

tomar $R_{(k)}^t = r_t + \frac{B_{(k)}^t}{t}$, donde el $\{B_{(k)}^t\}$ son independientes al azar de Bernoulli variables con parámetro $r_t - r_t$. La variabilidad en el número total de copias puede reducirse dejando que el $\{B_{(k)}^t\}_{t=1}^{\infty}$ sea tal que el número total de las partículas remuestreadas son fijas. También es posible combinar remuestreo bootstrap y enriquecimiento, como en el *Rosenbluth enriquecido con podado* método de [18]. El Capítulo 9 explora más a fondo los méritos de estos métodos de "partículas Monte Carlo" en los que se combina un esquema de muestreo secuencial con un paso de remuestreo/división.

5.10 FILTRADO NO LINEAL PARA MODELOS MARKOV OCULTOS

Esta sección describe una aplicación de SIS y SIR al filtrado no lineal. Muchos problemas de ingeniería, ciencias aplicadas, estadística y econometría se pueden formular como *modelos ocultos de Markov* (HMM). En su forma más simple, un HMM es un proceso estocástico $\{X_t, Y_t\}$, donde X_t (que puede ser multidimensional) representa la *verdadero* estado de algún sistema y Y_t representa el *observado* estado del sistema en un tiempo discreto t .

Generalmente se supone que $\{X_t\}$ es una cadena de Markov, digamos con distribución inicial $P(X_0)$ y probabilidades de transición de un paso $P(X_t/X_{t-1})$. Es importante tener en cuenta que el estado real de la cadena de Markov sigue siendo *oculto*, de ahí el nombre HMM. Toda la información sobre el sistema es transmitida por el proceso. $\{Y_t\}$. Suponemos que, dado X_0, \dots, X_t , la observación Y_t depende solo de X_t

a través de algún pdf condicional $P(Y_t/X_t)$. Tenga en cuenta que hemos utilizado aquí un estilo de notación bayesiano en el que todas las densidades de probabilidad (condicionales) están representadas por el *mismo símbolo* P . Usaremos esta notación en el resto de esta sección.

Denotamos por $\mathbf{X}_{1:t} := (X_1, \dots, X_t)$ y $\mathbf{Y}_{1:t} := (Y_1, \dots, Y_t)$ las sucesiones no observables y observables hasta el tiempo t , respectivamente, y de manera similar para sus equivalentes en minúsculas.

El HMM se representa gráficamente en la Figura 5.4. Éste es un ejemplo de un *red bayesiana*. La idea es que los bordes indiquen la estructura de dependencia entre dos variables. Por ejemplo, dados los estados X_1, \dots, X_t , la variable aleatoria Y_t es condicionalmente independiente de X_1, \dots, X_{t-1} , porque no hay un borde directo desde Y_t a cualquiera de estas variables. Así tenemos $P(Y_t/\mathbf{X}_{1:t}) = P(Y_t/X_t)$, y más generalmente

$$P(\mathbf{y}_{1:t}/\mathbf{x}_{1:t}) = P(y_1/X_1) \cdots P(y_t/X_t) = P(\mathbf{y}_{1:t-1}/\mathbf{x}_{1:t-1}) P(y_t/X_t). \quad (5.86)$$

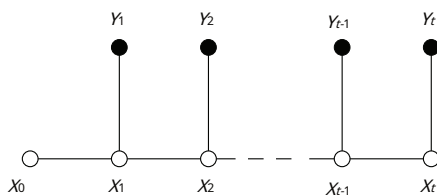


Figura 5.4: Una representación gráfica del HMM.

Resumiendo, tenemos

$$\begin{aligned} X_t &\sim P(X_t/X_{t-1}) & Y_t &\sim P(Y_t/X_t) & \text{(ecuación de estado),} \\ y_t &/X_t & & & \text{(ecuación de observación).} \end{aligned} \quad (5.87)$$

■ EJEMPLO 5.18

Un ejemplo de (5.87) es el siguiente modelo popular:

$$\begin{aligned} X_t &= \phi_1(X_{t-1}) + \varepsilon_{1t}, & Y_t &= \phi_2(X_t) \\ & & & + \varepsilon_{2t}, \end{aligned} \quad (5.88)$$

dónde $\phi_1(\cdot)$ y $\phi_2(\cdot)$ tienen funciones vectoriales dadas y ε_{1t} y ε_{2t} son independientes de vectores aleatorios gaussianos bidimensionales con media cero y matrices de covarianza C_1 y C_2 , respectivamente.

Nuestro objetivo, basado en un resultado de $y_{1:t}$ de $\mathbf{Y}_{1:t}$, es determinar, o estimar *en línea*, las siguientes cantidades:

1. El pdf condicional conjunto $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ y, como caso especial, el condicional marginal pdf $R(X_t/\mathbf{y}_{1:t})$, que se denomina *filtrado* pdf.
2. El rendimiento esperado

$$= \int_{R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})} [H(\mathbf{X}_{1:t})] = \int H(\mathbf{X}_{1:t}) R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t}) d\mathbf{X}_{1:t}. \quad (5.89)$$

Es bien sabido [11] que el pdf condicional $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ o el pdf filtrado $R(X_t/\mathbf{y}_{1:t})$ se puede encontrar explícitamente solo para los siguientes dos casos particulares:

- (a) Cuando $\phi_1(X)$ y $\phi_2(X)$ en (5.88) son lineales, la fdp de filtrado se obtiene de la célebre *filtro Kalman*. El filtro de Kalman se explica en la Sección A.6 del Apéndice.
- (b) Cuando el X_t puede tomar sólo un número finito, digamos k , de valores posibles, por ejemplo, como en señales binarias, podemos calcular $R(X_t/\mathbf{y}_{1:t})$ eficientemente con complejidad $O(k^2t)$. Las aplicaciones se pueden encontrar en comunicación digital y reconocimiento de voz; ver, por ejemplo, la Sección A.7 del Apéndice.

Porque el pdf de destino $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ para el modelo de espacio de estado general (5.87) es difícil de obtener con exactitud, es necesario recurrir a los métodos de Monte Carlo. Para poner el problema de filtrado no lineal en el marco secuencial de Monte Carlo de la Sección 5.8, primero escribimos $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ en forma secuencial, similar a (5.78). Un candidato natural para el pdf "auxiliar" en el momento t es el pdf condicional $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$. Es decir, sólo las observaciones hasta el momento t son usados. Por la regla de Bayes tenemos para cada $t=1, \dots, \text{norte}$,

$$\begin{aligned} & \frac{R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})}{R(\mathbf{X}_{1:t-1}/\mathbf{y}_{1:t-1})} \\ &= \frac{R(\mathbf{y}_{1:t}/\mathbf{X}_{1:t}) R(\mathbf{X}_{1:t})}{R(\mathbf{y}_{1:t}) R(\mathbf{y}_{1:t-1}/\mathbf{X}_{1:t-1}) R(\mathbf{X}_{1:t-1})} \\ &= \frac{R(\mathbf{y}_{1:t-1}/\mathbf{X}_{1:t-1}) R(y_t/X_t) R(\mathbf{X}_{1:t-1}) R(X_t/X_{t-1})}{R(\mathbf{y}_{1:t-1}) R(y_t/\mathbf{y}_{1:t-1})} \frac{R(\mathbf{y}_{1:t-1})}{R(\mathbf{y}_{1:t-1}/\mathbf{X}_{1:t-1}) R(\mathbf{X}_{1:t-1})} \\ &= \frac{R(y_t/X_t) R(X_t/X_{t-1})}{R(y_t/\mathbf{y}_{1:t-1})}, \end{aligned} \quad (5.90)$$

donde también hemos usado (5.86) y el hecho de que $R(X_t/\mathbf{X}_{1:t-1}) = R(X_t/X_{t-1})$, $t=1, 2, \dots$, por la propiedad de Markov.

Este resultado es de poca utilidad para un cálculo exacto de $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$, ya que requiere el cálculo de $R(y_t/\mathbf{y}_{1:t-1})$, que implica la evaluación de integrales complicadas. Sin embargo, si ambas funciones (pdfs) $R(X_t/X_{t-1})$ y $R(y_t/X_t)$ se puede evaluar exactamente (lo cual es una suposición razonable), entonces SIS se puede usar para simular aproximadamente desde $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ de la siguiente manera: Sea $gramo(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ sea el pdf de muestreo de importancia. Suponemos que, similar a (5.73), podemos escribir $gramo(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ recursivamente como

$$gramo(\mathbf{X}_{1:t}/\mathbf{y}_{1:t}) = gramo(X_0/y_0) \prod_{s=1}^t gramo(X_s/X_{s-1}, y_s). \quad (5.91)$$

Luego, por analogía con (5.76), y usando (5.90) (eliminando la constante de normalización $R(y_t/\mathbf{y}_{1:t-1})$), podemos escribir el peso de importancia W_t de un camino $\mathbf{X}_{1:t}$ generado a partir de $gramo(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ recursivamente como

$$W_t = W_{t-1} \frac{R(y_t/X_t) R(X_t/X_{t-1})}{gramo(X_t/X_{t-1}, y_t)} = W_{t-1} t u_t. \quad (5.92)$$

Una elección natural para el pdf de muestreo de importancia es

$$gramo(X_t/\mathbf{X}_{1:t-1}, \mathbf{y}_{1:t}) = R(X_t/X_{t-1}), \quad (5.93)$$

en cuyo caso el peso incremental se simplifica a

$$t u_t = R(y_t/X_t). \quad (5.94)$$

Con esta elección de distribución de muestreo, simplemente estamos adivinando los valores del proceso oculto $\{X_t\}$ sin prestar atención a los valores observados.

Una vez que se elige la densidad de muestreo de importancia, el muestreo del pdf objetivo $R(\mathbf{X}_{1:t}/\mathbf{y}_{1:t})$ procede como se describe en la Sección 5.8. Para más detalles, se remite al lector interesado a [11], [27] y [32].

■ EJEMPLO 5.19 Seguimiento de solo rodamientos

Supongamos que queremos rastrear un objeto (por ejemplo, un submarino) frente a un dispositivo que solo informa el *ángulo* al objeto (ver Figura 5.5). I las medidas de los ángulos son ruidosas. Suponemos que se conocen las velocidades po iniciales y que el objeto se mueve a una velocidad constante

Dejar $X_t = (pags_{1t}, v_{1t}, pags_{2t}, v_{2t})$ sea el vector de posiciones y velocidades (discretas) del objeto objetivo en el tiempo $t=0, 1, 2, \dots$, y deja Y_t Sea el ángulo medido. El problema es rastrear el estado desconocido del objeto X_t basado en las medidas $\{Y_t\}$ y las condiciones iniciales.

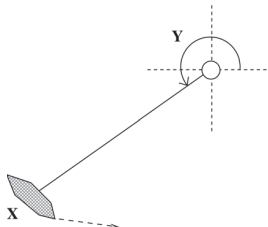


Figura 5.5: Seguimiento del objeto mediante mediciones ruidosas del ángulo.

El proceso (X_t, Y_t) , $t=0,1,2, \dots$ se describe mediante el siguiente sistema:

$$X_t = HACHA_{t-1} + \varepsilon_{1t}$$

$$Y_t = \arctan(\text{pags}_{1t}, \text{pags}_{2t}) + \varepsilon_{2t}.$$

Aquí $\arctan(u, v)$ denota el arco tangente de cuatro cuadrantes, es decir, $\arctan(v/u) + C$, donde C es $0, \pm\pi$, o $\pm\pi/2$, dependiendo del cuadrante en el que (u, v) mentiras. Los vectores de ruido aleatorio $\{\varepsilon_{1t}\}$ se supone que son NORTE($0, C_1$) distribuido, y el ruido de medición ε_{2t} es NORTE($0, \sigma_2$) repartido. Todas las variables de ruido son independientes entre sí. La matriz A es dado por

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

El problema es encontrar el pdf condicional $P(X_t/y_{1:t})$ y, en particular, el estado esperado del sistema $MI[X_t/y_{1:t}]$.

Indicamos cómo se puede solucionar este problema vía SIS. Usar (5.93) para la distribución muestral significa simplemente que X_t se extrae de un NORTE($HACHA_{t-1}, C_1$) distribución. Como consecuencia de (5.94), el peso incremental, $tu_t = P(y_t/X_t)$, es igual al valor y_t de la pdf normal con media $\arctan(\text{pags}_{1t}, \text{pags}_{2t})$ y diferencia σ_2 . El procedimiento SIS correspondiente se resume a continuación. Nota que se da el trámite del SIS paralelo, en el que el $\{W_t^{(k)}\}_{k=1}^n$ y $\{X_t^{(k)}\}_{k=1}^n$ son calculado al mismo tiempo ejecutando *norte* procesos paralelos.

Algoritmo 5.10.1: Procedimiento SIS para seguimiento solo de rodamientos

aporte: Tamaño de la muestra *norte*, matrices A y C_1 , parámetro σ_2 y distribución de el estado inicial.

producción: Estimador X_t de la posición esperada $MI[X_t/y_{1:t}]$.

1 Inicializar $X_{(k)0}$ y establecer $W_{(k)0} \leftarrow 0, k=1, \dots, \text{norte}$

2 **por** $t=1$ **a** *norte* **hacer**

3 **por** $k=1$ **a** *norte* **hacer**

4 $X_{(k)t} \sim \text{NORTE}(HACHA_{(k)t-1}, C_1)$ (

5 $tu_t \leftarrow \frac{1}{\sigma_2} \frac{1}{\sqrt{2\pi}} \text{Exp} \left[-\frac{1}{2} \frac{(y_t - \arctan(\text{pags}_{1t}, \text{pags}_{2t}))^2}{\sigma_2^2} \right]$)

6 $W_{(k)t} \leftarrow W_{(k)t-1} tu_t$

7 $X_t \leftarrow \left(\sum_{k=1}^{\text{norte}} W_{(k)t} X_{(k)t} \right) / \left(\sum_{k=1}^{\text{norte}} W_{(k)t} \right)$

8 **devolver** X_t

Como ilustración numérica, considere el caso donde $\sigma_2=0.005$ y

$$C_1 = \sigma_2 \begin{bmatrix} 1/4 & 1/2 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 0 & 0 & 1/4 & 1/2 \\ 0 & 0 & 1/2 & 1 \end{bmatrix},$$

con $\sigma_1=0.001$. Vamos $X_0 \sim N(\mathbf{m}_0, \Sigma_0)$, con $\mathbf{m}_0 = (-0.05, 0.001, 0.2, -0.055)$, y

$$\Sigma_0 = 0.12 \begin{bmatrix} 0.52 & 0 & 0 & 0 \\ 0 & 0.0052 & 0 & 0 \\ 0 & 0 & 0.32 & 0 \\ 0 & 0 & 0 & 0.012 \end{bmatrix}.$$

El panel izquierdo en la Figura 5.6 muestra cómo el proceso estimado $\{X_i\}$, obtenido a través de SIS, rastrea el proceso real $\{X_i\}$ sobre $norte=25$ pasos de tiempo, utilizando un tamaño de muestra de $norte=10,000$. En el panel derecho se muestra el resultado de SIR con remuestreo bootstrap (Algoritmo 5.9.2), para el mismo tamaño de muestra que en el caso SIS. Vemos que la posición real se rastrea a lo largo del tiempo con mayor precisión.

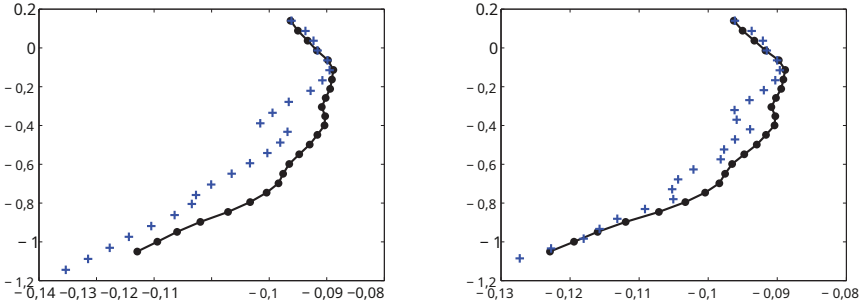


Figura 5.6: Comparación del rendimiento de los algoritmos SIS (izquierda) y SIR (derecha) para el problema de seguimiento de solo rumbos, utilizando un tamaño de muestra de $norte=10^4$ más de 25 pasos de tiempo.

Tanto para SIS como para SIR, a medida que aumenta el tiempo, el seguimiento rápidamente se vuelve más inestable. Esto es una consecuencia de la degeneración de la razón de verosimilitud. De hecho, después de algunas iteraciones, solo un puñado de muestras contiene la mayor parte del peso de importancia. Esto produce una alta variabilidad entre muchas corridas y proporciona estimaciones menos confiables. El paso de remuestreo mitiga algo de esta degeneración. Se han propuesto varias otras técnicas de remuestreo heurístico (p. ej., véase [11]).

5.11 MÉTODO DE LA RAZÓN DE VEROSIMILITUD DE LA TRANSFORMACIÓN

la *razón de verosimilitud de transformación* (TLR) es un método simple, conveniente y unificando manera de construir estimadores de muestreo de importancia eficientes. Para motivar el método TLR, consideramos la estimación de

$$= \text{MI}[H(\mathbf{X})], \quad (5.95)$$

dónde $\mathbf{X} \sim \mathcal{H}(\mathbf{X})$. Considere primero el caso donde \mathbf{X} es unidimensional (escribimos X en vez de \mathbf{X}). Dejar F ser el cdf de X . De acuerdo con el método IT, podemos escribir

$$X = F^{-1}(tu), \quad (5.96)$$

dónde $tu \sim \text{tu}(0,1)$ y F^{-1} es el inverso de la cdf F . Sustituyendo $X = F^{-1}(tu)$ en $= \text{MI}[H(X)]$, obtenemos

$$= \text{MI}[H(F^{-1}(tu))] = \text{MI}[H(tu)].$$

Nótese que a diferencia de $=MI[H(X)]$, donde la expectativa se toma con respecto a $H(X)$, en $= [H(tu)]$, la expectativa se toma con respecto al uniforme $tu(0, 1)$ distribución. La extensión al caso multidimensional es simple.

Dejar $h(tu, \nu)$ sea otra densidad en $(0,1)$, parametrizado por algún parámetro de referencia ν , con $h(tu, \nu) > 0$ para todos $0 < tu < 1$ (tenga en cuenta que tu es una variable y no un parámetro). Un ejemplo es el $Beta(\nu, 1)$ distribución, con densidad

$$h(tu, \nu) = \nu tu^{\nu-1}, \quad tu \in (0, 1),$$

con $\nu > 0$, o el $Beta(1, \nu)$ distribución, con densidad

$$h(tu, \nu) = \nu(1-tu)^{\nu-1}, \quad tu \in (0, 1).$$

Usando $Beta(1, \nu)$ como el pdf de muestreo de importancia, podemos escribir como

$$= \min_t [H(tu) W(tu, \nu)], \quad (5.97)$$

dónde $tu \sim h(tu, \nu)$, y

$$W(tu, \nu) = \frac{1}{h(tu, \nu)} \quad (5.98)$$

es la razón de verosimilitud. El estimador de razón de verosimilitud de está dado por

$$= \text{norte}^{-1} \sum_{k=1}^{\text{norte}} H(tu_k) W(tu_k, \nu), \quad (5.99)$$

dónde $tu_1, \dots, tu_{\text{norte}}$ es una muestra aleatoria de $h(tu, \nu)$, llamamos a (5.99) la *razón de verosimilitud de transformada inversa* (ITLR) estimador; véase Kroese y Rubinstein [23].

Supongamos, por ejemplo, que $X \sim \text{Weib}(\alpha, \lambda)$, es decir que X tiene la densidad

$$f_X(x; \alpha, \lambda) = \alpha \lambda (\lambda x)^{\alpha-1} \exp(-(\lambda x)^\alpha). \quad (5.100)$$

Tenga en cuenta que se puede generar una variable aleatoria de Weibull usando la transformación

$$X = \lambda^{-1} Z^{1/\alpha}, \quad (5.101)$$

donde la variable aleatoria Z tiene un $\text{Exp}(1)$ distribución. Aplicando el método IT, obtenemos

$$X = F^{-1}(tu) = \lambda^{-1} (-\ln(1-tu))^{1/\alpha}, \quad (5.102)$$

y $H(tu) W(tu, \nu)$ en (5.99) se reduce a $H(\lambda^{-1} (-\ln(1-tu))^{1/\alpha}) / h(tu, \nu)$.

El método TLR es una extensión natural del método ITLR. Comprende dos pasos. El primero es un sencillo *cambio de variable* paso, y el segundo involucra una aplicación de la técnica SLR al pdf transformado.

Para aplicar el primer paso, simplemente escribimos X en función de otro vector aleatorio, digamos como

$$X = \text{GRAMO}(Z). \quad (5.103)$$

si definimos

$$H(Z) = H(\text{GRAMO}(Z)),$$

entonces estimar (5.95) es equivalente a estimar

$$= MI[H(Z)]. \quad (5.104)$$

Note que las expectativas en (5.95) y (5.104) se toman con respecto a la densidad original de \mathbf{X} y la densidad transformada de \mathbf{Z} . Como ejemplo, considere nuevamente un caso unidimensional y sea $X \sim \text{Weib}(\alpha, \lambda)$. Recordando (5.101), tenemos $H(\mathbf{Z}) = H(\lambda^{-1} Z^{1/\alpha})$, y por lo tanto $\text{MI}[H(\lambda^{-1} Z^{1/\alpha})]$.

Para aplicar el segundo paso, supongamos que \mathbf{Z} tiene una densidad $h(\mathbf{z}; \boldsymbol{\theta})$ en alguna clase de densidades $\{h(\mathbf{z}; \boldsymbol{\eta})\}$. Luego, podemos buscar estimar de manera eficiente a través del muestreo de importancia, por ejemplo, utilizando el método de razón de verosimilitud estándar. En particular, por analogía con (5.58), obtenemos el siguiente estimador:

$$\hat{\tau} = \frac{1 \sum_{\text{norte}}}{\text{norte}} H(\mathbf{Z}) \tilde{W}(\mathbf{Z}_k; \boldsymbol{\theta}, \boldsymbol{\eta}) \quad , \quad (5.105)$$

dónde

$$W(\mathbf{Z}_k; \boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{h(\mathbf{Z}_k; \boldsymbol{\theta})}{h(\mathbf{Z}_k; \boldsymbol{\eta})}$$

y $\mathbf{Z}_k \sim h(\mathbf{z}; \boldsymbol{\eta})$. Llamaremos al estimador SLR (5.105) basado en la transformación (5.103), *estimador TLR*. Como ejemplo, considérese nuevamente la $\text{Weib}(\alpha, \lambda)$ caso. Usando (5.101), podríamos tomar $h(\mathbf{z}; \boldsymbol{\eta}) = \eta \text{mi}_{1-\eta} z$ como el pdf de muestreo, con $\eta = \theta = 1$ como parámetro nominal. Por tanto, en este caso, en (5.105) se reduce a

$$= \frac{1 \sum_{\text{norte}}}{\text{norte}} H(\lambda^{-1} Z_k^{1/\alpha}) W(\mathbf{Z}_k; \boldsymbol{\theta}, \boldsymbol{\eta}), \quad (5.106)$$

con

$$W(\mathbf{Z}_k; \boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{h(\mathbf{Z}_k; \boldsymbol{\theta})}{h(\mathbf{Z}_k; \boldsymbol{\eta})} = \frac{\theta \text{mi}_{1-\theta} Z_k}{\eta \text{mi}_{1-\eta} Z_k}$$

y $\mathbf{Z}_k \sim \text{Exp}(\eta)$.

Para encontrar el vector de parámetros óptimo $\boldsymbol{\eta}$ del estimador TLR (5.105), podemos resolver, por analogía a (5.63), el programa CE

$$\underset{\boldsymbol{\eta}}{\text{máximo}} D(\boldsymbol{\eta}) = \underset{\boldsymbol{\eta}}{\text{máx. mi}_{\tau}} H(\mathbf{Z}) W(\mathbf{Z}; \boldsymbol{\theta}, \boldsymbol{\tau}) \text{ en } h(\mathbf{Z}; \boldsymbol{\eta}), \quad (5.107)$$

y de manera similar para la contraparte estocástica de (5.107).

Ya que \mathbf{Z} se puede distribuir de manera bastante arbitraria, su distribución se elige típicamente de una familia exponencial de distribuciones (consulte la Sección A.3 del Apéndice), para la cual la solución óptima $\boldsymbol{\eta}$ de (5.107) se puede obtener analíticamente en una forma conveniente y simple. A continuación presentamos el algoritmo TLR para estimar

$\text{mi}_{\tau}[H(\mathbf{X})]$, asumiendo que \mathbf{X} es un vector aleatorio con componentes independientes distribuidas de forma continua. La clave es encontrar una función de transformación GRAMO tal que $\mathbf{X} = \text{GRAMO}(\mathbf{Z})$, con $\mathbf{Z} \sim h(\mathbf{z}; \boldsymbol{\theta})$. Por ejemplo, podemos tomar \mathbf{Z} con todos los componentes iid y distribuidos de acuerdo con una familia exponencial (p. ej., $\text{Exp}(1)$).

Algoritmo 5.11.1: Método de relación de probabilidad de transformación (TLR)

aporte :Función $GRAMO$ tal que $\mathbf{X} = GRAMO(\mathbf{Z})$, con $\mathbf{Z} \sim h(\mathbf{z}; \boldsymbol{\theta})$. Tamaños de muestra $norte$ y $norte2$. Parámetro inicial $\boldsymbol{\tau}$. **producción**:

Estimador de $=MI[H(\mathbf{X})] = MI[H(\mathbf{Z})]$.

1 Generar una muestra aleatoria $\mathbf{Z}_1, \dots, \mathbf{Z}_{norte}$ de $h(\cdot; \boldsymbol{\tau})$.

2 Resuelva la contrapartida estocástica del programa (5.107) (para un parámetro familia exponencial parametrizada por la media, aplique directamente la solución analítica (5.68). Iterar si es necesario. Denote la solución por $\hat{\boldsymbol{\eta}}$.

3 Generar una muestra aleatoria (más grande) $\mathbf{Z}_1, \dots, \mathbf{Z}_{norte2}$ de $h(\cdot; \hat{\boldsymbol{\eta}})$ y estimar $=MI[H(GRAMO(\mathbf{Z}))]$ mediante el estimador TLR (5.105), tomando $\boldsymbol{\eta} = \hat{\boldsymbol{\eta}}$.

