BEIO

Boletín de Estadística e Investigación Operativa

Volumen 37
Número 1
Marzo 2021

ISSN: 2387-1725

A. Forte F. Parreño	Editorial	1
J. E. Ruiz Castro C. J. Acal González A. M. Aguilera del Pino	Phase-Type distributions: computational aspects and applications in electronics	3
L. J. R. Esparza J. C. Macías Ponce R. A. Kú Carrillo S. E. Delgadillo Alemán A. E. Giles Flores	Ranking of the Academic Departments of the Autonomous University of Aguascalientes	19
S. Carrascosa García	Coherence between surveys and register-based data in labour market Statistics	38
R. Medel Esquivel J. A. Vázquez I. Gómez-Vargas R. García Salcedo	An introduction to Markov Chain Monte Carlo	47
P. Grima X. Tort-Martorell R. Behar Gutiérrez	Matemáticas y Estadística. Al César lo que es del Cesar...	75

Revista Oficial de la Sociedad de Estadística e Investigación Operativa

Boletín de Estadística e Investigación Operativa
Vol. 37, No. 1, Marzo 2021, pp. 47-37

Historia y Enseñanza

An introduction to Markov Chain Monte Carlo

Ricardo Medel Esquivel and Isidro Gómez-Vargas
CICATA-Legaria
Instituto Politécnico Nacional (México)
✉ rmedele1500@alumno.ipn.mx,, ✉ igomezv0701@alumno.ipn.mx,

J. Alberto Vázquez
Instituto de Ciencias Físicas
Universidad Nacional Autónoma de México
✉ javazquez@icf.unam.mx

Ricardo García Salcedo
Departamento de Ingeniería Civil, División de Ingeniería[1]
Universidad de Guanajuato (México)
✉ rigarcias@ipn.mx

Abstract

In this paper we present a didactic introduction to the Markov Chain Monte Carlo method. We do not assume that the reader has prior knowledge about the subject, so we also present the necessary fundamentals of probability theory, a historical outline of the evolution of the Monte Carlo method and practical examples on numerical integration and statistical inference implemented in Python.

Keywords: Markov chains, Monte Carlo method, Random numbers.
AMS Subject classifications: 65C05.

1. Introducción

Bajo el nombre de método de Monte Carlo (MC) se designa a una familia de técnicas computacionales basadas en el muestreo aleatorio (en la generación de números pseudoaleatorios) y empleadas para hallar soluciones aproximadas de una gran variedad de problemas matemáticos surgidos en diversos campos de la ciencia (por ejemplo en la investigación, en genética y biología, en astronomía y

[1]Estancia sabática.

© 2021 SEIO

cosmología, en los métodos bayesianos de la estadística, la minería de datos y el aprendizaje automático, entre otros). No hay pues un método MC sino muchos de ellos y puede decirse que, en general, se usan en tres modalidades distintas como [30]:

1. técnica de integración numérica;

2. método de muestreo aleatorio para simulación u optimización;

3. procedimiento de prueba de hipótesis.

La historia del método MC es fascinante. El contexto en el cual Stanislaw M. Ulam lo ideó y se empleó por vez primera coincide con dos hechos científicos de importancia mayor en la historia mundial del siglo XX: la construcción de la primera bomba atómica (el proyecto Manhattan) y la disponibilidad de una de las primeras computadoras de uso general: la ENIAC (Electronic Numerical Integrator And Computer). Coincidieron entonces la necesidad práctica de un método de cálculo poderoso y la herramienta adecuada para llevarlo a cabo.

En la actualidad, sobre todo en relación con la estadística bayesiana, las versiones del método MC más utilizadas se basan en la construcción de cadenas de Markov para obtener muestras de distribuciones de probabilidad específicas; estas extensiones del método MC se conocen como métodos de Monte Carlo vía Cadenas de Markov (MCMC, por sus siglas en inglés).

Presentamos aquí una primera aproximación a los métodos MCMC que pretende ser concisa, accesible y práctica para cualquier persona interesada en estos métodos y que cuente con un mínimo de requerimientos técnicos. Deseamos mostrar las ideas básicas del método y señalar porqué es plausible su funcionamiento, más que discutir consecuencias, alcances y dificultades del mismo. La estructura del artículo es la siguiente. En la Sección 2 se presenta una breve reseña del desarrollo histórico de los métodos MC y MCMC. En la Sección 3 se citan los teoremas de probabilidad fundamentales para comprender el método MC; en el Apéndice se ofrece una referencia más detallada, para aquellos lectores con menos bagaje sobre teoría de probabilidad. La Sección 4 es una recopilación de algunas ideas básicas sobre la generación de números pseudoaleatorios, que son el fundamento de todos los métodos MC. En la Sección 5 se ponen en práctica las ideas esenciales de los métodos de integración MC, mediante un ejemplo resuelto por tres aproximaciones distintas. La Sección 6 es un resumen de la teoría de cadenas de Markov, relevante para la implementación moderna de los métodos MC. Finalmente, en la Sección 7 se presenta otro ejemplo, esta vez para encontrar los parámetros de un ajuste lineal mediante un muestreo MCMC realizado con un algoritmo Metropolis-Hastings. Los códigos utilizados están disponibles en [8].

2. Reseña histórica

Stanislaw Ulam cuenta que la idea del método de Monte Carlo se le ocurrió cuando jugaba al solitario con un mazo de cartas, mientras se recuperaba de una enfermedad en 1946 [3, 18, 29]. Analizando el juego, se percató de que para tener una idea de la probabilidad de que salga un solitario (cosa en la que no influye la habilidad del jugador) era más práctico ir echando las cartas o experimentar con el proceso para tomar nota de cuántas veces salía, que tratar de determinar todas las combinaciones posibles, pues crecen exponencialmente y son difíciles de calcular, salvo en casos excepcionales. Esto le llevó a darse cuenta de que en problemas complejos es mejor un muestreo aleatorio que el análisis completo de todas las posibilidades.

En 1946, Ulam le planteó esta idea a John von Neumann –quien había jugado un papel muy importante en el desarrollo de la ENIAC– durante un largo viaje en automóvil y lo convenció de las posibilidades de los esquemas de cálculo probabilistas. Juntos, Ulam y von Neumann desarrollaron las matemáticas del nuevo método y publicaron el primer artículo sobre éste en 1949 [20]. El nombre de Monte Carlo, al parecer, fue sugerido por Nicholas Metropolis, en alusión al casino de Monte Carlo, lugar al cual era aficionado un tío de Ulam [18].

Sin embargo, las investigaciones históricas evidencian que el método ya era conocido y se había usado previamente para resolver algunos problemas de estadística. Incluso podría remontarse hasta el año 1773 con el experimento de la aguja de Buffon, que sirve para estimar el número π (aunque esta historia podría ser apócrifa) [12]. Lo que sí está documentado es que hacia 1901 Kelvin usó una simulación MC rudimentaria, valiéndose de cartas numeradas para estimar una distribución de velocidades; también Fermi se valió de cálculos manuales tipo MC en su trabajo sobre la fisión nuclear en la década de 1930 [23]. Y existen informes de que el método fue propuesto independientemente por J. E. Mayer para tratar problemas de la física del estado líquido [19].

En 1952 Metropolis y sus colaboradores propusieron el primer algoritmo de muestreo MCMC y lo implementaron en el MANIAC (siglas en inglés de Mathematical Analyzer, Numerical Integrator and Computer) de los Álamos para derivar, de la distribución de Boltzmann y del modelo de esfera dura, algunas propiedades físicas de un sistema simulado compuesto por 224 partículas ubicadas en una retícula de área unitaria [17, 19]. La aplicación del algoritmo de Metropolis al modelo que describe las transiciones de fase de un ferromagneto, conocido como modelo de Ising bidimensional, es quizá la más difundida, tanto por su utilidad pedagógica como por resultar fundamental en los trabajos de reconstrucción de imágenes [23].

Físicos y químicos, principalmente, desarrollaron generalizaciones y alternativas del algoritmo de Metropolis, sobre todo en relación con el modelo de Ising. Este trabajo de exploración y generalización culminó en 1970 cuando W.

