

The background of the slide features a vibrant, abstract pattern of oil droplets suspended in water. The colors are primarily shades of blue, purple, and orange, creating a marbled effect. The droplets vary in size and are scattered across the frame, with some larger ones on the right side.

Métodos Numéricos 2

# Gibbs Sampling

Estuardo Menéndez  
Lorena Beltrán

Noviembre, 2021

# Contenido

## 1 El Origen

## 2 Antecedentes

2.1 Método de Monte Carlo

2.2 Algoritmo de Metrópolis

2.3 Algoritmo de Metrópolis - Hastings

2.4 Gibbs sampler para procesamiento de imágenes

## 3 Gibbs Sampling

## 4 Simulated Annealing

## 5 Ejemplos

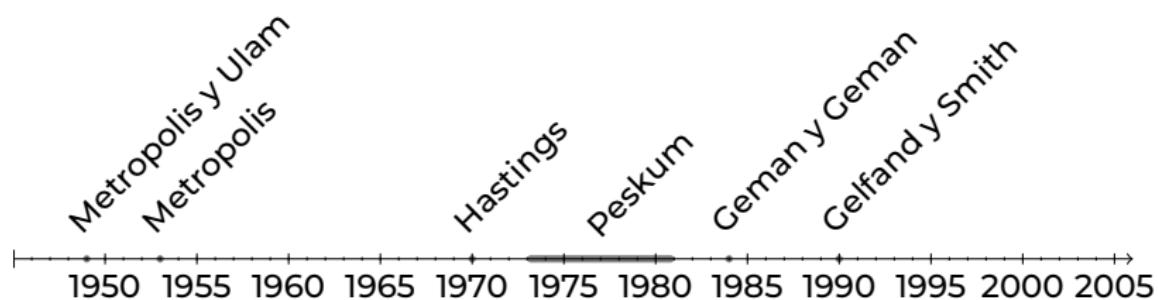
5.1 Circuitos

5.2 Comparación de diferentes configuraciones del método

5.3 Ley de Decrecimiento

## 6 Referencias

# El árbol genealógico de GS



# Método de Monte Carlo (1949)

Nicholas Metropolis y Stanisław Ulam proponen por primera vez el **Método Montecarlo** con su paper publicado en el *Journal of the American Statistical Association* (Vol 44)

## Método Monte Carlo

"El Método es, esencialmente, un acercamiento estadístico al estudio de ecuaciones diferenciales, o de manera más general, de las ecuaciones integrodiferenciales que ocurren en varias ramas de las ciencias naturales" (Metropolis, Ulam, 1949).

Bajo la intención de ofrecer una alternativa más adecuada a los problemas de física del continuo, se emplearon los recursos de cálculo y las ecuaciones diferenciales parciales en temas como hidrodinámica y electromagnetismo. Por otro lado, se tenían las *teorías cinéticas*, donde se aproximaba el continuo usando un número grande y finito de partículas.

# Método de Monte Carlo (1949)

¿Qué ocurre en un caso intermedio?

## Ejemplo

Asúmase que se tiene un medio donde se introduce una partícula nuclear capaz de producir partículas semejantes con una distribución de energía y dirección de movimiento.

El comportamiento de las partículas de este sistema es modelado por ecuaciones diferenciales cinéticas de los gases denominadas *ecuaciones de Boltzmann*.

Entonces, la idea general del método es evitar procedimientos de integración múltiple o multiplicación de matrices de probabilidad para sustituirlos como una cadena de eventos. Entonces, se toma una muestra de las posibles cadenas y se hace el estudio de las distribuciones de probabilidad en un tiempo dado.

Los Métodos de Cadenas de Markov Monte Carlo (MCMC) estás diseñados para adaptarse muestreos aleatorios sistemáticos para distribuciones de altas dimensiones. La diferencia con métodos de Monte Carlo es que estos no son tan efectivos luego de aumentar la dimensión del problema ya que su efectividad como modelo probabilístico decae.