4 devolver

El Algoritmo TLR 5.11.1 asegura que tan pronto como la transformación $\mathbf{X} = GRAMO(\mathbf{Z})$, se puede estimar utilizando el estimador TLR (5.105) en lugar del estimador SLR (5.58). Aunque la precisión de ambos estimadores (5.105) y (5.58) es la misma (Rubinstein y Kroese [35]), la ventaja del primero es su universalidad y su capacidad para evitar la carga computacional al mismo tiempo que entrega directamente la solución analítica del estocástico. contraparte del programa (5.107).

5.12 PREVENCIÓN DE LA DEGENERACIÓN DEL MUESTREO DE IMPORTANCIA

En esta sección, mostramos cómo prevenir la degeneración de los estimadores de muestreo de importancia a través de la *método de detección*. La degeneración de las razones de verosimilitud en problemas de simulación Monte Carlo de alta dimensión es uno de los temas centrales en la simulación Monte Carlo; véase también la Observación 5.7.1.

Para motivar el método de selección, considere nuevamente el ejemplo 5.15 y observe que solo los dos primeros parámetros de muestreo importantes del vector de cinco dimensiones $\hat{\mathbf{v}} = (\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_4, \hat{v}_5)$ son sustancialmente diferentes de los del vector de parámetros nominales $\mathbf{tu} = (tu_1, tu_2, tu_3, tu_4, tu_5)$. La razón es que las derivadas parciales de con respecto a tu_1 y tu_2 son significativamente más grandes que los que se refieren a tu_3 , tu_4 , y tu_5 . Llamamos a tales elementos *tu* y *tu* elementos de cuello de botella. Con base en esta observación, podríamos usar en lugar del vector de muestreo de importancia $\hat{\mathbf{v}} = (\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_4, \hat{v}_5)$ el vector $\hat{\mathbf{v}} = (\hat{v}_1, \hat{v}_2, tu_3, tu_4, tu_5)$, reduciendo el número de parámetros de muestreo de importancia de cinco a dos. Esto no solo tiene ventajas computacionales (entonces resolveríamos una varianza bidimensional o un programa de minimización CE en lugar de uno de cinco dimensiones), sino que también conduce a una mayor reducción de la varianza, ya que el término de razón de verosimilitud W con dos términos de producto es menos “ruidoso” que el que tiene cinco términos de producto. Los elementos de cuello de botella se pueden identificar como aquellos elementos i que tienen el mayor *perturbación relativa* $\delta_i = (\hat{v}_i - tu_i)/tu_i$.

El algoritmo 5.12.1 presenta el algoritmo de detección de dos etapas para estimar

$$= \int tu[H(\mathbf{X})] = \int H(\mathbf{X}) f(\mathbf{X}; \mathbf{tu}) d\mathbf{X},$$

basado en CE, y denotado como CE-SCR. Su contraparte de VM, el algoritmo VM-SCR, es similar. Para simplificar, supongamos que los componentes de \mathbf{X} son independientes y que cada componente se distribuye de acuerdo con una familia exponencial unidimensional que está parametrizada por la media; el caso dependiente podría tratarse

similar. Es más, $H(\mathbf{X})$ se supone que es una función monótonamente creciente en cada componente de \mathbf{X} . Una consecuencia de los supuestos anteriores es que el vector de parámetros \mathbf{v} tiene dimensión n .

En la primera etapa del algoritmo (líneas 1 a 9), identificamos y estimamos los parámetros de cuello de botella, dejando los parámetros que no son cuello de botella como están, y en la segunda etapa (líneas 10 a 12), calculamos y devolvemos el estimador de muestreo de importancia

$$\hat{B} = \frac{1}{norte} \sum_{k=1}^{norte} H(\mathbf{X})_k W_B(\mathbf{X}_{kB}; \hat{\mathbf{v}}_B), \quad (5.108)$$

dónde $\mathbf{X}_{k\alpha}$ es el subvector de elementos cuello de botella de \mathbf{X}_k y W_α es la razón de verosimilitud correspondiente. Nótese que, por analogía con la Proposición A.4.2 del Apéndice, si cada componente del vector aleatorio \mathbf{X} es de una familia exponencial de un parámetro parametrizada por la media y si $H(\mathbf{X})$ es una función monótonamente creciente en cada componente de \mathbf{X} , entonces cada elemento del parámetro óptimo \mathbf{CEv} es al menos tan grande como el correspondiente de \mathbf{detu} . Dejamos la demostración como ejercicio para el lector.

Algoritmo 5.12.1: Algoritmo de detección de dos etapas CE-SCR

aporte: Tamaño de la muestra n , función de rendimiento H , tolerancia d , número de repeticiones d .

producción: Estimador B del rendimiento esperado $MI[H(\mathbf{X})]$.

```

1 Inicializar  $B \leftarrow \{1, \dots, n\}$ .

```

2por $t=1$ a dhacer

3 Generar una muestra $\mathbf{X}_1, \dots, \mathbf{X}_n$ de $P(\mathbf{X}; \mathbf{t}_u)$. **por** $i =$

4	1 a n o r t e h a c e r
---	--------------------------------

$$5 \quad \hat{V}_i \leftarrow \frac{\sum_{k=1}^{K_{norte}} H(\mathbf{X}_k) X_{i,k}}{\sum_{k=1}^{K_{norte}} H(\mathbf{X}_k)}$$

6 $di \leftarrow \frac{v_i - t_{ii}}{t_{ii}}$ // calcular la perturbación relativa

7	si $d_i < \delta$ después
---	---------------------------

8			$\hat{v}_i = t u_i$
---	--	--	---------------------

9	$B \leftarrow B \setminus \{i\}$	// retirar i del conjunto de cuello de botella
---	----------------------------------	--

10 Generar una muestra $\mathbf{X}_1, \dots, \mathbf{X}_n$ de $P(\mathbf{X}; \hat{\mathbf{v}})$

$$\mathbf{11}^{B \leftarrow norte} = \frac{1}{\sum_{k=1}^{norte} H(\mathbf{X}_k)} W_B(\mathbf{X}_k; \hat{\mathbf{v}}^B).$$

12devolver B

Es importante tener en cuenta lo siguiente:

1. Como se mencionó, bajo los presentes supuestos (componentes independientes, cada uno de una familia exponencial de un parámetro parametrizada por la media, y $H(\mathbf{X})$ aumentando monótonamente en cada componente), los componentes de la \mathbf{v} -son al menos tan grandes como los elementos correspondientes de \mathbf{t} . El algoritmo 5.12.1 tiene esto en cuenta y siempre identifica todos los elementos correspondiente a $d_i < 0$ como no cuello de botella.
2. Recuerde que las líneas 3 a 9 se realizan a propósito d veces. Esto nos permite determinar mejor los parámetros que no son cuellos de botella, ya que es probable que fluctúen alrededor de su valor nominal. t y por lo tanto d se volverá negativo o muy pequeño en una de las repeticiones.

En general, los modelos de simulación complejos de gran dimensión contienen parámetros tanto de cuello de botella como de no cuello de botella. El número de parámetros de cuello de botella suele ser menor que el número de parámetros que no son de cuello de botella. Imagine una situación en la que el tamaño (dimensión) del vector \mathbf{t} es grande, digamos 100, y el número de elementos de cuello de botella es solo de 10 a 15. Entonces, claramente, un estimador de muestreo de importancia basado solo en elementos de cuello de botella no solo será mucho más preciso que su contraparte de muestreo de importancia estándar que involucre las 100 razones de verosimilitud (que contengan tanto cuello de botella como no cuello de botella) sino que, en contraste con este último, no será degenerado.

El fenómeno del cuello de botella a menudo ocurre cuando se necesita estimar la probabilidad de un evento atípico en el sistema, como la probabilidad de un evento raro. Esto se tratará en el Capítulo 8. Por ejemplo, si se observa una falla en un sistema de confiabilidad con elementos altamente confiables, entonces es muy probable que varios elementos (típicamente los menos confiables) que forman un corte mínimo en el modelo fallen todos simultáneamente. . Otro ejemplo es la estimación de una probabilidad de desbordamiento de búfer en una red de colas, es decir, la probabilidad de que el número total de clientes en todas las colas supere un número grande. De nuevo, si se produce un desbordamiento de búfer, es muy probable que se deba a una acumulación en la cola de cuello de botella, que es la más congestionada de la red.

5.12.0.1 Resultados numéricos A continuación presentamos estudios numéricos con Algorithm 5.12.1 para una generalización del sistema de puente en el Ejemplo 5.1, representado en la Figura 5.7. Implementaremos la detección para los métodos CE y VM. Recuerde que para el método CE el vector de parámetros $\mathbf{v}(\mathbf{y}|\mathbf{a})$ a menudo se puede actualizar analíticamente, en particular cuando la distribución muestral proviene de una familia exponencial. Por el contrario, para VM, la actualización suele implicar un procedimiento numérico.

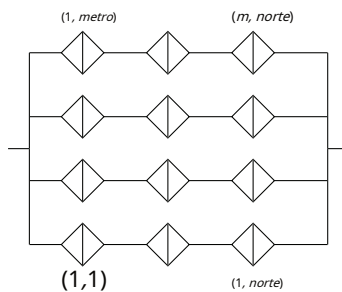


Figura 5.7: Un metro \times nortes sistema de puente

El sistema consta de metro \times nortes puentes dispuestos en una cuadrícula, y todos los puentes son de la forma en la Figura 5.1. Denote las longitudes de los bordes dentro del (y_0, j) -th puente por $X_{y_01}, \dots, X_{y_05}$. Entonces la longitud del camino más corto a través del puente (y_0, j) es

$$Y_{y_0} = \min \{ X_{y_01} + X_{y_04}, X_{y_02} + X_{y_05}, X_{y_01} + X_{y_03} + X_{y_05}, X_{y_02} + X_{y_03} + X_{y_04} \}. \quad (5.109)$$

Supongamos que queremos estimar la longitud máxima esperada de los caminos más cortos en todas las filas, es decir, $=MI[H(\mathbf{X})]$, con

$$H(\mathbf{X}) = \text{máx.}\{Y_{11}+\cdots+Y_{1\textit{norte}}, \ldots, Y_{\textit{metro}1}+\cdots+Y_{\textit{Minnesota}}\}.$$
 (5.110)

En nuestros resultados numéricos, asumimos que los componentes X_{jk} del vector aleatorio \mathbf{X} son independientes y que cada componente tiene un Weib (α, tu) distribución, es decir X_{jk} tiene la densidad

$$f(X; \alpha, tu) = \alpha tu (ux)^{\alpha-1} \text{mi}_{-(ux)^\alpha},$$

con $tu = tu_{jk}$. Recuerde que tal variable aleatoria de Weibull se puede generar usando la transformación $X = tu^{-1} Z^{1/\alpha}$, donde Z es una variable aleatoria distribuida $\text{Exp}(1)$. También suponemos que sólo tu es controlable, mientras que α es fijo y es igual a 0,2. Seleccionamos deliberadamente algunos elementos de tu user cuello de botella y establecer $d = 0.1$. Es importante señalar que el $\{X_{jk}\}$ están aquí *no* parametrizado por la media. Sin embargo, tomando $1/tu_{jk}$ como parámetro, estamos en el marco descrito anteriormente. En particular, las perturbaciones relativas se realizan con respecto a $\alpha/\hat{v}_{jk} \alpha/tu_{jk}$.

La Tabla 5.2 presenta el desempeño del Algoritmo 5.12.1 para el 1×1 (puente único) modelo (5.110). Aquí $tu_{11} = 1$ y $tu_{12} = 1$ se eligen como los parámetros de cuello de botella, mientras que los restantes (no cuello de botella) se establecen en 2. Las notaciones en la Tabla 5.2 son las siguientes:

- 1. *media, máx., y min* denota los valores medio, máximo y mínimo de la muestra de 10 estimaciones generadas independientemente de .
- 2. *RE* denota el error relativo de la muestra para , promediado sobre las 10 ejecuciones.
- 3. *UPC* denota el tiempo promedio de CPU en segundos basado en 10 ejecuciones.

Tabla 5.2: Desempeño del Algoritmo 5.12.1 para el modelo de puente único con muestras *norte* = *norte* = 500.

	CMC	CE	máquina virtual	CE-SCR	VM-SCR
Significar	4.052	3.970	3.734	3.894	3.829
máx.	8.102	4.327	4.201	4.345	4.132
mínimo	1.505	3.380	3.395	3.520	3.278
RE	0.519	0.070	0.078	0.076	0.068
UPC	0.00	0.04	0.21	0.05	0.13

De los resultados de la Tabla 5.2, se deduce que para este modelo relativamente pequeño, tanto CE como VM funcionan de manera similar a sus contrapartes de detección. Veremos además (en el Capítulo 8) que a medida que aumenta la complejidad del modelo, VM-SCR supera a sus tres alternativas, en particular CE-SCR. Tenga en cuenta que para este modelo, tanto CE como VM detectaron correctamente, en la primera etapa, los dos parámetros de cuello de botella. En particular, la Tabla 5.3 presenta una dinámica típica de detección de los dos parámetros de cuello de botella en la primera etapa del Algoritmo 5.12.1 para un modelo de puente único que tiene

un total de 5 parámetros. En la Tabla 5.3, t denota el número de replicación en la primera etapa, mientras que los 0 y 1 indican si los parámetros correspondientes se identifican como parámetros sin cuello de botella o cuello de botella, respectivamente. Como se puede observar, luego de dos repeticiones quedan cuatro parámetros de cuello de botella, luego de seis repeticiones se identifican tres como parámetros de cuello de botella, y luego de siete repeticiones el proceso se estabiliza, detectando correctamente los dos verdaderos parámetros de cuello de botella.

Tabla 5.3: Dinámica típica para detectar los parámetros de cuello de botella en la primera etapa del Algoritmo 5.12.1 para el modelo de puente.

tu_1	tu_2	tu_3	tu_4	tu_5	tu_1	tu_2	tu_3	tu_4	tu_5
0 1	1	1	1	1	5 1	1	0	1	0
1 1	1	0	1	1	6 1	1	0	1	0
2 1	1	0	1	1	7 1	1	0	0	0
3 1	1	0	1	0	8 1	1	0	0	0
4 1	1	0	1	0	9 1	1	0	0	0

La Tabla 5.4 presenta una evolución típica de \hat{v}_i en el modelo de puente único para los métodos VM y VM-SCR en la segunda etapa del Algoritmo 5.12.1.

Tabla 5.4: Evolución típica de \hat{v}_i para los métodos VM y VM-SCR.

máquina virtual						VM-SCR					
t	\hat{v}_1	\hat{v}_2	\hat{v}_3	\hat{v}_4	\hat{v}_5	t	\hat{v}_1	\hat{v}_2	\hat{v}_3	\hat{v}_4	\hat{v}_5
1.000	1.000	2.000	2.000	2.000		1.000	1.000	2	2	2	
1	0,537	0,545	2,174	2,107	1,615	1	0,555	0,599	2	2	2
2	0,346	0,349	2,071	1,961	1,914	2	0,375	0,402	2	2	2
3	0,306	0,314	1,990	1,999	1,882	3	0,315	0,322	2	2	2

Como se ve claramente, los parámetros de cuello de botella disminuyen unas tres veces después de la tercera iteración, mientras que los que no tienen cuello de botella fluctúan alrededor de su valor nominal. $tu=2$.

La Tabla 5.5 presenta el desempeño del Algoritmo 5.12.1 para los 3×Modelo de 10 puentes con seis cuellos de botella correspondientes a los elementos tu_{111} , tu_{112} , tu_{211} , tu_{212} , tu_{311} , tu_{312} . Establecimos $tu_{111}=tu_{112}=tu_{211}=tu_{212}=tu_{311}=tu_{312}=1$, mientras que los valores restantes (sin cuellos de botella) se establecen en 2. Tenga en cuenta nuevamente que en este caso, tanto CE como VM encontraron los seis cuellos de botella verdaderos.

Tabla 5.5: Desempeño del Algoritmo 5.12.1 para los 3×Modelo 10 con seis elementos de cuello de botella y tamaño de muestra $norte=norte_1=1000$.

	CMC	CE	máquina virtual	CE-SCR	VM-SCR
Significar	16.16	16.11	14.84	16.12	15.67
máx.	22,65	26,85	16.59	18.72	17.20
mínimo	11.13	7.007	12.59	14.63	14.80
RE	0.20	0.34	0.075	0.074	0.049
UPC	0.00	0.49	68.36	0.73	27.54

De los resultados de la Tabla 5.5, se deduce que, sin cribado, incluso el ingenuo Monte Carlo supera al CE estándar. Sin embargo, el uso de la detección da como resultado una mejora sustancial de la CE. Finalmente, VM-SCR supera a las cuatro alternativas restantes.

PROBLEMAS

5.1 Considere la integral
$$\int_a^b H(X) dX = (\text{segundo} - \text{un}) \text{MI}[H(X)], \text{ con } X \sim \text{tu}(un, b).$$
 Dejar $X_1, \dots, X_{\text{norte}}$ ser una muestra aleatoria de $\text{tu}(un, b)$. Considere los estimadores
$$\hat{I}_1 = \frac{1}{\text{norte}} \sum_{i=1}^{\text{norte}} H(X_i) \text{ y } \hat{I}_2 = \frac{1}{2\text{norte}} \sum_{i=1}^{\text{norte}} \{H(X_i) + H(b+a-X_i)\}.$$
 probar que si $H(X)$ es monotónico en X , después

$$\text{Var}(\hat{I}_1) - \text{Var}(\hat{I}_2).$$

En otras palabras, usar variables aleatorias antitéticas es más preciso que usar CMC.

5.2 Estime la longitud esperada del camino más corto para la red de puentes del ejemplo 5.1. Utilice tanto el estimador CMC (5.8) como el estimador antitético (5.9). Para ambos casos, tome un tamaño de muestra de $\text{norte} = 100.000$. Suponga que las longitudes de los enlaces X_1, \dots, X_n se distribuyen exponencialmente, con media 1, 1, 0.5, 2, 1.5. Compare los resultados.

5.3 Las variables aleatorias comunes (CRV) se utilizan a menudo al estimar derivadas o gradientes de funciones. Como ejemplo de juguete, considere la estimación de la derivada (tu) de la función $(tu) = \text{MI}[Y]$, donde $Y \sim \text{Exp}(2tu)$. Por eso, $(tu) = -1/(2tu)$. Una forma sencilla de estimar (tu) es primero aproximarlo con el *diferencia hacia adelante*

$$\frac{(tu+h) - (tu)}{h}$$

Para pequeños h , y luego para estimar ambos $(tu+h)$ y (tu) a través de la simulación Monte Carlo. Por ejemplo, generar $X_1, \dots, X_{\text{norte}} \sim \text{Exp}(tu+h)$ y $Y_1, \dots, Y_{\text{norte}} \sim \text{Exp}(tu)$ de forma independiente, y tomar $(X - Y)/\text{hora}$ como estimador de la diferencia directa (y por lo tanto como estimador sesgado de (tu)). Sin embargo, es mejor dibujar cada par (X_i, Y_i) con los CRN, por ejemplo, dejando $X_i \sim \text{Exp}(tu)$ y $Y_i \sim \text{Exp}(2tu)$, donde $tui \sim \text{tu}(0,1)$, $i = 1, \dots, \text{norte}$.

a) Implementar los dos estimadores de diferencia directa para el caso $tu=1$ y $h=10^{-2}$, tomando $\text{norte}=10^6$. Estime y compare los errores relativos de ambos estimadores.

b) Calcule los errores relativos exactamente (sin usar simulación).

5.4 Use el método de medias por lotes para estimar el tiempo de espera estacionario esperado en un *CURRO* cola a través de la ecuación de Lindley para el caso en que los tiempos entre llegadas son $\text{Exp}(1/2)$ distribuidos y los tiempos de servicio son $\text{tu}[0.5, 2]$ distribuido. Realice una simulación de *METRO* = 10.000 clientes, descartando el primero $k=100$ observaciones. Examine en qué medida se puede lograr la reducción de la varianza mediante el uso de variables aleatorias antitéticas.

5.5 Ejecute el problema del camino más corto estocástico en el ejemplo 5.4 y estime el rendimiento $= \text{MI}[H(\mathbf{X})]$ de 1000 repeticiones independientes, usando el dado (C_1, C_2, C_3, C_4) como el vector de variables de control, asumiendo que $X_i \sim \text{Exp}(1)$, $i = 1, \dots, 5$. Compare los resultados con los obtenidos con el método CMC.

5.6 Estime el tiempo de espera esperado del cuarto cliente en un *CURRO*1 cola para el caso de que los tiempos entre llegadas sean $\text{Exp}(1/2)$ distribuidos y los tiempos de servicio son $\text{tu}[0.5,2]$ distribuido. Use la ecuación de Lindley y las variables de control, como se describe en el ejemplo 5.5. Generar $n_{\text{orte}}=1000$ repeticiones de W_4 y proporcione un intervalo de confianza del 95% para $\text{MI}[W_4]$.

5.7 Demuestre que para cualquier par de variables aleatorias (U, V),

$$\text{Var}(tu) = \text{MI}[\text{Var}(U / V)] + \text{Var}(\text{MI}[U / V]).$$

[Sugerencia: use los hechos que $\text{MI}[tu2] = E[E[tu2 / V]]$ y $\text{Var}(X) = \text{MI}[X^2] - (\text{MI}[X])^2$.]

5.8 Dejar $R \sim \text{GRAMO}(pags)$ y definir $S_R = \sum_{i=1}^R X_i$, donde X_1, X_2, \dots es una secuencia de $\text{iidExp}(\lambda)$ variables aleatorias que son independientes de R .

- Muestre esa $S_R \sim \text{Exp}(\lambda \text{ pag})$. [Sugerencia: la forma más fácil es usar métodos de transformación y condicionamiento.]
- Para $\lambda = 1$ y $pags = 1/10$, estimación $\text{PAGS}(S_R > 10)$ usando CMC con un tamaño de muestra de $n_{\text{orte}} = 1000$.
- Repita b), ahora usando el estimador condicional de Monte Carlo (5.22). Compare los resultados con los de a) y b).

5.9 Considere la suma aleatoria S_R en el problema 5.8, con parámetros $pags = 0.25$ y $\lambda = 1$. Estimación $\text{PAGS}(S_R > 10)$ mediante estratificación utilizando estratos correspondientes a la partición de eventos $\{R=1\}, \{R=2\}, \dots, \{R=7\}$, y $\{R > 7\}$. Asignar un

Total de $n_{\text{orte}} = 10.000$ muestras a través de ambos $n_{\text{orte}} = pags \cdot n_{\text{orte}}$ y lo optimo $n_{\text{orte}} =$ en (5.35), y comparar los resultados. Para el segundo método, use una ejecución de simulación de tamaño 1000 para estimar las desviaciones estándar $\{\sigma_i\}$.

5.10 Demuestre que la solución al programa de minimización

$$\sum_{n_{\text{orte}}=1, \dots, N_{\text{orte}}^{\text{metro}}=1} \min \frac{\text{metro } pags \cdot \sigma_i}{n_{\text{orte}}} \quad \text{tal que} \quad n_{\text{orte}1} + \dots + n_{\text{orte}^{\text{metro}}} = N,$$

viene dada por (5.35). Esto justifica el muestreo estratificado Teorema 5.5.1.

5.11 Use el Algoritmo 5.4.2 y (5.26) para estimar la confiabilidad de la red de confiabilidad del puente en el Ejemplo 4.2 en la página 110 a través de la permutación Monte Carlo. Considere dos casos, donde las confiabilidades del enlace están dadas por $\text{pags} = (0.3, 0.1, 0.8, 0.1, 0.2)$ y $\text{pags} = (0.95, 0.95, 0.95, 0.95, 0.95)$, respectivamente. Tome un tamaño de muestra de $n_{\text{orte}} = 2000$.

5.12 Repita el Problema 5.11, usando el Algoritmo 5.4.3. Compare los resultados.

5.13 Este ejercicio analiza la contraparte del Algoritmo 5.4.3 que involucra caminos mínimos en lugar de cortes mínimos. Un vector de estado \mathbf{X} en el modelo de confiabilidad de la Sección 5.4.1 se llama *ruta de vectores* si $H(\mathbf{X}) = 1$. Si además $H(\mathbf{y}) = 0$ para todos $\mathbf{y} < \mathbf{X}$, después \mathbf{X} se llama *el vector de ruta mínima*. El conjunto correspondiente $A = \{i: X_i = 1\}$ se llama *el conjunto mínimo de rutas*; es decir, un conjunto mínimo de caminos es un conjunto mínimo de componentes cuyo *marcha* asegura el funcionamiento del sistema. Si $A_1, \dots, A_{\text{metro}}$

denote todos los conjuntos de rutas mínimas, entonces el sistema está funcionando si y solo si todos los componentes de al menos un conjunto de rutas mínimas están funcionando.

a) Muestre esa

$$H(\mathbf{X}) = \max_k \prod_{i \in A_k} X_i = 1 - \prod_{k=1}^{\text{metro}} \left(1 - \prod_{i \in A_k} X_i \right). \quad (5.111)$$

b) Definir

$$Y_k = \prod_{i \in A_k} X_i, \quad k=1, \dots, m,$$

eso es, Y_k es el indicador del evento que todo componente en la serie A_k esté en marcha. Aplicar la Proposición 5.4.1 a la suma $S = \sum_{k=1}^m Y_k$ e idear un algoritmo similar al Algoritmo 5.4.3 para estimar la confiabilidad $r = \text{PAGS}(S > 0)$ del sistema.

c) Pruebe este algoritmo en la red de confiabilidad del puente en el ejemplo 4.2.

5.14 Demostrar (ver (5.44)) que la solución de

$$\min_{\text{gramo}} \left(\frac{H(\mathbf{X})}{f(\mathbf{X})} \right)$$

es

$$\text{gramo}(\mathbf{X}) = \int \frac{H(\mathbf{X})/f(\mathbf{X})}{\int H(\mathbf{X})/f(\mathbf{X}) d\mathbf{X}} d\mathbf{X}.$$

5.15 Dejar $Z \sim \text{NORTE}(0,1)$. Estimar $\text{PAGS}(Z > 4)$ a través de un muestreo de importancia, utilizando el siguiente pdf de muestreo exponencial desplazado:

$$\text{gramo}(X) = \min\{X-4, X\}.$$

Elegir n suficientemente grande para obtener una precisión de al menos tres dígitos significativos y comparar con el valor exacto.