K. Hastings logró fundamentar matemáticamente el algoritmo de Metropolis [10, 11, 25], y éste pasó a ser conocido como algortimo Metropolis-Hastings. El artículo de Hastings mostró el potencial del algoritmo como herramienta de muestreo de propósito general, más allá de los ámbitos de la física estadística, sin embargo pasarían varios lustros hasta que todas sus implicaciones fuesen valoradas acertadamente.

En 1983 S. Kirkpatrick y colaboradores introdujeron el algoritmo de Recocido Simulado [13] para abordar problemas de optimización combinatoria, en los cuales se desea minimizar una función objetivo, llamada costo, en cierta región, denominada espacio de búsqueda. Un ejemplo típico es el problema del vendedor viajero, en el cual se busca minimizar la distancia de un recorrido que ha de pasar por un conjunto de vértices dado. Kirkpatrick y colaboradores decidieron pensar la función objetivo como un tipo de energía, de esta manera pudieron aprovechar las intuiciones de la física estadística y aplicar el algoritmo de Metropolis. Con este algoritmo el uso de los MCMC se extendió a la investigación de operaciones, la biología, economía e ingeniería eléctrica [23].

El algoritmo de Recocido Simulado, así como el modelo de Ising, trabajan sobre una retícula dividida en celdas. Esto inspiró a los hermanos Donald y Stuart Geman a aplicar las ideas del algoritmo de Metropolis y el Recocido Simulado, en combinación con un enfoque estadístico bayesiano, para atacar el problema de la restauración de imágenes digitales; su enfoque los llevó a encontrar en 1984 una variante del algoritmo de Metropolis que culminó dos siglos de historia del enfoque estadístico bayesiano: el algoritmo de Gibbs [6].

La restauración de imágenes digitales por medio del algoritmo de Gibbs constituyó la primera aplicación popular conjunta de los métodos estadísticos bayesianos y los MCMC, aunque ya en 1968 los MCMC habían sido utilizados secretamente como una herramienta de la estadística bayesiana en la localización de un submarino de combate de propulsión atómica, el U.S.S. Scorpion. Este episodio histórico, así como un relato pormenorizado de los avatares del teorema de Bayes, desde sus inicios hasta la síntesis que supuso el algortimo de Gibbs, pueden encontrarse en el libro de Sharon Bertsch McGrayne [16].

En la década de 1990 el método de Monte Carlo experimentó una completa renovación al combinarse con los métodos bayesianos de inferencia, así como por la disponibilidad creciente de computadoras potentes y económicas; fue en esta época, evidenciado ya su potencial, cuando los MCMC se popularizaron como herramienta invaluable de la inferencia estadística de propósito general. El software especializado en análisis estadístico bayesiano BUGS (Bayesian Inference Using Gibbs Sampling), lanzado en 1992 y aún disponible en internet como OpenBUGS [28] contribuyó enormemente a la popularidad de los MCMC al hacerlos accesibles al público, principalmente para aplicaciones en Bioestadística.

Los lenguajes de programación modernos han permitido generar una ga-

ma muy amplia de innovaciones en relación con los MCMC y también formas alternas de muestreo aleatorio, cada vez más eficientes. R es un lenguaje de programación orientado a la estadística; por otro lado, Python se ha revelado como un lenguaje de programación muy versátil, popular en muchas áreas de la ciencia y la ingeniería: módulos científicos desarrollados en Python, R y otros lenguajes modernos se crean constantemente.

PyMC [22], por ejemplo, es un módulo de Python especializado en análisis estadístico bayesiano y contiene métodos MCMC de propósito general. Su desarrollo comenzó en 2003 con la intención de hacer accesibles los MCMC a los investigadores no especializados en estadística y estaba dirigido principalmente a los ecologistas. La publicación de su primera versión ocurrió en 2005; actualmente (2020), la versión 4.0 de PyMC está en desarrollo. Además existen otros módulos, muchos de ellos con aplicaciones a disciplinas específicas, como MontePython [1], especializado en Cosmología.

En la actualidad los métodos MCMC tienen un uso generalizado en el análisis intensivo de datos en áreas de la ciencia que trabajan con grandes cantidades de datos. Sin embargo, la complejidad de los modelos y las enormes cantidades de datos a tratar han orientado la búsqueda de técnicas de muestreo más eficientes que las derivadas de los métodos MC.

A manera de resumen de esta sección, la Tabla 1 enlista algunos de los acontecimientos más importantes en la historia del Método de Monte Carlo.

Nota: La lista de acontecimientos no es totalizadora.
Fuente: Elaboración propia.

Año	Acontecimiento
1949	Método de Monte Carlo
1953	Algoritmo Metropolis
1970	Algoritmo Metropolis-Hastings
1983	Algoritmo de Recocido Simulado
1984	Muestreo de Gibbs
1992	BUGS
2000-2020	Módulos especializados en Python y R

Tabla 1: Hitos en la historia de los métodos de Monte Carlo.

3. Antecedentes de probabilidad

En esta sección se citan las definiciones y los teoremas de probabilidad indispensables para comprender el funcionamiento del método MC. En el apéndice, siguiendo el excelente resumen que hace S. M. Ross [26], ofrecemos informa-

ción más completa para aquellos lectores poco familiarizados con la teoría de probabilidad.

Definición 3.1. *El valor esperado de $g(x)$, función de la variable aleatoria X, para el caso en que X adopte valores discretos de una función densidad de probabilidad $p(x)$, se define como*

$$\mathbb{E}[g(X)] := \sum_x g(x)p(x),$$

y para el caso en que X sea una variable aleatoria continua con función densidad de probabilidad $f(x)$

$$\mathbb{E}[g(X)] := \int_{-\infty}^{\infty} g(x)f(x)dx. \tag{3.1}$$

Definición 3.2. *Para una variable aleatoria X con media μ se define la varianza como*

$$Var(X) := \mathbb{E}[(X-\mu)^2].$$

Definición 3.3. *La covarianza de dos variables aleatorias, X de media μ_x y Y de media μ_y, se define como*

$$Cov(X,Y) := \mathbb{E}[(X-\mu_x)(Y-\mu_y)].$$

La desigualdad de Chebyshev (ver Apéndice para mayores detalles) permite arribar a los teoremas más importantes para el método de Monte Carlo ordinario, que citamos a continuación.

Teorema 3.1. ***Ley débil de los grandes números.***
Sea $X_1, X_2, ...$ una sucesión de variables aleatorias independientes e idénticamente distribuidas con media μ. Entonces, para cada $\epsilon > 0$

$$P\Big\{\Big|\frac{X_1+...+X_n}{n} - \mu\Big| < \epsilon\Big\} \to 0, \quad \textit{cuando } n \to \infty.$$

La generalización de este resultado es:

Teorema 3.2. ***Ley fuerte de los grandes números.***
Bajo las condiciones del teorema anterior, se tiene con probabilidad 1, que

$$\lim_{n\to\infty} \frac{X_1+...+X_n}{n} = \mu. \tag{3.2}$$

Y un resultado de gran utilidad para realizar inferencia estadística es el:

Teorema 3.3. ***Teorema central del límite.***
Dada una sucesión de variables aleatorias independientes e idénticamente distribuidas, $X_1, X_2, ...$, con media finita μ y varianza finita σ^2 se cumple que

$$\lim_{n\to\infty} P\Big\{\frac{X_1+...+X_n-n\mu}{\sigma\sqrt{n}} < x\Big\} = \phi(x), \tag{3.3}$$

donde $\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-x^2}{2}} dx$*, para* $-\infty < x < \infty$*, es la distribución normal o gaussiana.*

La ley fuerte de los grandes números es el fundamento de la integración por el método de Monte Carlo. Nótese, en primer lugar, que la Ecuación (3.1) permite interpretar una integral $\int_a^b g(x)dx$ como el valor esperado, μ, de la función $g(x)$ respecto a una distribución de probabilidad uniforme definida para $a \leq x \leq b$; es decir, si $f(x) = Unif(a,b) = \frac{1}{b-a}$, multiplicando la integral anterior por el factor $(b-a)$, se tiene

$$\int_a^b g(x)dx = (b-a)\int_a^b g(x)f(x)dx = (b-a)\mathbb{E}[g(X)] \text{ con } X \sim Unif(a,b).$$

Por otro lado, la ley fuerte de los grandes números (Ecuación 3.2), permite aproximar este último valor esperado mediante el promedio de una muestra de tamaño n de la variable aleatoria distribuida uniformemente, si se toma n suficientemente grande. Así:

$$\int_a^b g(x)dx = (b-a) \lim_{n\to\infty} \frac{X_1 + ... + X_n}{n} \text{ con } X \sim Unif(a,b).$$

La cuestión relevante, ahora, es cómo generar las variables aleatorias uniformemente distribuidas, $X_1, ..., X_n$. Exploramos esto en la siguiente sección.

