoIssledovatelskogo Instituta Matematiki i Mekhaniki (1938)

[16] Troutt, M.D.: *A theorem on the density of the density ordinate and an alternative interpretation of the box-muller method.* Taylor & Francis (1991)

[17] Troutt, M.D.: *Vertical density representation and a further remark on the box-muller method.* Statistics Vol. 24, Issue 1, (1993), pp. 81-83

[18] http://en.wikipedia.org/wiki/Von_Neumann (21 de Marzo 2011)

[19] http://en.wikipedia.org/wiki/George_Marsaglia (21 de Marzo 2011)

[20] Martino, L.: *Novel schemes for adaptive rejection sampling.* Doctoral Thesis. Universidad Carlos III de Madrid (2011)

[21] Chen, R.: *Another look at rejection sampling through importance sampling.* Statistics & Probability Letters Vol. 72, Issue 4, (15 May 2005), pp. 277-283

[22] Marshall, A.W. *The use of multi-stage sampling schemes in Monte Carlo computation.* Springer (1956)

[23] Mackay, D.J.C.: *Introduction to Monte Carlo Methods.* Springer (1986)

[24] http://en.wikipedia.org/wiki/John_Strutt,_3rd_Baron_Rayleigh (15 de Abril 2011)

[25] http://en.wikipedia.org/wiki/J._Robert_Oppenheimer (20 de Abril 2011)

[26] http://en.wikipedia.org/wiki/Manhattan_Project (20 de Abril 2011)

[27] http://en.wikipedia.org/wiki/Nicholas_Constantine_Metropolis (20 de Abril 2011)

[28] http://en.wikipedia.org/wiki/Enrico_Fermi (20 de Abril 2011)

[29] Sánchez del Río, C.: *Física cuántica (I).* EUDEMA, S.A. (Ediciones de la Universidad Complutense, S.A.), (1991)

[30] Metropolis, N. and Ulam, S.: *The Monte Carlo method.* Journal of the American Statical Association, Vol. 44, Issue 247, (September 1949), pp. 335-341

[31] Hastings, W. K.: *Monte Carlo sampling methods using Markov chains and their applications.* Biometrika Vol. 57, Issue 1, (1970), pp. 97-109

[32] Barker, A.A.: *Monte Carlo calculations of the radial distribution functions for a proton-electron plasma.* Australian Journal of Physics Vol. 18, Issue 2, (1965), pp. 119-134

[33] http://en.wikipedia.org/wiki/Charles_Stein_(statistician) (23 de Abril 2011)

[34] http://en.wikipedia.org/wiki/Stein's_example (25 de Abril 2011)

[35] Peskun, P.H.: *Optimun Monte Carlo sampling using Markov chains.* Biometrika Vol. 60, Issue 3, (May 1973), pp. 607-612

[36] Roberts, G.O. and Tweedie, R.L.: *Exponential Convergence of Langevin Distributions and Their Discrete Approximations.* Bernoulli Vol. 2, Issue 4, (December 1996), pp. 341-363

[37] Pandolfi, S., Bartolucci, F. and Friel, N.: *A generalization of the Multiple-Try Metropolis algorithm for Bayesian estimation and model selection.* JMLR Workshop and Conference Proceedings Vol. 9, ISSN 1533-7928, (2010), pp. 581-588

[38] Liu, Jun S.: *Monte Carlo strategies in scientific computing.* Springer (2001)

[39] Qin, Z.S. and Liu, J.S.: *Multipoint Metropolis Method with Application to Hybrid Monte Carlo.* Journal of Computational Physics Vol. 172, Issue 2, (20 September 2001), pp. 827-840

[40] Gilks, W.R., Roberts, G.O. and George, E.I.: *Adaptive Direction Sampling.* Journal of the Royal Statistical Society. Series D (The Statistician) Vol. 43, No. 1, Special Issue: Conference on Practical Bayesian Statistics, (1992 (3)) (1994), pp. 179-189

[41] Roberts, G.O. and Gilks, W.R.: *Convergence of adaptive direction sampling.* Journal of Multivariate Analysis Vol. 49, Issue 2, (May 1994), pp. 287-298

[42] Liu, J.S., Liang, F. and Wong, W.H.: *The multiple-try method and local optimization in Metropolis sampling.* Journal of the American Statistical Association Vol. 95, No. 449 (March 2000), pp. 121-134

[43] Swendsen, R.H. and Wang, J.S.: *Replica Monte Carlo simulation of spin glasses.* Physical Review Letters Vol. 57, Issue 21, (1986), pp. 2607-2609

[44] Geyer, C.J.: *In Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface.* American Statistical Association, New York, (1991), pp. 156

[45] Earl, D.J., and Deem, M.W.: *Parallel tempering: Theory, applications, and new perspectives.* Phys. Chem. Chem. Phys., Vol. 7, Issue 23, (2005), pp. 3910-3916

[46] Artur B.A.: *The theory behind tempered Monte Carlo methods* (unpublished notes, 2005)

[47] Sugita, Y., and Okamoto, Y.: *Replica-exchange molecular dynamics method for protein folding.* Chemical Physics Letters Vol. 314, Issues 1-2, (26 November 1999), pp. 141-151