5.16 de Pearson χ^2 medida de discrepancia entre densidades gramo y h . Se define como

$$d(g, h) = \frac{1}{2} \int \left[\frac{\text{gramo}(\mathbf{X}) - h(\mathbf{X})}{h(\mathbf{X})} \right]^2 d\mathbf{X}.$$

Verifique que el programa VM (5.43) sea equivalente a minimizar el Pearson χ^2 medida de discrepancia entre el pdf de varianza cero gramo en (5.45) y la importancia de la densidad de muestreo gramo . En este sentido, los métodos CE y VM son similares, ya que el método CE minimiza la distancia Kullback-Leibler entre gramo y gramo .

5.17 Repita el problema 5.2 usando un muestreo de importancia, donde las longitudes de los enlaces se distribuyen exponencialmente con medias μ_1, \dots, μ_n . Escriba las fórmulas de actualización CE deterministas y estímelas a través de una ejecución de simulación de tamaño 1000 utilizando $\mathbf{w} = \mathbf{t}$.

5.18 Considere la familia exponencial natural ((A.9) en el Apéndice). que $\mathbf{t} = \mathbf{t}(\mathbf{X})$, $\mathbf{v} = \mathbf{v}(\mathbf{X})$, se reduce a resolver Espectáculo

$$\min_{\boldsymbol{\theta}} \left[H(\mathbf{X}) \left(\frac{\nabla Q(\boldsymbol{\theta})}{Q(\boldsymbol{\theta})} + \mathbf{t}(\mathbf{X}) \right) \right] = 0. \quad (5.112)$$

5.19 Como aplicación de (5.112), suponga que queremos estimar la expectativa de $H(\mathbf{X})$, con $X \sim \text{Exp}(\lambda_0)$. Muestre que el parámetro óptimo CE correspondiente es

$$\lambda^* = \frac{\min_{\lambda} [H(\mathbf{X})]}{\min_{\lambda} [H(\mathbf{X})X]}.$$

Compare con (A.15) en el Apéndice. Explique cómo estimar λ mediante simulación.

5.20 Dejar $X \sim \text{Weib}(\alpha, \lambda)$. Deseamos estimar $\text{mi}_{\lambda_0}[H(X)]$ a través del método SLR, generando muestras a partir de $\text{Weib}(\alpha, \lambda)$ — cambiando así el parámetro de escala λ pero manteniendo el parámetro de escala α fijado. Use la Ec. (5.112) y la Tabla A.1 en el Apéndice para mostrar que la elección óptima de CE para λ es

$$\lambda_* = \left(\frac{\text{mi}_{\lambda_0}[H(X)]}{\text{mi}_{\lambda_0}[H(X)X^\alpha]} \right)^{1/\alpha}.$$

Explique cómo podemos estimar λ mediante simulación.

5.21 Dejar $X_1, \dots, X_{\text{norte}}$ n variables aleatorias distribuidas. Dejar $\mathbf{X} = (X_1, \dots, X_{\text{norte}})$ y $S(\mathbf{X}) = X_1 + \dots + X_{\text{norte}}$. Deseamos estimar $\text{PAGS}(S(\mathbf{X}) - y)$ vía muestreo de importancia, utilizando $X_i \sim \text{Exp}(\theta)$, para todos i . Muestre que el parámetro óptimo de CE θ es dado por

$$\theta_* = \frac{\text{MI}[yO(S(\mathbf{X}) - y)]}{\text{MI}[yO(S(\mathbf{X}) - y)X]},$$

con $\bar{X} = (X_1 + \dots + X_{\text{norte}})/n$ indicando la expectativa bajo la distribución original (donde cada $X_i \sim \text{Exp}(1)$).

5.22 Considere el problema 5.20. Definir $\text{GRAMO}(z) = z^{1/\alpha}/\lambda_0 y H(z) = H(\text{GRAMO}(z))$.

- Demuestra que si $Z \sim \text{Exp}(1)$, entonces $\text{GRAMO}(Z) \sim \text{Weib}(\alpha, \lambda_0)$.
- Explicar cómo estimar a través del método TLR.
- Muestre que el parámetro óptimo de CE para Z es dado por

$$\theta_* = \frac{\text{mi}_{\eta}[H(Z)W(Z; 1, \eta)]}{\text{mi}_{\eta}[H(Z)ZW(Z; 1, \eta)]},$$

dónde $W(Z; 1, \eta)$ es la relación de $\text{Exp}(1)$ y $\text{Exp}(\eta)$ pdf.

5.23 Assú que el rendimiento esperado se puede escribir como $w = \sum_{i=1}^{\text{metro}} a_i y_i$, aquí $y_i = H(\mathbf{X})$ y $a_i, i=1, \dots, \text{metro}$ son coeficientes conocidos. Dejar $q(\mathbf{X}) = \sum_{i=1}^{\text{metro}} a_i H(\mathbf{X})$. Para cualquier pdf g dominante $q(\mathbf{X})$, la variable aleatoria

$$L = \frac{\sum_{i=1}^{\text{metro}} H(\mathbf{X})}{a_i g(\mathbf{X})} = \frac{q(\mathbf{X})}{g(\mathbf{X})},$$

dónde $\mathbf{X} \sim g$, es un estimador insesgado de w — tenga en cuenta que solo hay una muestra. Pruebal que alcanza la varianza más pequeña cuando $g = q$, con

$$g(\mathbf{X}) = \int \frac{q(\mathbf{X})}{Q(\mathbf{X})} d\mathbf{X},$$

y eso

$$\text{Var}_g(L) = \int \frac{q(\mathbf{X})^2}{Q(\mathbf{X})} d\mathbf{X} - w^2.$$

5.24 El método de acertar o fallar. Suponga que la función de rendimiento de la muestra, H , está acotado en el intervalo $[0, b]$, digamos, $0 \leq H(X) \leq b$ por $X \in [0, b]$. Dejar

$= \int H(x) dx = b \mathbb{E}[H(X)]$, con $X \sim \text{Unif}[0, b]$. Defina un estimador de θ por

$$\hat{h} = \frac{\sum_{i=1}^{\text{norte}} Y_i \mathbf{1}_{\{Y_i < H(X_i)\}}}{\text{norte}},$$

dónde $\{(X_i, Y_i) : i=1, \dots, n\}$ es una secuencia de puntos u el rectángulo $[0, b] \times [0, c]$ (ver Figura 5.8). el estimador *estimador*, desde un punto (X, Y) es aceptado o rechazado el punto cae dentro o fuera del área sombreada en la Figura 5.8 el estimador de acertar o fallar tiene una varianza mayor que el C

$$\hat{h} \approx \frac{b}{\text{norte}} \sum_{i=1}^{\text{norte}} H(X_i)$$

con $X_1, \dots, X_{\text{norte}}$ una muestra aleatoria de $\text{Unif}[0, b]$.

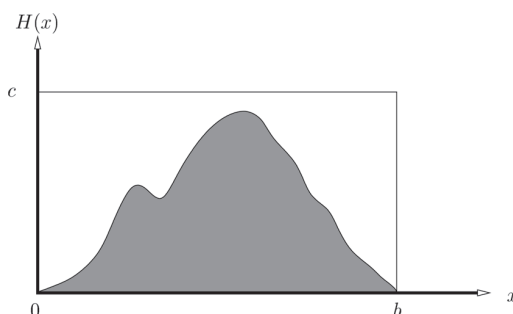


Figura 5.8: El método de acertar o fallar.

Otras lecturas

El artículo fundamental sobre las técnicas de reducción de la varianza es el de Kahn y Marshall [20]. Hay muchos buenos libros de texto de Monte Carlo con capítulos sobre técnicas de reducción de varianza. Entre ellos se encuentran [13], [17], [21], [22], [25], [29], [32], [33] y [40]. Para un estudio exhaustivo de las técnicas de reducción de la varianza, consulte Fishman [13] y Rubinstein [34]. Asmussen y Glynn [2] brindan un tratamiento moderno de reducción de varianza y simulación de eventos raros. Véase también el Capítulo 9 de [24], que explica en detalle la amplia variedad de técnicas de reducción de la varianza que se pueden utilizar para resolver un problema de estimación única. Libros influyentes sobre el remuestreo de importancia secuencial son [27] y [11]. Multilevel Monte Carlo es un área de investigación floreciente. El artículo reciente de Rhee y Glynn [31] explica cómo la aleatorización de los niveles de un algoritmo Monte Carlo multinivel de niveles infinitos puede lograr teóricamente un estimador de sesgo cero. Botev [6] describe la técnica de reducción de varianza de última generación para el muestreo de distribuciones normales multivariadas truncadas de alta dimensión.

Una introducción a los modelos de confiabilidad se puede encontrar en [15]. Para obtener más información sobre la reducción de la varianza en presencia de distribuciones de colas pesadas, consulte también [1], [3], [4] y [10].

Hay una gran literatura sobre la estimación del número de SAW. Aunque actualmente no existe una fórmula conocida para el número exacto de SAW de longitud *norte*, existen muchos métodos de aproximación. Los algoritmos más avanzados son los pivotes. Pueden manejar sierras de tamaño 107; ver [8], [28]. Para una encuesta reciente, véase [41].

En [26] se proporcionó un precursor del Algoritmo 5.12.1, donde los elementos de cuello de botella se identificaron mediante la estimación de gradientes de la función de rendimiento.

REFERENCIAS

1. S. Asmussen. Distribuciones estacionarias por tiempos de primer paso. En JH Dshalalow, editor, *Avances en colas: teoría, métodos y problemas abiertos*, páginas 79–102, Nueva York, 1995. CRC Press.
2. S. Asmussen y PW Glynn. *Simulación estocástica*. Springer-Verlag, Nueva York, 2007.
3. S. Asmussen y DP Kroese. Algoritmos mejorados para la simulación de eventos raros con colas pesadas. *Avances en probabilidad aplicada*, 38(2):545–558, 2006.
4. S. Asmussen, DP Kroese y RY Rubinstein. Colas pesadas, muestreo de importancia y entropía cruzada. *Modelos estocásticos*, 21(1):57–76, 2005.
5. S. Asmussen y RY Rubinstein. Propiedades de complejidad de la simulación de eventos raros en estado estacionario en modelos de colas. En JH Dshalalow, editor, *Avances en colas: teoría, métodos y problemas abiertos*, páginas 429–462, Nueva York, 1995. CRC Press.
6. ZI Botev. La ley normal bajo restricciones lineales: simulación y estimación vía minimax tilting. *Revista de la Sociedad Real de Estadística (B)*, 79:1–24, 2017.
7. H.-P. Chan y T.-L. Lai. Una teoría general de filtros de partículas en modelos ocultos de Markov y algunas aplicaciones. *Anales de estadísticas*, 41(6):2877–2904, 2013.
8. N. Clisby. Implementación eficiente del algoritmo de pivote para caminatas autoevitables. *Revista de física estadística*, 140:349–392, 2010.
9. GT Cochran. *Técnicas de muestreo*. John Wiley & Sons, Nueva York, 3ª edición, 1977.
10. PT de Boer, DP Kroese y RY Rubinstein. Un método rápido de entropía cruzada para estimar desbordamientos de búfer en redes de colas. *Ciencias de la gestión*, 50(7):883–895, 2004.
11. A. Doucet, N. de Freitas y N. Gordon. *Métodos secuenciales de Monte Carlo en la práctica*. Springer-Verlag, Nueva York, 2001.
12. T. Elperin, IB Gertsbakh y M. Lomonosov. Estimación de la fiabilidad de la red mediante modelos de evolución de grafos. *Transacciones IEEE sobre confiabilidad*, 40(5):572–581, 1991.
13. GS Hombre-pep. *Monte Carlo: Conceptos, Algoritmos y Aplicaciones*. Springer-Verlag, Nueva York, 1996.
14. S. Gal, RY Rubinstein y A. Ziv. Sobre la optimalidad y eficiencia de los números aleatorios comunes. *Matemáticas. computar Simul.*, 26(6):502–512, 1984.
15. IB Gertsbakh. *Teoría de la confiabilidad estadística*. Marcel Dekker, Nueva York, 1989.
16. MB Giles. Simulación de ruta Monte Carlo multinivel. *La investigación de operaciones*, 56(8):607–617, 2008.
17. P. Glasserman. *Métodos Monte Carlo en Ingeniería Financiera*. Springer-Verlag, Nueva York, 2004.

18. P. Grassberger. Método de Rosenbluth enriquecido con poda: simulaciones de ϕ -polímeros de longitud de cadena de hasta 1000000. *física Rev.E*, 56:3682–3693, septiembre de 1997.
19. D. Gross y CM Harris. *Fundamentos de la teoría de colas*. John Wiley & Sons, Nueva York, 2ª edición, 1985.
20. M. Kahn y AW Marshall. Métodos para reducir el tamaño de la muestra en los cálculos de Monte Carlo. *La investigación de operaciones*, 1:263–278, 1953.
21. JPC Kleijnen. *Técnicas Estadísticas en Simulación, Parte 1*. Marcel Dekker, Nueva York, 1974.
22. JPC Kleijnen. Análisis de simulación con números aleatorios comunes: una nota sobre Heikes et al. *simuletero*, 11:7–13, 1976.
23. DP Kroese y RY Rubinstein. El método de relación de probabilidad de transformación para la simulación de eventos raros con colas pesadas. *Sistemas de colas*, 46:317–351, 2004.
24. DP Kroese, T. Taimre y ZI Botev. *manual de métodos de monte carlo*. John Wiley & Sons, 2011.
25. AM Law y WD Kelton. *Modelado y análisis de simulación*. McGraw-Hill, Nueva York, 3ª edición, 2000.
26. D. Lieber, RY Rubinstein y D. Elmakis. Estimación rápida de eventos raros en redes estocásticas. *Transacción IEEE sobre confiabilidad*, 46:254–265, 1997.
27. JS Liu. *Estrategias de Monte Carlo en Computación Científica*. Springer-Verlag, Nueva York, 2001.
28. NN Madrás. *Conferencias sobre los métodos de Monte Carlo*. Sociedad Matemática Estadounidense, 2002.
29. DL McLeish. *Monte Carlo Simulación y Finanzas*. John Wiley & Sons, Nueva York, 2005.
30. MF Neutros. *Soluciones de matriz geométrica en modelos estocásticos: un enfoque algorítmico*. Publicaciones de Dover, Nueva York, 1981.
31. C.-H. Rhee y PW Glynn. Estimación imparcial con convergencia de raíz cuadrada para modelos SDE. *La investigación de operaciones*, 63(5):1026–1043, 2015.
32. CP Roberto y G. Casella. *Métodos estadísticos de Monte Carlo*. Springer, Nueva York, 2ª edición, 2004.
33. SM Ross. *Simulación*. Academic Press, Nueva York, 3ª edición, 2002.
34. RY Rubinstein. *Simulación y el Método Monte Carlo*. John Wiley & Sons, Nueva York, 1981.
35. RY Rubinstein y DP Kroese. *El método de entropía cruzada: un enfoque unificado para la optimización combinatoria, la simulación Monte Carlo y el aprendizaje automático*. Springer-Verlag, Nueva York, 2004.
36. RY Rubinstein y R. Marcus. Eficiencia de variables de control multivariado en simulación Monte Carlo. *La investigación de operaciones*, 33:661–667, 1985.
37. RY Rubinstein y B. Melamed. *Simulación y modelado modernos*. John Wiley & Sons, Nueva York, 1998.
38. RY Rubinstein y A. Shapiro. *Sistemas de eventos discretos: análisis de sensibilidad y optimización estocástica a través del método de función de puntuación*. John Wiley & Sons, Nueva York, 1993.
39. RY Rubinstein, M. Samorodnitsky y M. Shaked. Variables antitéticas, dependencia multivariante y simulación de sistemas estocásticos complejos. *Ciencias de la gestión*, 31:66–77, 1985.

40. MI Sobol. *Introducción al Método Monte Carlo*. CRC Press, Boca Ratón, FL, 1994.
41. EJJ van Rensburg. Métodos de Montecarlo para la marcha autoevasiva. *Journal of Physics A: Matemática y Teórica*, 42(32):323001, 2009.
42. FT Wall y JJ Erpenbeck. Nuevo método para el cálculo estadístico de dimensiones de polímeros. *Revista de física química*, 30(3):634–637, 1959.
43. P. Wesseling. *Una introducción a los métodos de redes múltiples*. John Wiley & Sons, 1992.
44. W. Whitt. Distribuciones bivariadas con marginales dadas. *Anales de estadísticas*, 4(6):1280–1289, 1976.

CAPÍTULO 6

CADENA MARKOV MONTE CARLO

6.1 INTRODUCCIÓN

En este capítulo presentamos un poderoso método genérico, llamado *Cadena de Markov Montecarlo* (MCMC), para *aproximadamente* generar muestras a partir de una distribución arbitraria. Esto, como aprendimos en la Sección 2.5, no suele ser una tarea fácil, en particular cuando \mathbf{x} es un vector aleatorio con componentes dependientes. Una ventaja adicional de MCMC es que solo requiere la especificación del pdf de destino hasta una constante (normalización).

El método MCMC se debe a Metropolis et al. [18]. Fueron motivados por problemas computacionales en física estadística, y su enfoque utiliza la idea de generar una cadena de Markov cuya distribución límite es igual a la distribución objetivo deseada. Hay muchas modificaciones y mejoras del algoritmo original de Metropolis [18], en particular el algoritmo introducido por Hastings [11]. Hoy en día, cualquier enfoque que produzca una cadena de Markov ergódica cuya distribución estacionaria sea la distribución objetivo se denomina MCMC o *Muestreo en cadena de Markov* [20]. Los algoritmos MCMC destacados son los muestreadores Metropolis-Hastings y Gibbs, siendo este último particularmente útil en el análisis bayesiano. Finalmente, el muestreo MCMC es el ingrediente principal en el popular *recocido simulado* técnica [1] para optimización discreta y continua.

El resto de este capítulo está organizado de la siguiente manera: en la sección 6.2 presentamos el algoritmo clásico de Metropolis-Hastings, que simula una cadena de Markov tal que su distribución estacionaria coincide con la distribución objetivo. Un caso especial importante es el *pega y corre* muestreador, discutido en la Sección 6.3. La sección 6.4 trata de la *Muestreador de Gibbs*, donde la cadena de Markov subyacente se construye en base a

en una secuencia de distribuciones condicionales. La Sección 6.5 explica cómo muestrear las distribuciones que surgen en los modelos de Ising y Potts, que se utilizan ampliamente en la mecánica estadística, y la Sección 6.6 trata de las aplicaciones de MCMC en las estadísticas bayesianas. En la Sección 6.7 mostramos que los muestreadores de Metropolis-Hastings y Gibbs pueden verse como casos especiales de un algoritmo MCMC general y presentamos los muestreadores de corte y de salto reversible. La sección 6.8 trata sobre el método clásico de recocido simulado para encontrar el mínimo global de una función multiextremal, que se basa en el método MCMC. Finalmente, la Sección 6.9 presenta el método de muestreo perfecto, para muestrear exactamente de una distribución objetivo en lugar de aproximadamente.

6.2 ALGORITMO METRÓPOLIS-HASTINGS

La idea principal detrás del algoritmo Metropolis-Hastings es simular una cadena de Markov de modo que la distribución estacionaria de esta cadena coincida con la distribución objetivo.

Para motivar el método MCMC, supongamos que queremos generar una variable aleatoria X tomando valores en $X = \{1, \dots, m\}$, según una distribución objetivo $\{\pi_i\}$, con

$$\pi_i = \frac{b_i}{\sum_{j \in X} b_j}, \quad (6.1)$$

donde se supone que $\sum_{j \in X} b_j$ es estrictamente positivos, b_j son grandes, y la normalización constante $C = \sum_{j \in X} b_j$ es difícil de calcular. Siguiendo a Metropolis et al. [18], construimos una cadena de Markov $\{X_t, t=0, 1, \dots\}$ en X cuya evolución se basa en una matriz de transición arbitraria $q = (q_{y0})$ de la siguiente manera:

- Cuando $X_t = i$, generar una variable aleatoria Y satisfactorio $PAGS(Y=j) = q_{y0}, j \in X$. De este modo, Y se genera a partir de la *metrodistribución* de puntos dada por la i -ésima fila de q .
- Si $Y \neq j$, dejar $\{$

$$X_{t+1} = \begin{cases} j & \text{con probabilidad } a_{y0} = \min \left\{ \frac{\pi_j q_{ji}}{\pi_i q_{jy}}, 1 \right\} \\ i & \text{con probabilidad } 1 - a_{y0}. \end{cases}$$

Resulta que $\{X_t, t=0, 1, \dots\}$ tiene una matriz de transición de un paso $PAGS = (pags_{y0})$ dada por

$$pags_{y0} = \begin{cases} q_{y0} a_{y0} & \text{si } i \neq j, \\ 1 - \sum_{k \in X} q_{y0} a_{y0} & \text{si } i = j. \end{cases} \quad (6.2)$$

Ahora es fácil comprobar (ver Problema 6.1) que, con a_{y0} como anteriormente,

$$\pi_i pags_{y0} = \pi_j pags_{ji}, \quad y0, j \in X. \quad (6.3)$$

En otras palabras, *ecuaciones de equilibrio detalladas* (1.38) se cumple, y por lo tanto la cadena de Markov es reversible en el tiempo y tiene probabilidades estacionarias $\{\pi_i\}$. Además, esta distribución estacionaria es también la *limitando* distribución si la cadena de Markov es irreducible y aperiódica. Tenga en cuenta que no hay necesidad de la constante de normalización C en (6.1) para definir la cadena de Markov.

La extensión del enfoque MCMC anterior para generar muestras a partir de un pdf multidimensional arbitrario $P(\mathbf{X})$ (en vez de π_i) es sencillo. En este caso, la función de transición de probabilidad no negativa $q(\mathbf{X}, \mathbf{y})$ (tomando el lugar de q_{y0}

arriba) es a menudo llamado el *propuesta o instrumental* función. Al ver esta función como un pdf condicional, también podemos escribir $q(\mathbf{y}|\mathbf{X})$ en vez de $q(\mathbf{X}, \mathbf{y})$. La probabilidad $\alpha(\mathbf{X}, \mathbf{y})$ se llama el *probabilidad de aceptación*. El algoritmo original de Metropolis [18] fue sugerido para funciones propuestas simétricas, es decir, para $q(\mathbf{X}, \mathbf{y}) = q(\mathbf{y}, \mathbf{X})$. Hastings modificó el algoritmo MCMC original para permitir funciones de propuesta no simétricas. Tal algoritmo se llama *Algoritmo de Metropolis-Hastings*. A la cadena de Markov correspondiente la llamamos *Cadena Metrópolis-Hastings Markov*.

En resumen, el algoritmo de Metropolis-Hastings, que, al igual que el método de aceptación-rechazo, se basa en una estrategia de prueba y error, es el siguiente:

Algoritmo 6.2.1: Algoritmo de Metropolis-Hastings

aporte : Estado inicial \mathbf{X}_0 y tamaño de la muestra n . pdf objetivo $F(\mathbf{X})$ y propuesta función $q(\mathbf{X}, \mathbf{y})$.
producción: cadena de Markov $\mathbf{X}_1, \dots, \mathbf{X}_n$ distribuidas aproximadamente según $F(\mathbf{X})$.

1 **por** $t=0$ **a** $n-1$ **hacer**

```

2   Generar  $\mathbf{Y} \sim q(\mathbf{X}_t, \mathbf{y})$  // dibujar una probabilidad
3    $\alpha \leftarrow \min\left\{\frac{F(\mathbf{Y})q(\mathbf{X}_t, \mathbf{Y})}{F(\mathbf{X}_t)q(\mathbf{Y}, \mathbf{X}_t)}, 1\right\}$  //  $\alpha$  de aceptación de propuesta
4   Generar  $u \sim \text{tu}(0, 1)$  si
5    $u \leq \alpha$  después
6   |  $\mathbf{X}_{t+1} \leftarrow \mathbf{Y}$ 
7   más
8   |  $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t$ 
```

9 **volver** $\mathbf{X}_1, \dots, \mathbf{X}_n$

El algoritmo produce una secuencia $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$ de *dependiente* variables aleatorias, con \mathbf{X}_t distribuidas aproximadamente según $F(\mathbf{X})$, para grande t .

Dado que el algoritmo 6.2.1 es del tipo aceptación-rechazo, su eficiencia depende de la probabilidad de aceptación $\alpha(\mathbf{X}, \mathbf{y})$. Idealmente, nos gustaría $q(\mathbf{X}, \mathbf{y})$ para reproducir el pdf deseado $F(\mathbf{y})$ con la mayor fidelidad posible. Un enfoque común [20] es primero parametrizar $q(\mathbf{X}, \mathbf{y})$ como $q(\mathbf{X}, \mathbf{y}; \theta)$ y luego usar métodos de optimización estocástica para maximizar esto con respecto a θ . A continuación consideramos varias opciones particulares de $q(\mathbf{X}, \mathbf{y})$.

■ EJEMPLO 6.1 Muestreador de independencia

El algoritmo MCMC de tipo Metropolis más simple se obtiene eligiendo la función propuesta $q(\mathbf{X}, \mathbf{y})$ ser independiente de \mathbf{X} , eso es, $q(\mathbf{X}, \mathbf{y}) = \text{gramo}(\mathbf{y})$ para algunos pdf $\text{gramo}(\mathbf{y})$. Así, partiendo de un estado anterior \mathbf{X} un estado candidato \mathbf{Y} se genera a partir de $\text{gramo}(\mathbf{y})$ y aceptado con probabilidad

$$\alpha(\mathbf{X}, \mathbf{Y}) = \min \left\{ \frac{F(\mathbf{Y})\text{gramo}(\mathbf{X})}{F(\mathbf{X})\text{gramo}(\mathbf{Y})}, 1 \right\}.$$

Este procedimiento es muy similar a los métodos originales de aceptación-rechazo del Capítulo 2 y, como en ese método, es importante que la distribución de la propuesta gramo esté cerca del objetivo F . Tenga en cuenta que, en contraste con el método de aceptación-rechazo, el muestreador de independencia produce *dependiente* muestras

■ EJEMPLO 6.2 Muestreo Uniforme

Ser capaz de muestrear *uniformemente* de algún conjunto discreto Y es importante en muchas aplicaciones; ver, por ejemplo, los algoritmos para contar en el Capítulo 9. Un procedimiento general simple es el siguiente: Defina un *vecindario* estructura en Y . Se permite cualquier estructura de vecindad, siempre que la cadena Metrópolis-Hastings Markov resultante sea irreducible y aperiódica. Dejar $norte_x$ ser el número de vecinos de un estado X . Para la distribución propuesta, simplemente elegimos cada posible vecino del estado actual X con igual probabilidad. Eso es, $q(X, y) = 1/norte_x$. Desde el pdf de destino $f(X)$ aquí es constante, la probabilidad de aceptación es

$$\alpha(X, y) = \min\{norte_x/norte_y, 1\}.$$

Por construcción, la distribución límite de la cadena Metropolis-Hastings Markov es la distribución uniforme en Y .