4. Números pseudoaleatorios

Existen dispositivos mecánicos capaces de generar muestras o secuencias de números aleatorios, usualmente en relación con algunos juegos de azar: dados, ruletas, tómbolas, etcétera. En una computadora es posible implementar algoritmos para generar secuencias de números que, bajo ciertas pruebas estadísticas, sean prácticamente indistinguibles de las muestras generadas por procesos realmente aleatorios. Para hacer evidente la distinción, se llama números pseudoaleatorios a dichas secuencias producidas en una computadora, y a los algoritmos utilizados en esta tarea se les denomina generadores de números aleatorios.

Técnicamente, se dice que una secuencia de n números es aleatoria si no puede predecirse el valor de la entrada n_{k+1} dadas las entradas anteriores n_i, con $i = 0, 1, ..., k$.

Además, una secuencia pseudoaleatoria es una secuencia de números generada de manera determinista que debe resultar, bajo ciertos criterios estadísticos, indistinguible de una verdadera secuencia aleatoria.

Nótese que esta definición no hace referencia alguna a la distribución de probabilidad (que llamaremos distribución padre) generadora de la muestra de números pseudoaleatorios. Sin embargo, los números pseudoaleatorios fundamentales para fines prácticos son los que siguen una distribución uniforme, que se conocen como números rectangulares o, sencillamente, números aleatorios.

En 1946 John von Neumann propuso el primer algoritmo generador de números pseudoaleatorios: el método del cuadrado central. Consiste en proponer un número de n dígitos, al que se le llama semilla, se eleva éste al cuadrado y del resultado se toman los n dígitos centrales; el procedimiento se repite con el nuevo número de n dígitos, hasta generar toda una secuencia de números de n dígitos. La Tabla 2 esboza un ejemplo.

Nota: Números generados de la semilla $X_0 = 1981$.
Fuente: Elaboración propia.

N	X_N	$(X_N)^2$	X_{N+1}
0	1981	39**2436**1	2436
1	2436	59**3409**6	3409
2	3409	11**6212**81	6212
3	6212	38**5889**44	5889
4	5889	34**6803**21	6803
...	...	...	...

Tabla 2: Método del cuadrado central.

Este algoritmo funciona a cierto nivel, pero puede tener fallas severas. Por ejemplo, si se toma el número 3792 como semilla y se eleva al cuadrado, se obtiene 14379264; luego, esta semilla produce la secuencia no aleatoria 3792, 3792, 3792,... [5].

Otro método para producir números pseudoaleatorios, el generador congruencial multiplicativo, consiste en comenzar con una semilla x_0 y calcular de manera recursiva x_n, para $n \geq 1$:

$$x_n = ax_{n-1} \text{ módulo } m,$$

siendo a y m enteros positivos y donde la operación módulo m indica que x_n es el residuo de dividir ax_{n-1} entre m.

Existen también generadores congruenciales lineales y mixtos, del tipo:

$$x_n = ax_{n-1} + c \quad \text{módulo} \quad m.$$

En estos generadores los números a, c y m se deben elegir siguiendo ciertas reglas basadas en la aritmética modular y de acuerdo con las características de la memoria de la computadora para evitar que la secuencia alcance rápidamente el período que es característico de las estructuras congruenciales y, en consecuencia, deje de ser una muestra aleatoria [2].

Los generadores anteriores producen números pseudoaleatorios distribuidos uniformemente. Para generar números distribuidos bajo otras funciones de probabilidad se usan técnicas más sofisticadas, como el método de la transformación

inversa o el algoritmo de aceptación y rechazo, ambos ampliamente usados hasta nuestros días y propuestos también por John von Neumann, en una carta dirigida a Ulam fechada en 1947 [3]. El método de la transformada inversa opera bajo el siguiente resultado [26]:

Teorema 4.1. *Sea U una variable aleatoria uniforme en $(1, 0)$. Para cualquier función de distribución continua F, la variable aleatoria X definida como*

$$X = F^{-1}(U),$$

donde F^{-1} es valor de x tal que $F(x) = u$, tiene distribución F.

Este método aplica cuando puede determinarse la forma analítica de la inversa de la distribución de probabilidad padre de los números pseudoaleatorios. Sin embargo, esto no siempre es posible, incluso para funciones de distribución tan usuales como la distribución normal. En todos los casos en que no puede aplicarse el método de la transformada inversa es necesario un método particular. La referencia [15] es un manual que recopila técnicas para generar números pseudoaleatorios para muchas distribuciones de probabilidad.

El algoritmo de aceptación y rechazo, por otro lado, es una forma eficaz para generar muestras de una variable de distribución padre $f(x)$ y contiene el germen de las ideas para técnicas más elaboradas, así como para introducir el uso de las cadenas de Markov. Consiste en proceder indirectamente, a partir de una distribución $g(x)$, que previamente ya se sepa generar; cada valor se genera con una probabilidad proporcional a $\frac{f(y)}{g(y)}$, según el siguiente esquema. Se toma una constante c, de modo que para toda y se cumpla $\frac{f(y)}{g(y)} \leq c$ y se procede de manera recursiva, según el Algoritmo de Aceptación y Rechazo, mostrado en la Tabla 3.

Fuente: [26, 27].

Aceptación y Rechazo
Paso 1: Generar $Y \sim g(\cdot)$.
Paso 2: Generar un número aleatorio $U \sim Unif(0, 1)$.
Paso 3: Si $U < f(Y)/cg(Y)$, $X = Y$. Otro caso, ir a Paso 1.

Tabla 3: Algoritmo 4.1.

Al final de la sección anterior vimos que para estimar numéricamente la integral $\int_a^b g(x)dx$ podemos usar una muestra de una variable con distribución de probabilidad uniforme. Los números pseudoaleatorios con esta distribución son, por ello, fundamentales para el método de integración MC; también lo son, como indica el método de la transformada inversa, fundamentales para

generar muestras de distribuciones de probabilidad no uniformes. Muestrear una distribución de probabilidad más complicada puede tener un objetivo más allá de la integración, y estar más relacionado con conocer la distribución de probabilidad de los parámetros de un modelo. Los métodos MCMC suelen estar más enfocados a esta segunda tarea, como veremos en la Sección 7.

Los lenguajes de programación modernos tienen incorporados generadores de números pseudoaleatorios (ver la Figura 1). Por esta razón se les suele asumir como algo dado, sin embargo para aplicaciones específicas es una buena práctica analizar su procedimiento de generación para evitar errores estadísticos [14].

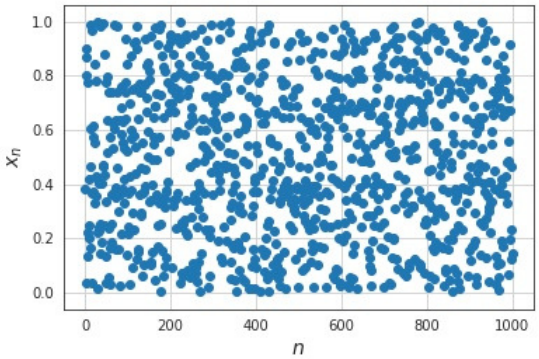


Figura 1: Representación gráfica de una muestra de 1000 números aleatorios en el intervalo $(0, 1)$ generados en Python.
Fuente: Elaboración propia.

