Physics 127c: Statistical Mechanics

Monte Carlo Methods

Monte Carlo Integration

Monte Carlo is most basically a way of doing integrals or sums.

Consider first a one dimensional integral

$$I = \int_0^1 f(x)dx = \langle f(x) \rangle$$

where we have added the suggestive $\langle\rangle$ meaning the average over a uniform distribution of x in the interval $0 \le x \le 1$. We will evaluate the integral on a computer as the discrete sum

$$I \approx \frac{1}{M} \sum_{i=1}^{M} f(x_i).$$

We can imagine two simple methods:

Method 1: take equally spaced x_i with separation $h = 1/M$;

Method2: generate M random x_i from a uniform distribution.

In the latter case since $f(x_i)$ is a random variable, the central limit theorem tells us that I is a Gaussian random variable for large M with variance

$$\sigma_I^2 = \frac{1}{M}\sigma_f^2$$

so that the error in I goes down as $1/\sqrt{M}$ and is smaller if the variance σ_f^2 of f is smaller.

For a one dimensional integration the Monte Carlo method is not compelling. However consider a d dimensional integral evaluated with M points. For a uniform mesh each dimension of the integral gets $M^{1/d}$ points, so that the separation is $h = M^{-1/d}$. The error in the integration over one h^d cube is of order h^{d+2}, since we are approximating the surface by a linear interpolation (a plane) with an $O(h^2)$ error. The total error in the integral is $Mh^{d+2} = M^{-2/d}$. The error in the Monte Carlo method remains $M^{-1/2}$, so that this method wins for $d > 4$.

We can reduce the error in I by reducing the effective σ_f. This is done by concentrating the sampling where $f(x)$ is large, using a weight function $w(x)$ (i.e. $w(x) > 0$, $\int_0^1 w(x) = 1$)

$$I = \int_0^1 \frac{f(x)}{w(x)} w(x)\, dx \Rightarrow \frac{1}{M} \sum_{i=1}^{M} \frac{f(x_i)}{w(x_i)}$$

where in the last sum we *generate the x_i with distribution $w(x)$*—known as *importance sampling*. How do we generate the x_i? We can try by introducing the auxiliary variable y defined by

$$\frac{dy}{dx} = w(x), \;\; y(x = 0) = 0,$$

and then $y(x = 1) = 1$ follows by integrating. From this find $x(y)$. Now generate a set of y_i with uniform distribution $\bar{P}(y) = 1$. Then since

$$\bar{P}(y)dy = P(x)dx$$

the set $x_i = x(y_i)$ are distributed with $w(x)$.

Unfortunately this simple scheme is not always possible, e.g. for the common case of a Gaussian $w(x) \propto \exp[-x^2/2\sigma^2]$. In this particular case a cute trick can be used, which is worth mentioning since Gaussian distributions are so common. Generate a 2d distribution $P(x_1, x_2) \propto \exp[-(x_1^2 + x_2^2)/2\sigma^2]$. In polar coordinates this becomes

$$P(r, \theta) dr d\theta \propto e^{-r^2/2\sigma^2} r \, dr \, d\theta$$

or with $u = r^2/2\sigma^2$

$$P(u, \theta) \propto e^{-u} du d\theta.$$

Now $dy/du = e^{-u}$ can be solved to give $u = -\ln[(1-y)]$. So if we generate y_i with a uniform distribution on the interval $[0, 1]$ and evaluate $r_i = \sigma\sqrt{-2\ln(1-y_i)}$, and θ_i with a uniform distribution on the interval $[0, 2\pi]$, the variables $x_1 = r\cos\theta$ and $x_2 = r\sin\theta$ have a Gaussian distributions (and both numbers can be used to generate two successive entries in the list of x_i).

Monte Carlo in Classical Statistical Mechanics

Classical statistical mechanics is just a very big sum! For example, canonical averages are

$$\langle A \rangle = \frac{\sum_{\text{states}} e^{-\beta H} A}{\sum_{\text{states}} e^{-\beta H}}.$$

We cannot enumerate *all* the states (e.g. 2^N states for an Ising spin system) for any N large enough to be interesting. Monte Carlo methods instead generate a subset of microstates $\vec{x}_l$ with probability distribution $P(\vec{x}_l)$, and estimate averages

$$\langle A \rangle \approx \frac{\sum_{l=1}^{M} e^{-\beta H(\vec{x}_l)} A(\vec{x}_l)/P(\vec{x}_l)}{\sum_{l=1}^{M} e^{-\beta H(\vec{x}_l)}/P(\vec{x}_l)}.$$

(For concreteness, think of $\vec{x}_l$ as being a particular configuration of N spins in the Ising model, for example.) It is natural to choose $P(\vec{x}_l)$ to be $P_{eq}(\vec{x}_l) = Z^{-1} e^{-\beta H(\vec{x}_l)}$ so that

$$\langle A \rangle \approx \frac{1}{M} \sum_{l=1}^{M} A(\vec{x}_l).$$

But how do we generate the $\vec{x}_l$?

In the *Metropolis method* the $\vec{x}_l$ are generated as a Markov process, with $\vec{x}_{l+1}$ generated from $\vec{x}_l$ via a suitably chosen transition probability $W(\vec{x}_l \to \vec{x}_{l+1})$. A sufficient condition to guarantee that P_{eq} will be maintained by this process is to impose the *principle of detailed balance* as a constraint on the W for every pair of states $\vec{x}_r$, $\vec{x}_s$

$$P_{eq}(\vec{x}_r) W(\vec{x}_r \to \vec{x}_s) = P_{eq}(\vec{x}_s) W(\vec{x}_s \to \vec{x}_r),$$

i.e. the condition that the equilibrium distribution ratio between $\vec{x}_r$ and $\vec{x}_s$ is maintained by direct transitions between these states. Note that it is not necessary to impose this condition: other constraints can be found involving transitions to other states that will maintain P_{eq}. Note also that we are saying nothing about how the physical system maintains P_{eq}: this is a question of the dynamics of the system, which is not needed to evaluate classical canonical averages.