■ EJEMPLO 6.3 Muestreador de paseo aleatorio

En el muestreador de caminata aleatoria, el estado de la propuesta Y , para un estado actual dado X , es dado por $Y = X + Z$, donde Z se genera típicamente a partir de alguna distribución esféricamente simétrica (en el caso continuo), como $NORTE(0, \Sigma)$. Tenga en cuenta que la función propuesta es simétrica en este caso; de este modo,

$$\alpha(X, y) = \min \left\{ \frac{f(y)}{f(X)}, 1 \right\}. \quad (6.4)$$

■ EJEMPLO 6.4

Deje que el vector aleatorio $X = (X_1, X_2)$ tienen el siguiente pdf bidimensional:

$$f(X) = C \exp(-\{X_2^2 - X_1 X_2 + X_2^2 - 8X_1 - 8X_2\}/2), \quad (6.5)$$

dónde $C \approx 1/20216.335877$ es una constante de normalización. El gráfico de esta densidad se representa en la Figura 6.1.

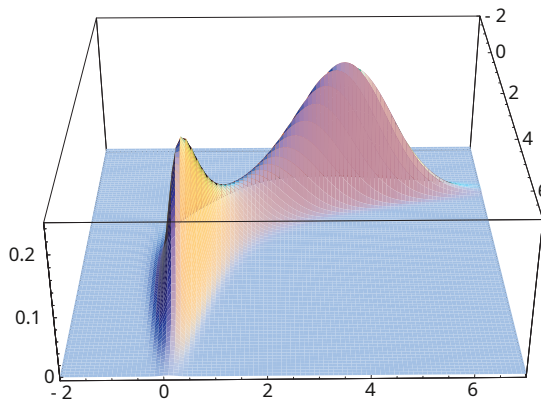


Figura 6.1: La densidad $f(X_1, X_2)$.

Supongamos que deseamos estimar $\mu = \mathbb{E}[X_1]$ a través del estimador CMC

$$\hat{\mu} = \frac{1}{n} \sum_{t=1}^n X_t$$

usando el muestreador de paseo aleatorio para generar una muestra dependiente \mathbf{X}_t de $\mathcal{F}(\mathbf{X})$. Una elección simple para el incremento \mathbf{Z} es dibujar los componentes de \mathbf{Z} independientemente, de un $\text{NORTE}(0, a^2)$ distribución para algunos $a > 0$. Tenga en cuenta que, si a se elige demasiado pequeño, digamos menos de 0.5, los componentes de las muestras estarán fuertemente correlacionados positivamente, lo que conducirá a una gran variación para $\hat{\mu}$. Por otra parte, para a demasiado grande, digamos mayor que 10, la mayoría de las muestras serán rechazadas, lo que nuevamente conducirá a una baja eficiencia. A continuación elegimos un valor moderado de a , decimos $a=2$. El muestreador de caminata aleatoria ahora se resume de la siguiente manera:

Algoritmo 6.2.2: Muestreador de paseo aleatorio

```

1 Inicializar  $\mathbf{X}_0 \leftarrow (\mathbf{X}_{01}, \mathbf{X}_{02})$ .
2  $\text{port} \leftarrow 0$  hacer  $n$  veces
3   Dibujar  $\mathbf{Z}_1, \mathbf{Z}_2 \sim \text{NORTE}(0, 1)$  de forma
4   independiente. Establecer  $\mathbf{Z} \leftarrow (\mathbf{Z}_1, \mathbf{Z}_2)$  y  $\mathbf{Y} \leftarrow \mathbf{X}_t + \mathbf{Z}$ .
5   Establecer  $\alpha \leftarrow \min\left(\frac{\mathcal{F}(\mathbf{Y})}{\mathcal{F}(\mathbf{X}_t)}, 1\right)$ .
6   Generar  $u \sim \text{unif}(0, 1)$ . si  $u \leq \alpha$  después
7      $\mathbf{X}_{t+1} \leftarrow \mathbf{Y}$ 
8   más
9      $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t$ 
10
```

Ejecutamos este algoritmo para producir $n=10^5$ muestras. Los últimos cientos de estos se muestran en el gráfico de la izquierda de la Figura 6.2. Vemos que las muestras siguen de cerca el gráfico de contorno del pdf, lo que indica que se ha muestreado la región correcta. Esto es corroborado por el gráfico de la derecha de la Figura 6.2, donde vemos que el histograma de los X_1 valores está cerca del pdf verdadero (línea continua).

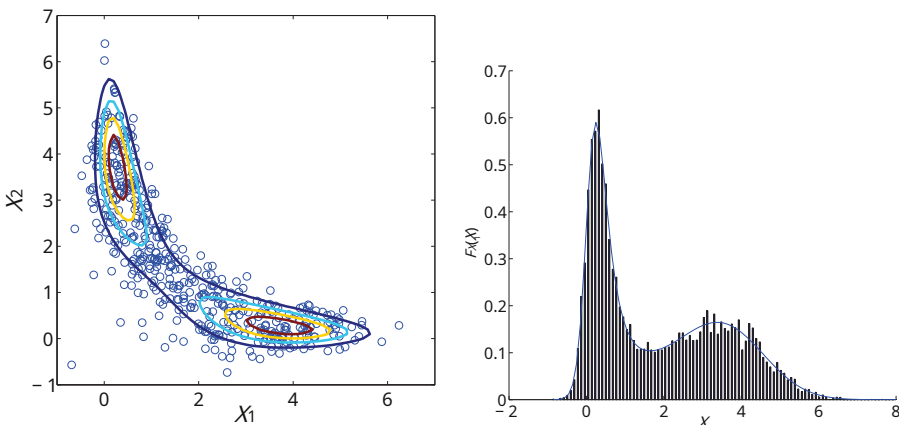


Figura 6.2: El gráfico de la izquierda muestra algunas muestras del muestreador de paseo aleatorio junto con varias líneas de contorno de \mathcal{F} . El gráfico de la derecha muestra el histograma de los X_1 valores junto con la densidad real de X_1 .

Obtuvimos una estimación = 1.89 (el valor real es $MI[X_1] \approx 1.85997$). Para obtener un IC, podemos usar (4.21), donde se estima la varianza asintótica, o emplea el método de medias por lotes de la Sección 4.4.2.1. La figura 6.3 muestra la función de (auto)covarianza estimada $\hat{R}(k)$ por $k=0,1, \dots, 400$. Vemos que hasta alrededor de 100 las covarianzas no son despreciables. Por lo tanto, para estimar la varianza de $\hat{\mu}$, necesitamos incluir todos los términos distintos de cero en (4.20), no solo la varianza $\hat{R}(0)$ de X_1 . Sumando los primeros 400 rezagos, obtuvimos una estimación de 10.41 para la varianza asintótica. Esto da un error relativo estimado de 0.0185 y un IC del 95% de (1.82, 1.96). Se encontró un IC similar al usar el método de medias de lotes con 500 lotes de tamaño 200.

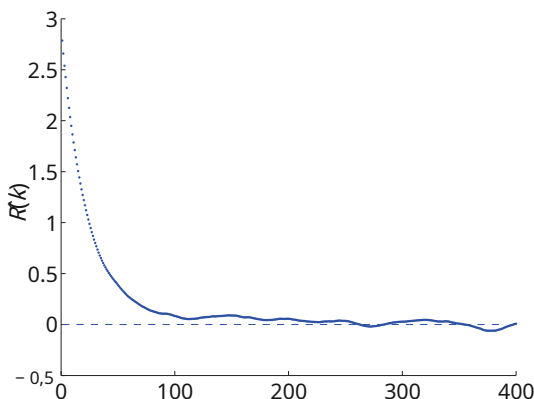


Figura 6.3: La función de covarianza estimada para el $\{X_n\}$ para retrasos k hasta 400.

Si bien MCMC es un método genérico y se puede utilizar para generar muestras aleatorias prácticamente a partir de cualquier distribución objetivo, independientemente de su dimensionalidad y complejidad, los problemas potenciales con el método MCMC son:

1. Las muestras resultantes suelen estar altamente correlacionadas.
2. Por lo general, lleva una cantidad de tiempo considerable hasta que la cadena de Markov subyacente se estabilice en su estado estacionario.
3. Las estimaciones obtenidas a través de muestras MCMC a menudo tienden a tener variaciones mucho mayores que las obtenidas a partir del muestreo independiente de la distribución objetivo. Se han hecho varios intentos para superar esta dificultad. Para más detalles véase, por ejemplo, [14] y [20].

Observación 6.2.1 En este punto debemos recalcar que aunque es una práctica común usar MCMC para tomar muestras de $f(\mathbf{X})$ para estimar cualquier expectativa $= \mathbb{E}[f(\mathbf{X})]$, la *actua*l objetivo para estimar es $\text{gramo}(\mathbf{X}) \propto H(\mathbf{X}) / f(\mathbf{X})$. Es decir, el muestreo de *gramo*(\mathbf{X}) da un estimador de varianza mínima (varianza cero en el caso $H(\mathbf{X}) = 0$). Por lo tanto, es importante distinguir claramente entre el uso de MCMC para generar desde algunos pdf difíciles $f(\mathbf{X})$ y usando MCMC para estimar una cantidad como $\text{gramo}(\mathbf{X})$. Para este último problema se pueden utilizar técnicas mucho más eficientes, como el muestreo por importancia; además, se puede obtener adaptativamente un pdf de muestreo de buena importancia, como con los métodos CE y TLR.

6.3 MUESTREO DE HIT-AND-RUN

los *pega y corre* sampler, iniciado por Robert Smith [25], se encuentra entre los primeros samplers MCMC en la categoría de *muestreadores de línea* [2]. Como en el apartado anterior, el objetivo es muestrear a partir de una distribución objetivo $P(\mathbf{X})$ en $X \subset \mathbb{R}^n$. Los muestreadores de línea brindan la oportunidad de llegar a toda la región factible X en un solo paso

En primer lugar, describimos el sampler hit-and-run original para generar a partir de una *uniforme* distribución en una región abierta delimitada X de \mathbb{R}^n . En cada iteración, a partir de un punto actual \mathbf{x} , a *vectores de dirección* \mathbf{d} se genera uniformemente en la superficie de un *nórt* hiperesfera dimensional. La intersección de la línea bidireccional correspondiente (a través de \mathbf{x}) y la caja adjunta de X define un segmento de línea L . el siguiente punto \mathbf{y} luego se selecciona uniformemente de la intersección de $L \cap X$.

La figura 6.4 ilustra el algoritmo hit-and-run para generar uniformemente a partir del conjunto X (la región gris), que está delimitada por un cuadrado. Dado el punto \mathbf{x} en X , una dirección aleatoria \mathbf{d} se genera, que define el segmento de línea $L = \text{ultravioleta}$. Entonces un punto \mathbf{y} se elige uniformemente en $L \cap X$, por ejemplo, por el método de aceptación-rechazo; es decir, se genera un punto uniformemente en L y luego acepta este punto sólo si se encuentra en X .

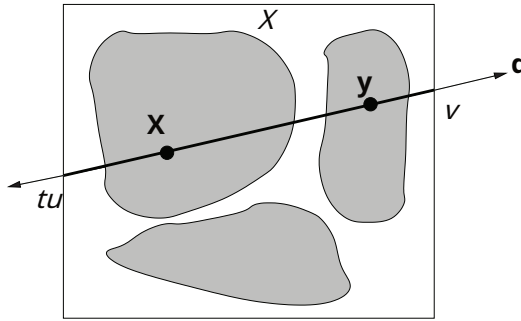


Figura 6.4: Ilustración del algoritmo hit-and-run en un cuadrado en dos dimensiones.

Smith [25] demostró que golpear y correr genera asintóticamente puntos uniformemente distribuidos sobre *arbitrario* regiones abiertas de \mathbb{R}^n . Una propiedad deseable de hit-and-run es que puede alcanzar globalmente cualquier punto del conjunto en un solo paso; es decir, existe una probabilidad estrictamente positiva de muestrear cualquier vecindario del conjunto. Esta propiedad, junto con una propiedad de simetría, es importante para derivar la distribución límite. Lovász [15] demostró que el atropello y la fuga de un cuerpo convexo en *nórt* dimensiones produce un punto de muestra distribuido aproximadamente uniformemente en tiempo polinomial, $O(n^3)$, el límite más conocido para dicho algoritmo de muestreo. Señaló que, en la práctica, el algoritmo hit-and-run parece ofrecer la convergencia más rápida a una distribución uniforme [15, 16]. Hit-and-run es único en el sentido de que solo se necesita un tiempo polinomial para salir de una esquina; a diferencia de, *paseo de pelota* tarda un tiempo exponencial en salir de una esquina [17].

Tenga en cuenta que el algoritmo hit-and-run descrito anteriormente es un caso especial del Algoritmo Metropolis-Hastings 6.2.1, donde la función de propuesta $q(\mathbf{x}, \mathbf{y})$ es simétrico y el objetivo $P(\mathbf{X})$ es constante. De ello se deduce que cada punto candidato es

aceptado con probabilidad 1. Para generar a partir de un *genera/pdf* continuo estrictamente positivo $R(\mathbf{X})$, uno puede simplemente modificar el algoritmo uniforme hit-and-run anterior aceptando el candidato \mathbf{y} con probabilidad

$$\alpha(\mathbf{X}, \mathbf{y}) = \min\{R(\mathbf{y})/R(\mathbf{X}), 1\}, \quad (6.6)$$

como en el Algoritmo 6.2.1 (nótese que $q(\mathbf{y}, \mathbf{X})/q(\mathbf{X}, \mathbf{y})$ es igual a 1). Por lo tanto, el algoritmo general hit-and-run con el criterio de aceptación anterior de Metropolis se resume de la siguiente manera [21]:

Algoritmo 6.3.1: Pega y corre

aporte : región delimitada X , objetivo pdf F en X , tamaño de la muestra

norte. **salida**: $\mathbf{X}_1, \dots, \mathbf{X}_{norte}$ distribuidas aproximadamente según F .

1 Inicializar $\mathbf{X}_0 \in X$.

2 **por** $t=0$ **a** *norte* -1 **hacer**

3 **repetir**

4 Generar una dirección aleatoria \mathbf{d} según una distribución uniforme en la unidad *norte* hiperesfera dimensional.

5 Dejar $METRO_t \leftarrow \{\mathbf{X} \in X: \mathbf{X} = \mathbf{X}_t + \lambda \mathbf{d}_t, \lambda \in \mathbb{R}\}$.

6 **Hasta que** $METRO_t = \emptyset$

7 Generar un punto candidato \mathbf{Y} distribuido uniformemente sobre el conjunto de líneas $METRO_t$.

8 Generar $u \sim \text{tu}(0, 1)$.

9 **si** $u < \min\{R(\mathbf{Y})/R(\mathbf{X}_t), 1\}$ **después**

10 $\mathbf{X}_{t+1} \leftarrow \mathbf{Y}$

11 **más**

12 $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t$

13 **volver** $\mathbf{X}_1, \dots, \mathbf{X}_{norte}$

Chen y Schmeiser [5] describen cómo se puede generalizar el muestreador hit-and-run para muestrear cualquier pdf en cualquier región delimitada o no delimitada X .

El algoritmo hit-and-run se puede integrar dentro de un marco de optimización para generar dos algoritmos de optimización global: *al esconderse* [21] y *mejorar el golpe y la fuga* [27]. Este último se ha aplicado con éxito a problemas prácticos, incluido el diseño de materiales compuestos y la optimización de la forma, y se ha demostrado que tiene una complejidad polinomial, en promedio, para una clase de programas cuadráticos. En la Sección 6.8, mostramos cómo convertir un muestreador MCMC en un algoritmo de optimización mediante recocido simulado.

6.4 MUESTREO GIBBS

los *Muestreador de Gibbs* (Geman y Geman [7]) utiliza una metodología algo diferente del algoritmo Metropolis-Hastings y es particularmente útil para generar *norte*- vectores aleatorios dimensionales. La característica distintiva del muestreador de Gibbs es que la cadena de Markov subyacente se construye, de manera determinista o aleatoria, a partir de una secuencia de distribuciones condicionales.

El muestreo de Gibbs es ventajoso si es más fácil tomar muestras de las distribuciones condicionales que de la distribución conjunta. La idea esencial de la muestra de Gibbs: actualizar una parte del elemento anterior manteniendo las otras partes

fijo: es útil en muchos casos en los que la variable de estado es una variable aleatoria que toma valores en un espacio general, no solo en \mathbb{R}^{norte} (ver [12]).

Supongamos que deseamos muestrear un vector aleatorio $\mathbf{X} = (X_1, \dots, X_{norte})$ según un pdf de destino $F(\mathbf{X})$. Dejar $F(X_i | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_{norte})$ representan la función de densidad de probabilidad condicional de la i -ésima componente, X_i , dados los otros componentes $X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_{norte}$. La muestra de Gibbs se da a continuación.

Algoritmo 6.4.1: Muestreador de Gibbs

aporte : Punto inicial \mathbf{X}_0 , tamaño de la muestra $norte$, y pdf de destino F .

salida: $\mathbf{X}_1, \dots, \mathbf{X}_{norte}$ distribuidas aproximadamente según F .

1 **port** ← 0 **a norte** - 1 **hacer**

2 Dibujar Y_i del pdf condicional $F_{Y_i | X_{y0,2}, \dots, X_{Tennessee}}$. **por** i

3 = 2 **a norte** **hacer**

4 Dibujar Y_i del pdf condicional $F_{Y_i | Y_1, \dots, Y_{i-1}, X_{y0+1}, \dots, X_{Tennessee}}$.

5 $\mathbf{X}_{t+1} \leftarrow \mathbf{Y}$

evolver $\mathbf{X}_1, \dots, \mathbf{X}_{norte}$

Tenga en cuenta que en la muestra de Gibbs *todos* se aceptan muestras, en contraste con el algoritmo Metropolis-Hastings. Veremos en la Sección 6.7 que bajo condiciones suaves la distribución limitante del proceso $\{\mathbf{X}_t, t=1, 2, \dots\}$, generado a través de la muestra de Gibbs, es precisamente $F(\mathbf{X})$. Además, bajo algunas otras condiciones simples, se puede demostrar (ver [14], [20]) que la convergencia a la fdp deseada es geoméricamente rápida.

■ EJEMPLO 6.5 Ejemplo 6.4 (Continuación)

Mostraremos cómo muestrear fácilmente desde el pdf F en (6.5) a través del muestreador de Gibbs. Empezamos escribiendo

$$F(x, y) = C_1(y) \exp \left(- \frac{1+y_2}{2} \left(x - \frac{4}{1+y_2} \right)^2 \right),$$

dónde $C_1(y)$ depende sólo de y ; vemos que, condicionado a y , X_i tiene una distribución normal con expectativa $4/(1+y_2)$ y varianza $1/(1+y_2)$. La distribución condicional de Y dado X sigue de la misma manera. La muestra de Gibbs correspondiente es, por lo tanto, la siguiente:

Algoritmo 6.4.2: Muestreador de Gibbs para el ejemplo 6.4

1 Inicializar Y_0 .

2 **port** ← 0 **a norte** - 1 **hacer**

3 Dibujar $Z_1, Z_2 \sim \text{NORTE}(0,1)$.

4 $X_{t+1} \leftarrow Z_1 / \sqrt{\frac{1+Y_2+4/(1+Y_2)}{t}}$

5 $Y_{t+1} \leftarrow Z_2 / \sqrt{\frac{1+X_2+4/(1+X_2)}{t+1}}$

Observación 6.4.1 (Muestreadores de Gibbs sistemáticos y aleatorios) Tenga en cuenta que el Algoritmo 6.4.1 presenta un *sistemático* Muestreador de Gibbs por coordenadas. es decir, el vector

Se actualiza en un orden determinista: $1, 2, \dots, n, 1, 2, \dots$. En el *aleatorio* muestreador de Gibbs por coordenadas, las coordenadas se eligen al azar, por ejemplo, generándolas independientemente de un uniforme discreto *norte*-punto pdf. En ese caso, la muestra de Gibbs puede verse como una instancia de la muestra de Metropolis-Hastings, es decir, con la función de transición

$$q(\mathbf{X}, \mathbf{y}) = \frac{1}{norte} F(y_i | X_1, \dots, X_{y_0-1}, X_{i+1}, \dots, X_{norte}) = \frac{1}{norte} \frac{F(\mathbf{y})}{F(y_i)},$$

ya que $\mathbf{y} = (X_1, \dots, X_{y_0-1}, y_i, X_{i+1}, \dots, X_{norte})$. Ya que $\sum_{\mathbf{y}} F(\mathbf{y})$ también se puede escribir como $F(\mathbf{X})$, tenemos

$$q(\mathbf{X}, \mathbf{y}) = \frac{F(\mathbf{y}) q(\mathbf{y}, \mathbf{X})}{F(\mathbf{X}) q(\mathbf{X}, \mathbf{y})} = \frac{F(\mathbf{y}) F(\mathbf{X})}{F(\mathbf{X}) F(\mathbf{y})} = 1,$$

de modo que la probabilidad de aceptación $\alpha(\mathbf{X}, \mathbf{y})$ es 1 en este caso.

Aquí hay otro ejemplo de una aplicación de la muestra de Gibbs.

■ EJEMPLO 6.6 Red Cerrada de Colas en Forma de Producto

Considerar *metro* clientes que se mueven entre *norte* colas en una red de colas cerrada. Denotamos por $X_i(t)$ el número de clientes en cola $y_0 = 1, \dots, norte$, y deja $\mathbf{X}(t) = (X_1(t), \dots, X_{norte}(t))$ y $\mathbf{X} = (X_1, \dots, X_{norte})$. Es bien sabido [22] que si el límite

$$\lim_{t \rightarrow \infty} \text{PAGS}(\mathbf{X}(t) = \mathbf{X}) = \pi(\mathbf{X})$$

existe, entonces, para tiempos de servicio distribuidos exponencialmente, la función de densidad de probabilidad conjunta discreta $\pi(\mathbf{X})$ se puede escribir en forma de producto como

$$\pi(\mathbf{X}) = C \prod_{i=1}^{norte} F(X_i), \quad \text{por } \sum_{i=1}^{norte} X_i = m, \quad (6.7)$$

donde el $\{F(X_i), X_i \geq 0\}$ son *conocido* pdf discretos, y C es una constante de normalización. Para un ejemplo concreto, vea el Problema 6.11.

La C constante es en general difícil de calcular. Para proceder, escribir $S(\mathbf{X}) = \sum_{i=1}^{norte} X_i$ y $\mathbf{X} = \{\mathbf{X}: S(\mathbf{X}) = metro\}$, tenemos

$$C^{-1} = \sum_{\mathbf{X} \in \mathbf{X}} \prod_{i=1}^{norte} F(X_i), \quad (6.8)$$

que requiere la evaluación del p(producto) de *norte* pdf para cada \mathbf{X} en el conjunto \mathbf{X} . Este conjunto tiene un total de $X^*/norte$ elementos (vea el problema 6.10), que rápidamente crece mucho.

Ahora mostramos cómo calcular C basado en el muestreo de Gibbs. Para aplicar el muestreador de Gibbs, necesitamos poder generar muestras a partir de la distribución condicional de X_i dado el otro componente \mathbf{X}_{-i} . Tenga en cuenta que sólo tenemos que generar X_1, \dots, X_{n-1} , ya que $X_{norte} = m - \sum_{k=1}^{n-1} X_k$. Para $i = 1, \dots, norte - 1$, nosotros tener

$$F(X_i | X_1, \dots, X_{y_0-1}, X_{i+1}, \dots, X_{n-1}) \propto F(X_i) F_{norte} \left(\sum_{k=1}^{n-1} X_k \right) \quad (6.9)$$

por $X_i \in \{0, 1, \dots, \text{metro} - x_1, \dots, -X_{y_0-1} - X_{i+1} - \dots - X_{n-1}\}$. El muestreo de estos pdf condicionales a menudo se puede hacer de manera eficiente, en particular cuando el $\{F_i\}$ son miembros de una familia exponencial; véase también el Problema 6.11.

Ahora que podemos muestrear (aproximadamente) de $\pi(\mathbf{X})$, es sencillo estimar la constante de normalización C al observar que

$$\left[\frac{1}{\prod_{i=1}^n F_i(X_i)} \right] = \sum_{\mathbf{X} \in \mathcal{X}_*} \prod_{i=1}^n \frac{1}{F_i(X_i)} C \prod_{i=1}^n F_i(X_i) = X_*/C.$$

Esto sugiere el siguiente estimador para C , obtenido de una muestra aleatoria $\mathbf{X}_1, \dots, \mathbf{X}_{\text{norte}}$ de π :

$$C = \frac{(\text{metro} + \text{norte} - 1)^{-1} \sum_{k=1}^{\text{norte}} \prod_{i=1}^{\text{metro}} \frac{1}{F_i(X_{ki})}}{\text{norte} - 1},$$

dónde X_{ki} es el i -ésimo componente de \mathbf{X}_k .

6.5 MODELOS DE MACETAS Y MACETAS

6.5.1 Modelo de salida

El modelo de Ising es uno de los modelos más populares y estudiados en mecánica estadística. Describe la interacción de imanes idealizados, llamados *giros*, que se encuentran en una red de dos o tres dimensiones. En el caso bidimensional básico, los giros se encuentran en la red $\{1, \dots, n\} \times \{1, \dots, n\}$, y cada uno de los n^2 sitios tienen cuatro vecinos más cercanos, posiblemente incluidos los sitios limítrofes, que "envuelven" al otro lado de la red, creando lo que se denomina *tubo*. Consulte la Figura 6.5, donde los cuatro sitios de color gris claro son los vecinos del sitio de color gris oscuro.

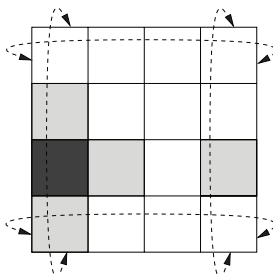


Figura 6.5: Los sitios limítrofes se envuelven. Los vecinos del sitio gris oscuro son los sitios gris claro.

Dejar $\{1, \dots, \text{norte}\}$ ser una enumeración de los sitios. Cada giro puede ser en uno de dos estados: -1 o 1 . Cada uno de los 2^{norte} configuraciones de giros $\mathbf{s} = (s_1, \dots, s_{\text{norte}})$ lleva una cantidad de *energía total*

$$m(\mathbf{s}) = -A \sum_{i \leftrightarrow j} s_i s_j - B \sum_i s_i,$$

dónde A y B son constantes $\sum_{i \rightarrow j} S_i S_j$ en la mayoría de las llamadas $A=1$ y $B=0$, que ahora haremos asumir. Las cantidades $\sum_{i \rightarrow j} S_i S_j$ se llaman *energía de interacción* y $\sum_{i \rightarrow j} S_i$ indica que la suma es tomada el control de pares vecinos (y, j) .