5. Método de Monte Carlo ordinario

Como ya se dijo en la Sección 3, la idea central del método de integración Monte Carlo es la siguiente: si deseamos calcular la integral $I = \int_a^b g(x)dx$ para alguna función $g(x)$ difícil de integrar analíticamente, podemos hacer la siguiente sustitución:

$$I = \int_a^b g(x)dx = \int_a^b w(x)f(x)dx,$$

con $w(x) = g(x)(b-a)$ y $f(x) = \frac{1}{(b-a)}$. Por tanto, la integral transformada puede interpretarse como un múltiplo del valor esperado de $g(x)$ bajo la distribución uniforme:

$$\mathbb{E}(g(X)) = \frac{I}{(b-a)}, \quad \text{con} \quad X \sim Unif(a, b). \tag{5.1}$$

Si tomamos una muestra de números pseudoaleatorios uniformemente distribuidos, digamos $X_1, ..., X_N \sim Unif(a, b)$, por la ley de los grandes números

(Ecuación 3.2), para N suficientemente grande tendremos que, con probabilidad 1:

$$\frac{1}{N}\sum_{i=1}^{N} g(X_i) \to \mathbb{E}(g(X)) = \frac{I}{(b-a)}. \tag{5.2}$$

Ejemplo 5.1. *Para ilustrar el método de integración MC calculemos*

$$\int_0^1 \sqrt{arctan(x)}dx,$$

que no es integrable por métodos analíticos [24]. Programas orientados al cálculo científico, como Maple y Mathematica, no dan una solución analítica de la integral anterior. Se puede estimar esta integral numéricamente de varias maneras: por ejemplo, mediante sumas de Riemann, implementando el MC de acuerdo a la Ecuación (5.2) o implementando un MC en la región bidimensional que acota la función $f(x) = \sqrt{arctan(x)}$ *en el plano cartesiano.*

Con sumas de Riemann la aproximación numérica a la integral se puede elaborar con base en una partición regular, $b_0 = 0, b_1, b_2, ..., b_n = 1$*, del intervalo de integración, según la conocida fórmula del Cálculo:*

$$I = \frac{1}{n}\{g(b_1) + ... + g(b_n)\}, \tag{5.3}$$

donde $g(x) = \sqrt{arctan(x)}$ *y* $b_i = i/n$ *para* $i = 1, 2, ..., n$.

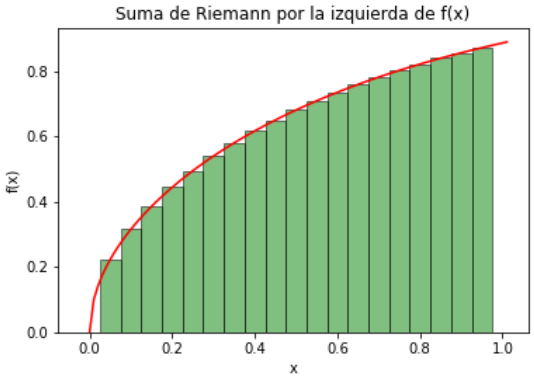


Figura 2: Sumas de Riemann para el área bajo la curva $f(x) = \sqrt{arctan(x)}$. Fuente: Elaboración propia.

La Figura 2 muestra el esquema gráfico de esta aproximación.

El MC basado en la ley de los grandes números, Ecuación (5.2), suele llamarse método MC de la Media Muestral. La aproximación numérica se obtiene de la siguiente expresión:

$$I = \frac{1}{n}\{g(x_1) + ... + g(x_n)\}, \tag{5.4}$$

donde $g(x) = \sqrt{arctan(x)}$ *y* $x_i \sim Unif(0,1)$. *Las Ecuaciones (5.3) y (5.4) tienen aspectos muy similares, sin embargo su interpretación es muy distinta, pues en el primer caso los valores de la variable representan puntos igualmente espaciados en el intervalo de integración* $(0,1)$ *mientras que en el segundo caso los valores de la variable consisten en una muestra de números pseudoaleatorios uniformes para el mismo intervalo* $(0,1)$*; entonces, de la Ecuación (5.1):*

$$I = (b-a)\mathbb{E}[g(X)].$$

Y un estimador insesgado (ver Apéndice) de esta integral es:

$$\theta_1 = (b-a)\frac{1}{n}\Sigma_{i=1}^{n} g(X_i) \;\; con \;\; X_i \sim Unif(0,1), \tag{5.5}$$

el cual tiene una varianza dada por:

$$Var(\theta_1) = \frac{1}{n}\left[(b-a)\int_a^b g^2(x)dx - I^2\right].$$

Entonces este método puede implementarse con el Algoritmo MC de la Media Muestral, presentado en la Tabla 4, para estimar $\int_a^b g(x)dx$ *[27].*

Fuente: [27].

Monte Carlo de la Media Muestral
Paso 1: Generar una secuencia $\{U_i\}_{i=1}^n \sim Unif(0,1)$.
Paso 2: Calcular $X_i = a + U_i(b-a)$.
Paso 3: Calcular $g(X_i)$, para $i = 1, ..., n$.
Paso 4: Estimar I calculando la media muestral θ_1 (Ecuación 5.5).

Tabla 4: Algoritmo 5.1.

La Figura 3 muestra el histograma de los resultados de esta integración MC repetida 1000 veces. Para hacer evidente la distribución estadística subyacente a este experimento, en este caso hemos tomado un valor pequeño para el tamaño de las muestras de números pseudoaleatorios $(n = 50)$.

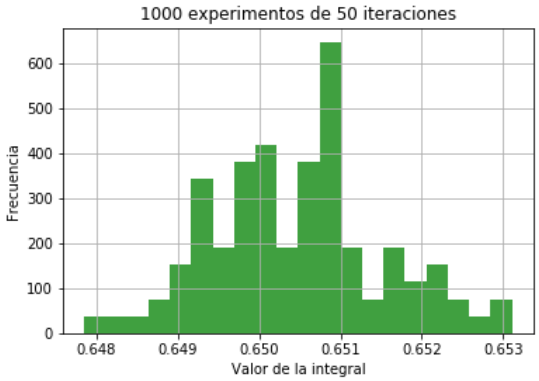


Figura 3: Histograma de resultados para la aproximación de $\int_0^1 \sqrt{arctan(x)}dx$ por el método MC de la Media Muestral.
Fuente: Elaboración propia.

Obsérvese que el histograma sugiere una distribución normal para los resultados de este experimento, lo que está de acuerdo con el teorema central del límite (Ecuación 3.3).

Nótese que la ley fuerte de los grandes números lleva a pensar que en la estimación numérica expresada por la Ecuación (5.4) el error, la diferencia entre I y μ, inversamente proporcional a $\sqrt{n}$, puede prácticamente anularse si n se hace suficientemente grande, y esta idea parece respaldada por el teorema central del límite; sin embargo, esto no se verifica en la práctica porque el método de MC trabaja no sobre números pseudoaletorios sino sobre transformaciones de ellos y los números pseudoaleatorios tienen períodos definitivamente finitos, por tanto suele ser necesario recurrir a diversas estrategias estadísticas para estimar los errores de aproximación y los intervalos de confianza [4].

