

# Práctica computacional - Modelo de Ising

## 1. Introducción

Consideremos una grilla de  $N = L \times L$  sitios en cada uno de los cuales hay un spin que solo puede tomar los valores  $S = 1, -1$ . Estos spines tienen interacciones tipo Ising dadas por el siguiente Hamiltoniano:

$$H = -J \sum_{\text{n.n}} S_i S_j - B \sum_i S_i \quad (1)$$

Donde la primer suma es sobre primeros vecinos y  $J$  y  $B$  son parámetros que dan la energía de interacción entre spines, entre cada uno de ellos y un campo magnético externo, respectivamente. Supongamos que queremos calcular numéricamente algún observable macroscópico a una dada temperatura  $\beta^{-1}$ , como por ejemplo el valor medio de la magnetización total:

$$\langle M \rangle = \sum_{S_1, \dots, S_N} p_{S_1, \dots, S_N} (S_1 + S_2 + \dots + S_N). \quad (2)$$

Esta vez la suma es sobre todas las configuraciones posibles de los spines, y  $p_{S_1, \dots, S_N}$  es la probabilidad de cada configuración, que según el ensamble canónico, es:

$$p_{S_1, \dots, S_N} = \frac{1}{Z} e^{-\beta H(S_1, \dots, S_N)}, \quad (3)$$

donde  $Z$  es la función de partición:

$$Z = \sum_{S_1, \dots, S_N} e^{-\beta H(S_1, \dots, S_N)}. \quad (4)$$

Si quisieramos evaluar la fórmula de la ecuación 2 tal cual esta planteada, necesitaríamos recorrer todos los estados posibles del sistema (al igual que para calcular exactamente la función de partición, necesaria para calcular las probabilidades  $p_{S_1, \dots, S_N}$ ). El problema es que la cantidad de estados es  $2^N$ , y solo para  $N = 100$  este número es ridículamente grande. Ninguna computadora podría recorrer todos estos estados en un tiempo razonable. Pero, como sabemos, hacer eso tampoco es necesario, ya que solo una fracción de los estados posibles contribuirá apreciablemente a la suma en la ecuación 2. Entonces, dado que no podemos recorrer todos los estados, una

estrategia posible sería solo considerar algunos términos de la ecuación 2, de una forma tal que estos términos sean representativos de los estados que típicamente recorre el sistema cuando se encuentra a temperatura  $\beta^{-1}$ . Estos términos, naturalmente, son aquellos para los cuales la probabilidad  $p_{S_1, \dots, S_N}$  es mayor. Dicho de otra forma, necesitamos ‘muestrear’ la distribución de probabilidad  $p_{S_1, \dots, S_N}$ , es decir generar estados de forma tal que la fracción de veces que se genera el estado  $(S_1, \dots, S_N)$  sea  $p_{S_1, \dots, S_N}$ . El algoritmo de Metrópolis, que pertenece a la familia mas general de algoritmos de Montecarlo, es una solución a este problema.

## 2. Algoritmo de Metrópolis

El algoritmo de Metrópolis genera estados secuencialmente, de forma que cada estado es generado a partir del anterior de forma estocástica. La secuencia de estados generados es entonces una cadena de Markov, y el proceso de generación es tal que la distribución estacionaria es la distribución de Boltzmann que se desea muestrear. El algoritmo es muy sencillo, y para el caso de un sistema de Ising 2D se describe de la siguiente manera:

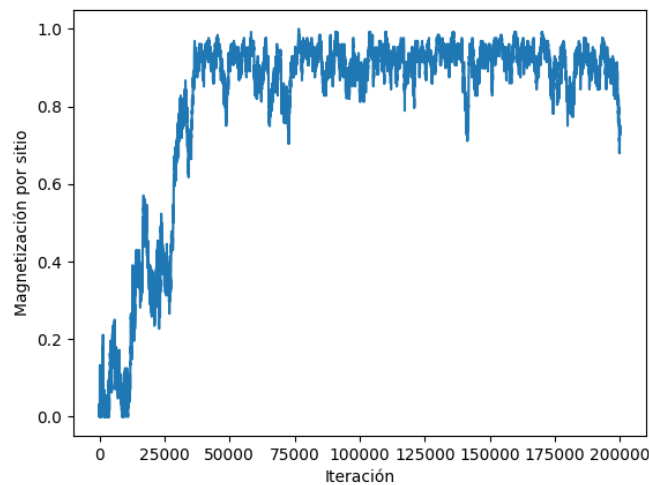
1. Se selecciona un estado inicial al azar o con algún criterio. Este estado se guarda en una matriz  $S$  de  $L \times L$  cuyos elementos son 1 o  $-1$ .
2. Se elige aleatoriamente un sitio de la red. Es decir se eligen coordenadas  $i$  y  $j$  según una distribución uniforme entre 1 y  $L$  (o entre 0 y  $L - 1$ , según como se indexen las matrices en el lenguaje que usen)
3. Se calcula la diferencia de energía  $\Delta E$  que resultaría de invertir el estado del spin en la posición  $(i, j)$ .
4. Si  $\Delta E \leq 0$ , se invierte el spin, es decir que el nuevo estado de la red,  $S'$ , es igual a  $S$  excepto en el sitio  $(i, j)$ , donde  $S'(i, j) = -S(i, j)$ .
5. Si  $\Delta E > 0$ , el spin se invierte con probabilidad  $p = e^{-\beta \Delta E}$ . El nuevo estado de la red es de nuevo  $S'$ , que con probabilidad  $1 - p$  será igual a  $S$ , y con probabilidad  $p$  sera distinto sólo en el valor del sitio  $(i, j)$ .
6. Se repiten los pasos 2-5 con el nuevo estado  $S'$ , tantas veces como sea necesario.

Este algoritmo genera una secuencia de estados  $S_1, S_2, S_3, \dots$ , donde cada estado depende solo del anterior de una manera estocástica (un dado estado  $S_k$  puede dar lugar a muchos estados distintos  $S_{k+1}$  con distintas probabilidades). El punto crucial del algoritmo es que las probabilidades de transición de un estado  $a$  a un estado  $b$  satisfacen la condición:

$$\frac{P(a \rightarrow b)}{P(b \rightarrow a)} = e^{-\beta(E_b - E_a)} = \frac{e^{-\beta E_b} / Z}{e^{-\beta E_a} / Z} = \frac{P_b}{P_a} \quad (5)$$

Esta condición se conoce como 'balance detallado', e implica que el proceso estocástico alcanza un estado estacionario donde la probabilidad de observar el estado  $a$  es  $P_a \propto e^{-\beta E_a}$ . Notar que en ningún paso del algoritmo, ni en la condición de balance detallado, es necesario conocer la constante de normalización de la distribución de probabilidad que se desea muestrear (en este caso, esta constante es la función de partición canónica). Por lo tanto, la condición de balance detallado nos asegura que cuando el proceso de generación de estados alcance un estado estacionario (lo cuál se espera que suceda luego de un número suficiente de interacciones) la probabilidad de cada estado estará dada por el factor de Boltzmann.

*Ejercicio: Tiempo de equilibración.* Estudie como evoluciona la magnetización y la energía del sistema en función del número de iteraciones, comenzando de una configuración aleatoria. ¿Cuántas iteraciones son necesarias para que estas magnitudes alcancen un equilibrio? ¿Depende este tiempo del tamaño del sistema o de la temperatura? Como ejemplo se muestra la magnetización total en función del número de iteraciones para  $L = 16, T = 2, J = 1, B = 0$ .



### 3. Encontrando la transición de fase

En la sección anterior vimos como generar estados representativos del sistema a una dada temperatura. Para cada uno de estos estados podemos calcular su energía, su magnetización, o algún otro observable de interés. De esta forma se puede estudiar numéricamente el comportamiento del sistema a distintas temperaturas. En particular, vamos a estar interesados en el comportamiento crítico del sistema cerca de una transición de fase. Para el modelo de Ising en 2D sin campo externo, una solución exacta fue

obtenida por primera vez por Lars Onsager en 1944. A partir de esta solución sabemos que existe una transición de segundo orden entre una fase ordenada a bajas temperaturas, y una fase desordenada para altas temperaturas. Esta transición ocurre a una temperatura crítica  $T_c$  que satisface  $k_b T_c / J = 2 / \ln(1 + \sqrt{2}) \simeq 2,2691 \dots$  (de aquí en mas se toma  $k_b = 1$ ).