En equilibrio térmico, la distribución de los espines, digamos π , sigue la ley de Boltzmann: $\pi(\mathbf{s}) \propto \text{Exp}(-E(\mathbf{s})/T)$, donde T es una temperatura fija. En otras palabras, tenemos

$$\pi(\mathbf{s}) = \frac{\exp\left(-\frac{1}{T} \sum_{i \rightarrow j} S_i S_j\right)}{Z},$$

dónde Z es la constante de normalización, llamada *parte* cantidades particulares de interés son la *energía media por la magnetización media por espín* $\langle S_i \rangle$ y $\langle S_i^2 \rangle$. Estos a través de la simulación de Monte Carlo, siempre que se pueda muestrear la distribución π (vea abajo).

En la Figura 6.6 una muestra de π se da (blancos $K=1$, la llamada *temperatura crítica* $T=2/\ln(1+2) \approx$



Figura 6.6: Configuración de Ising a la temperatura crítica.

A continuación definimos el *Modelo Potts*—que puede verse como una generalización del modelo de Ising, y explicar cómo generar muestras a partir de este modelo extendido y, por lo tanto, en particular, cómo generar la Figura 6.6.

6.5.2 Modelo de macetas

Dejar $\{1, \dots, J\}$ sea una enumeración de posiciones espaciales (sitios), y sea $\psi_{y \circ}$ ser alguna función simétrica y positiva que relacione los sitios entre sí, por ejemplo,

$$\psi_{y \circ} = \begin{cases} \beta > 0 & \text{si } y, j \text{ son vecinos, caso} \\ 0 & \text{contrario.} \end{cases} \quad (6.10)$$

Asignar a cada sitio un color" X_i . Supongamos que hay k tales colores, etiquetados $\{1, \dots, K\}$. Definir $\mathbf{X} = (X_1, \dots, X_j)$, y deja \mathcal{X} ser el espacio de tales configuraciones. En \mathcal{X} definimos el pdf objetivo $F(\mathbf{X}) \propto \prod_{i \sim j} \psi_{y_{ij}}(X_i, X_j)$ con

$$H(\mathbf{X}) = \sum_{y_{ij} < j} \psi_{y_{ij}}(X_i, X_j) \quad .$$

Para ver que el modelo de Ising es un caso especial del modelo de Potts, defina $\mathcal{X} = \{y_{ij} \in \{1, \dots, K\} \mid y_{ij} = y_{ji}\}$ un \mathcal{X} como en (6.10), con $\beta = 4/T$. Después

$$\frac{1}{T} \sum_{i \sim j} s_i s_j = \frac{1}{T} \sum_{i \sim j} \left(\frac{1}{2} \sum_{y_{ij} \in \mathcal{X}} y_{ij} (X_i - X_j) - \frac{1}{2} \sum_{y_{ij} < j} \psi_{y_{ij}}(X_i, X_j) + \text{constante} \right),$$

de modo que $\pi(\mathbf{s}) = F(\mathbf{X})$.

A continuación, mostramos cómo generar una muestra a partir del pdf de destino $F(\mathbf{X})$. Para ello, definimos variables aleatorias auxiliares $Y_{y_{ij}}, 1 \leq y_{ij} < j$, tal que condicionado a $\mathbf{X} = \mathbf{X}$ la $\{Y_{y_{ij}}\}$ son independientes y cada uno $Y_{y_{ij}}$ se distribuye uniformemente en el intervalo $[0, a_{y_{ij}}]$, con $a_{y_{ij}} = \exp(\psi_{y_{ij}}(X_i, X_j)) - 1$. En otras palabras, el condicional pdf de $\mathbf{Y} = \{Y_{y_{ij}}\}$ dado $\mathbf{X} = \mathbf{X}$ es

$$F(\mathbf{y} | \mathbf{X}) = \prod_{y_{ij} < j} \frac{\mathbb{I}_{\{y_{ij} \in [0, a_{y_{ij}}]\}}}{a_{y_{ij}}} = \prod_{y_{ij} < j} \mathbb{I}_{\{y_{ij} \in [0, a_{y_{ij}}]\}} \quad \text{mi } -H(\mathbf{X}).$$

La importancia de esto es que el pdf conjunto de \mathbf{X} y \mathbf{Y} ahora es simplemente

$$F(\mathbf{X}, \mathbf{y}) = F(\mathbf{X}) F(\mathbf{y} | \mathbf{X}) \propto \begin{cases} 1 & \text{si } y_{ij} \in [0, a_{y_{ij}}] \text{ para todos } y_{ij} < j, \\ 0 & \text{de lo contrario.} \end{cases}$$

En otras palabras, (\mathbf{X}, \mathbf{Y}) es *uniformemente* repartido. Más importante, porque $F(\mathbf{X}, \mathbf{y}) \propto F(\mathbf{X}, \mathbf{y})$, encontramos eso \mathbf{X} y \mathbf{y} se distribuye uniformemente en el conjunto $A = \{\mathbf{X}, \mathbf{y} : y_{ij} \in [0, a_{y_{ij}}] \text{ para todos } y_{ij} < j\}$. ahora bien $y_{ij} \in [0, 1]$ o $y_{ij} \in [1, \exp(\psi_{y_{ij}})]$. En el primer caso, para cualquier $\mathbf{X} \in A$, las coordenadas X_i, X_j rango sobre todos los colores, y por la uniformidad, cada color es igualmente probable. Pero en este último caso, X_i debe ser igual a X_j . Así, para un dado \mathbf{y} , los sitios $y_{ij}, j(\text{con } y_{ij} < j)$ para cualquier $y_{ij} > 1$ se pueden agrupar en grupos, y dentro de cada grupo, los sitios tienen colores idénticos. Además, dado \mathbf{y} , los colores dentro de los grupos son independientes y uniformemente distribuidos en $\{1, \dots, K\}$. Lo mismo ocurre con los colores de las posiciones restantes, que se pueden ver como sitios de un grupo.

Por lo tanto, podemos generar fácilmente ambos \mathbf{X} y \mathbf{y} y \mathbf{X} . Como consecuencia, podemos usar el muestreador de Gibbs para muestrear (aproximadamente) $F(\mathbf{X}, \mathbf{y})$; es decir, muestreamos iterativamente de $F(\mathbf{X}, \mathbf{y})$ y $F(\mathbf{y} | \mathbf{X})$. Finalmente, para obtener una muestra \mathbf{X} de $F(\mathbf{X})$, generamos (\mathbf{X}, \mathbf{Y}) a través de la muestra de Gibbs y simplemente ignore \mathbf{Y} .

Para simplificar aún más las cosas, tenga en cuenta que en lugar del valor exacto $Y_{y_{ij}}$ basta con conocer solo la variable $B_{y_{ij}} = Y_{y_{ij}} \mathbb{I}_{\{Y_{y_{ij}} > 1\}}$. GRAM

iven $\mathbf{X} = \mathbf{X}$, $B_{y_{ij}}$ tiene un $\text{Ber}(1 - \exp(-\psi_{y_{ij}}))$

distribución si $X_i = X_j$, y $B_{y_{ij}} = 0$ de lo contrario. Esto conduce al siguiente llamado algoritmo de Swendsen-Wang:

Algoritmo 6.5.1: Swendsen-Wang

- 1 Dado $\{X_i\}$, generar $B_{y_{ij}} \sim \text{Ber}(y_{ij} \in \mathcal{X} \mid X_i = X_j) (1 - \exp(-\psi_{y_{ij}}))$ para $1 \leq y_{ij} < j$.
- 2 Dado $\{B_{y_{ij}}\}$, generar $X_i, i = 1, \dots, j$ agrupando todos los sitios y eligiendo cada color de grupo de forma independiente y uniforme de $\{1, \dots, K\}$.

Observación 6.5.1 (Aumento de datos) La idea anterior de introducir una *auxiliar* variable y para hacer muestreo de $F(X)$ más fácil también se conoce como *aumento de datos*. El método de composición descrito en la Sección 2.3.3 puede verse como otro ejemplo de aumento de datos. Para ilustrar, supongamos que queremos muestrear de la mezcla pdf

$$F(X) = \sum_{i=1}^K p_{\text{ags}} F_i(X).$$

Dejar Y sea la variable aleatoria discreta que toma valores en $\{1, \dots, K\}$ correspondiente a las probabilidades $\{p_{\text{ags}}\}$. El método de composición facilita la toma de muestras del pdf conjunto de X y Y : fiprimero, dibujar Y de acuerdo a $\{p_{\text{ags}}\}$ y luego muestra X condicionado a $Y=i$, es decir, muestra de $F_i(X)$. Simplemente ignorando Y , obtenemos una muestra de $F(X)$.

6.6 ESTADÍSTICAS BAYESIANAS

Una de las principales áreas de aplicación del método MCMC es la estadística bayesiana. El pilar del enfoque bayesiano es la regla de Bayes (1.6), que, en términos de pdf, puede escribirse como

$$F(y/X) = \int \frac{F(X/y)F(y)}{F(X/y)F(y)} dy \propto F(X/y)F(y). \quad (6.11)$$

En otras palabras, para dos variables aleatorias cualesquiera X y Y , la distribución condicional de Y dado $X=X$ es proporcional al producto de la fdp condicional de X dado $Y=y$ y el pdf de Y . Tenga en cuenta que en lugar de escribir $F_X, F_Y, F_{X/Y}, F_{Y/X}$ en la fórmula anterior, hemos utilizado el *misma letra* F para el pdf de X , Y y los pdf condicionales. Este estilo particular de notación es típico en el análisis bayesiano y puede ser de gran valor descriptivo, a pesar de su aparente ambigüedad. Usaremos esta notación siempre que trabajemos en un entorno bayesiano.

La importancia de (6.11) queda clara cuando se emplea en el contexto de la estimación de parámetros bayesianos, a veces denominada *aprendizaje bayesiano*. El siguiente ejemplo explica las ideas.

■ EJEMPLO 6.7 Lanzamiento de monedas y aprendizaje bayesiano

Recuerde el experimento aleatorio básico del Ejemplo 1.1 en la página 3, donde lanzamos una moneda sesgada n veces. Supongamos que los resultados son X_1, \dots, X_n , con $X_i = 1$ si el i -th tiro es cara y $X_i = 0$ de lo contrario, $i=1, \dots, n$. Dejar p_{ags} denote la probabilidad de caras. Queremos obtener información sobre p_{ags} de los datos $X = (X_1, \dots, X_n)$, por ejemplo, construya un CI.

La idea crucial es resumir la información sobre p_{ags} a través de una densidad de probabilidad $F(p_{\text{ags}})$. Por ejemplo, si no sabemos nada sobre p_{ags} , nosotros tomamos $F(p_{\text{ags}})$ distribuida uniformemente en $(0,1)$ intervalo, es decir, $F(p_{\text{ags}}) = 1, 0 < p_{\text{ags}} < 1$. En efecto, tratamos p_{ags} como una variable aleatoria. Ahora, obviamente, los datos X afectará nuestro conocimiento de p_{ags} , y la forma de actualizar esta información es usando la fórmula de Bayes:

$$F(p_{\text{ags}}/X) \propto F(X/p_{\text{ags}})F(p_{\text{ags}}).$$

la densidad $F(p_{\text{ags}})$ se llama el *previo* densidad, $F(p_{\text{ags}}/X)$ se llama el *densidad posterior*, y $F(X/p_{\text{ags}})$ se conoce como el *probabilidad*. En nuestro caso, dado p_{ags} , la

$\{X_i\}$ son independientes y $\text{Ber}(pags)$ distribuido, por lo que

$$R(\mathbf{X}/pags) = \prod_{i=1}^{norte} pags^{x_i} (1-pags)^{1-x_i} = pags^s (1-pags)^{n-s},$$

con $s = X_1 + \dots + X_{norte}$ que representa el número total de éxitos. Luego, usando un uniforme previo ($R(pags) = 1$) obtenemos un pdf posterior

$$R(pag/\mathbf{X}) = c.p.s(1-pags)^{n-s},$$

cual es el pdf del $(\text{Beta}(s+1, n-s+1))$ distribución. La constante de normalización es $C = \frac{1}{(norte+1) \text{Beta}(s+1, n-s+1)}$

Un CI bayesiano para $pags$ ahora se forma tomando los cuantiles apropiados de la función de densidad de probabilidad posterior. Como ejemplo, supongamos que $norte=100$ y $s=1$. Luego, un IC del 95% unilateral izquierdo para $pags$ es $[0, 0.0461]$, donde 0.0461 es el cuantil 0.95 del $\text{Beta}(2, 100)$ distribución. Para estimar $pags$, podemos tomar el valor para el cual el pdf es máximo, el llamado *moda* del pdf. En este problema la moda es 0.01, coincidiendo con la media muestral. La figura 6.7 muestra un gráfico de la función de densidad de probabilidad posterior de este problema.

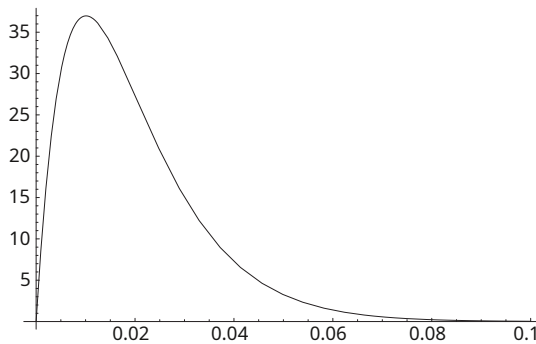


Figura 6.7: PDF posterior para $pags$, con $norte=100$ y $s=1$.

Generalizando el ejemplo anterior, una situación típica en la que se puede utilizar MCMC (en particular, el muestreo de Gibbs) en estadísticas bayesianas es la siguiente: supongamos que queremos muestrear a partir de una densidad posterior $R(\boldsymbol{\theta}/\mathbf{X})$, donde los datos \mathbf{X} son dados (fijos) y $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$ es el parámetro de interés. Suponga que es fácil tomar muestras de $R(\theta_i/\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_k, \mathbf{X})$ para todos i . Entonces, podemos usar el muestreador de Gibbs para obtener una muestra $\boldsymbol{\theta}$ de $R(\boldsymbol{\theta}/\mathbf{X})$. El siguiente ejemplo, adaptado de Gelman et al. [6], ilustra la idea general.

■ EJEMPLO 6.8 Problema de interrupción de Poisson

Supongamos que las variables aleatorias X_1, \dots, X_{norte} describen el número de desastres en $norte$ años subsecuentes. En algún año al azar la tasa de desastres cambia de λ_1 a λ_2 . tal k a menudo se llama un *punto de cambio*. Nuestro conocimiento previo de λ se resume en un $\text{Gama}(a, \eta)$, donde el parámetro de forma a es conocida. Sucesivamente, η está dado por un $\text{Gama}(b, C)$ distribución, donde tanto

by C son conocidos. Dejar $\lambda = (\lambda_1, \lambda_2)$ y $\eta = (\eta_1, \eta_2)$. nos dan los datos $\mathbf{X} = (X_1, \dots, X_{\text{norte}})$, y el objetivo es simular a partir de la distribución posterior de $\theta = (\lambda_1, \lambda_2, \eta_1, \eta_2, k)$ dado \mathbf{X} .

Para el modelo tenemos la siguiente estructura jerárquica:

1. tiene algunos pdf discretos $F(k)$ en $1, \dots, \text{norte}$.
2. dado k , la $\{\eta_i\}$ son independientes y tienen $\text{Gama}(b_i, C_i)$ distribución para $i=1, 2$.
3. dado k y η , la $\{\lambda_i\}$ son independientes y tienen $\text{Gama}(a_i, \eta_i)$ distribución para $i=1, 2$.
4. Dado k, η , y λ , la $\{X_i\}$ son independientes y tienen $\text{Poi}(\lambda_1)$ distribución para $i=1, \dots, k$, y un $\text{Poi}(\lambda_2)$ distribución para $i=k+1, \dots, \text{norte}$.

Del punto 4 se sigue que

$$\begin{aligned} F(\mathbf{X} / \lambda, \eta, k) &= \prod_{i=1}^k \frac{\lambda_1^{X_i}}{X_i!} \prod_{i=k+1}^{\text{norte}} \frac{\lambda_2^{X_i}}{X_i!} \\ &= \lambda_1^{-\sum_{i=1}^k X_i} \lambda_2^{-\sum_{i=k+1}^{\text{norte}} X_i} \prod_{i=1}^k \frac{1}{X_i!} \prod_{i=k+1}^{\text{norte}} \frac{1}{X_i!}. \end{aligned}$$

Además, por la regla del producto (1.4), la función de densidad de probabilidad conjunta viene dada por

$$\begin{aligned} F(\mathbf{X}, \lambda, \eta, k) &\propto F(k) \lambda_1^{-\sum_{i=1}^k X_i} \lambda_2^{-\sum_{i=k+1}^{\text{norte}} X_i} \prod_{i=1}^2 \frac{1}{X_i!} \\ &\propto \lambda_1^{-\sum_{i=1}^k X_i} \lambda_2^{-\sum_{i=k+1}^{\text{norte}} X_i} \prod_{i=1}^2 \frac{1}{X_i!} \\ &\propto \lambda_1^{-\sum_{i=1}^k X_i} \lambda_2^{-\sum_{i=k+1}^{\text{norte}} X_i} \prod_{i=1}^2 \frac{1}{X_i!} \end{aligned}$$

Como consecuencia,

$$F(\lambda_1 / \lambda_2, \eta, k, \mathbf{X}) \propto \lambda_1^{-\sum_{i=1}^k X_i} \lambda_2^{-\sum_{i=k+1}^{\text{norte}} X_i} \prod_{i=1}^2 \frac{1}{X_i!}$$

En otras palabras, $(\lambda_1 / \lambda_2, \eta, k, \mathbf{X}) \sim \text{Gama}(a_1 + \sum_{i=1}^k X_i, k + \eta_1)$. De manera similar manera, tenemos

$$\begin{aligned} (\lambda_2 / \lambda_1, \eta, k, \mathbf{X}) &\sim \text{Gama}(a_2 + \sum_{i=k+1}^{\text{norte}} X_i, n - k + \eta_2), \\ (\eta_1 / \lambda, \eta_2, k, \mathbf{X}) &\sim \text{Gama}(a_1 + b_1, h + C_1), (\eta_2 / \lambda, \eta_1, k, \mathbf{X}) \sim \text{Gama}(a_2 + b_2, h + C_2), \\ F(k / \lambda, \eta, \mathbf{X}) &\propto F(k) \lambda_1^{-\sum_{i=1}^k X_i} \lambda_2^{-\sum_{i=k+1}^{\text{norte}} X_i} \end{aligned}$$

Por lo tanto, el muestreo de Gibbs se puede usar para muestrear desde el pdf posterior $F(\lambda, \eta, k / \mathbf{X})$.

6.7 OTROS MUESTREADORES MARKOV

Existen muchas variantes de los muestreadores Metropolis-Hastings y Gibbs. Sin embargo, todos los algoritmos MCMC conocidos se pueden describir a través del siguiente marco: Considere una cadena de Markov $\{(\mathbf{X}_{\text{norte}}, \mathbf{Y}_{\text{norte}}), \text{norte}=0, 1, 2, \dots\}$ En el set $\mathbf{X} \times \mathbf{Y}$, donde

Sea el objetivo fijado y Y sea un conjunto auxiliar. Dejar $R(X)$ sea el pdf de destino. Cada transición de la cadena de Markov consta de dos partes. El primero es $(X, \tilde{y}) \rightarrow (X, y)$, según una matriz de transición q ; el segundo es $(X, y) \rightarrow (X', y)$, según una matriz de transición R . En efecto, la matriz de transición **PAGS** de la cadena de Markov viene dada por el producto **código QR**. Ambos pasos se ilustran en la Figura 6.8 y se explican a continuación.

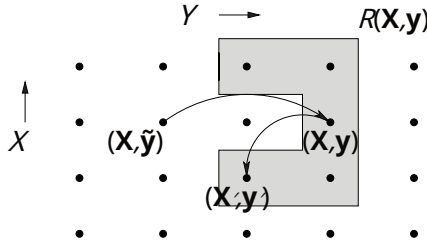


Figura 6.8: Cada transición de la cadena de Markov consta de dos pasos: el q -paso, seguido por el R -paso.

El primer paso, el q -paso, cambia el y -coordina pero deja el X -coordenada intacta. En particular, q es de la forma $q[(X, \tilde{y}), (X, y)] = q_X(\tilde{y}, y)$, donde q_X es una matriz de transición en Y . Dejar q_X sea una distribución estacionaria para q_X , suponiendo que exista.

El segundo paso, el R -paso, está determinada por la distribución estacionaria q_X y la estructura del vecindario en el set $X \times Y$. Específicamente, definimos para cada punto (X, y) un conjunto de vecinos $R(X, y)$ tal que si (X, y) es vecino de (X, y) entonces lo contrario también es cierto; véase la figura 6.8, donde el área sombreada indica el conjunto de vecinos de (X, y) . El paso crucial ahora es definir la matriz de transición. R como

$$R[(X, y), (X, y)] = C(X, y) R(X) q_X(y) \text{ para todos } (X, y) \in R(X, y),$$

dónde $C(X, y) = \sum_{(X, y) \in R(X, y)} R(X) q_X(y)$. Tenga en cuenta que $C(X, y) = C(X, y)$ cuando (X, y) y (X, y) pertenecen al mismo conjunto de vecinos. Con esta elección de q y R se puede demostrar (vea el Problema 6.15) que la cadena de Markov tiene una distribución estacionaria

$$m(X, y) = R(X) q_X(y), \quad (6.12)$$

que es también la distribución límite, siempre que la cadena sea irreducible y aperiódica. En particular, al ignorar la y -coordenada, vemos que la fdp límite de X_{norte} es el objetivo requerido $R(X)$. Esto lleva a lo siguiente *Muestreador de Markov generalizado*[12]:

Algoritmo 6.7.1: Muestreador generalizado de Markov

1 inicializar (X_0, Y_0) .

2 $port = 0$ a $norte - 1$ hacer

3 Dado (X_t, Y_t) , generar Y de $q_X(Y_t, y)$ Dado Y , generar $(X_{t+1}, Y$ // q -paso

4 $_{t+1})$ de $R[(X_t, Y), (X, y)]$ // R -paso

Observación 6.7.1 denotando $R(\mathbf{X}, \mathbf{y}) = R(\mathbf{X}, \mathbf{y}) \mid \{\mathbf{X}, \mathbf{y}\}$, el muestreador se puede generalizar aún más (ver [12]) redefiniendo R como

$$R[(\mathbf{X}, \mathbf{y}), (\mathbf{X}', \mathbf{y}')] = \begin{cases} s(\mathbf{X}, \mathbf{y}) Q(\mathbf{X}, \mathbf{y}) q(\mathbf{X}') & \text{si } (\mathbf{X}', \mathbf{y}') \in R(\mathbf{X}, \mathbf{y}), \\ 1 - \sum_{(\mathbf{z}, \mathbf{k}) \in R(\mathbf{X}, \mathbf{y})} R[(\mathbf{X}, \mathbf{y}), (\mathbf{z}, \mathbf{k})] & \text{si } (\mathbf{X}', \mathbf{y}') = (\mathbf{X}, \mathbf{y}), \end{cases} \quad (6.13)$$

dónde s es una función arbitraria tal que, primero, $s(\mathbf{X}, \mathbf{y}) = s(\mathbf{X}', \mathbf{y}')$ para todos $(\mathbf{X}', \mathbf{y}') \in R(\mathbf{X}, \mathbf{y})$, segundo, las cantidades anteriores son todas probabilidades.

El marco generalizado de muestras de Markov hace posible obtener muchas muestras diferentes de una manera simple y unificada. Damos dos ejemplos: el sampler de rebanadas y el sampler de salto reversible.

6.7.1 Muestreador de rebanadas

Supongamos que deseamos generar muestras a partir del pdf.

$$f(\mathbf{X}) = b \prod_{k=1}^m f_k(\mathbf{X}), \quad (6.14)$$

dónde b es una constante conocida o desconocida y las $\{f_k\}$ son funciones positivas conocidas — no necesariamente densidades. Empleamos el Algoritmo 6.7.1, donde en el q -paso generamos, para un dado $\mathbf{X} = \mathbf{X}$, un vector $\mathbf{Y} = (Y_1, \dots, Y_m)$ dibujando independientemente cada uno Y_k componente Y_k de la distribución uniforme en $[0, f_k(\mathbf{X})]$. De este modo

$q(\mathbf{y}) = 1 / \prod_{k=1}^m f_k(\mathbf{X}) = b / f(\mathbf{X})$. A continuación, dejamos $R(\mathbf{X}, \mathbf{y}) = \{(\mathbf{X}', \mathbf{y}') : f_k(\mathbf{X}') = y_k, k=1, \dots, m\}$. Entonces, (desde $f(\mathbf{X}) q(\mathbf{y}) = b$),

$$R[(\mathbf{X}, \mathbf{y}), (\mathbf{X}', \mathbf{y}')] = \frac{1}{|R(\mathbf{X}, \mathbf{y})|}.$$

En otras palabras, en el R -paso, dado \mathbf{X}, \mathbf{y} , dibujamos \mathbf{X}' uniformemente del conjunto $\{\mathbf{X}' : f_k(\mathbf{X}') = y_k, k=1, \dots, m\}$. Esto da lo siguiente *muestreador de rebanadas*.

Algoritmo 6.7.2: Muestreador de rebanadas

aporte : pdf f de la forma (6.14), punto inicial \mathbf{X}_0 y tamaño de la muestra n . **salida :** $\mathbf{X}_1, \dots, \mathbf{X}_n$ distribuidas aproximadamente según f .

1 **por** $t=0$ **a** $n-1$ **hacer**

2 **por** $k=1$ **a** m **hacer**

3 Dibujar $U_k \sim \text{tu}(0,1)$.

4 $Y_k \leftarrow U_k f_k(\mathbf{X}_t)$

5 Dibujar \mathbf{X}_{t+1} uniformemente del conjunto $\{\mathbf{X} : f_k(\mathbf{X}) = Y_k, k=1, \dots, m\}$.

6 **evolver** $\mathbf{X}_1, \dots, \mathbf{X}_n$

■ EJEMPLO 6.9 Muestreador de rebanadas

Supongamos que ahora necesitamos generar una muestra del pdf de destino

$$f(x) = C \frac{x^{m-1}}{1+x}, \quad x > 0,$$

utilizando el muestreador de rebanadas con $F_1(x) = x/(1+x)$ y $F_2(x) = m - x$.