El siguiente método MC es el más difundido porque tiene una interpretación geométrica, lo cual incrementa su valor pedagógico. Suele ser llamado método MC de Acierto y Error. Consiste en interpretar la integral como el área bajo la curva en una región rectangular, en nuestro ejemplo: $\{0 \leq x \leq 1\} \times \{0 \leq g(x) \leq c\}$, para algún valor c que acote la función en todo el intervalo de integración, y generar n puntos aleatorios (x, y) sobre dicho rectángulo. Luego, la probabilidad de acertar a la región bajo la curva es:

$$p = \frac{\int_a^b g(x)dx}{c(b-a)} = \frac{I}{c(b-a)}.$$

El parámetro p puede ser estimado de:

$$\hat{p} = \frac{n_a}{n},$$

donde n_a es el número de puntos bajo o sobre la curva, es decir, los aciertos del

muestreo. Y la integral puede ser, a su vez, estimada por:

$$I \approx \theta_2 = c(b-a)\frac{n_a}{n}, \tag{5.6}$$

Como la generación de cada uno de los puntos aleatorios es independiente de los otros, este procedimiento define un experimento de Bernoulli con probabilidad p de acertar; entonces θ_2 es un estimador insesgado de I porque:

$$\mathbb{E}(\theta_2) = c(b-a)\mathbb{E}(\frac{n_a}{n}) = pc(b-a) = I.$$

En este caso la varianza es:

$$Var(\theta_2) = \frac{1}{n}\Big[c(b-a) - I\Big],$$

mientras que la desviación estándar es:

$$\sigma_{\theta_2} = \frac{\sqrt{I(c(b-a)-I)}}{\sqrt{n}},$$

y puede observarse que la precisión del estimador es de orden $n^{-\frac{1}{2}}$.

Cuando n se hace suficientemente grande y se aplica el teorema central del límite a la variable $\hat{\theta} = \frac{\theta_2 - I}{\sigma_{\theta_2}}$ se tiene que el intervalo de confianza (ver Apéndice) con nivel $1-2\alpha$ para I es [27]:

$$\theta_2 \pm z_\alpha \frac{\sqrt{\hat{p}(1-\hat{p})}(b-a)c}{\sqrt{n}},$$

donde $z_\alpha = \phi^{-1}(\alpha)$ es la inversa de la distribución normal. Este método se puede implementar de acuerdo al Algoritmo MC de Acierto y Error, mostrado en la Tabla 5, para estimar $\int_a^b g(x)dx$ [27]:

Fuente: [27].

Monte Carlo de Acierto y Error
Paso 1: Generar una secuencia $\{U_i\}_{i=1}^{2n} \sim Unif(0,1)$.
Paso 2: Conformar pares (U_i, U_{n+i}), para $i = 1, ..., n$.
Paso 3: Calcular $X_i = a + U_i(b-a)$, para $i = 1, ..., n$.
Paso 4: Calcular $g(X_i)$, para $i = 1, ..., n$.
Paso 5: Contar los aciertos n_a tales que $g(X_i) > cU_{n+i}$.
Paso 6: Estimar I calculando la media muestral θ_2 (Ecuación 5.6).

Tabla 5: Algoritmo 5.2.

La Figura 4 muestra el resultado de este algoritmo para la función $g(x) = \sqrt{arctan(x)}$. Los tres métodos descritos se programaron en Python para tres valores de n y se obtuvieron los resultados registrados en la Tabla 6. Los códigos están disponibles en [8].

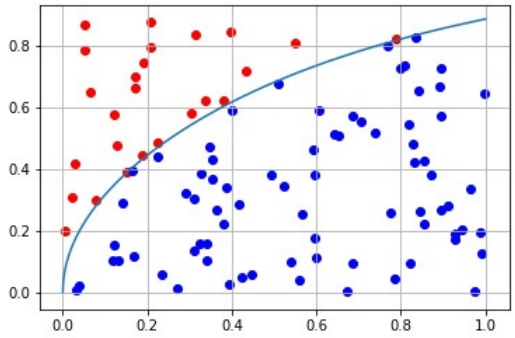


Figura 4: Estimación de $\int_0^1 \sqrt{arctan(x)}dx$ por el método MC de Acierto y Error. El área bajo la curva se estima como la razón del número de puntos bajo la curva (en azul) al número total de puntos (rojos más azules).
Fuente: Elaboración propia.

Nota: Los códigos están disponibles en [8].
Fuente: Elaboración propia.

Método	$n = 1000$	$n = 10000$	$n = 100000$
Sumas de Riemann	0.62937	0.62978	0.62982
MC media muestral	0.62560	0.63297	0.62989
MC acierto y error	0.62756	0.63001	0.62946

Tabla 6: Comparación de tres aproximaciones distintas de $\int_0^1 \sqrt{arctan(x)}dx$.

6. Cadenas de Markov

La fuente principal de esta sección es el libro de Wasserman [31], y su objetivo es brindar alguna familiaridad con las definiciones y los teoremas sobre cadenas de Markov más útiles en los métodos MCMC.

Definición 6.1. *Un proceso estocástico es una colección de variables aleatorias* $\{X_t : t \in T\}$*, donde* X_t *toma valores de un espacio de estados,* $\mathcal{X}$*, indexados por el conjunto* T*, llamado tiempo, que puede ser discreto o continuo.*

Definición 6.2. *Un proceso estocástico* $\{X_n : n \in T\}$ *es una cadena de Markov si*

$$P(X_n = x|X_0, ..., X_{n-1}) = P(X_n = x|X_{n-1}),$$

para todo n *y todo* x*.*

Es decir, en una cadena de Markov la probabilidad de alcanzar el valor X_n depende únicamente del valor previo X_{n-1}. Esta condición se conoce como la *propiedad de Markov* y representa un modelo probabilístico sencillo pero de gran potencia.

Definición 6.3. *Las cantidades*

$$p_{ij} \equiv P(X_{n+1} = j | X_n = i),$$

se llaman probabilidades de transición y la matriz $\mathbf{P}$ *cuyas entradas* (i, j) *son los elementos* p_{ij} *se llama matriz de transición.*

Una cadena de Markov puede imaginarse como una sucesión de valores de una variable aleatoria:

$$X_0 \to X_1 \to X_2 \to ... \to X_n \to ...,$$

en la cual cada valor de la variable tiene un solo padre. La cantidades clave de las cadenas de Markov son las probabilidades de ir de un estado de la cadena (un valor de la variable) al siguiente.

Definición 6.4. *Un estado se llama recurrente si*

$$P(X_n = i \text{ para algún } n \geq 1 | X_0 = i) = 1.$$

En otro caso, se dice que el estado es transitorio.

Las características importantes al generar cadenas de Markov, y por ende para aplicarlas a los métodos MC, son las referentes a su convergencia.

Definición 6.5. *Una cadena de Markov es irreducible si para cada par de estados* i *y* j *hay una probabilidad positiva de que el proceso transite del estado* i *al estado* j*.*

Definición 6.6. *Supongamos que* $X_0 = i$*. El tiempo de recurrencia se define como*

$$T_{ij} = min\{n > 0 : X_n = j\},$$

si X_n *siempre retorna al estado* i*, de lo contrario se define* $T_{ij} = \infty$*.*

Definición 6.7. *El tiempo medio de recurrencia de un estado* i *se define como*

$$m_i = \mathbb{E}(T_{ij}) = \Sigma_n n f_{ii}(n),$$

donde $f_{ij} = P(X_1 \neq j, X_2 \neq j, ..., X_{n-1} \neq j, X_n = j | X_0 = i)$*.*

Definición 6.8. *Un estado recurrente se llama nulo si* $m_i = \infty$*, en otro caso es llamado no nulo o positivo.*

Definición 6.9. *Una cadena de Markov se denomina aperiódica si no oscila regularmente entre valores de la variable.*

Definición 6.10. *Un estado se llama ergódico si es recurrente, no nulo y aperiódico. Y se dice que una cadena en sí misma es ergódica si todos sus estados son ergódicos.*

Ahora, un vector $\pi = (\pi_i : i \in \mathcal{X})$ cuyas entradas no negativas sumen 1 puede pensarse como un función discreta de probabilidad. El siguiente par de definiciones son las más relevantes para los fines de los métodos MCMC.

Definición 6.11. *π es una distribución estacionaria si $\pi = \pi\mathbf{P}$. Y se dice que la cadena tiene una distribución límite si*

$$\mathbf{P}^n \to \left\{ \begin{array}{c} \pi \\ \pi \\ \vdots \\ \pi \end{array} \right\},$$

para algún π, esto es, si $\pi_j = lim_{n\to\infty} P_{ij}^n$ existe y es independiente de i.

