Lecture

Metropolis Monte Carlo Algorithm

We saw that the goal is to compute thermodynamic average values, like $\langle E \rangle$, $\langle N \rangle$, $\langle \frac{p^2}{2m} \rangle$, and so on. If we can compute the partition function then useful averages can be obtained by taking partial derivatives of the partition function. If we cannot compute it, we can measure those average values in the lab. If this is hard also, and we still did not give up, we can use a computer. One way to do it is to approximate the partition function with the help of some numerical procedure. This can be also difficult. What is left so us is to simply generate the states j with the probability $\frac{1}{Z} e^{-\beta E_j}$ and then take whatever average value you want.

<u>Goal</u>

Generate a long sequence of states

$$j_1, j_2, j_3, \ldots, j_{100000}$$

with probability $\frac{1}{Z} e^{-\beta E_j}$

Then the average value of a state-dependant quantity, $f(j)$ will be

$$\langle f \rangle = \frac{f(j_1) + \cdots + f(j_{100000})}{100000}$$

<u>Example</u> We have a solid composed of 3 magnetic moments in a magnetic field B.

$B \Uparrow \quad \uparrow\downarrow\downarrow$

We have 2^3 possible states each with its own energy

$\uparrow\uparrow\uparrow$ State ① $E_1 = -3\mu B$

$\uparrow\uparrow\downarrow$ State ② $E_2 = -\mu B$

$\uparrow\downarrow\uparrow$ State ③ $E_3 = -\mu B$

$\downarrow\uparrow\uparrow$ State ④ $E_4 = -\mu B$

$\uparrow\downarrow\downarrow$ State ⑤ $E_5 = \mu B$

$\downarrow\uparrow\downarrow$ State ⑥ $E_6 = \mu B$

$\downarrow\downarrow\uparrow$ State ⑦ $E_7 = \mu B$

$\downarrow\downarrow\downarrow$ State ⑧ $E_8 = 3\mu B$

The probabilities are

① $\frac{1}{Z} e^{+3\beta\mu B}$

②, ③, ④ $\frac{1}{Z} e^{\beta\mu B}$

⑤, ⑥, ⑦ $\frac{1}{Z} e^{-\beta\mu B}$

⑧ $\frac{1}{Z} e^{-3\mu B}$

with $Z = e^{3\beta\mu B} + 3e^{\beta\mu B} + 3e^{-\beta\mu B} + e^{-3\mu B}$

the states' probabilities are NOT EQUAL

$$p_1 = \frac{1}{2} e^{3\beta\mu B} > P_2 = P_3 = P_4 = \frac{1}{2} e^{\beta\mu B} > P_5 = P_6 = P_7 =$$

$$= \frac{1}{2} e^{-\beta\mu B} > P_8 = \frac{1}{2} e^{-3\mu B}$$

As a consequence, a sequence of states must contain a lots of ①'s, less ②'s, ③'s and ④'s and even less ⑧'s. A sequence of states is a mixture of states; the number of appearances of ①'s, and ②'s, and ... ⑧'s must be proportional with the probabilities $p_1, p_2, \ldots, p_8$ of appearance of each state.

①, ⑤, ①, ②, ④, ⑥, ①, ②, ③, ⑧, ①, ①, ①, ⑤, ⑥, ⑧, ①, ②, ①, ④, ①, ④, ②, ③, ①, ⑦, ③, ... and so on

this is a sequence of states

Notice that the number of 1's is higher than the number of 5's, for example ($P_1 > P_5$).

Once we generated a sequence of states we can use it to find any average value. For example, the average value of the <u>number of up spins</u> is (using the above sequence)

$$\langle \text{number of up spins} \rangle =$$

$$= \frac{3 \cdot 9 + 2 \cdot 4 + 2 \cdot 2 + 2 \cdot 3 + 1 \cdot 2 + 1 \cdot 2 + 1 \cdot 1 + 0 \cdot 2}{26}$$

number of 1's in the above sequence of states

number of 2's

#3 #4 #5 #6 #7 #8

number of spins up in the state 1

number of spins up in the state 2

- - -

Conclusion

We need to construct an algorithm capable of generating a sequence of states with the desired probabilities. Metropolis, Rosenbluth, Teller published such an algorithm in 1953.