Supongamos que en la iteración t , $X_t = z$, y u_1, u_2 se generan en las líneas 2-4. En la línea 5, X_{t+1} se extrae uniformemente del conjunto $\{X_t F_1(X_t)/F_1(z) - tu_1, F_2(X_t)/F_2(z) - tu_2\}$, lo que implica los límites $X_{t+1} \geq z - tu_1$ y $X_{t+1} \leq z + tu_2$. Desde $z > 0$ y $0 \leq tu_1, tu_2$, el último límite es mayor que el primero, el intervalo que se dibujará en la línea 5 es $(z - tu_1, z + tu_2)$.

La figura 6.9 muestra un histograma de $n=105$ muestras generadas a través de la muestra de corte, junto con el verdadero pdf $f(x)$. Vemos que los dos están en estrecho acuerdo.

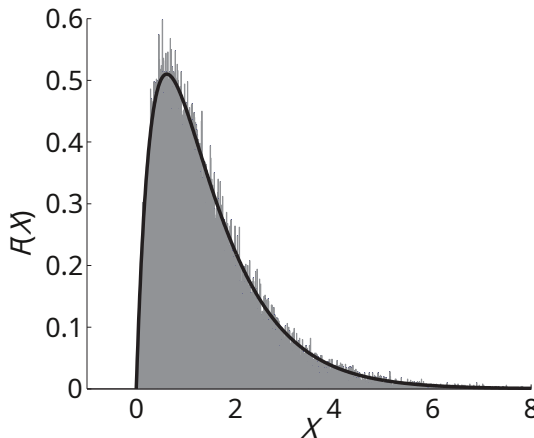


Figura 6.9: Densidad real e histograma de muestras producidas por el muestreador de cortes.

6.7.2 Muestreador de salto reversible

Muestreadores de salto reversibles[9] son útiles para muestrear espacios de destino que contienen vectores de diferentes dimensiones. Esto ocurre a menudo en la inferencia bayesiana cuando se consideran diferentes modelos para los datos.

■ EJEMPLO 6.10 Datos de regresión

Dejar datos y_1, \dots, y_{norte} ser los resultados de variables aleatorias independientes $\{Y_i\}$ de la forma

$$Y_i = \sum_{j=0}^{norte} \beta_j u_{ij} + \varepsilon_i, \quad \varepsilon_i \sim \text{NORTE}(0,1), \quad i=1, \dots, norte, \quad (6.15)$$

dónde t_1, \dots, t_{norte} son variables conocidas, y $METRO \in \{0, \dots, METRO_{\text{máximo}}\}$ y los parámetros $\{\beta_{metro}\}$ son desconocidos. Dejar $\mathbf{y} = (y_1, \dots, y_{norte})$ y $\boldsymbol{\beta} = (\beta_0, \dots, \beta_{METRO})$. Tomando priores uniformes (es decir, constantes) para $\{\beta_{metro}\}$ y $METRO$, tenemos el pdf conjunto

$$R(\mathbf{y}, m, \boldsymbol{\beta}) \propto \exp \left[-\frac{1}{2} \sum_{i=1}^{norte} \left(y_i - \sum_{j=0}^{metro} \beta_j t_{ij} \right)^2 \right]. \quad (6.16)$$

Dejar $\mathbf{X} = (metro, \boldsymbol{\beta})$. Nuestro objetivo es sacar del pdf posterior $R(\mathbf{X}|\mathbf{y}) = R(metro, \boldsymbol{\beta}|\mathbf{y})$. Esto arroja información no solo sobre los parámetros, sino también sobre qué modelo (expresado por $METRO$) es más apropiado. Nótese aquí que la dimensionalidad de \mathbf{X} depende crucialmente de $metro$, por lo que el muestreo estándar de Gibbs o Metropolis-Hastings no es adecuado.

El muestreador de salto reversible salta entre espacios de diferentes dimensiones de acuerdo a un conjunto de saltos permitidos (también llamados *se mueve*). En el ejemplo anterior, podríamos, por ejemplo, permitir solo saltos entre vectores que difieren en dimensión por como máximo 1; eso es, $\beta_0 \rightarrow \beta_0$, $\beta_0 \rightarrow (\beta_0, \beta_1)$, $(\beta_0, \beta_1) \rightarrow \beta_0$, y así.

Para formular el muestreador de salto reversible en el marco del muestreador de Markov generalizado, definimos $\mathbf{X} \times METRO$, donde $METRO$ es el conjunto de movimientos, y escribe un elemento genérico como $(\mathbf{z}, metro)$. En el q -paso que damos $q(\cdot | (\mathbf{z}, metro)) = pags(\mathbf{z} | metro) q_{metro}(\mathbf{X}, \mathbf{z})$. Es decir, un movimiento de tipo $metro$ se selecciona de acuerdo con algún pdf discreto $pags(\mathbf{z} | metro)$. Por ejemplo, la dimensión de \mathbf{X} se reduce, aumenta o se deja sin cambios. Entonces un nuevo \mathbf{z} se selecciona de acuerdo con alguna función de transición $q_{metro}(\mathbf{X}, \mathbf{z})$. Tenga en cuenta que el pdf estacionario para el q -paso en $(\mathbf{z}, metro)$ entonces se convierte $pags(\mathbf{z} | metro) q_{metro}(\mathbf{X}, \mathbf{z})$. Los R -el paso se determina definiendo $R(\mathbf{X}, (\mathbf{z}, metro)) = \{(\mathbf{X}, (\mathbf{z}, metro)), (\mathbf{z}, (\mathbf{X}, metro))\}$, donde $metro$ es el movimiento inverso de $metro$, es decir, $deza \mathbf{X}$. Entonces (6.13) se reduce a

$$R((\mathbf{X}, (\mathbf{z}, metro)), (\mathbf{z}, (\mathbf{X}, metro))) = \frac{S(\mathbf{X}, (\mathbf{z}, metro))}{1 + 1/S}, \quad (6.17)$$

con $\frac{R(\mathbf{z}) pags(metro) q_{metro}(\mathbf{z}, \mathbf{X})}{pags(\mathbf{z} | metro) q_{metro}(\mathbf{X}, \mathbf{z})}$. Tomando $S(\mathbf{X}, (\mathbf{z}, metro)) = \min\{1, 1 + 1/S\}$ reduce el lado derecho de (6.17) más allá de $\min\{1, 1\}$. La transición $(\mathbf{X}, (\mathbf{z}, metro)) \rightarrow (\mathbf{z}, (\mathbf{X}, metro))$ puede interpretarse como aceptación del elemento propuesto \mathbf{z} . En efecto, q se utiliza para proponer un nuevo elemento de acuerdo con el movimiento $metro$ y función de transición q , y R se utiliza para aceptarlo o rechazarlo de acuerdo con el índice de aceptación anterior. Por lo tanto, el muestreador de salto reversible puede verse como una generalización del muestreador Metropolis-Hastings. Esto da el Algoritmo 6.7.3.

Observación 6.7.2 (Coincidencia de dimensiones) Cuando se trata de variables aleatorias continuas, es importante asegurarse de que las densidades de transición estén definidas correctamente. Supongamos que $\dim(\mathbf{X}) = d$ y $\dim(\mathbf{z}) = d' > d$. Una posible forma de generar una transición $\mathbf{X} \rightarrow \mathbf{z}$ es dibujar primero un $(d-d)$ -vector aleatorio \mathbf{tu} según alguna densidad $gramo(\mathbf{tu})$ y luego dejar $\mathbf{z} = \varphi(\mathbf{X}, \mathbf{tu})$ para alguna biyección φ . Esto se conoce como *coincidencia de dimensiones*—la dimensión de $(\mathbf{X}, \mathbf{tu})$ debe coincidir con el de \mathbf{z} . Tenga en cuenta que por (1.20) la densidad de transición está dada por $q(\mathbf{X}, \mathbf{z}) = gramo(\mathbf{tu}) / |j_{\mathbf{X}, \mathbf{tu}}(\varphi)|$, donde $|j_{\mathbf{X}, \mathbf{tu}}(\varphi)|$ es el valor absoluto del determinante de la matriz de Jacobi de φ a $(\mathbf{X}, \mathbf{tu})$.

Algoritmo 6.7.3:Muestreador de salto reversible

aporte :pdf $\{p_{\text{ags}}\}$, funciones de transición $\{q_{\text{metro}}\}$, pdf de destino F , punto inicial \mathbf{X}_0 ,
y tamaño de la muestra n_{orte} .

salida: $\mathbf{X}_1, \dots, \mathbf{X}_{n_{\text{orte}}}$ distribuidas aproximadamente según F .

```

1 port ← 0 a norte - 1 hacer
2   Generar  $\text{metro} \sim p_{\text{ags}}(\text{metro})$ .
3   Generar  $\mathbf{Z} \sim q_{\text{metro}}(\mathbf{X}_t, \mathbf{Z})$ . Dejar  $\text{metro}$  ser el movimiento inverso de  $\mathbf{Z}$  a  $\mathbf{X}_t$ .
4    $\alpha \leftarrow \min \frac{R(\mathbf{Z}) p_{\text{ags}}(\text{metro}) q_{\text{metro}}(\mathbf{Z}, \mathbf{X}_t)}{R(\mathbf{X}_t) p_{\text{ags}}(\text{metro}) q_{\text{metro}}(\mathbf{X}_t, \mathbf{Z})}, 1$  // probabilidad de aceptación
5
6   Generar  $tu \sim \text{tu}(0, 1)$ . si
7      $tu < \alpha$  después
8        $\mathbf{X}_{t+1} \leftarrow \mathbf{Z}$ 
9     más
10     $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t$ 
11 volver  $\mathbf{X}_1, \dots, \mathbf{X}_{n_{\text{orte}}}$ 

```

■ EJEMPLO 6.11 Ejemplo 6.10 (Continuación)

Ilustramos el muestreador de salto reversible usando datos de regresión $\mathbf{y} = (y_1, \dots, y_{n_{\text{orte}}})$ de la forma (6.15), con $tu = (y_0 - 1)/20$, $i = 1, \dots, 101$, $\beta_0 = 1$, $\beta_1 = 0.3$, y $\beta_2 = -0.2$. Los datos se muestran en la Figura 6.10. Aunque es obvio que un modelo constante ($\text{metro} = 0$) no se ajusta a los datos, no está claro si un modelo lineal ($\text{metro} = 1$) o un modelo cuadrático ($\text{metro} = 2$) es más apropiado. Para evaluar los diferentes modelos, podemos ejecutar un muestreador de salto reversible para producir muestras del pdf posterior $R(\mathbf{X} | \mathbf{y})$, que (hasta una constante de normalización) viene dada por el lado derecho de (6.16). Una implementación muy básica es la siguiente:

Algoritmo 6.7.4:Muestreador de salto reversible para el ejemplo 6.10

```

1 Inicializar  $\mathbf{X}_0 = (\text{metro}; \boldsymbol{\beta})$ 
2 port ← 0 a norte - 1 hacer
3   Generar  $\text{metro} \in \{0, 1, 2\}$  con igual probabilidad.
4   Generar  $\boldsymbol{\beta}$  desde un  $(\text{metro} + \text{PDF normal de 1})$  dimensión  $\text{gramo}_{\text{metro}}$  con independiente  
   componentes, con media 0 y varianzas  $\sigma^2$ .
5    $\mathbf{Z} \leftarrow (\text{metro}; \boldsymbol{\beta})$ 
6    $\alpha \leftarrow \min \frac{R(\mathbf{Z} | \mathbf{y}) \text{gramo}_{\text{metro}}(\boldsymbol{\beta}) R(\mathbf{X}_t)}{R(\mathbf{X}_t | \mathbf{y}) \text{gramo}_{\text{metro}}(\boldsymbol{\beta})}, 1$  // probabilidad de aceptación
7   Generar  $tu \sim \text{tu}(0, 1)$ . si
8      $tu < \alpha$  después
9        $\mathbf{X}_{t+1} \leftarrow \mathbf{Z}$ 
10    más
11     $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t$ 
12     $(\text{metro}; \boldsymbol{\beta}) \leftarrow \mathbf{X}_{t+1}$ 

```

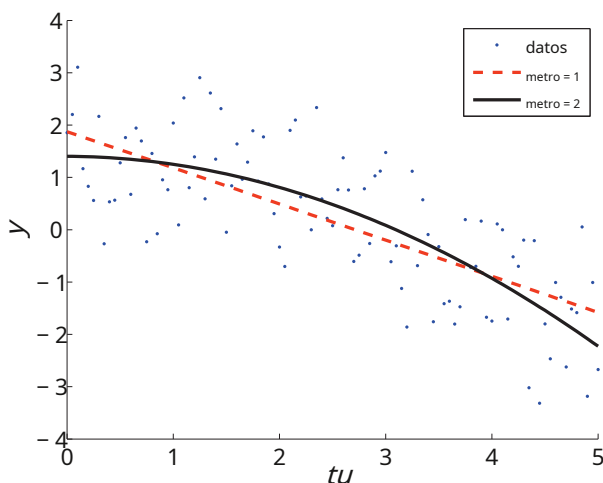


Figura 6.10: Datos de regresión y curvas ajustadas.

El procedimiento anterior, con $norte=10$ y $\sigma=2$, produjo 22,136 vectores bidimensionales β y 77.834 vectores tridimensionales, dando probabilidades posteriores de 0,221 y 0,778 para los modelos 1 y 2, respectivamente. La probabilidad posterior para el modelo constante fue insignificante (0,0003). Esto indica que el modelo cuadrático tiene el mejor ajuste. Los parámetros de regresión β se estiman a través de las medias muestrales de la $\{\beta_i\}$ por $metro=1$ o 2 y se encuentra que son (1.874, -0.691) y (1.404, -0.011, -0.143). Las curvas de regresión correspondientes se representan en la Figura 6.10.

6.8 RECOCIDO SIMULADO

recocido simulado es una técnica de optimización popular basada en MCMC. Esta técnica utiliza muestreo MCMC para encontrar un modo de una densidad $P(\mathbf{X})$ (un punto \mathbf{X} donde $P(\mathbf{X})$ es máxima). Implica definir una familia de densidades de la forma $F_T(\mathbf{X}) \propto [P(\mathbf{X})]^{1/T}$, donde el parámetro T se llama *la temperatura* de la distribución. El muestreo MCMC se usa para dibujar un solo elemento $\mathbf{X}_{(k)}$ de F_T para temperaturas cada vez más bajas T_1, T_2, \dots cada elemento $\mathbf{X}_{(k)}$ se utiliza como elemento inicial de la siguiente cadena. A medida que se reduce la temperatura, las distribuciones alcanzan un pico pronunciado en los máximos globales de F . Por lo tanto, la $\{\mathbf{X}_{(k)}\}$ convergen en un punto. Pueden converger a un máximo local, pero esta posibilidad se reduce mediante una cuidadosa selección de temperaturas sucesivas. La secuencia de temperaturas, o *cronograma de recocido*, por lo tanto, es fundamental para el éxito del método. Una opción común para el programa de recocido es una progresión geométrica, comenzando con una temperatura inicial específica y multiplicando por un *factor de enfriamiento* en el intervalo (0,1) después de cada iteración.

El recocido simulado también se puede aplicar a problemas de optimización no probabilística. Dada una función objetivo $S(\mathbf{X})$, se define una distribución de Boltzmann a través de la densidad $P(\mathbf{X}) \propto e^{-S(\mathbf{X})}$ o $P(\mathbf{X}) \propto e^{S(\mathbf{X})}$, dependiendo de si el objetivo es minimizar o maximizar S . Óptimos globales de S luego se obtienen buscando la moda de la distribución de Boltzmann. Ilustramos el método a través de dos

ejemplos trabajados, uno basado en el sampler de Metropolis-Hastings y el otro en el sampler de Gibbs.

■ EJEMPLO 6.12 Problema del vendedor ambulante

El problema del viajante de comercio (TSP) se puede formular de la siguiente manera: Considere un gráfico ponderado $G = (V, E)$ con n nodos, etiquetados $1, 2, \dots, n$. Los nodos representan ciudades y los bordes representan los caminos entre las ciudades. Cada borde $e \in E$ tiene peso o costo c_e , que representa la longitud de la carretera. El problema es encontrar el más corto *recorrido* que visita todas las ciudades exactamente una vez excepto la ciudad inicial, que también es la ciudad final. En la Figura 6.11 se da un ejemplo, donde las líneas en negrita forman un posible recorrido.

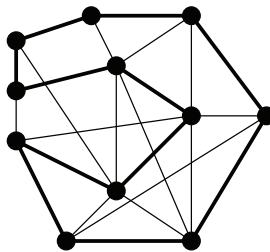


Figura 6.11: Encuentra el recorrido más corto X visitando todos los nodos.

Sin pérdida de generalidad, podemos suponer que la gráfica es *completa* (completamente conectado) porque, si no está completo, siempre podemos agregar algunos costos (distancias) iguales a $+\infty$. Dejar X sea el conjunto de todos los recorridos posibles, y sea $S(X)$ la duración total del recorrido $X \in X$. Podemos representar cada recorrido a través de una *permutación* de $(1, \dots, n)$. por ejemplo, para $n=4$, la permutación $(1, 3, 2, 4)$ representa el recorrido $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1$. Por tanto, identificaremos un recorrido con su permutación correspondiente. Por lo tanto, el objetivo es minimizar

$$\min_{X \in X} S(X) = \min_{X \in X} \left\{ \sum_{i=1}^{n-1} C_{X_i, X_{i+1}} + C_{X_n, 1} \right\}. \quad (6.18)$$

Tenga en cuenta que el número de elementos en X suele ser muy grande, ya que $|X| = n!$.

El TSP se puede resolver mediante recocido simulado. Primero, definimos el pdf de destino para que sea el pdf de Boltzmann $P(X) = C e^{-S(X)/T}$. En segundo lugar, definimos una estructura de vecindad en el espacio de permutaciones X , llamémosla *opciones*. Aquí los vecinos de una permutación arbitraria X se encuentran (1) seleccionando dos índices diferentes de $\{1, \dots, n\}$ y (2) invertir el camino de X entre esos dos índices. Por ejemplo, si $X = (1, 2, \dots, 10)$ y se seleccionan los índices 4 y 7, luego $y = (1, 2, 3, 7, 6, 5, 4, 8, 9, 10)$; consulte la figura 6.12. Otro ejemplo es: si $X = (6, 7, 2, 8, 3, 9, 10, 5, 4, 1)$ y se seleccionan los índices 6 y 10, luego $y = (6, 7, 2, 8, 3, 1, 4, 5, 10, 9)$.

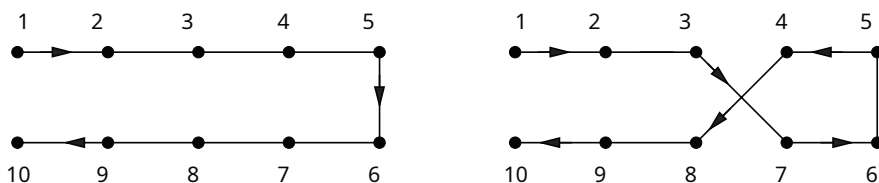


Figura 6.12: Ilustración de la estructura de vecindad de 2 opciones.

En tercer lugar, aplicamos el algoritmo Metropolis-Hastings para tomar muestras del objetivo. Necesitamos proporcionar una función de transición. $q(\mathbf{X}, \mathbf{y})$ de \mathbf{X} a uno de sus vecinos. Por lo general, los dos índices para la vecindad de 2 opciones se seleccionan de manera uniforme. Esto se puede hacer, por ejemplo, dibujando una permutación uniforme de $(1, \dots, \text{norte})$ (ver Sección 2.10) y luego seleccionando los dos primeros elementos de esta permutación. La función de transición es aquí constante: $q(\mathbf{X}, \mathbf{y}) = q(\mathbf{y}, \mathbf{X}) = 1/\text{norte}$. De ello se deduce que en este caso la probabilidad de aceptación es

$$\alpha = \min \left\{ \frac{f(\mathbf{y})}{f(\mathbf{X})}, 1 \right\} = \begin{cases} 1 & \text{si } S(\mathbf{y}) - S(\mathbf{X}) \leq 0 \\ \exp(-(S(\mathbf{y}) - S(\mathbf{X}))/T) & \text{si } S(\mathbf{y}) - S(\mathbf{X}) > 0 \end{cases} \quad (6.19)$$

A medida que vamos bajando la temperatura T , la distribución de Boltzmann se vuelve cada vez más concentrada alrededor del minimizador global. La práctica común es disminuir la temperatura a medida que $T_{t+1} = \beta T_t$ para algunos $\beta < 1$ cerca de 1, como $\beta = 0.99$. Esto conduce al siguiente algoritmo genérico de recocido simulado con muestreo Metropolis-Hastings:

Algoritmo 6.8.1: Recocido simulado: muestreo de Metropolis-Hastings

aporte : Función objetiva S , estado inicial \mathbf{X}_0 , temperatura inicial T_0 , número de iteraciones norte , función propuesta simétrica $q(\mathbf{X}, \mathbf{y})$, constante β .

producción: Valor mínimo aproximado de S y el minimizador correspondiente.

1 port $\leftarrow 0$ **a norte** - 1 **hacer**

2 Generar un nuevo estado \mathbf{Y} de la propuesta simétrica $q(\mathbf{X}_t, \mathbf{y})$. **si** $S(\mathbf{Y}) < S(\mathbf{X}_t)$ **después**

3 | $\mathbf{X}_{t+1} \leftarrow \mathbf{Y}$

4 **más**

5 | Dibujar $tu \sim \text{tu}(0, 1)$.

6 | **si** $tu < \exp(-(S(\mathbf{Y}) - S(\mathbf{X}_t))/T_t)$ **después**

7 | | $\mathbf{X}_{t+1} \leftarrow \mathbf{Y}$

8 | **más**

9 | | $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t$

10 | $T_{t+1} \leftarrow \beta T_t$

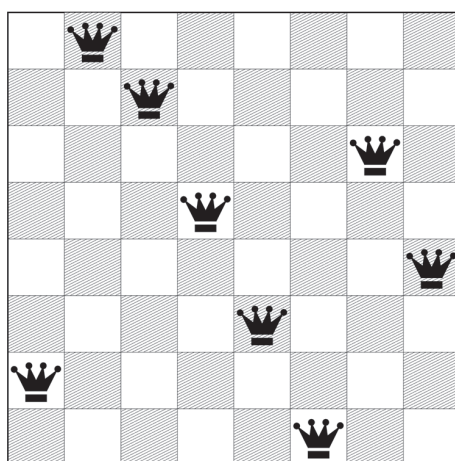
12 **devolver** $S(\mathbf{X}_{\text{norte}})$ y $\mathbf{X}_{\text{norte}}$

En lugar de detenerse después de un número fijo norte de iteraciones, es útil detenerse cuando los valores de función consecutivos están más cerca que cierta distancia ϵ entre sí, o cuando el valor de la función mejor encontrado no ha cambiado en un número fijo d de iteraciones.

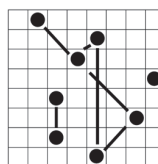
EJEMPLO 6.13 *norte*-Problema de reinas

En el *norte*-problema de reinas, el objetivo es arreglar *norte* reinas en un *norte* × *norte* tablero de ajedrez de tal manera que ninguna reina pueda capturar a otra reina. En la Figura 6.13 se da una ilustración para el caso *norte* = 8. Tenga en cuenta que la configuración de la Figura 6.13 no resuelve el problema. Nosotros tomamos *norte* = 8 de ahora en adelante. Tenga en cuenta que cada fila del tablero de ajedrez debe contener exactamente una reina. Indicar la posición de la reina en la *i*-ésima fila por X_i ; de esta manera cada configuración puede ser representada por un vector $\mathbf{X} = (X_1, \dots, X_8)$. Por ejemplo, $\mathbf{X} = (2, 3, 7, 4, 8, 5, 1, 6)$ corresponde a la configuración grande de la Fig. dado en la misma figura. Ahora podemos formular una función $S(\mathbf{X})$ que representa la a

Para esto simplemente sumamos el número de diagonales de reina que tienen al menos dos reinas pres en la Figura 6.13 solo hay dos diagonales $(2-1) + (2-1) = 2$. Tenga en cuenta que las soluciones mínimas son $\mathbf{X} = (5, 1, 8, 6, 3, 7, 2, 4)$.

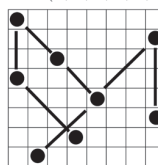


$$\mathbf{x} = (2, 5, 4, 8, 3, 7, 3, 5)$$



$$S(\mathbf{x}) = 5$$

$$\mathbf{x} = (1, 8, 3, 1, 5, 8, 4, 2)$$



$$S(\mathbf{x}) = 7$$

Figura 6.13: Coloque las ocho reinas de manera que ninguna reina pueda capturar a otra.

A continuación, mostramos cómo se puede resolver este problema de optimización mediante recocido simulado utilizando el muestreador Gibbs. Como en el ejemplo anterior de TSP, cada iteración del algoritmo consiste en tomar muestras del pdf de Boltzmann $P(\mathbf{X}) = \frac{e^{-S(\mathbf{X})}}{Z}$ a través del muestreador Gibbs, seguido de una disminución de la temperatura. Esto conduce al siguiente algoritmo genérico de recocido simulado que utiliza el muestreo de Gibbs:

Algoritmo 6.8.2: Recocido simulado: Muestreo de Gibbs

```

1 Inicializar  $X_0$ .
2  $t \leftarrow 0$ 
3 tiempo  $S(X_t) > 0$  hacer
4   Dibujar  $Y_1$  del pdf condicional  $P_{Y_1} / X_{Y_0,2}, \dots, X_{Tennessee}$ . por  $i$ 
5   = 2 a norte hacer
6     Dibujar  $Y_i$  del pdf condicional  $P_{Y_i} / Y_1, \dots, Y_{i-1}, X_{Y_{i+1}}, \dots, X_{Tennessee}$ .
7    $X_{t+1} \leftarrow Y$ 
8    $T_{t+1} \leftarrow \beta T_t$ 
9    $t \leftarrow t+1$ 
10 volver  $X_t$ 

```

Tenga en cuenta que en la línea 6 cada Y_i se extrae de una distribución discreta en $\{1, \dots, n\}$ con probabilidades proporcionales a $e^{-S(Z_1)/T_t}, \dots, e^{-S(Z_{norte})/T_t}$, donde cada uno Z_k es igual al vector $(Y_1, \dots, Y_{i-1}, k, X_{Y_{i+1}}, \dots, X_{Tennessee})$.