### Def. Cadena de Markov

Proceso evolutivo en el que para cada paso el proceso puede estar en cualquiera de los estados; en el paso siguiente, el proceso puede permanecer en su estado presente o cambiar a uno de los otros estados. Y al momento de avanzar, su probabilidad depende únicamente del estado presente y no de la historia pasada del proceso.

# Algoritmo Metropolis (1953)

La definición del primer algoritmo MCMC es asociado a la segunda computadora MANIAC construida en Los Álamos bajo el mando de Metropolis (1952). Al año siguiente, se publicó en el *Journal of Chemical Physics* la computación de integrales de la forma

$$\mathcal{J} = \int F(\theta) \exp\{-E(\theta)/kT\} d\theta \Bigg/ \int \exp\{-E(\theta)/kT\} d\theta$$

en  $\mathbb{R}^{2N}$  usando  $\theta$  para un conjunto de  $N$  partículas en  $\mathbb{R}^2$  definiendo energía como

$$E(\theta) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N V(d_{ij})$$

donde  $V$  es la función potencial y  $d_{ij}$  la distancia eucliana entre las partículas. Igualmente, se empleó la constante de Boltzman con un factor de normalización. Y esto con la idea de proponer el modelo para las armas termonucleares.

# Algoritmo de Metrópolis-Hastings

El método de Metropolis fue generalizado por Hastings (1970) y por Peskun (1973,1981) para aplicaciones estadísticas. Definiendo una metodología para cadenas de Markov finitas y reversibles determinando una analogía discretizada para el lado continuo. Para esto, se definió la probabilidad genérica de pasar de un estado  $i$  a un estado  $j$  como

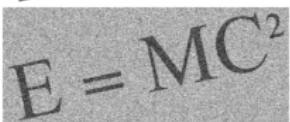
$$\alpha_{ij} = \frac{s_{ij}}{1 + (\pi_i/\pi_j)(q_{ij}/q_{ji})}$$

En donde  $s_{ij} = s_{ji}$ ,  $\pi_i$  es el objetivo y  $q_{ij}$  es la propuesta. A partir de este punto, se podrá usar los métodos de MCMC para dimensiones más grandes.

# Procesamiento de Imágenes(1984)

En 1984 Geman y Geman hicieron uso de métodos basados en MCMC, en particular Gibbs Sampling. En este exploran el uso de los campos aleatorios de Markov (MRF) como herramienta para restaurar imágenes que han sido sometidas a ruido de algún tipo. El supuesto original es que, al tomar segmentos de la imagen, estos se comportan, a grandes rasgos como un MRF, por lo que podemos usar métodos de aproximación basados en la teoría de MCMC.

$$E = MC^2$$


$$E = MC^2$$

$$E = MC^2$$

# GIBBS SAMPLING y aplicaciones varias

# Gibbs Sampling

La idea básica de Gibbs Sampling es evitar el problema que se tiene con altas dimensionalidades por medio de particionar el vector objetivo en secciones tal que estas sean linealmente independientes. Esto es  $x \in X \subset \mathbb{R}^d$  se escribe en  $K$  bloques tal que  $x = (x_1, x_2, x_3, \dots, x_K)$ . Luego se denota:

$$f_K(x_i|x_1, x_2, x_3, \dots, x_K) \quad i = 1, 2, \dots, K$$

Como el conjunto de distribuciones condicionales. Bajo las condiciones de regularidad de MCMC (convergencia del método y de los promedios) este conjunto expresa la distribución de la función  $f(x)$  bajo el siguiente teorema:

## Teorema (Hammersley-Clifford)

Si  $f(x) > 0$  para cualquier  $x \in \mathbb{X}$  entonces la distribución de  $f(x)$  esta determinada de manera única por medio de su conjunto de condicionales. Más precisamente:

$$f(x) = f(y) \prod_{k=1}^K \frac{f_{j_k}(x_{j_k} | x_{j_1}, x_{j_2}, \dots, x_{j_{k-1}}, y_{j_{k+1}}, \dots, y_{j_K})}{f_{j_k}(y_{j_k} | x_{j_1}, x_{j_2}, \dots, x_{j_{k-1}}, y_{j_{k+1}}, \dots, y_{j_K})} \quad x \in \mathcal{X}$$

Para cada permutación de  $j \in \{0, 1, 2, \dots, K\}$  y  $y \in \mathbb{X}$

Por lo tanto, se sabe que el proceso convergerá a la distribución deseada.

# Gibbs Sampling

**Algoritmo:** (Gibbs Sampling )

$$x = x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_K^{(0)})$$

**for**  $t = 0, 1, 2, \dots$

1 Simular  $x_1^{(t+1)} \sim f_1(x_1 | x_2^{(t)}, \dots, x_K^{(t)})$ .

2 Simular  $x_2^{(t+1)} \sim f_2(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots, x_K^{(t)})$ .

3 Simular  $x_3^{(t+1)} \sim f_2(x_3 | x_1^{(t+1)}, x_2^{(t+1)}, x_4^{(t)}, \dots, x_K^{(t)})$ .

⋮

K Simular  $x_K^{(t+1)} \sim f_K(x_K | x_1^{(t+1)}, \dots, x_{K-1}^{(t+1)})$ .

Lo más común para el criterio de paro son iteraciones. Dependiendo de la aplicación del método que se esté haciendo existen ciertos métodos específicos, los cuales dan una mejor idea de cuando el método debería parar. Uno de ellos, utilizado en procesamiento de imágenes, es el gradiente ralo anterior (basada en la anterior iteración del modelo).

# Ejemplos

# Ejemplos

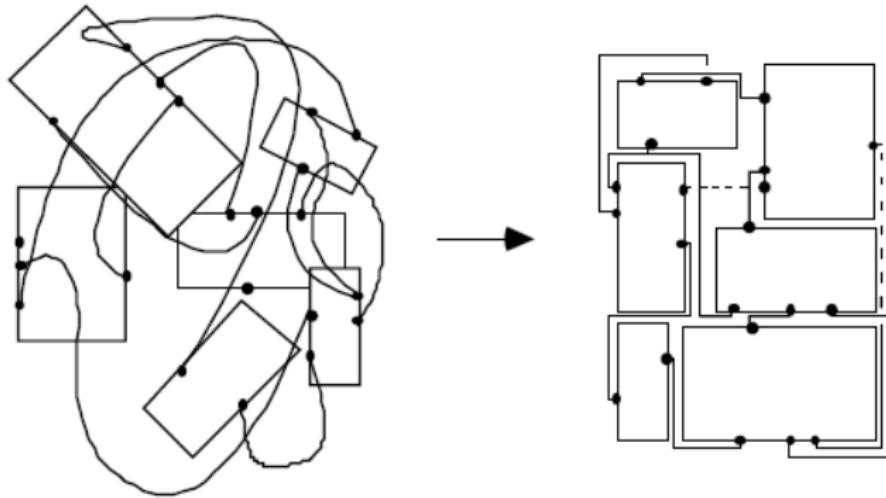


Figure: Configuraciones de conexiones de un circuito con componentes de diferentes tamaños