A common algorithm is to choose

$$W(\vec{x}_r \to \vec{x}_s) \propto \begin{cases} e^{-\beta \delta H} & \text{if } \delta H > 0 \\ 1 & \text{if } \delta H \geq 0 \end{cases}$$

with $\delta H = H(\vec{x}_s) - H(\vec{x}_r)$, then we have

$$\frac{W(\vec{x}_r \to \vec{x}_s)}{W(\vec{x}_s \to \vec{x}_r)} = \left\{ \begin{array}{ll} e^{-\beta\delta H} & \text{for } H(\vec{x}_s) > H(\vec{x}_r) \\ \frac{1}{e^{-\beta(-\delta H)}} & \text{for } H(\vec{x}_s) < H(\vec{x}_r) \end{array} \right\} = e^{-\beta[H(\vec{x}_s) - H(\vec{x}_r)]}$$

with the final result independent of whether δH is positive or negative.

We also must show that the $\vec{x}_l \ldots$ does indeed converge to the distribution P_{eq}. To do this consider an ensemble of the Markoff process or chains (i.e. many repetitions of the Monte Carlo scheme). Suppose at a given step there are N_r chains in the state r and N_s chains in the state s where $H(\vec{x}_r) < H(\vec{x}_s)$. We have some way of generating transitions between r and s. If we first ignore $\delta H = H(\vec{x}_s) - H(\vec{x}_r)$, the rates $r \to s$ and $s \to r$ must be equal (call this $W^{(0)}_{rs} = W^{(0)}_{sr}$). Then

$$\begin{aligned} W(\vec{x}_r \to \vec{x}_s) &= W^{(0)}_{rs} e^{-\beta\delta H}, \\ W(\vec{x}_s \to \vec{x}_r) &= W^{(0)}_{rs}. \end{aligned}$$

The number of transition at this step is

$$\begin{aligned} N_{r\to s} &= N_r W^{(0)}_{rs} e^{-\beta\delta H}, \\ N_{s\to r} &= N_s W^{(0)}_{rs}, \end{aligned}$$

so that the net number of transition is

$$\Delta N_{r\to s} = N_{r\to s} - N_{s\to r} = N_r W^{(0)}_{rs} \left(\frac{e^{-\beta H(\vec{x}_s)}}{e^{-\beta H(\vec{x}_r)}} - \frac{N_s}{N_r} \right).$$

Clearly this expression is zero for the equilibrium distribution, and also we see that if N_s/N_r is *too small*, the sign is such that the ratio is *increased*, and vice versa.

A simple example will clarify the discussion. Consider the Ising model with nearest neighbor interactions $H = -\frac{1}{2} J \sum_{i,\delta} s_i s_{i+\delta}$. The Monte Carlo procedure is:

Step 1: Flip a single, randomly chosen spin (so $W^{(0)}_{rs} = W^{(0)}_{sr}$). Call this spin i.

Step 2: Calculate $\delta H = -\frac{1}{2} J \sum_{\delta=1}^{2d} s_{\iota+\delta}(\pm 2)$ with the $+$ sign if the ith spin is flipped up, and the $-$ sign if the spin is flipped down.

Step 3: Keep the new configuration according to the Metropolis algorithm, i.e. if $\delta H < 0$ keep the new state, whereas if $\delta H > 0$, draw a random number z uniformly distributed on $[0, 1]$ and keep the new state if $z < e^{-\beta\delta H}$, otherwise keep the old state.

This process generates a sequence of states that can be used (after a number of iterations to allow convergence to the estimate of P_{eq}) to calculate canonical averages. Alternatively to step 1, we could sweep through the lattice systematically, flipping each spin in turn to be tested by the Metropolis algorithm.

A number of caveats should be considered:

1. Successive configurations will not be statistically independent. This is no problem calculating the mean (each iteration is an unbiased sample) but the error in the estimate is *not* given by $\sigma/\sqrt{N_s}$ with σ^2 the measured variance and N_s the number of samples.

2. The random number generator better be good—there are unfortunate examples in the literature where incorrect answers were generated because the "random" numbers in fact had subtle correlations.

3. The single spin updates become very inefficient at low temperatures if low energy transitions that involve many (correlated) spin flips are important, such as the cluster flips we found in the 1d Ising model, or long wavelength spin-wave type distortions in other models. This is because $e^{-\beta\delta H}$ for a single spin flip will almost always be very small, so that the "dynamics" freezes. A more sophisticated way of generating possible updates is needed.

4. Feasible system sizes are limited in practice. Even with modern computers a $10^2 \times 10^2 \times 10^2$ would be considered large, and finite size corrections may be quite important—note that 5% of the particles are on a surface in this system of 10^6 spins!

Further Reading

A good reference for this section is chapter 8 of *Computational Physics* by S. Koonin. A more advanced reference is *Monte Carlo Simulation in Statistical Physics* by Binder and Heerman. There are numerous Java implementations of the Metropolis algorithm on the 2d Ising model on the internet—do a Google search. Two examples are

http://threeplusone.com/code/ising.html
http://stp.clarku.edu/simulations/ising2d/