Se pueden usar otros muestreadores MCMC en el recocido simulado. por ejemplo, en el *escondite* se utiliza el algoritmo [21], el muestreador general hit-and-run (Sección 6.3). La investigación motivada por el uso de hit-and-run y hit-and-run discreto en el recocido simulado ha dado como resultado el desarrollo de un programa de enfriamiento derivado teóricamente que utiliza los valores registrados obtenidos durante el curso del algoritmo para actualizar la temperatura de forma adaptativa. [23, 24].

6.9 MUESTREO PERFECTO

Volviendo al principio de este capítulo, supongamos que deseamos generar una variable aleatoria X tomando valores en $\{1, \dots, m\}$ según una distribución objetivo $\pi = \{\pi_i\}$. Como se mencionó, uno de los principales inconvenientes del método MCMC es que cada muestra X_i es solo *asintóticamente* distribuido de acuerdo a π , es decir, $\lim_{t \rightarrow \infty} P(S_t = i) = \pi_i$. A diferencia de, *muestreo perfecto* es una técnica MCMC que produce muestras exactas de π .

Dejar $\{X_t\}$ ser una cadena de Markov con espacio de estado $\{1, \dots, m\}$, matriz de transición P y distribución estacionaria π . Deseamos generar $\{X_t, t=0, -1, -2, \dots\}$ de una manera que X_0 tiene la distribución deseada. podemos dibujar X_0 desde el *metro*-distribución de puntos correspondiente a la X_{-1} -ésima fila de P ; ver Algoritmo 2.7.1. Esto se puede hacer a través del método de TI, que requiere la generación de una variable aleatoria $tu \sim tu(0,1)$. Similarmente, X_{-1} se puede generar a partir de $X_{-2} \sim tu_{-1} \sim tu(0,1)$. En general, vemos que para cualquier tiempo negativo $-t$, la variable aleatoria X_0 depende de X_{-t} y las variables aleatorias independientes $tu_{-t+1}, \dots, tu_0 \sim tu(0,1)$.

A continuación, consideremos *metro* copias dependientes de la cadena de Markov, a partir de cada uno de los estados $1, \dots, m$ y usando el mismo números al azar $\{tu_i\}$ —como en el método CRV. Entonces, si dos caminos coinciden, *ojuntarse*, en algún momento, a partir de ese momento, ambos caminos serán idénticos. Se dice que los caminos son *acoplado*. El punto principal del método de muestreo perfecto es que si la cadena es ergódica (en particular, si es aperiódica e irreducible), entonces *con probabilidad 1 existe un tiempo negativo $-T$ tal que todas las rutas se habrán fusionado antes o en el tiempo 0*. La situación se ilustra en la figura 6.14.

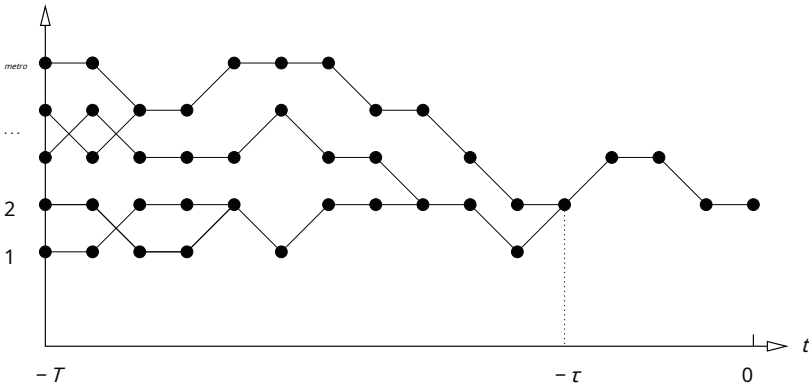


Figura 6.14: Todas las cadenas de Markov se han fusionado en el tiempo $-\tau$.

Dejamos \mathbf{tu} representar el vector de todos tu_t , $t=0$. Para cada \mathbf{tu} sabemos que existe, con probabilidad 1, un $-\tau(\mathbf{tu}) < 0$ tal que por tiempo 0 todos \mathbf{metro} cadenas acopladas definidas por \mathbf{tu} se han fusionado. Además, si empezamos en el tiempo $-T$, a *estacionario* versión de la cadena de Markov, usando de nuevo el mismo \mathbf{tu} , esta cadena estacionaria debe, en el momento $t=0$, se han fusionado con los demás. Así, cualquiera de los \mathbf{metro} cadenas tiene en el tiempo 0 la misma distribución que la cadena estacionaria, que es π .

Tenga en cuenta que para construir T , no necesitamos saber el todo (vector infinito) \mathbf{tu} . En cambio, podemos trabajar hacia atrás desde $t=0$ generando tu_{-1} primero, y comprobando si $-T=-1$. Si no es así, genera tu_{-2} y comprobar si $-T=-2$, y así sucesivamente. Esto conduce al siguiente algoritmo, debido a Propp y Wilson [19], llamado *acoplamiento del pasado*:

Algoritmo 6.9.1: Acoplamiento del pasado

```

1 Generar  $tu_0 \sim \mathbf{tu}(0,1)$ .
2  $tu_0 \leftarrow tu_0$ 
3  $t \leftarrow -1$ 
4 no fusionado  $\leftarrow$  verdadero
5 tiempo no fusionado hacer
6   Generar  $\mathbf{metro}$  Cadenas de Markov  $\{X_i, i = -t, \dots, 0\}$ , a partir de  $t$  de cada uno de
      los estados  $1, \dots, \mathbf{metro}$ , y usando el mismo vector aleatorio  $\mathbf{tu}_{t+1}$ . Si todas las
7   cadenas se han fusionado antes o en el tiempo 0 después
8   | no fusionado  $\leftarrow$  falso
9   | más
10  | Generar  $tu_t \sim \mathbf{tu}(0,1)$ .  $tu_t$ 
11  |  $\leftarrow (tu_t, \mathbf{tu}_{t+1})$   $t \leftarrow t - 1$ 
12  |
13 devolver  $X_0$ 
```

Aunque el muestreo perfecto parece realmente "perfecto" en el sentido de que devuelve una muestra exacta del objetivo π en lugar de una aproximación, las aplicaciones prácticas de la técnica son, en la actualidad, bastante limitadas. La técnica no solo es difícil o imposible de usar para la mayoría de los sistemas de simulación continua, sino que también es mucho más intensiva desde el punto de vista computacional que el simple MCMC.

PROBLEMAS

6.1 Verifique que la ecuación de equilibrio local (6.3) sea válida para el algoritmo Metropolis-Hastings.

6.2 Al ejecutar un algoritmo MCMC, es importante saber cuándo el transitorio (*oquemado*) el período ha terminado; de lo contrario, los análisis estadísticos de estado estacionario como los de la Sección 4.4.2 pueden no ser aplicables. En la práctica, esto se hace a menudo mediante una inspección visual de la ruta de la muestra. Ejecute el muestreador de recorrido aleatorio con distribución objetivo normal NORTE(10,1) y propuesta $Y \sim \text{NORTE}(X, 0.01)$. Tome un tamaño de muestra $n_{\text{NORTE}} = 5000$. Determine aproximadamente cuando el proceso alcanza la estacionariedad.

6.3 Una herramienta útil para examinar el comportamiento de un proceso estacionario $\{X_t\}$ obtenido, por ejemplo, de una simulación MCMC, es la función de covarianza $R(t) = \text{Cov}(X_t, X_0)$; ver Ejemplo 6.4. Estime la función de covarianza para el proceso del problema 6.2 y grafique los resultados. En Matlab *procesamiento de la señal* caja de herramientas, esto se implementa bajo la función `Mxcov.m`. Pruebe diferentes distribuciones de propuestas del formulario $\text{NORTE}(x, \sigma)$ y observe cómo cambia la función de covarianza.

6.4 Implementar el muestreador de independencia con un $\text{Exp}(1)$ objetivo y un $\text{Exp}(\lambda)$ propuesta de distribución para varios valores de λ . Similar a la situación de muestreo de importancia, las cosas salen mal cuando la distribución de muestreo se aleja demasiado de la distribución objetivo, en este caso cuando $\lambda > 2$. Para cada corrida, use un tamaño de muestra de 105 y empezar con $X = 1$.

- a) Para cada valor $\lambda = 0.2, 1, 2$ y 5, grafique un histograma de los datos y compárelo con el pdf verdadero.
- b) Para cada valor de λ dada arriba, calcule la media de la muestra y repita esto para 20 corridas independientes. Haz un diagrama de puntos de los datos (trázalos en una línea) y observa las diferencias. Obsérvese que por $\lambda = 5$ la mayoría de las medias de la muestra están por debajo de 1 y, por lo tanto, subestiman la expectativa real de 1, pero algunas son significativamente mayores. Obsérvese también el comportamiento de las correspondientes funciones de autocovarianza, tanto entre los diferentes λ s y, para $\lambda = 5$, dentro de las 20 corridas.

6.5 Implemente el muestreador de paseo aleatorio con un $\text{Exp}(1)$ distribución objetivo, donde Z (en la propuesta $Y = X + Z$) tiene una distribución exponencial doble con parámetro λ . Realice un estudio similar al del problema 6.4 para diferentes valores de λ , decir $\lambda = 0.1, 1, 5, 20$. Observa que (en este caso) el muestreador de paseo aleatorio tiene un comportamiento más estable que el muestreador independiente.

6.6 Dejar $X = (X, Y)$ sea un vector de columna aleatorio con una distribución normal bivariada con un vector de expectativa $\mathbf{0} = (0, 0)$ y matriz de covarianza

$$\Sigma = \begin{pmatrix} 1 & \\ & 1 \end{pmatrix}.$$

- a) Muestra esa $(Y | X = x) \sim \text{NORTE}(x, 1 - x)$ y $(X | Y = y) \sim \text{NORTE}(y, 1 - y)$.
- b) Escriba una muestra sistemática de Gibbs para dibujar 104 muestras de la distribución bivariada $\text{NORTE}(\mathbf{0}, \Sigma)$ y trazar los datos para $\rho = 0.7$ y 0.9 .

6.7 Una característica notable del muestreador de Gibbs es que las distribuciones condicionales del algoritmo 6.4.1 contienen información suficiente para generar una muestra a partir de la distribución conjunta. El siguiente resultado (por Hammersley y Clifford [10]) muestra

que es posible expresar directamente el pdf conjunto en términos de los pdf condicionales. A saber,

$$F(x, y) = \frac{F_{Y|X}(y/x)}{\int_{F_{Y|X}(y/x)} F_{X|Y}(x/y) dy}.$$

Demuestra esto. Generalizar esto al *norte*-caso dimensional.

6.8 En el modelo de Ising la *magnetización esperada por espines* dado por

$$METRO(T) = \frac{1}{norte} \lim_{norte \rightarrow \infty} \sum_i S_i,$$

dónde π es la distribución de Boltzmann a temperatura T . Estimar $METRO(T)$, por ejemplo a través del algoritmo de Swendsen-Wang, para varios valores de $T \in [0, 5]$, y observe que la gráfica de $METRO(T)$ cambia bruscamente alrededor de la temperatura crítica $T \approx 2.61$. Toma $norte = 20$ y utilice límites periódicos.

6.9 Ejecute el subprograma Java de Peter Young en <http://física.ucsc.edu/~peter/java/ising/ising.html>

para obtener una mejor comprensión de cómo funciona el modelo Ising.

6.10 Como en el ejemplo 6.6, sea $X^* = \{X\}$: $\sum_{i=1}^{norte} X_i = m$, $X_i \in \{0, \dots, m\}$, $y \sigma = 1, \dots, n\}$. Demuestra que este conjunto tiene $m \cdot norte + n - 1$ elementos.

6.11 En un modelo simple para una red de colas cerradas con *norte* colas y *metro* clientes, se supone que los tiempos de servicio son independientes y están distribuidos exponencialmente, con tasa m para cola i , $i = 1, \dots, norte$. Después de completar el servicio en la cola i , el cliente pasa a la cola j con probabilidad p_{agsyo} . los $\{p_{agsyo}\}$ son los llamados *probabilidades de enrutamiento*.

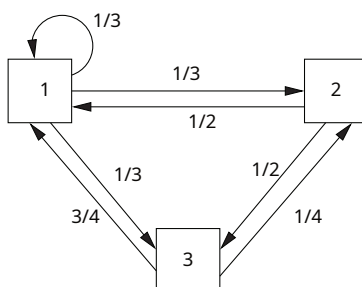


Figura 6.15: Una red de colas cerrada.

Se puede demostrar (por ejemplo, ver [13]) que la distribución estacionaria del número de clientes en las colas es de forma producto (6.7), con F siendo el pdf del GRAMO $(1 - y_i/\mu_i)$ distribución; de este modo $F(X) \propto (y_i/\mu_i)^x$. Aquí el $\{y_i\}$ son constantes que se obtienen del siguiente conjunto de *Flordabajo saldo* ecuaciones:

$$y_i = \sum_j y_j p_{agsji}, i = 1, \dots, norte, \quad (6.20)$$

que tiene un espacio de solución unidimensional. Sin pérdida de generalidad, y_i se puede establecer en 1 para obtener una solución única.

Considere ahora el caso específico de la red representada en la Figura 6.15, con $norte=3$ colas. Suponga que las tarifas de servicio son $m_1=2, m_2=1, y m_3=1$. Las probabilidades de enrutamiento se dan en la figura.

- Muestre que una solución a (6.20) es $(\mu_1, \mu_2, \mu_3) = (1, 10/21, 4/7)$.
- Para $metro=50$ determinar la constante de normalización exacta C .
- Implemente el procedimiento del ejemplo 6.6 para estimar C a través de MCMC, y compare la estimación $metro=50$ con el valor exacto.

6.12 Dejar X_1, \dots, X_{norte} ser una muestra aleatoria de la $NORTE(\mu, \sigma^2)$ distribución. Considere el siguiente modelo bayesiano:

- $R(\mu, \sigma^2) = 1/\sigma^2$;
- $(X_i/\mu, \sigma^2) \sim \text{NORTE}(\mu, \sigma^2), i=1, \dots, norte$ independientemente.

Tenga en cuenta que el anterior para (μ, σ^2) es incorrecto. Es decir, no es un pdf en sí mismo, pero si aplicamos obstinadamente la fórmula de Bayes, sí produce un pdf posterior adecuado. En cierto sentido, transmite la menor cantidad de información sobre μ y σ^2 . Dejar $\mathbf{X} = (X_1, \dots, X_{norte})$ representar los datos. La función de densidad de probabilidad posterior está dada por

$$R(\mu, \sigma^2/\mathbf{X}) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^{norte} (X_i - m)^2\right\} \frac{1}{\sigma^2}.$$

Deseamos muestrear de esta distribución a través del muestreador de Gibbs.

- Muestra esa $(\mu/\sigma^2, \mathbf{X}) \sim \text{NORTE}(\bar{x}, \sigma^2/norte)$, donde \bar{x} es la media de la muestra.
- Prueballo

$$R(\sigma^2/\mu, \mathbf{X}) \propto \frac{1}{(\sigma^2)^{norte/2+1}} \exp\left\{-\frac{nV_m}{2\sigma^2}\right\}, \quad (6.21)$$

dónde $V_m = \sum_{i=1}^{norte} (X_i - m)^2/norte$ es la varianza muestral clásica para valores conocidos m . En otras palabras, $(1/\sigma^2/\mu, \mathbf{X}) \sim \text{Gama}(norte/2, nV_m/2)$.

- Implementar un muestreador Gibbs para muestrear de la distribución posterior, tomando $norte=100$. Ejecute el muestreador durante 10 iteraciones. Trazar los histogramas de $R(\mu/\mathbf{X})$ y $R(\sigma^2/\mathbf{X})$, y encuentre las medias muestrales de estos posteriores. Compáralas con las estimaciones clásicas.

d) Demuestre que la función de densidad de probabilidad posterior verdadera dem , dados los datos, es

$$R(\mu/\mathbf{X}) \propto \frac{1}{\mu^{norte/2+1}} \exp\left\{-\frac{1}{\mu} \sum_{i=1}^{norte} X_i\right\}$$

dónde $V = \sum_{i=1}^{norte} (X_i - \bar{x})^2/norte$. [Sugerencia: para evaluar la integral

$$R(\mu/\mathbf{X}) = \int_0^\infty R(\mu, \sigma^2/\mathbf{X}) d\sigma^2$$

escribelo primero como $(2\pi)^{-norte/2} \int_0^\infty t^{norte/2-1} \exp(-1/2t.C) dt$, donde $C = nV_m$, por aplicando el cambio de variable $t=1/\sigma^2$. Demuestre que la última integral es proporcional a $C^{-norte/2}$. Finalmente, aplique la descomposición $V_m = (\bar{x}-\mu)^2 + V$]

6.13 Suponer que $R(\theta/\mathbf{X})$ es la función de densidad de probabilidad posterior de algún problema de estimación bayesiano. Por ejemplo, θ podría representar los parámetros de un modelo de regresión basado en los datos \mathbf{X} . Un uso importante para el pdf posterior es hacer predicciones

sobre la distribución de otras variables aleatorias. Por ejemplo, suponga que el pdf de alguna variable aleatoria Y depende de θ a través del pdf condicional $R(y|\theta)$. los pdf predictivo de Y dado X se define como

$$R(y|X) = \int R(y|\theta) R(\theta|X) d\theta,$$

que puede ser visto como la expectativa de $R(y|\theta)$ bajo el pdf posterior. Por lo tanto, podemos usar la simulación Monte Carlo para aproximar $R(y|X)$ como

$$R(y|X) \approx \frac{1}{n_{\text{norte}}} \sum_{i=1}^{n_{\text{norte}}} R(y|\theta_i)$$

donde la muestra $\{\theta_i, i=1, \dots, n\}$ se obtiene de $R(\theta|X)$ (por ejemplo, a través de MCMC).

Como aplicación concreta, suponga que los datos de medición independientes $-0.4326, -1.6656, 0.1253, 0.2877, -1.1465$ vienen de algunos NORTE (μ, σ^2) distribución. Definir $\theta = (\mu, \sigma^2)$. Dejar $Y \sim \text{NORTE}(\mu, \sigma^2)$ ser una nueva medida. Estime y dibuje el pdf predictivo $R(y|X)$ de una muestra $\theta_1, \dots, \theta_{n_{\text{norte}}}$ obtenido mediante el muestreador de Gibbs del problema 6.12. Tomar $n_{\text{norte}}=10.000$. Compare esto con el pdf gaussiano de "sentido común" con expectativa \bar{X} (media muestral) y varianzas s^2 (varianza muestral).

6.14 En el *Poisson inflado a cero* (ZIP) modelo, datos aleatorios $X_1, \dots, X_{n_{\text{norte}}}$ se supone que son de la forma $X = R_i Y_i$, donde el $\{Y_i\}$ tener un $\text{Poi}(\lambda)$ distribución y $\{R_i\}$ tener un $\text{Ber}(p_{\text{ags}})$ distribución, todos independientes entre sí. Dado un resultado $X = (X_1, \dots, X_{n_{\text{norte}}})$, el objetivo es estimar tanto λ y p_{ags} . Considere el siguiente modelo jerárquico de Bayes:

- $p_{\text{ags}} \sim \text{tu}(0, 1)$ (antes de p_{ags}),
- $(\lambda/p_{\text{ags}}) \sim \text{Gama}(a, b)$ (antes de λ),
- $(r_i/p_{\text{ags}}, i) \sim \text{Ber}(p_{\text{ags}})$ independientemente (del modelo anterior),
- $(X_i/r_i, \lambda, p_{\text{ags}}) \sim \text{Poi}(\lambda r_i)$ independientemente (del modelo anterior),

dónde $r = (r_1, \dots, r_{n_{\text{norte}}})$ y a, b son parámetros conocidos. Resulta que

$$R(X, r, \lambda, p_{\text{ags}}) = \frac{b^a \lambda^{a-1} \Gamma(a)}{\Gamma(a)} \prod_{i=1}^{n_{\text{norte}}} \frac{\text{mi}_{-i}(\lambda r_i)}{\lambda^{r_i}} \frac{p_{\text{ags}}^{r_i} (1-p_{\text{ags}})^{1-r_i}}{X_i!}.$$

Deseamos tomar una muestra del pdf posterior $R(\lambda, p_{\text{ags}}|X)$ utilizando el muestreador de Gibbs.

a) Muestra esa

1. $(\lambda/p_{\text{ags}}, r, X) \sim \text{Gama}(a + \sum_{i=1}^{n_{\text{norte}}} X_i, b + \sum_{i=1}^{n_{\text{norte}}} r_i)$.
2. $(p_{\text{ags}}|\lambda, r, X) \sim \text{Beta}(1 + \sum_{i=1}^{n_{\text{norte}}} r_i, n_{\text{norte}} + 1 - \sum_{i=1}^{n_{\text{norte}}} r_i)$.
3. $(r_i|\lambda, p_{\text{ags}}, X) \sim \text{Ber}(\frac{p_{\text{ags}} \lambda r_i}{p_{\text{ags}} \lambda r_i + (1-p_{\text{ags}}) X_i})$.

b) Genere una muestra aleatoria de tamaño $n_{\text{norte}}=100$ para el modelo ZIP usando parámetros $p_{\text{ags}}=0.3$ y $\lambda=2$

c) Implemente el muestreador de Gibbs, genere una muestra grande (dependiente) de la distribución posterior y utilícela para construir IC bayesianos del 95 % para p_{ags} y λ usando los datos en b). Compare sus resultados con los valores reales.

6.15 Muestra esa m en (6.12) satisface las ecuaciones de equilibrio locales

$$m(\mathbf{X}, \mathbf{y}) R[(\mathbf{X}, \mathbf{y}), (\mathbf{X}', \mathbf{y}')] = m(\mathbf{X}', \mathbf{y}') R[(\mathbf{X}, \mathbf{y}), (\mathbf{X}', \mathbf{y}')].$$

De este modo m es estacionario con respecto a R , eso es, $mR = m$. Muestra esa m también es estacionario con respecto a q . Muestre también que m es estacionario con respecto a $P_{\text{AGS}} = \text{código QR}$.

6.16 Para mostrar que el muestreador sistemático de Gibbs es un caso especial del muestreador generalizado de Markov, tome Y ser el conjunto de índices $\{1, \dots, n\}$, y definir para el q -paso

$$q_{\mathbf{X}}(s, s') = \begin{cases} 1 & \text{si } y = y+1 \text{ o } y = 1, y = \text{norte, de lo} \\ 0 & \text{contrario.} \end{cases}$$

Sea el conjunto de posibles transiciones $R(\mathbf{X}, y)$ sea el conjunto de vectores $\{(\mathbf{X}', y)\}$ tal que todas las coordenadas de \mathbf{X}' son los mismos que los de \mathbf{X} a excepción de posiblemente el y -ésima coordenada.

a) Demuestre que la distribución estacionaria de $q_{\mathbf{X}}$ es $q_{\mathbf{X}}(y) = 1/n$ norte, por $y = 1, \dots, n$ norte.

b) Muestra esa

$$R[(\mathbf{X}, y), (\mathbf{X}', y')] = \frac{R(\mathbf{X})}{\sum_{(z, y) \in R(\mathbf{X}, y)} R(z)}, \quad \text{por } (\mathbf{X}, y) \in R(\mathbf{X}, y).$$

c) Comparar con el Algoritmo 6.4.1.

6.17 Demuestre que el algoritmo Metropolis-Hastings es un caso especial del muestreador de Markov generalizado. [Sugerencia: deje que el conjunto auxiliar Y ser una copia del conjunto de destino X , dejar $q_{\mathbf{X}}$ corresponden a la función de transición del algoritmo Metropolis-Hastings (es decir, $q_{\mathbf{X}}(\cdot; \mathbf{y}) = q(\mathbf{X}, \mathbf{y})$), y definir $R(\mathbf{X}, \mathbf{y}) = \{(\mathbf{X}, \mathbf{y}), (\mathbf{y}, \mathbf{X})\}$. Use argumentos similares a los del muestreador de salto de Markov (ver (6.17)) para completar la demostración.]

6.18 Los algoritmos MCMC de Barker y Hastings difieren del muestreador simétrico Metropolis solo en que definen la tasa de aceptación $\alpha(\mathbf{X}, \mathbf{y})$ ser, respectivamente, $R(\mathbf{y}) / (R(\mathbf{X}) + R(\mathbf{y}))$ y $s(\mathbf{X}, \mathbf{y}) / (1 + s(\mathbf{X}, \mathbf{y}))$ en lugar de $\min\{R(\mathbf{y}) / R(\mathbf{X}), 1\}$. Aquí, $s(\mathbf{X}, \mathbf{y}) = R(\mathbf{y}) q(\mathbf{y}, \mathbf{X}) / (R(\mathbf{X}) q(\mathbf{X}, \mathbf{y}))$ y s es cualquier función simétrica tal que $0 \leq \alpha(\mathbf{X}, \mathbf{y}) \leq 1$. Muestre que ambos son casos especiales del muestreador de Markov generalizado. [Sugerencia: tomar $Y = X$.]

6.19 Implementar el algoritmo de recocido simulado para el *norte*-problema de reinas sugerido en el ejemplo 6.13. ¿Cuántas soluciones puedes encontrar?

6.20 Implemente el algoritmo de templado simulado basado en Metropolis-Hastings para el TSP del ejemplo 6.12. Ejecute el algoritmo en algunos problemas de prueba en

<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

6.21 Escriba un algoritmo de recocido simulado basado en el muestreador de paseo aleatorio para maximizar la función

$$S(X) = \left| \frac{\text{pecados}(10X) + \text{porques}(5X+1)}{X^2 - X + 1} \right|, X \in \mathbb{R}.$$

Utilizar una $NORTE(x, \sigma)$ función propuesta, dado el estado actual X . Empezar con $X=0$. Trazar el mejor valor actual de la función contra el número de evaluaciones de S para varios valores de σ y varios programas de recocido. Repita los experimentos varias veces para evaluar qué funciona mejor.

Otras lecturas

MCMC es una de las principales herramientas de computación estadística y análisis bayesiano. Se puede encontrar una discusión completa de las técnicas MCMC en [20], y las aplicaciones prácticas se discuten en [8]. Véase también [4]. Para obtener más detalles sobre el uso de MCMC en el análisis bayesiano, consulte [6]. Una referencia clásica sobre recocido simulado es [1]. Algoritmos de búsqueda global más generales se pueden encontrar en [26]. Un artículo influyente sobre la detección de la estacionariedad en las cadenas de Markov, que está estrechamente relacionado con el muestreo perfecto, es [3].