Para hacer una primera caracterización de la transición de fase vamos a considerar cuatro cantidades relevantes:

- Valor medio de la magnetización:  $\langle M \rangle$
- Desvío estandar de la magnetización:  $\sigma_M = \sqrt{\langle M^2 \rangle - \langle M \rangle^2}$
- Valor medio de la energía:  $\langle E \rangle$
- Desvío estandar de la energía:  $\sigma_E = \sqrt{\langle E^2 \rangle - \langle E \rangle^2}$

Por ejemplo, si para una dada temperatura se generaron  $K$  estados (luego de una cantidad de prudente de iteraciones para asegurar que el sistema termalice), y para cada uno de ellos se calculó la magnetización, se tienen  $K$  valores  $M_1, M_2, M_3, \dots, M_K$ , y el valor medio será  $\langle M \rangle = (\sum_{k=1}^K M_k) / K$ .

*Ejercicio: Energía y magnetización en función de la temperatura.* Grafique en función de la temperatura la energía y magnetización media para un sistema con  $L = 16$ . Interprete los resultados.

¿Cómo calcularía el calor específico  $C_V$  y la susceptibilidad magnética  $\chi_M$ ? ¿Cómo están relacionadas estas cantidades con las desviaciones estandar de la energía y la magnetización? Grafique  $C_V$  y  $\chi_M$  en función de la temperatura y compare los resultados con los esperados a partir de la solución exacta y de la teoría de campo medio. Estime el valor de la temperatura crítica para  $L = 16$  y  $L = 32$ .

## 4. Longitud de Correlación

Otra característica distintiva de las transiciones de fase de 2do orden es el desarrollo de correlaciones de largo rango dentro del sistema. Para estudiar este fenómeno consideramos la función:

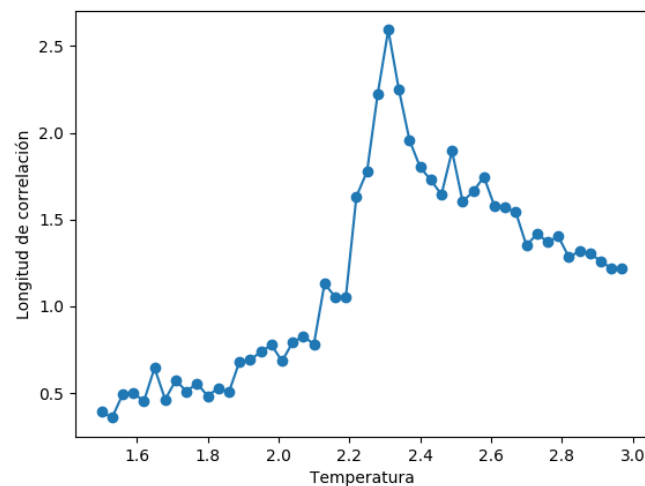
$$C(\bar{r}_1, \bar{r}_2) = \langle S_1 S_2 \rangle - \langle S_1 \rangle \langle S_2 \rangle, \quad (6)$$

que expresa la correlación entre un spin en la posición  $\bar{r}_1$  y otro en la posición  $\bar{r}_2$ . Dada la simetría del sistema, en nuestro caso esta función solo puede depender de la variable  $R = |\bar{r}_1 - \bar{r}_2|$ . Para sistemas que tienen interacciones locales y a altas temperaturas se espera que estas correlaciones decaigan exponencialmente con  $R$ :

$$C(R) \propto e^{-R/\xi}, \quad (7)$$

donde  $\xi$  es la distancia de correlación, que depende en general de la temperatura (en realidad hay otra dependencia en  $1/R$  que vamos a obviar). En el límite termodinámico,  $\xi \rightarrow \infty$  para  $T \rightarrow T_c$ , lo cual indica que en el punto crítico existen correlaciones entre spines sin importar cuan separados estén. Por supuesto, en un sistema finito la longitud de correlación no diverge pero se espera observar un máximo en la temperatura crítica.