To find the algorithm, we pretend that we already have a few states generated for us by somebody else

(2) (1) (8)

We need to add the fourth states. In other words we need the probability to have the state (1) or (2) or ... (8) AFTER (8)

(2) (1) (8) $\longrightarrow$ (1) $\qquad T_{8\to 1}$ = Transition probability from state (8) into (1)

(2) (1) (8) $\longrightarrow$ (2) $\qquad T_{8\to 2}$ = Transition probability from state (8) into (2)

⋮

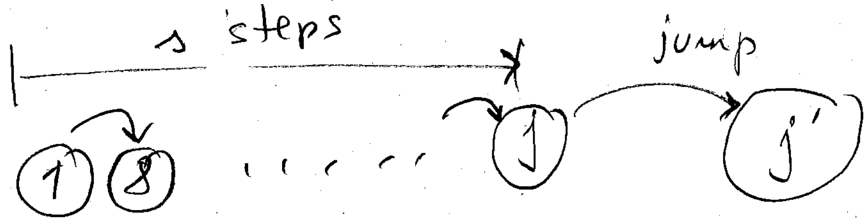
(2) (1) (8) $\longrightarrow$ (8) $\qquad T_{8\to 8}$ = Transition probability from state (8) into (8)

If we know these transition probabilities $T_{8\to 1}, T_{8\to 2}, \ldots, T_{8\to 8}$, we can extend the sequence for example, from (2) (1) (8) to (2) (1) (8) (3) by a random transition that happened to be (3).

In general thus, we need

$$T_{j\to j'}$$ = transition probability from state j into state j'

These transition probability are the unknowns and, as usual, the unknowns are solutions of an equation. So now we are looking for an equation for $T_{j\to j'}$. This is not so difficult to find if we imagine that the process of generating a sequence of states is like a random walk.



Let $P(j,s)$ be the probability to be in the state j after s steps. Then

$$P(j,s+1) = \underbrace{\sum_{j'\neq j} P(j',s)\, T_{j'\to j}}_{\substack{\text{probability to be}\\ \text{in } j' \text{ after } s \text{ steps}\\ \text{and then jump into}\\ j \text{ at the } s+1 \text{ step}}} + \underbrace{\Big(1-\sum_{j'\neq j} T_{j\to j'}\Big) P(j,s)}_{\substack{\text{probability to}\\ \text{be in } j \text{ after}\\ s \text{ steps and}\\ \text{stay in } j \text{ at}\\ \text{the } s+1 \text{ step}}}$$

to stay is equal to 1 minus not to stay

not to stay = jump anywhere = $\sum_{j'\neq j} T_{j\to j'}$

Moving $P_j(s)$ from the right hand side to the left hand side, we get

$$\boxed{P(j, s+1) - P(j, s) = \sum_{j' \neq j}' \left(P(j', s) T_{j' \to j} - T_{j \to j'} P(j, s) \right)}$$

This is the equation we need. It has a name: THE MASTER EQUATION.

Next idea

The probability in the equilibrium statistical physics is time independent. The system is at equilibrium! For our state sequence generating process, the time is the step number. Time independance means s-independant.

$$P(j, s) = P(j)$$

So, the above equation gives

$$0 = \sum_{j' \neq j}' \left(P(j') T_{j' \to j} - T_{j \to j'} P(j) \right)$$

or

$$\boxed{\sum_{j' \neq j}' P(j') T_{j' \to j} = \sum_{j' \neq j}' T_{j \to j'} P(j)} \quad (*)$$

Given $P(j)$ find $T_{j' \to j}$ that solves the equation (*). This equation is the equation that we were looking for $T_{j \to j'}$

Now that we have the equation we need to find a solution. The good news is that we do not need all solutions. One solution will be enough to build an algorithm for (*) upon it. The simplest requirement for (*) to hold is

$$\boxed{P(j')\, T_{j' \to j} = P(j)\, T_{j \to j'}} \qquad (**)$$

The interpretation of (**) is immediate: the probability to be in j and jump into j' equals the probability to be in j' and jump into j. The condition (**) is called the DETAILED BALANCE CONDITION.

Now comes Metropolis and his collaborators who give us a possible solution of (**). The equation (**) has many solutions, like (*); however one solution is again enough for building an algorithm.

Because
$$P(j) = \frac{1}{Z} e^{-\beta E_j}$$
we have from (**)

$$\frac{T_{j \to j'}}{T_{j' \to j}} = \frac{P(j')}{P(j)} = \frac{e^{-\beta E_{j'}}}{e^{-\beta E_j}} = e^{-\beta (E_{j'} - E_j)}$$

Metropolis solution is

$$T_{j \to j'} = \begin{cases} e^{-\beta (E_{j'} - E_j)} & \text{if } E_{j'} > E_j \\ 1 & \text{if } E_{j'} < E_j \end{cases}$$

Now we can state the algorithm, which is on the next page. Notice that on the decision box $E_{j'} < E_j$, the YES answer results in $T_{j \to j'} = 1$, so for sure we accept the jump from j to j'. If the answer to $E_{j'} < E_j$ is NO then we jump into j' with a transition probability $e^{-\beta(E_{j'} - E_j)}$. To implement this situation we need a uniform random number r and a second decision box.

Note that we will reject the first several states since it takes a while for the random walk to reach the stationary distribution.

Metropolis algorithm