REFERENCIAS

1. EHL Aarts y JHM Korst. *Máquinas de recocido simulado y Boltzmann*. John Wiley & Sons, Chichester, 1989.
2. DJ Aldous y J. Fill. *Cadenas de Markov reversibles y paseos aleatorios en gráficos*. En la preparación de. <http://www.stat.berkeley.edu/users/aldous/book.html>, 2007.
3. S. Asmussen, PW Glynn y H. Thorisson. Detección estacionaria en el problema transitorio inicial. *Transacciones ACM en Modelado y Simulación por Computadora*, 2(2):130–157, 1992.
4. S. Brooks, A. Gelman, G. Jones y X.-L. Meng. *Manual de Cadena de Markov Monte Carlo*. Prensa CRC, 2011.
5. M.-H. Chen y BW Schmeiser. Muestreo general de Monte Carlo de golpe y ejecución para evaluar integrales multidimensionales. *Cartas de investigación de operaciones*, 19(4):161–169, 1996.
6. A. Gelman, JB Carlin, HS Stern y DB Rubin. *Análisis de datos bayesianos*. Chapman & Hall, Nueva York, 2ª edición, 2003.
7. S. Geman y D. Geman. Relajación estocástica, distribución de Gibbs y restauración bayesiana de imágenes. *Transacciones IEEE en PAMI*, 6:721–741, 1984.
8. WR Gilks, S. Richardson y DJ Spiegelhalter. *Cadena de Markov Monte Carlo en la práctica*. Chapman & Hall, Nueva York, 1996.
9. PJ Green. Cálculo de Monte Carlo de cadena de Markov de salto reversible y determinación del modelo bayesiano. *Biometrika*, 82(4):711–732, 1995.
10. J. Hammersley y M. Clifford. Campos de Markov en grafos finitos y redes. Manuscrito inédito, 1970.
11. WK Hastings. Métodos de muestreo Monte Carlo utilizando cadenas de Markov y sus aplicaciones. *Biometrika*, 57:92–109, 1970.
12. JM Keith, DP Kroese y D. Bryant. Un muestreador de cadena de Markov generalizado. *Metodología y Computación en Probabilidad Aplicada*, 6(1):29–53, 2004.
13. FPKelly. *Reversibilidad y Redes Estocásticas*. John Wiley & Sons, Chichester, 1979.

14. JS Liu. *Estrategias de Monte Carlo en Computación Científica*. Springer-Verlag, Nueva York, 2001.
15. L. Lovász. Hit-and-run mezclas rápido. *Programación Matemática*, 86:443–461, 1999.
16. L. Lovász y SS Vempala. Hit-and-run es rápido y divertido. Informe técnico, Microsoft Research, SMS-TR, 2003.
17. L. Lovász y S. Vempala. Golpe y fuga desde una esquina. *Revista SIAM de Computación*, 35(4):985–1005, 2006.
18. M. Metropolis, AW Rosenbluth, MN Rosenbluth, AH Teller y E. Teller. Cálculos de ecuaciones de estado por máquinas de computación rápida. *Revista de física química*, 21:1087–1092, 1953.
19. JG Propp y DB Wilson. Muestreo exacto con cadenas de Markov acopladas y aplicaciones a la mecánica estadística. *Estructuras aleatorias y algoritmos*, 1 y 2:223–252, 1996.
20. CP Roberto y G. Casella. *Métodos estadísticos de Monte Carlo*. Springer-Verlag, Nueva York, 2ª edición, 2004.
21. HE Romeijn y RL Smith. Recocido simulado para optimización global restringida. *Revista de optimización global*, 5:101–126, 1994.
22. SM Ross. *Simulación*. Academic Press, Nueva York, 3ª edición, 2002.
23. Y. Shen. *Recocido de búsqueda adaptativa con métodos de muestreo Hit-and-Run para algoritmos de optimización global estocástica*. Tesis doctoral, Universidad de Washington, 2005.
24. Y. Shen, S. Kiatsupaibul, ZB Zabinsky y RL Smith. Un programa de enfriamiento derivado analíticamente para el recocido simulado. *Revista de optimización global*, 38(3):333–365, 2007.
25. RL Smith. Procedimientos eficientes de Monte Carlo para generar puntos uniformemente distribuidos en regiones delimitadas. *La investigación de operaciones*, 32:1296–1308, 1984.
26. ZB Zabinsky. *Búsqueda adaptativa estocástica para optimización global*. Editores académicos de Kluwer, Dordrecht, 2003.
27. ZB Zabinsky, RL Smith, JF McDonald, HE Romeijn y DE Kaufman. Mejora de hit-and-run para la optimización global. *Revista de optimización global*, 3:171–192, 1993.

CAPÍTULO 7

ANÁLISIS DE SENSIBILIDAD Y OPTIMIZACIÓN MONTE CARLO

7.1 INTRODUCCIÓN

Como se discutió en el Capítulo 3, muchos sistemas complejos del mundo real en ciencia e ingeniería se pueden modelar como *sistemas de eventos discretos*. El comportamiento de tales sistemas se identifica a través de una secuencia de eventos discretos, lo que hace que el sistema cambie de un estado a otro. Los ejemplos incluyen sistemas de tráfico, sistemas de fabricación flexibles, sistemas de comunicaciones por computadora, sistemas de inventario, líneas de producción, sistemas coherentes de por vida, redes de técnicas de revisión y evaluación de programas (PERT) y redes de flujo. Un sistema de eventos discretos se puede clasificar como *estático* o *dinámico*. Los primeros se llaman *sistemas estáticos de eventos discretos* (DESS), mientras que estos últimos se denominan *sistemas dinámicos de eventos discretos* (DEDS). La principal diferencia es que los DESS no evolucionan con el tiempo, mientras que los DEDS sí. La red PERT es un ejemplo típico de un DESS, siendo el rendimiento de muestra, por ejemplo, el camino más corto en la red. Una red de colas, como la red Jackson de la Sección 3.4.1, es un ejemplo de un DEDS, siendo el rendimiento de muestra, por ejemplo, el retraso (tiempo de espera de un cliente) en la red. En este capítulo nos ocuparemos principalmente de DESS. Para un estudio completo de DESS y DEDS, se remite al lector a [12], [17] y [21].

Debido a su complejidad, la evaluación del desempeño de los sistemas de eventos discretos generalmente se realiza mediante simulación y, a menudo, se asocia con la estimación de una función de respuesta (\mathbf{tu}) = $\text{mitu}[H(\mathbf{X})]$, donde la distribución del rendimiento de la muestra $H(\mathbf{X})$ depende del parámetro de control o de referencia $\mathbf{tu} \in V$. *Sensibilidad*

análisis se ocupa de evaluar las sensibilidades (gradientes, hessianas, etc.) de la función de respuesta (**tu**) con respecto al vector de parámetros **tu**, y se basa en la función de puntuación y la información de Fisher. Proporciona orientación para las decisiones operativas y de diseño y juega un papel importante en la selección de parámetros del sistema que optimizan ciertas medidas de rendimiento.

Para ilustrar, considere los siguientes ejemplos:

1. **Redes estocásticas.** El análisis de sensibilidad se puede emplear para minimizar el camino medio más corto en la red con respecto, digamos, a los parámetros del enlace de la red, sujeto a ciertas restricciones. Las redes PERT y las redes de flujo son ejemplos comunes. En el primero, las variables de entrada y salida pueden representar la duración de las actividades y la duración mínima del proyecto, respectivamente. En este último, pueden representar capacidades de flujo y capacidades de flujo máximas.
2. **Sistemas de semáforos.** La medida del desempeño podría ser el retraso promedio de un vehículo cuando avanza desde un origen determinado a un destino determinado o el número promedio de vehículos que esperan una luz verde en una intersección determinada. Los parámetros de sensibilidad y decisión pueden ser la velocidad promedio a la que los vehículos llegan a las intersecciones y la velocidad de los cambios de luz de verde a rojo. Algunas cuestiones de rendimiento de interés son:
 - (a) ¿Cuál será la demora promedio del vehículo si la tasa entre llegadas en una intersección dada aumenta (disminuye), digamos, en un 10-50%? ¿Cuál sería el impacto correspondiente de agregar uno o más semáforos al sistema?
 - (b) ¿Qué parámetros son los más significativos para causar cuellos de botella (alta congestión en el sistema), y cómo se pueden prevenir o eliminar estos cuellos de botella con mayor eficacia?
 - (c) ¿Cómo se puede minimizar el retraso promedio en el sistema, sujeto a ciertas restricciones?

Distinguiremos entre los llamados *distributivo* parámetros de sensibilidad y la *estructural* unos. En el primer caso, estamos interesados en las sensibilidades de los rendimiento esperado

$$(\mathbf{tu}) = \text{mitu}[H(\mathbf{X})] = \int H(\mathbf{X})f(\mathbf{X};\mathbf{tu}) d\mathbf{X} \quad (7.1)$$

con respecto al vector de parámetros **tu** del pdf $f(\mathbf{X};\mathbf{tu})$, mientras que en este último caso, estamos interesados en las sensibilidades del rendimiento esperado

$$(\mathbf{tu}) = \text{MI}[H(\mathbf{X};\mathbf{tu})] = \int H(\mathbf{X};\mathbf{tu})f(\mathbf{X}) d\mathbf{X} \quad (7.2)$$

con respecto al vector de parámetros **tu** en el rendimiento de la muestra $H(\mathbf{X};\mathbf{tu})$. Como ejemplo, considere un *CURRO1* cola. En el primer caso, **tu** podría ser el vector de las tasas entre llegadas y de servicio, y en el segundo caso, **tu** podría ser el tamaño del búfer. Tenga en cuenta que a menudo el vector de parámetros **tu** incluye tanto la distributiva como la parámetros estructurales. En tal caso utilizaremos la siguiente notación:

$$(\mathbf{tu}) = \text{mitu}_2[H(\mathbf{X};\mathbf{tu}_1)] = \int H(\mathbf{X};\mathbf{tu}_1)f(\mathbf{X};\mathbf{tu}_2) d\mathbf{X}, \quad (7.3)$$

dónde **tu** = (**tu**₁, **tu**₂). Tenga en cuenta que (**tu**) en (7.1) y (7.2) pueden considerarse casos particulares de (**tu**) en (7.3), donde los tamaños correspondientes de los vectores **tu**₁ o **tu**₂ igual a 0.

■ EJEMPLO 7.1

Dejar $H(\mathbf{X}; tu_3, tu_4) = \max\{X_1 + tu_3, X_2 + tu_4\}$, donde $\mathbf{X} = (X_1, X_2)$ es un vector bidimensional con componentes independientes y $X_i \sim F(X; tu_i)$, $i = 1, 2$. En este ejemplo tu_1 y tu_2 son parámetros de distribución, y tu_3 y tu_4 son estructurales.

Considere el siguiente problema de minimización utilizando la representación (7.3):

$$\begin{aligned}
 &\text{minimizar} && o(\mathbf{tu}) = \text{mitu}_1[H_0(\mathbf{X}; \mathbf{tu}_2)], && \mathbf{tu} \in V, \\
 (\text{PAGSo}) \quad &\text{sujeto a:} && f_j(\mathbf{tu}) = \text{mitu}_1[H_j(\mathbf{X}; \mathbf{tu}_2)] - 0, && j = 1, \dots, k, \\
 &&& f_j(\mathbf{tu}) = \text{mitu}_1[H_j(\mathbf{X}; \mathbf{tu}_2)] = 0, && j = k+1, \dots, \text{METRO},
 \end{aligned} \tag{7.4}$$

dónde $H_j(\mathbf{X})$ es el j -rendimiento de muestra -th, impulsado por un vector de entrada $\mathbf{X} \in \mathbb{R}_{\text{norte}}$ con pdf $F(\mathbf{X}; \mathbf{tu}_1)$, $\mathbf{tu} = (\mathbf{tu}_1, \mathbf{tu}_2)$ es un vector de parámetros de decisión que pertenece a algún conjunto de parámetros $V \subset \mathbb{R}_{\text{metro}}$.

Cuando la función objetivo $o(\mathbf{tu})$ y las funciones de restricción $f_j(\mathbf{tu})$ están disponibles analíticamente, (Po) se convierte en un problema de programación no lineal estándar, que se puede resolver de forma analítica o numérica mediante técnicas de programación no lineal estándar. Por ejemplo, la optimización del sistema de colas de Markovian cae dentro de este dominio. Aquí, sin embargo, se supondrá que la función objetivo y algunas de las funciones de restricción en (Po) no están disponibles analíticamente (típicamente debido a la complejidad del sistema subyacente), por lo que se debe recurrir a métodos de optimización estocástica, particularmente la optimización de Monte Carlo.

El resto de este capítulo está organizado de la siguiente manera: la Sección 7.2 trata del análisis de sensibilidad de DESS con respecto a los parámetros de distribución. Aquí presentamos al célebre *función de puntuación* (SF) método. La sección 7.3 trata de la optimización basada en simulación para programas del tipo (Po) cuando los valores esperados $\text{mitu}_1[H_j(\mathbf{X}, \mathbf{tu}_2)]$

se sustituyen por sus correspondientes medias muestrales. La versión basada en simulación de (Po) se llama el *contraparte estocástica* del programa original (Po). El énfasis principal se pondrá en la contraparte estocástica del programa sin restricciones (Po). Aquí mostramos cómo el método de la contraparte estocástica puede aproximarse de manera bastante eficiente a la verdadera solución óptima desconocida del programa (Po) usando una sola simulación. Nuestros resultados se basan en [16, 18, 19], donde se establecen los fundamentos teóricos del método de la contraparte estocástica. Es interesante notar que Geyer y Thompson [3] descubrieron de forma independiente el método de la contraparte estocástica en 1995. Lo usaron para hacer inferencias estadísticas para una configuración particular sin restricciones del programa general (Po). La Sección 7.4 presenta una introducción al análisis de sensibilidad y la optimización basada en simulación de DESS. Se hace especial hincapié en el análisis de sensibilidad con respecto a los parámetros de distribución de las cadenas de Markov utilizando la versión dinámica del método SF. Para un estudio exhaustivo del análisis de sensibilidad y la optimización de DESS, incluidos los diferentes tipos de colas y modelos de inventario, se remite al lector a [17].

7.2 MÉTODO DE LA FUNCIÓN DE PUNTUACIÓN PARA EL ANÁLISIS DE SENSIBILIDAD DE DESS

En esta sección presentamos el célebre *método de función de puntuación (SF)* para el análisis de sensibilidad de DESS. El objetivo del método SF es estimar el gradiente y las derivadas superiores de (\mathbf{tu}) con respecto al vector de parámetros distribucionales \mathbf{tu} , donde el rendimiento esperado viene dado (ver (7.1)) por

$$(\mathbf{tu}) = \text{mitu}[H(\mathbf{X})],$$

con $\mathbf{X} \sim F(\mathbf{X}; \mathbf{tu})$. Como veremos a continuación, el enfoque SF permite la estimación de *todos* sensibilidades (gradientes, hessianas, etc.) de una *ejecución de simulación única* (experimento) para un DESS con decenas y muy a menudo con cientos de parámetros. Seguimos de cerca [17].

Considere primero el caso donde \mathbf{tu} es escalar (denotado por lo tanto tu en vez de \mathbf{tu}) y supongamos que el conjunto de parámetros V es un intervalo abierto en la recta real. Supongamos que para todos \mathbf{X} el pdf $F(\mathbf{X}; tu)$ es continuamente diferenciable en tu y que existe una función integrable $h(\mathbf{X})$ tal que

$$\left| H(\mathbf{X}) \frac{dF(\mathbf{X}; tu)}{dtu} \right| \leq h(\mathbf{X}) \quad (7.5)$$

para todos $tu \in V$. Entonces, bajo condiciones moderadas [19], los operadores de diferenciación y expectativa (integración) son intercambiables, de modo que la diferenciación de (tu) rendimientos

$$\begin{aligned} \frac{d(tu)}{dtu} &= \frac{d}{dtu} \int H(\mathbf{X}) F(\mathbf{X}; tu) d\mathbf{X} = \int H(\mathbf{X}) \frac{dF(\mathbf{X}; tu)}{dtu} d\mathbf{X} \\ &= \int H(\mathbf{X}) \frac{\frac{dF(\mathbf{X}; tu)}{dtu}}{F(\mathbf{X}; tu)} F(\mathbf{X}; tu) d\mathbf{X} = \text{mitu} \left[H(\mathbf{X}) \frac{d \ln F(\mathbf{X}; tu)}{dtu} \right] \\ &= \text{mitu}[H(\mathbf{X}) S(tu; \mathbf{X})], \end{aligned}$$

dónde

$$S(tu; \mathbf{X}) = \frac{d \ln F(\mathbf{X}; tu)}{dtu}$$

es el *función de puntuación* (SF); véase también (1.57). Esto es visto como una función de tu para una dada \mathbf{X} .

Consideremos a continuación el caso multidimensional. Argumentos similares nos permiten representar el gradiente y las derivadas de orden superior de (\mathbf{tu}) en la forma]

$$\nabla_k(\mathbf{tu}) = \text{mitu} \left[H(\mathbf{X}) S_{(k)}(\mathbf{tu}; \mathbf{X}) \right], \quad (7.6)$$

dónde

$$S_{(k)}(\mathbf{tu}; \mathbf{X}) = \frac{\nabla^k F(\mathbf{X}; \mathbf{tu})}{F(\mathbf{X}; \mathbf{tu})} \quad (7.7)$$

es la *k-función de puntuación de orden th*, $k=0, 1, 2, \dots$. En particular, $S_{(0)}(\mathbf{tu}; \mathbf{X}) = 1$ (por definición), $S_{(1)}(\mathbf{tu}; \mathbf{X}) = S(\mathbf{tu}; \mathbf{X}) = \nabla \ln F(\mathbf{X}; \mathbf{tu})$, y $S_{(2)}(\mathbf{tu}; \mathbf{X})$ se puede representar como

$$\begin{aligned} S_{(2)}(\mathbf{tu}; \mathbf{X}) &= \nabla S(\mathbf{tu}; \mathbf{X}) + S(\mathbf{tu}; \mathbf{X}) S(\mathbf{tu}; \mathbf{X}) \\ &= \nabla^2 \ln F(\mathbf{X}; \mathbf{tu}) + \nabla \ln F(\mathbf{X}; \mathbf{tu}) \nabla \ln F(\mathbf{X}; \mathbf{tu}), \end{aligned} \quad (7.8)$$

dónde $\nabla \ln f(\mathbf{X}; \mathbf{tu})$ representa la transpuesta del vector columna $\nabla \ln f(\mathbf{X}; \mathbf{tu})$ de derivadas parciales de $\ln f(\mathbf{X}; \mathbf{tu})$. Tenga en cuenta que todas las derivadas parciales se toman con respecto a los componentes del vector de parámetros \mathbf{tu} .

La tabla 7.1 muestra las funciones de puntuación $S(\mathbf{tu}; X)$ calculado a partir de (7.6) para las distribuciones de uso común dadas en la Tabla A.1 en el Apéndice. Nosotros tomamos \mathbf{tu} usando los parámetros usuales para cada distribución. Por ejemplo, para el Gama(α, λ) y NORTE(μ, σ) distribuciones, tomamos $\mathbf{tu} = (\alpha, \lambda)$ y $\mathbf{tu} = (\mu, \sigma)$, respectivamente.

Tabla 7.1: Funciones de puntuación para distribuciones de uso común.

Distribución	$f(X; \mathbf{tu})$	$S(\mathbf{tu}; X)$
Exp(λ)	$\lambda e^{-\lambda x}$	$\lambda - 1 - X$
Gama(α, λ)	$\frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}$	$(\alpha - 1 - X) - \frac{1}{\Gamma(\alpha)} \frac{\Gamma(\alpha, \lambda x)}{\Gamma(\alpha)}$
NORTE(μ, σ)	$\frac{1}{\sigma} \sqrt{\frac{2}{\pi}} \frac{m - x}{\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$	$(\sigma^{-2}(x - \mu), -\sigma^{-1} + \sigma^{-3}(x - \mu)^2)$
Weib(α, λ)	$\alpha \lambda (\lambda x)^{\alpha-1} e^{-(\lambda x)^\alpha}$	$(\alpha - 1 + \ln(\lambda x)[1 - (\lambda x)^\alpha], \alpha [1 - (\lambda x)^\alpha])$
Compartimiento($norte, pags$)	$\sum_{x=0}^{norte} \frac{pags^x (1-pags)^{n-x}}{X!}$	$\frac{x - np}{pags(1-pags)}$
Poi(λ)	$\frac{\lambda^x e^{-\lambda}}{x!}$	$\frac{X}{\lambda} - 1$
GRAMO($pags$)	$pags(1-pags)^{x-1}$	$\frac{1 - p \text{íxeles}}{pags(1-pags)}$

En general, las cantidades $\nabla_k(\mathbf{tu})$, $k=0, 1, \dots$, no están disponibles analíticamente, ya que la respuesta (\mathbf{tu}) no está disponible. Sin embargo, se pueden estimar mediante simulación como

$$\hat{\nabla}_k(\mathbf{tu}) = \frac{1}{norte} \sum_{i=1}^{norte} H(\mathbf{X}_i) S_k(\mathbf{tu}; \mathbf{X}_i). \quad (7.9)$$

Se ve fácilmente que la función $\nabla_k(\mathbf{tu})$ y todas las sensibilidades $\nabla_k(\mathbf{tu})$ pueden estimarse a partir de una única simulación, ya que en (7.6) todas ellas se expresan como expectativas con respecto a la misma fdp, $f(\mathbf{X}; \mathbf{tu})$.

Los siguientes dos ejemplos de juguetes proporcionan más detalles sobre la estimación de $\nabla(\mathbf{tu})$. Ambos ejemplos son solo ilustrativos, ya que $\nabla_k(\mathbf{tu})$ está disponible analíticamente.

■ EJEMPLO 7.2

Dejar $H(\mathbf{X}) = X$, con $X \sim \text{Ber}(pags = tu)$, donde $tu \in [0, 1]$. Utilizando la Tabla 7.1 para la Compartimiento(1, $pags$) distribución, encontramos fácilmente que el estimador de $\nabla(tu)$ es

$$\hat{\nabla}(tu) = \frac{1}{norte} \sum_{i=1}^{norte} X_i \frac{X_i - tu}{tu(1-tu)} = \frac{1}{Naciones Unidas} \sum_{i=1}^{norte} X_i - 1, \quad (7.10)$$

dónde $X_1, \dots, X_{\text{norte}}$ es una muestra aleatoria de $\text{Ber}(tu)$. en la segunda ecuación usamos el hecho de que $X_2 = X_i$. El signo de aproximación en (7.10) se sigue de la ley de los grandes números.

Suponer que $tu = 1/2$. Supongamos también que tomamos una muestra de tamaño $\text{norte} = 20$ de $\text{Ber}(1/2)$ y obtuvo los siguientes valores:

$$\{X_1, \dots, X_{20}\} = \{0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1\}.$$

De (7.10) vemos que la derivada muestral es $\hat{V}_1(1/2) = 1.1$, mientras que la verdadera uno es claramente $V_1(1/2) = 1$.

■ EJEMPLO 7.3

Dejar $H(\mathbf{X}) = X$, con $X \sim \text{Exp}(\lambda = tu)$. Este es también un ejemplo de juguete, ya que $V(tu) = -1/tu$. Vemos en la tabla 7.1 que $S(tu, X) = tu^{-1} - X$, y por lo tanto

$$\hat{V}(tu) = \frac{1}{\text{norte}} \sum_{i=1}^{\text{norte}} X_i (tu^{-1} - X_i) \approx -\frac{1}{tu} \quad (7.11)$$

es un estimador de $V(tu)$, dónde $X_1, \dots, X_{\text{norte}}$ es una muestra aleatoria de $\text{Exp}(tu)$.

■ EJEMPLO 7.4 Ejemplo 7.1 (Continuación)

Como antes, deja $H(\mathbf{X}; tu_3, tu_4) = \max\{X_1 + tu_3, X_2 + tu_4\}$, dónde $\mathbf{X} = (X_1, X_2)$ es un vector bidimensional con componentes independientes y $X_i \sim F(X, tu_i)$, $i = 1, 2$. Supongamos que estamos interesados en estimar $V(\mathbf{tu}_1)$ con respecto al vector de parámetros distribucionales $\mathbf{tu}_1 = (tu_1, tu_2)$. Tenemos

$$\hat{V}(\mathbf{tu}_1) = \frac{1}{\text{norte}} \sum_{i=1}^{\text{norte}} H(\mathbf{X}_i; tu_3, tu_4) S(\mathbf{tu}_1; \mathbf{X}_i),$$

dónde $S(\mathbf{tu}_1; \mathbf{X})$ es el vector columna $(S(tu_1; X_1), S(tu_2; X_2))$.

A continuación, aplicaremos la técnica de muestreo por importancia para estimar las sensibilidades $V_k(\mathbf{tu}) = \text{mi}_{\mathbf{tu}}[H(\mathbf{X}) S_{(k)}(\mathbf{tu}; \mathbf{X})]$ simultáneamente para varios valores de \mathbf{tu} . Con este fin, deja $\text{gramo}(\mathbf{X})$ sea la densidad de muestreo de importancia, y suponga, como de costumbre, que el apoyo de $\text{gramo}(\mathbf{X})$ contiene el apoyo de $H(\mathbf{X}) R(\mathbf{X}; \mathbf{tu})$ para todos $\mathbf{tu} \in V$. Después $V_k(\mathbf{tu})$ Se puede escribir como

$$V_k(\mathbf{tu}) = \text{mi}_{\text{gramo}}[H(\mathbf{X}) S_{(k)}(\mathbf{tu}; \mathbf{X}) W(\mathbf{X}; \mathbf{tu})], \quad (7.12)$$

dónde

$$W(\mathbf{X}; \mathbf{tu}) = \frac{F(\mathbf{X}; \mathbf{tu})}{\text{gramo}(\mathbf{X})} \quad (7.13)$$

es la razón de verosimilitud de $R(\mathbf{X}; \mathbf{tu})$ y $\text{gramo}(\mathbf{X})$. El estimador de razón de verosimilitud de $V_k(\mathbf{tu})$ Se puede escribir como

$$\hat{V}_k(\mathbf{tu}) = \frac{1}{\text{norte}} \sum_{i=1}^{\text{norte}} H(\mathbf{X}_i) S_{(k)}(\mathbf{tu}; \mathbf{X}_i) W(\mathbf{X}_i; \mathbf{tu}), \quad (7.14)$$