Definición 6.12. *Se dice que π satisface la condición del balance detallado si*

$$\pi_i p_{ij} = p_{ji}\pi_j. \tag{6.1}$$

Y los teoremas principales sobre convergencia de cadenas de Markov son los siguientes:

Teorema 6.1. *Una cadena de Markov ergódica e irreducible tiene una única distribución estacionaria π. Si g es una función acotada, entonces, con probabilidad 1:*

$$lim_{N\to\infty} \frac{1}{N} \Sigma_{n=1}^{N} g(X_n) \to \mathbb{E}_\pi(g) \equiv \Sigma_j g(j)\pi_j. \tag{6.2}$$

Teorema 6.2. *Si π satisface el balance detallado, entonces π es una distribución estacionaria.*

7. Monte Carlo vía Cadenas de Markov

La Ecuación (6.2), el teorema ergódico, es una generalización de la ley fuerte de los grandes números para cadenas de Markov. Es natural, entonces, pensar en utilizar estas últimas para calcular numéricamente integrales, en analogía al procedimiento mostrado en la Sección 5. La diferencia radica en que ahora al calcular numéricamente la integral:

$$\int_a^b g(x)f(x)dx,$$

la distribución de probabilidad $f(x)$ puede ser mucho más compleja y no importa su forma, las cadenas de Markov nos ayudarán a encontrar una muestra de valores X_i que sigan tal distribución. Esto se logra construyendo una cadena de Markov que tenga a $f(x)$ como distribución límite. Entonces, después de m iteraciones, una vez que la cadena de Markov ha convergido, se pueden desechar o *quemar* los primeros m valores de la cadena (en inglés se llama *burn in* a este segmento de la cadena) y el resto puede usarse para estimar el valor de expectación (y por ende la integral deseada), mediante la aproximación [7]:

$$\mathbb{E}_f(g(X)) = \frac{1}{n-m} \sum_{i=m+1}^{n} g(X_i) \text{ con } X_i \sim f(x).$$

Entonces, para implementar el método MCMC necesitamos un algoritmo para generar cadenas de Markov que tengan como distribución límite a $f(x)$ y también un criterio para determinar que las cadenas de Markov han convergido y que nos guíe en la elección del valor m en la quema de los primeros valores de la cadena.

El procedimiento más popular para generar cadenas de Markov es el algoritmo Metropolis-Hastings [7, 26], que tiene el siguiente esquema general mostrado en la Tabla 7:

Fuente: [7].

Metropolis-Hastings
Paso 1: Inicializar X_0, $t = 0$.
Paso 2: Repetir {
Generar un candidato $Y \sim q(\cdot\|X_t)$
Generar $U \sim U(0,1)$
Si $U \leq \alpha(X_t, Y)$, tomar $X_{t+1} = Y$
otro caso, tomar $X_{t+1} = X_t$
Incrementar t
}

Tabla 7: Algoritmo 7.1.

La función $q(\cdot|X_t)$ es una distribución que ya se sepa simular (por su simetría, suele elegirse la distribución normal para facilitar el proceso de simulación), $\pi(\cdot)$ es la función objetivo, es decir $f(\cdot)$, y $\alpha(X_t, Y)$, la probabilidad de aceptación, está definida como:

$$\alpha(X,Y) = min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right).$$

El algoritmo Metropolis-Hastings es una versión general para el MCMC, de la cual pueden derivarse otros algoritmos de muestreo. El algoritmo original de

Metropolis (1953), por ejemplo, se recupera al considerar solamente distribuciones de probabilidades simétricas, donde $q(Y|X) = q(X|Y)$, de modo que la probabilidad de aceptación se reduce a:

$$\alpha(X,Y) = min\left(1, \frac{\pi(Y)}{\pi(X)}\right).$$

Y como caso especial de este algoritmo están las caminatas aleatorias de Metropolis, donde $q(Y|X) = q(|X - Y|)$.

En todos los casos, es necesario notar que la cantidad $\alpha(X,Y)$ es fundamental para la construcción de las cadenas de Markov. La elección de la forma de $\alpha(X,Y)$, que es muy sencilla, garantiza que $\pi(\cdot)$ satisface la condición del balance detallado (Ecuación 6.1), y por tanto que es $\pi(\cdot)$, en sí misma, la distribución estacionaria de la cadena de Markov.

Por otro lado, un criterio muy utilizado para analizar la convergencia de las cadenas de Markov es la Prueba de Gelman-Rubin, que consiste en los siguientes pasos:

1. Generar $M \geq 2$ cadenas de Markov para generar las variables aleatorias θ que se desea muestrear, cada una con $2N$ iteraciones, partiendo de distintos puntos iniciales.

2. Descartar las primeras N iteraciones de cada cadena.

3. Calcular la varianza de cada cadena:

$$W = \frac{1}{M}\Sigma_{j=1}^{M} s_j^2,$$

   donde s^2 es la varianza de cada cadena, calculada tras descartar las primeras N iteraciones. Y calcular la varianza entre las cadenas:

$$B = \frac{N}{M-1}\Sigma_{j=1}^{M} s_j^2.$$

4. Calcular la varianza estimada de θ:

$$var(\theta) = (1 - \frac{1}{N})W + \frac{1}{N}B,$$

5. Calcular el factor:

$$R = \sqrt{\frac{var(\theta)}{W}}.$$

   Se acepta que las cadenas han convergido cuando $0.97 < R < 1.03$.

Los métodos MCMC son, pues, generadores de números pseudoaleatorios utilizados para simular funciones de probabilidad complicadas.

Aunque pueden utilizarse para integración numérica (que en ciertas aplicaciones puede ser algo muy complejo debido a la alta dimensión de las funciones a integrar), es más común que los MCMC se usen como métodos de exploración de las distribuciones estadísticas en sí mismas, generalmente para determinar sus valores óptimos globales (máximos y mínimos) o sus promedios. A continuación mostramos un ejemplo donde se explora una distribución estadística. En [9] se aborda el mismo tipo de análisis, enfatizando su relación con otras técnicas computacionales.

Ejemplo 7.1. *Para ilustrar la aplicación de los métodos MCMC vamos a considerar un ejemplo de ajuste de parámetros de un modelo lineal: dado un conjunto de datos encontraremos los dos parámetros (pendiente y ordenada al origen) del modelo lineal que mejor se ajusten estadísticamente para describirlos. El ejercicio consta de los siguientes pasos (el código utilizado está disponible en [8]):*

1. *Planteamos un modelo lineal $y = m_0x + b_0$, como prueba del método, eligiendo los parámetros $m_0 = 2$ y $b_0 = 3$. La prueba consiste en generar datos sintéticos a partir de este modelo y luego hacer inferencia estadística sobre esos datos, mediante el MCMC, para comprobar que podemos recuperar los parámetros elegidos.*

2. *Generamos datos sintéticos agregando una dispersión aleatoria a los valores de y: $y = m_0x + b_0 + \epsilon$, donde $\epsilon \sim N(0, \sigma)$, y también barras de error aleatorias hasta un tamaño elegido. El resultado se muestra en la Figura 5.*

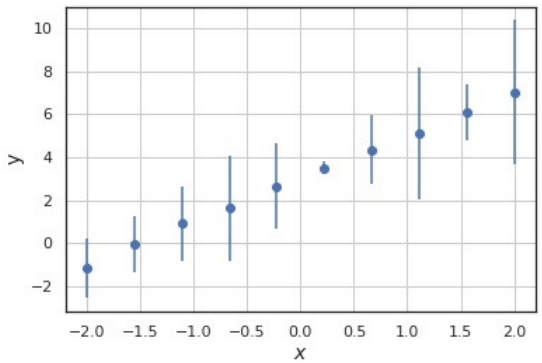


Figura 5: Datos sintéticos generados agregando dispersión y barras de error aleatorias a puntos de la recta $y = 2x + 3$.
Fuente: Elaboración propia.

