The Monte Carlo Method

Background

• *Idea:* A probabilistic method used to calculate quantities that may be too difficult to find analytically, such as mean values $\langle\phi\rangle$ of functions $\phi(s)$ of random variables s with probability distributions $P(s)$, when either P is not known sufficiently explicitly or the sums over all values of s cannot be evaluated. (In statistical mechanics, s is a state α, and $P(s) = \rho(\alpha)$ is usually the canonical distribution $\mathrm{e}^{-\beta E_\alpha}/Z$; an alternative method is Molecular Dynamics, often used for calculations with the microcanonical distribution.) The general idea of the Monte Carlo approach is to perform a numerical simulation in which one generates a large sample of states s_i at random according to $P(s)$, and then infers properties of the ensemble from those of the sample; for example, one can evaluate mean values as averages in that sample of states,

$$\langle\phi\rangle = \sum_s \phi(s)\, P(s) \approx \frac{1}{n}\sum_{i=1}^{n} \phi(s_i)\ .$$

• *Possible situations:* When $P(s)$ is a function of one real variable x in some interval $[a, b]$ or it has an otherwise simple form, for example it depends on several variables but it factorizes into 1-variable functions or it is a uniform distribution on a discrete space, one can directly generate variables with the right distribution.[1] In most cases of interest here, however, one has to use a different method: One starts by generating states with a different, easier distribution to work with, often the uniform one on a discrete space, and then uses some criterion to decide whether to accept or reject each configuration; it is then important to minimize the number of "wasted" ones by using some form of "importance sampling".

• *Straight Monte Carlo:* Generate configurations s uniformly at random, independently of each other, then apply some acceptance criterion. It is used when it is not too computationally intensive to generate a whole new configuration s from scratch every time, and one knows how to compute $P(s)$ (in statistical mechanics applications, this involves knowing Z), but the summation needed to obtain $\langle\phi\rangle$ cannot be evaluated.

• *Simple example:* Evaluating a mean value $\langle x\rangle = \sum_x x\, P(x)$ or an integral $I = \int_a^b f(x)\,\mathrm{d}x$ with this method.

Metropolis Algorithm

• *Idea:* Choose an initial configuration s_0, which in principle can be anything, then start generating small random steps that take the system to new configurations s, in practice setting up a fictitious dynamical system, a random walk. After generating each candidate step, apply an acceptance/rejection criterion adapted to the desired probability distribution to decide whether to keep it or not, then repeat the process. If the step is generated, and acceptance or rejection decided based solely on the configuration at the current time step and the candidate new configuration, then what we get is an example of a Markov process.

• *Markov process:* A stochastic process in which the transition probability $w_{ss'}$ between two states s and s' (the probability that, if the system is in state s at one time, it goes to s' in the next time step) only depends on the states involved.

• *Master equation:* Given the $w_{ss'}$s, the set of probabilities $P_s(t)$ that the system is in state s at time t must satisfy, by consistency, the master equation

$$\dot{P}_s = \sum\nolimits_{s'\neq s} (-w_{ss'}\, P_s + w_{s's}\, P_{s'})\ .$$

A system undergoing a Markov process is in equilibrium if $\dot{P}_\nu = 0$; in particular, it is sufficient that detailed balance, the condition $w_{ss'}\, P_s = w_{s's}\, P_{s'}$, be satisfied $\forall s, s'$ (this is a condition on both the system and the state).

---

[1] In the $P(x)$ case, generate a value y uniformly at random in [0,1], then calculate the value of x as $F^{-1}(y)$, where $F(x) := \int_a^x \mathrm{d}x\, P(x)$; since y has unit probability density $\mu_y(y) = 1$, the resulting probability density for x is $\mu_x(x) = |\mathrm{d}y/\mathrm{d}x|\, \mu_y(y(x)) = P(x)$, as desired.

