# Integración Monte Carlo vía Cadenas de Markov con Python


**Ricardo Medel Esquivel*, Ricardo García-Salcedo, J. Alberto Vázquez González**

**Instituto Politécnico Nacional, CICATA-Legaria, 11500, CDMX, México.**
**Instituto de Ciencias Físicas, UNAM, 62210, Cuernavaca, Morelos, México.**
***Contacto: rmedele1500@alumno.ipn.mx**


## Introducción

Monte Carlo es una familia de técnicas computacionales para determinar soluciones numéricas aproximadas a diversos problemas matemáticos.
En este trabajo abordamos la integración numérica y la simulación; usamos: 1) Monte Carlo (MC) como método de integración numérica, y 2) Monte Carlo vía Cadenas de Markov (MCMC) para simular una distribución de probabilidad.
Esto continua nuestra investigación sobre estadística bayesiana computacional, marco en el cual deseamos realizar ajuste de parámetros de modelos científicos.
El problema es:

- Calcular $I = \int_a^b g(x)dx = \int_0^1 \sqrt{arctan(x)}dx$.
- Simular $P(p|D) = \frac{p^{10+a-1}(1-p)^{4b-1}}{B(10+a,4+b)}$ con $a = b = 1$.

Palabras clave: *muestreo aleatorio, simulación, cadenas de Markov.*

## Metodología

Para la integración: En el *MC de acierto o fallo* se interpreta la integral como área bajo la curva y se generan puntos aleatorios sobre ella. La integral se estima como $I \approx \theta_1 = c(b-a)\frac{n_a}{n}$ y se implementa con el:

| **Algoritmo 1: MC de acierto y fallo** |
|---|
| 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 $\theta_1$. |

El *MC de la media muestral* se basa en la ley de los grandes números para hacer la aproximación $I \approx \theta_2 = (b-a)E[g(X)]$. Este método puede implementarse siguiendo el siguiente algoritmo [1].

| **Algoritmo 2: MC 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 $\theta_2$. |

Para la simulación: generamos cadenas de Markov con distribución estacionaria $q(\cdot|X_t) = P(p|D)$ y condición $\alpha(X_t, Y)$ de balance detallado.

**Algoritmo 3: Metropolis-Hastings**[2]

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 $t$
}

## Resultados

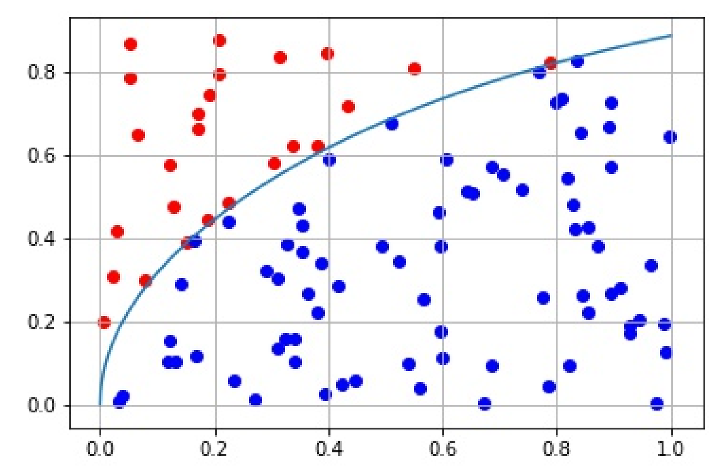

Fig. 1. MC de acierto y fallo para $\int_0^1 \sqrt{arctan(x)}dx$.

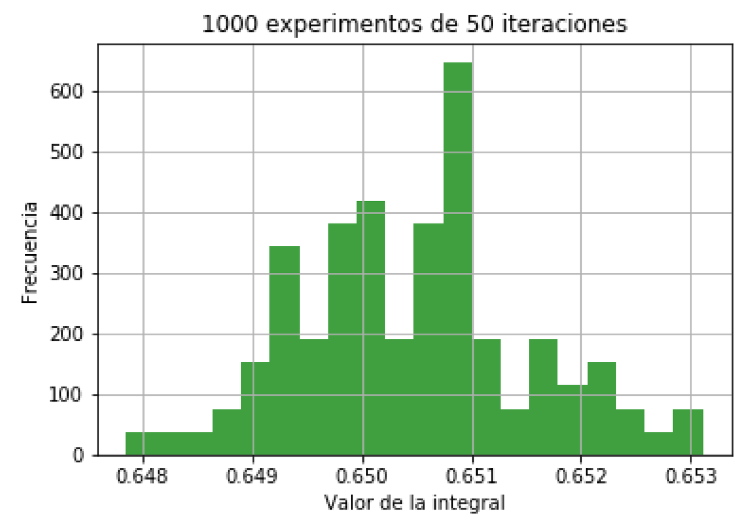


Fig. 2. Histograma del experimento MC de acierto y fallo para $\int_0^1 \sqrt{arctan(x)}dx$.

| Método | n=1000 | n=10000 | n=100000 |
|---|---|---|---|
| MC media muestral | 0.62560 | 0.63297 | 0.62989 |
| MC acierto o fallo | 0.62756 | 0.63001 | 0.62946 |

Tabla 1. Comparación de resultados de MC para $\int_0^1 \sqrt{arctan(x)}dx$.

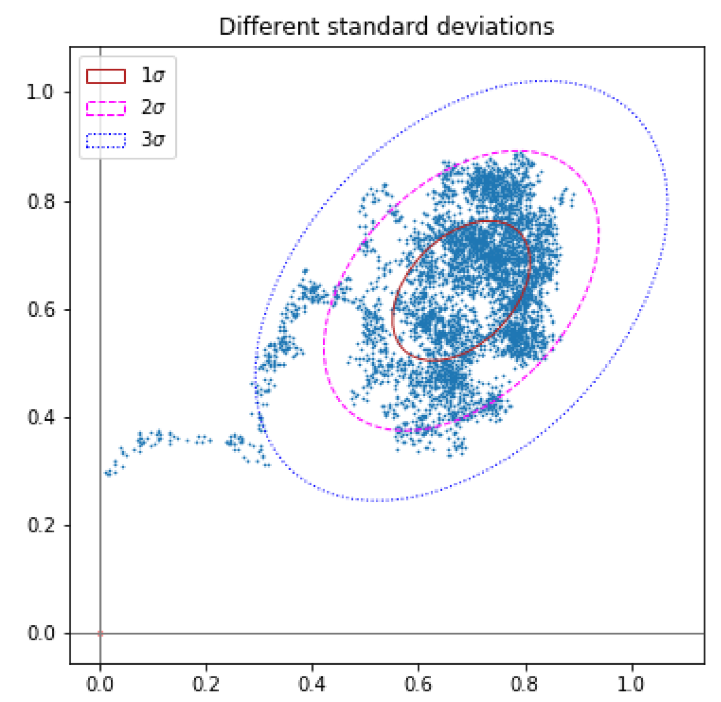


Fig. 3. Convergencia de las cadenas de Markov para $P(p|D) = \frac{p^{10+a-1}(1-p)^{4b-1}}{B(10+a,4+b)}$.

## Conclusión

MC y MCMC son herramientas fundamentales para la inferencia estadística computacional; cuestiones sobre convergencia y optimización quedan para trabajo futuro inmediato.

## Referencias


[1] Rubinstein, R. Y. (1981). Simulation and the Monte Carlo Method. John Wiley & Sons, New York, USA.

[2] Gilks, W. R. (1996), Richardson, S. & Spiegelhalter, D. J., Markov Chain Monte Carlo in practice, Chapman & Hall/CRC, New York.