*Ejercicio: Longitud de correlación en función de la temperatura.* Grafique la función de correlación entre un spin cualquiera y los otros de la misma fila o columna, para distintas temperaturas. Interprete los resultados. Estime la longitud de correlación para cada caso y grafique en función de la temperatura (estos resultados pueden ser excesivamente ruidosos si no se promedia sobre mucho estados, en la figura se muestra un ejemplo para  $L = 32$ ,  $J = 1$  y  $B = 0$ ).



## 5. Algunos comentarios técnicos

El algoritmo de Metrópolis básico explicado en la sección 2 tiene algunas limitaciones. En primer lugar sucede que los estados que genera no son independientes uno del otro, lo cual es obvio ya que los estados se generan modificando el estado anterior. Sin embargo, estas correlaciones disminuyen mientras más iteraciones haya entre un estado y otro. En general, habrá un número de iteraciones  $I_r$  tales que si dos estados están separados por un número de iteraciones mayor a  $I_r$ , estos pueden ser considerados como estadísticamente independientes. Por lo tanto, una forma directa de generar estados independientes sería solo considerar estados separados cada  $I_r$  iteraciones o más. Esto por supuesto tiene un costo computacional, ya que debemos correr más tiempo el algoritmo para generar un dado número de

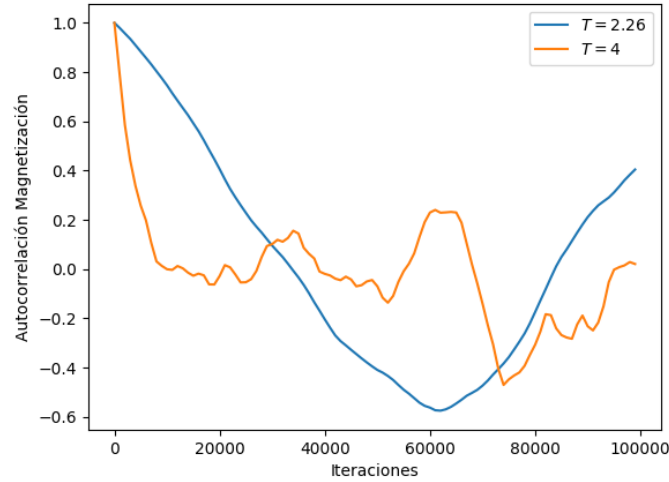


Figura 1: Autocorrelación de la serie temporal de la magnetización para un sistema con  $L = 32$  y dos temperaturas distintas.

estados para tomar datos.

Además, el número  $I_r$  de iteraciones necesarias para ‘olvidar’ las correlaciones depende de la temperatura. Cerca de la transición de fase, esta cantidad se hace muy grande, por lo que el tiempo que debería correrse el algoritmo para generar muestras independientes también se hace muy largo. Para ilustrar esto, en la figura 1 se muestra la función de autocorrelación para la serie de valores de magnetización total  $M$  generados por el algoritmo de Metrópolis para un sistema con  $L = 32$ . Se ve que para temperaturas altas ( $T = 4$ ), la autocorrelación decae a valores que fluctúan en torno a 0 para distancias entre estados mayores a aproximadamente  $I_r \simeq 10^4$  iteraciones. Sin embargo, cerca de la temperatura crítica la función de autocorrelación oscila entre valores positivos y negativos decayendo muy lentamente, y se ve claramente que el valor de  $I_r$  que debería tomarse es mayor a  $10^5$ . Este fenómeno se conoce como ‘critical slowing down’, y existen algunas maneras de evitarlo parcialmente. Una posibilidad es utilizar algoritmos basados en el reconocimiento de grupos de spines o ‘clusters’, que se definen como regiones formadas por spines contiguos con el mismo valor. Estos clusters son tratados entonces como una unidad, y la orientación de todos los spines correspondiente a un dado cluster es invertida de forma conjunta, de una forma similar al algoritmo de Metrópolis original.