3. *Ajustamos los datos al modelo valiéndonos de una prueba Chi-cuadrada, que es una generalización de la técnica de mínimos cuadrados [21]. Dado*

un conjunto de datos D_i y un modelo de estos datos, $y(x_i|\theta)$, que depende de un conjunto de parámetros θ, el ajuste del modelo está determinado por aquellos valores de los parámetros que minimizan la distribución:

$$\chi^2 \equiv \Sigma_{ij}(D_i - y_i(x_i|\theta))Q_{ij}(D_j - y_j(x_j|\theta)),$$

donde Q denota la inversa de la matriz de covarianza de los datos. En nuestro ejemplo, tenemos 10 datos que deseamos ajustar a un modelo lineal $y = mx + b$ con dos parámetros: m y b.

4. *Mediante un algoritmo Metropolis-Hastings realizamos un muestreo de la función χ^2 definida por los datos sintéticos y, tras una prueba de convergencia Gelman-Rubin, se determinan los valores de m y b que minimizan la función χ^2. La Figura 6 muestra las cadenas resultantes.*

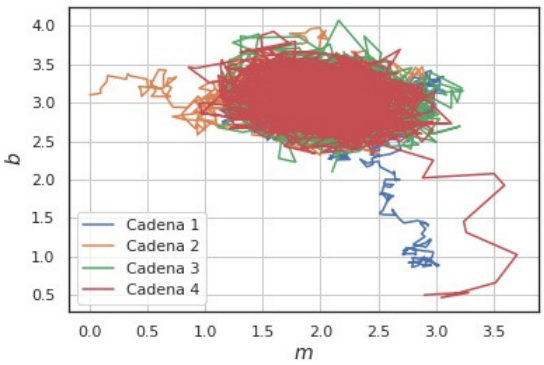


Figura 6: Cadenas de Markov en dos dimensiones ejecutadas para ajustar los datos sintéticos de la Figura 5 a un modelo lineal $y = mx + b$.
Fuente: Elaboración propia.

5. *Construimos las elipses de confianza (ver Apéndice) a partir de las cadenas generadas, las cuales se muestran en la Figura 7. Las elipses de confianza son regiones del espacio de parámetros (que en nuestro ejemplo es 2-dimensional) alrededor del punto de valores medios que contienen un porcentaje dado de la distribución de probabilidad. Es común utilizar la desviación estándar para cuantificar los niveles de confianza, de manera que las regiones de 1σ, 2σ y 3σ corresponden al $68.3\,\%$, $95.4\,\%$ y $99.5\,\%$, respectivamente. El mejor ajuste de los parámetros se obtuvo para $\theta_1 = m = 2.040$ y $\theta_2 = b = 3.029$ como valores que minimizan la función χ^2. Mientras que los valores medios son: $m = 1.951422$, $b = 3.027087$ con desviaciones medias 0.389912 y 0.280212, respectivamente, y matriz de covarianza:*

$$\begin{pmatrix} 0.1520314 & -0.00954926 \\ -0.00954926 & 0.07851859 \end{pmatrix}$$

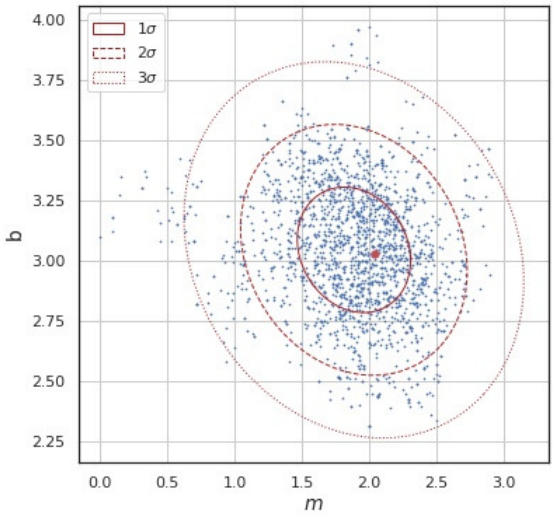


Figura 7: Elipses de confianza centradas en los valores medios $(1.951, 3.027)$ en el espacio de parámetros asociadas al ajuste de datos sintéticos de la Figura 5 para un modelo lineal $y = mx + b$. El punto $(2.040, 3.029)$ corresponde al mínimo de la distribución chi-cuadrada encontrado por el muestreo MCMC. Fuente: Elaboración propia.

8. Conclusión

Presentamos los métodos MC y MCMC de manera unificada, como resultantes de la aplicación de una misma idea esencial: la generación de números pseudoaleatorios para obtener muestras de distribuciones estadísticas. No obstante, mientras los métodos MC pueden requerir algoritmos muy particulares para simular distribuciones específicas, los métodos MCMC cuentan con algoritmos de aplicación mucho más general: entre ellos, el Metropolis-Hastings es uno de los más populares.

Las principales aplicaciones, integración numérica y exploración del espacio de una distribución estadística, se ilustran mediante ejemplos no triviales acompañados de códigos en Python que pueden implementarse y modificarse fácilmente, por lo cual éstos constituyen un primer paso en una exploración que puede iniciar el lector motivado.

Los autores sugerimos que se use este trabajo como material didáctico en cursos donde se estudien los métodos Monte Carlo, en vista de que la presentación unificada de estos métodos y la implementación práctica propuestos pueden contribuir a mejorar la comprensión del tema.

Agradecimientos

Este trabajo fue parcialmente apoyado por: CONACYT, ICF-UNAM, CICATA - Legaria del Instituto Politécnico Nacional (IPN) y el proyecto SIP20210500 del IPN. R.G.S. agradece el apoyo de las becas COFAA, EDI y Estancia Sabática del IPN, así como al proyecto FORDECYT-PRONACES-CONACYT No. CF-MG-2558591. J.A.V. agradece el apoyo proporcionado a los proyectos FOSEC SEP-CONACYT Investigación Básica A1-S-21925 y UNAM-DGAPA-PAPIIT IA102219. R.M.E. e I.G. V. agradecen el apoyo de las becas de posgrado CONACYT.

Referencias

[1] Audren, B. (2015). Monte Python. En: `https://baudren.github.io/`.

[2] Coss, R. (2017). *Simulación: Un enfoque práctico*, Limusa, México (México).

[3] Eckhardt, R. (1987). Stan Ulam, John Von Neumann, and the Monte Carlo Method. *Los Alamos Science*, **Special Issue(15)**, 131-136.

[4] Fishman, G. S. (1996). *Monte Carlo: concepts, algorithms and applications*, Springer-Verlag, New York (USA).

[5] Gardner, M. (1980). *Carnaval Matemático*, Alianza, Madrid (España).

[6] Geman, S., y Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6(6)**, 721-741.

[7] Gilks, W. R., Richardson, S., y Spiegelhalter, D. J. (1998). *Markov Chain Monte Carlo in practice*, Chapman & Hall/CRC, Boca Raton (USA).

[8] Gómez, I. (2019). IntroMCMC. En: `www.github.com/igomezv/`.

[9] Gómez, I., Medel, R., Vázquez, J. A., y García, R. (2019). Una Aplicación de las Redes Neuronales Artificiales en la Cosmología. *Komputer Sapiens*, **Año XI**(Vol. II), 12-16. En: `www.smia.mx/komputersapiens/`.

[10] Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57(1)**, 97-109.

[11] Hitchcock, D. B. (2003). A History of the Metropolis-Hastings Algorithm. *The American Statistician*, **57(4)**, 254-257.

[12] Johansen, A. M., y Evers, L. (2007). *Monte Carlo Methods: Lecture Notes*, University of Bristol. En: `www.warwick.ac.uk`.

[13] Kirkpatrick, S., Gelatt Jr., C. D., y Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, **220(4598)**, 671-680.

[14] Kneusel, R. T. (2018). *Random Numbers and Computers*, Springer (USA).

[15] Krishnamoorthy, K. (2006). *Handbook of statistical distributions with applications*, Chapman & Hall/CRC, Boca Raton (USA).

[16] McGrayne, S. B. (2012). *La teoría que nunca murió*, Crítica, Barcelona (España).

[17] Metropolis, N., y Ulam, S. (1952). A Property of Randomness of an Arithmetical function. *AECU*, **Technical Report(2038)**.

[18] Metropolis, N. (1987). The Beginning of the Monte Carlo Method. *Los Alamos Science*, **Special Issue(15)**, 125-130.