# Ejemplos

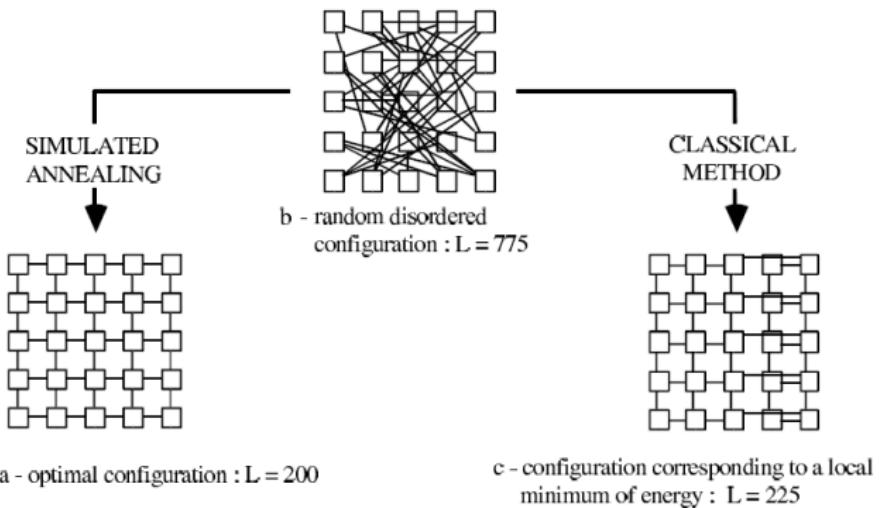


Figure: Comparación entre el método clásico para minimización y el Simulated Annealing

# Simulated Annealing: Física

El proceso algorítmico es un paralelo al proceso que se hace cuando se hace un Metropolis-Hastings. En base a una configuración inicial:

- ① Se hace una modificación elemental (un movimiento de parámetros o algun "giro")
- ② Si este cambio causa un decrecimiento en el nivel de energía, este es aceptado.
- ③ En caso contrario, si causa un incremento de  $\Sigma E$  en el nivel de energía, este también es aceptado con una probabilidad de  $e^{-\Delta E/T}$ , la constante de Boltzmann (donde la variable que cambia es T, la temperatura en ese set de parámetros iniciales).

Este procedimiento ayuda al algoritmo a viajar entre mínimos locales, dado que permite saltar entre niveles de temperatura y buscar valores más pequeños. Este proceso genera una cadena de Markov en el sentido que la siguiente cadena solo es dependiente de la que le sigue.

# Ley de decrecimiento

- ①  $T_{k+1} = \alpha T_k$  con  $\alpha = \text{constante}$  (geométrica)
- ②  $T_{k+1} = \alpha(T_k)T_k$
- ③  $T_{k+1} = \left(1 - T_k \frac{\Delta(T_k)}{\sigma^2(T_k)}\right) T_k$  donde  $\sigma^2(T_k) = \langle f_{T_k}^2 \rangle + \langle f_{T_k} \rangle^2$ ,  $f$  es la función a optimizar y  $\Delta(T_k)$  depende de la ley de adaptación tomada. Normalmente es constante.
- ④  $T_{k+1} = D_k T_k$  donde  $D_k = \min\left(D_0, \frac{E_k}{\langle E_k \rangle}\right)$ ,  $D_0$  es una constante tomada entre 0.5 y 0.9,  $\langle E_k \rangle$  es el promedio de energía de la configuración en el estado  $k$  y  $E_k$  es el mínimo de energía de la configuración en el estado  $k$ .

Los criterios de paro normalmente son el error relativo y el error absoluto entre las temperaturas o el nivel de energía en un estado decadente.

# Referencias

- Gammerman, D (2006). Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference.US: Taylor & Francis Group
- Gelfand, A., Hills, S., Racine-Poon, A. and Smith, A. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *J. Amer. Statist. Assoc.* 85 972–982.
- Geman, S.; Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. PAMI-6, NO. 6. 721-741
- Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their application. *Biometrika* 57 97–109.

# Referencias

- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953). Equations of state 14 C. ROBERT AND G. CASELLA calculations by fast computing machines. *J. Chem. Phys.* 21 1087–1092.
- Petrowski, D.; Taillard, S. (2005). *Metaheuristics for Hard Optimization*. Springer.
- Sucar, L. (2021). *Probabilistic Graphical Models*. Springer.
- Yue, C. (2018). *Markov Random Fields and Gibbs Sampling for Image Denoising*.