[48] Hukushima, K. and Nemoto, K.: *Exchange Monte Carlo Method and Application to Spin Glass Simulations.* J. Phys. Soc. Jpn. Vol. 65, (1996) pp. 1604-1608

[49] Candia, A. and Coniglio, A.: *Spin and density overlaps in the frustrated Ising lattice gas.* Phys. Rev. Vol. 65, Issue 1, (2001)

[50] http://en.wikipedia.org/wiki/Spin_glass (28 de Abril 2011)

[51] Neirotti, J.P., Calvo, F., Freeman, D.L. and Doll, J.D.: *Phase changes in 38-atom Lennard-Jones clusters. I. A parallel tempering study in the canonical ensemble.* Journal of Chemical Physics Vol. 112, Issue 23, ()

[52] Wang, Q., Yan, Q., Nealey, P.F. and de Pablo, J.J.: *Monte Carlo simulations of diblock copolymer thin films confined between two homogeneous surfaces.* Journal of Chemical Physics Vol. 112, Issue 1, ()

[53] Marinari, E. and Parisi, G.: *Simulated Tempering: A New Monte Carlo Scheme.* EPL (Europhysics Letters) Vol. 19, No. 6, (1992)

[54] Holland, J.H.: *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor (1975)

[55] Patton, W.F., Erdjument-Bromage, H., Marks, A.R., Tempst, P. and Taubman, M.B.: *Components of the Protein Synthesis and Folding Machinery Are Induced in Vascular Smooth Muscle Cells by Hypertrophic and Hyperplastic Agents IDENTIFICATION BY COMPARATIVE PROTEIN PHENOTYPING*

AND MICROSEQUENCING. The Journal of Biological Chemistry Vol. 270, (September 8 1995)

[56] Liang, F. and Wong, W.H.: *Evolutionary Monte Carlo: Aplications to C_p model sampling and change point problem.* The National University of Singapore and UCLA, 8118 Math Sciences. Statistica Sinica Vol. 10 (2000), pp. 317-342

[57] Liang, F. and Wong, W.H.: *Evolutionary Monte Carlo for protein folding simulations.* Journal of Chemical Physics Vol. 115, Issue 7, (2001)

[58] Goodman, J., Sokal, A.: *Multigrid Monte Carlo method: Conceptual foundations.* Journals Phys. Rev. D Vol. 40, Issue 6, (1989)

[59] Tanner, M.A. and Wong, H.W.: *The Calculation of Posterior Distributions by Data Augmentation.* Journal of the American Statistical Association. Vol. 82, No. 398, (1987)

[60] Billera, L.J. and Diaconis, P.: *A Geometric Interpretation of the Metropolis-Hastings Algorithm.* Statistical Science Vol. 16, No. 4 (November 2001), pp. 335-339

[61] Liang, F., Liu, C. and Chuanhai, J.: *Advanced Markov chain Monte Carlo methods.* John Wiley & Sons, (2010)

[62] Serrano Cádiz, A.: *Optimización estocástica mediante métodos de Monte Carlo.* Proyecto de Fin de Carrera. Universidad Carlos III de Madrid, Escuela Politécnica Superior, (Abril 2011)

[63] Iba, Y.: Population Monte Carlo algorithms. Trans. Jpn. Soc. Artif. Intell. 16, 279–286 (2000)

[64] Gel'fand, I.M., Shilov, G.E.: Generalized functions, 1–5, Academic Press (1966–1968)

[65] Frenkel, D. and Smit, B.: *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, San Diego, (1996)

[66] Lewandowski L., and Liu, J.S.: *The sample Metropolis- Hasting algorithm.* Thechnical report, Department of Statistics, Purdue University, (2008)

[67] Leach, A.R.: *Molecular Modelling: Principles and Applications.* Singapore: Addison-Weley Longman (1996)

[68] Karplus, M. and Petsko, G.A.: *Molecular dynamics simulations in biology.* Nature 347, 631-639 (18 October 1990)

[69] Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F. and Wootton, J.C.: *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment.* Science (8 October 1993), Vol. 262, no. 5131, pp. 208-214

[70] Alder, B.J. and Wainwright, T.E.: *Studies in Molecular Dynamics. I. General Method.* Journal of Chemical Physics, Vol. 31, Issue 2, (1959)

[71] Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P.: *Optimization by Simulated Annealing.* Science (13 May 1983), Vol. 220 no. 4598 pp. 671-680

[72] Gouriéroux, C. and Monfort, A.: *Simulation-based econometric method.* Econometric Theory (2000), 16: 131-138

[73] Geman, S. and Geman, D.: *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.* IEEE Trans. Pattern Anal. Machine Intell (1984)

[74] Metropolis, N., Rosenbluth, A.W., Rosenbluth M.N. and Teller, H.: *Equation of State Calculations by Fast Computing Machines.* The Journal of Chemical Physics, Vol. 21, No. 6. (June 1953)

[75] Efron, B.: *Bootstrap Methods: Another look at the jackknife.* The Annals of Statistics, Vol. 7, No. 1, (January 1979)

[76] Gelfand, A.E. and Smith, A.F.: *Sampling-Based Approaches to Calculating Marginal Densities.* Journal of the Americal Statistical Association, Vol. 85, No. 410, (June 1990)