• *Procedure:* Suppose you want to generate states s with a known probability distribution $P(s)$. Before you start, decide how each step will be generated, making sure that one can reach any point of configuration space in a finite number of steps. Then, the procedure is as follows:

⋆ Generate the first s_0 (in any way, random or not), and set $s = s_0$;
⋆ Calculate $P(s)$;
⋆ Choose a possible step $s \mapsto s'$;
⋆ Calculate $P(s')$;
⋆ If $P(s')/P(s) > 1$, accept the change, set $s = s'$ and repeat;
⋆ If $P(s')/P(s) < 1$, accept it with probability $P(s')/P(s)$.

• *Why this works:* Based on the procedure above,

$$w_{ss'} = \begin{cases} 1 & \text{if } P(s') \geq P(s) \\ P(s')/P(s) & \text{if } P(s') < P(s) \end{cases} , \qquad w_{s's} = \begin{cases} P(s)/P(s') & \text{if } P(s') \geq P(s) \\ 1 & \text{if } P(s') < P(s) \end{cases} .$$

In either case, we get that $w_{ss'}/w_{s's} = P(s')/P(s)$, which means that the detailed balance condition is satisfied. Thus, we have defined a stationary process with probabilities equal to $P_s = P(s)$, for all s.

• *Comments:* The advantage is a more efficient generation of configurations. The disadvantage is that each configuration is strongly correlated to the previous one, and it may take a large number of iterations to generate a significantly different one.

• *Simple example:* Evaluate the same mean value $\langle x \rangle = \sum_x x\, P(x)$ or integral $I = \int_a^b f(x)\, \mathrm{d}x$ as with the "straight" Monte Carlo example, but now using the Metropolis algorithm.

Main Example: The Case of a Canonical Ensemble

• *General idea:* If we want to calculate the mean value of some physical quantity $\phi(\alpha)$ in a thermal state at temperature T, the quantity of interest for the Metropolis acceptance criterion, $P_{s'}/P_s$, becomes

$$\rho(\alpha')/\rho(\alpha) = \mathrm{e}^{-\beta(E'_\alpha - E_\alpha)} ,$$

and we don't need to know Z to calculate it! The Metropolis algorithm is therefore very convenient to use. (Notice though that the Markov process used in this algorithm is not an actual time evolution of the system!)

• *Procedure:* With the identifications above, the steps are:

⋆ Generate the first state α_0 (in any way, random or not);
⋆ Set $\alpha = \alpha_0$;
⋆ Calculate $E(\alpha)$;
⋆ Choose a possible step $\alpha \mapsto \alpha'$;
⋆ Calculate $E(\alpha')$;
⋆ If $\Delta E = E_{\alpha'} - E_\alpha < 0$, accept the change, set $\alpha = \alpha'$ and repeat;
⋆ If $\Delta E := E_{\alpha'} - E_\alpha > 0$, accept it with probability $p = \mathrm{e}^{-\beta \Delta E}$.

• *Next:* Apply this procedure to the simulation of the 2-dimensional Ising model.

Comments

• *Statistical uncertainties:* Becaue each simulation is based on a finite sample, one must take into account the fact that there are statistical uncertainties σ_{MC} in the results, and $\langle A \rangle = \langle A \rangle_{\mathrm{MC}} \pm \sigma_{\mathrm{MC}}$.

• *Simulated Annealing:* One application of the Metropolis idea is a method to find the minimum of a function, such as the lowest-energy state of a system. Consider the variables the function depends on as canonical variables for a physical system, and the function to minimize as its energy. Run a Metropolis-type Monte Carlo simulation with the temperature initially set to a relatively high value, then lower it.

Reading

• *Pathria & Beale:* Chapter 16 (includes the Monte Carlo and Molecular Dynamics methods).
• *Other detailed discussions:* Chandler, Ch 6; Huang, Ch 12; Plischke & Bergersen, Ch 9.
• *Shorter treatments / mentions:* Also mentioned in Halley, pp 158–159.