[19] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., y Teller, A. H. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, **21(6)**, 1087-1092.

[20] Metropolis, N., y Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association*, **44(247)**, 335-341.

[21] Press, W. H., Teukolsky, S. A., Vetterling, W. T., y Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, New York (USA).

[22] PyMC3 (2019) En: `https://docs.pymc.io/`.

[23] Richey, M. (2010). The Evolution of Markov Chain Monte Carlo Methods. *The American Mathematical Monthly*, **117(5)**, 383-413.

[24] Riley, K. F., Hobson, M. P., y Bence, S. J. (2006). *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*, Cambridge University Press, New York (USA).

[25] Robert, C., y Casella, G. (2011). A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. *Statistical Science*, **26(1)**, 102-115.

[26] Ross, S. M. (1999). *Simulación*, Prentice Hall, México (México).

[27] Rubinstein, R. Y., y Kroese, D. P. (2008). *Simulation and The Monte Carlo Method*, John Wiley & Sons, New Jersey (USA).

[28] Thomas, A. (2010). Overview. En: `www.openbugs.net/`.

[29] Ulam, S. M. (2002). *Aventuras de un Matemático*, Nivola, Madrid (España).

[30] Upton, G., y Cook, I. (2002). *A Dictionary of Statistics*, Oxford University Press, New York (United States of America).

[31] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*, Springer, New York (USA).

Acerca de los autores

Ricardo Medel Esquivel es licenciado en física y matemáticas. Actualmente desarrolla un proyecto de análisis de datos cosmológicos en el programa de doctorado del Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Legaria (CICATA-Legaria) del Instituto Politécnico Nacional (IPN).

Isidro Gómez Vargas es licenciado en física y matemáticas por el IPN, ubicado en la Ciudad de México. En la actualidad, realiza su doctorado en Tecnología Avanzada (también en el IPN), donde estudia la estimación de parámetros cosmológicos mediante técnicas estadísticas y computacionales.

J. Alberto Vázquez es investigador del Instituto de Ciencias Físicas (ICF) de la UNAM. Previo a su incorporación al ICF, fue catedrático CONACYT asociado al CINVESTAV, IPN. Estancia postdoctoral en el Laboratorio Nacional Brookhaven, NY. Doctorado y Maestría otorgados por el Kavli Institute for Cosmology de la Universidad de Cambridge, UK. Su investigación se enfoca en la exploración y análisis de observaciones cosmológicas, en el estudio de modelos de Energía oscura e Inflacionarios, a través de la estadística Bayesiana y el computo científico.

Ricardo García Salcedo es profesor titular en el Departamento de Física Educativa en el CICATA-Legaria del IPN. Licenciado y maestro en Ciencias por la Universidad Autónoma del Estado de México, doctorado en Ciencias por el Depto. de Física del CINVESTAV-IPN. Su investigación se centra en la cosmología, especialmente los modelos de energía oscura y TIC en la enseñanza de la Física.

A. Apéndice

Esta sección es un brevísimo repaso de los fundamentos de la teoría de probabilidad; está basado en [26], que recomendamos ampliamente a los lectores interesados en los fundamentos de la simulación.

Asociado a cada evento A de un espacio muestral S hay un número $P(A)$, la probabilidad de ocurrencia del evento A, que satisface los siguientes axiomas:

1. $0 \leq P(A) \leq 1$.
2. $P(S) = 1$.
3. $P\Big(\bigcup_{i=1}^{n} A_i\Big) = \sum_{i=1}^{n} P(A), \qquad n = 1, 2, ..., \infty$.

La probabilidad conjunta de los eventos A y B (la probabilidad de que sucedan simultáneamente) se denota como $P(AB)$. La probabilidad condicional del evento A dado el evento B se escribe $P(A|B)$, y se cumple la relación:

$$P(A|B) = \frac{P(AB)}{P(B)}.$$

Y cuando los eventos A y B son independientes:

$$P(AB) = P(A)P(B).$$

Definición A.1. *El valor esperado, o media, de una variable aleatoria discreta X que asume alguno de los valores posibles $x_1, x_2, ..., x_n$, es*

$$\mathbb{E}[X] := \sum_i x_i P\{X = x_i\}.$$

Definición A.2. *El valor esperado, o media, para una variable continua X que puede tomar todos los valores x en $(-\infty, \infty)$, con función densidad de probabilidad f, se define como:*

$$\mathbb{E}[X] := \int_{-\infty}^{\infty} x f(x) dx.$$

Teorema A.1. *El valor esperado y la varianza (Definición 3.2) tienen las siguientes propiedades*

1. $\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$.
2. $\mathbb{E}[X_1 + X_2] = \mathbb{E}[X_1] + \mathbb{E}[X_2]$.
3. $Var(X) = \mathbb{E}[X^2] - \mu^2$.
4. $Var(aX + b) = a^2 Var(X)$.

donde a y b son constantes y X_1 y X_2 son variables aleatorias.

Teorema A.2. *La varianza y la covarianza (Definición 3.3) satisfacen las siguientes propiedades*

1. $Cov(X, Y) = \mathbb{E}[XY] + \mathbb{E}[X]\mathbb{E}[Y]$.
2. $Var(X \pm Y) = Var(X) + Var(Y) \pm 2Cov(X, Y)$.

3. $Var(X) \geq 0$.

Así, cuando X y Y son independientes, $Cov(X, Y) = 0$ y se tiene

$$Var(X + Y) = Var(X) + Var(Y).$$

Algunos de los resultados más importantes de la teoría de probabilidad son las desigualdades, pues con base en ellas se pueden hacer inferencias y determinar intervalos de confianza. Aquí nos interesan principalmente los resultados para variables continuas. Uno muy útil es el siguiente.

Teorema A.3. ***Desigualdad de Markov.***
Si X toma solo valores no negativos, entonces para cualquier $a > 0$

$$P\{X \geq a\} \leq \frac{\mathbb{E}[X]}{a}. \tag{A.1}$$

Demostración. Si X es una variable aleatoria no negativa, entonces

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xf(x)dx = \int_{0}^{\infty} xf(x)dx,$$

y separando en dos segmentos el intervalo de integración:

$$\begin{aligned}
\mathbb{E}[X] &= \int_0^a xf(x)dx + \int_a^\infty xf(x)dx \\
&\geq \int_a^\infty xf(x)dx \\
&\geq \int_a^\infty af(x)dx, \text{ pues } xf(x) \geq af(x) \text{ si } x \geq a \\
&= a\int_a^\infty f(x)dx = aP\{X \geq a\},
\end{aligned}$$

de donde se obtiene el resultado deseado. ■

La desigualdad de Markov permite probar fácilmente el siguiente resultado, de mayor utilidad práctica.

Teorema A.4. ***Desigualdad de Chebyshev.***
Si X es una variable aleatoria con media μ y varianza σ^2, entonces para cualquier $k > 0$

$$P\{|X - \mu| \geq k\sigma\} \leq \frac{1}{k^2}. \tag{A.2}$$

Demostración. Se obtiene aplicando la desigualdad de Markov (Ecuación A.1) a la variable no negativa $\frac{(X-\mu)^2}{\sigma^2}$, que tiene media igual a 1, como puede comprobarse a partir de la definición de la varianza y las propiedades del valor esperado (Teorema A.1). ■

La desigualdad de Chebyshev (Ecuación A.2) permite, considerando la hipótesis de que la varianza siempre tiene valores finitos, arribar a los resultados más importantes para el método de Monte Carlo ordinario: las leyes de los grandes números (Teoremas 3.1 y 3.2).

Definición A.3. *Un estimador es una función de la muestra usada para estimar un parámetro desconocido de la población. El sesgo de un estimador es la diferencia entre el valor esperado del estimador y el verdadero valor del parámetro a estimar. Un estimador es insesgado cuando su valor esperado es igual al parámetro a estimar.*

Definición A.4. *Se llama intervalo de confianza a un par de números en la recta real entre los cuales se estima que estará cierto valor desconocido con un determinado nivel de confianza. Las elipses de confianza generalizan este concepto al caso bidimensional.*

http://www.seio.es/BEIO