Mathematics 13: Monte Carlo: The Metropolis Algorithm

The Monte-Carlo technique is a method of performing a 'high dimensional' integration using random numbers and statistical analysis.

A very simple example of such a calculation is an estimation of π via:

$$\pi = 4 \int_0^1 dx \int_0^1 dy \theta \left[1 - x^2 - y^2\right]$$

We obtain a sequence of random numbers, uniformly distributed between zero and unity. We group the numbers into pairs which we interpret as points (x, y) in space. We then estimate π by:

$$\pi \sim \frac{4N_0}{N_{total}}$$

where N_{total} is the total number of pairs and N_0 is the number of such pairs within unit distance of the origin. It is crucial to understand what to expect from such a calculation.

We are analysing a particular case of a *binomial* distribution. Each 'point' has a probability, p say, of being accepted. The number of accepted states is controlled by:

$$[1 - p + px]^N = \sum_{m=0}^{N} \frac{N!}{m!(N-m)!}(1-p)^{N-m}(xp)^m = \sum_{m=0}^{N} a_m x^m$$

where m counts the number of acceptances and N is the number of attempts. In the limit that N becomes large we can employ *Stirling's Formula* for the factorials to yield:

$$a_m \sim \frac{N^{N+\frac{1}{2}}}{\sqrt{(2\pi)}\,[N-m]^{N-m+\frac{1}{2}}\, m^{m+\frac{1}{2}}} \times (1-p)^{N-m} p^m$$

which is best understood by analysing the exponent:

$$a_m \sim \frac{N^{N+\frac{1}{2}}}{\sqrt{(2\pi p[1-p])}} e^{-f(m)}$$

where:

$$f(m) = \left(N - m + \frac{1}{2}\right) \ln \frac{N-m}{1-p} + \left(m + \frac{1}{2}\right) \ln \frac{m}{p}$$

with derivatives:

$$\frac{df}{dm} = \ln \frac{m(1-p)}{(N-m)p} + \frac{1}{2}\left[\frac{1}{m} - \frac{1}{N-m}\right]$$

$$\frac{d^2 f}{dm^2} = \frac{1}{m} + \frac{1}{N-m} - \frac{1}{2}\left[\frac{1}{m^2} + \frac{1}{(N-m)^2}\right]$$

The distribution is dominated by the region where $f(m)$ is maximal, and near this point:

$$f(m) \sim (N+1)\ln N + \frac{1}{2}(m - Np)^2 \frac{1}{Np(1-p)} + \ldots$$

and so as N becomes large, so:

$$a_m \sim \frac{\exp\left[-\frac{1}{2}\frac{(m-NP)^2}{Np(1-p)}\right]}{\sqrt{[2\pi Np(1-p)]}}$$

a *Gaussian* distribution. This leads to:

$$< \frac{m}{N} >= p$$

and:

$$< \left(\frac{m}{N} - p\right)^2 >= \frac{p(1-p)}{N}$$

and the error $\sim \sqrt{\left[\frac{p(1-p)}{N}\right]}$. There are two important aspects to this error. Firstly, it is largest when $p = \frac{1}{2}$ and acceptance rate is about equal to rejection rate. Secondly, the error $\sim N^{-1/2}$ and so to get an error of ϵ we need $N \sim \epsilon^{-2}$ which is a *very* large number for appreciable accuracy: 10^{-6} requires 10^{12} calculations!

Note that:

$$\pi = \frac{2}{\prod_{n=1}^{\infty} x_n} = \frac{2}{x_1 x_2 x_3 ...}$$

with:

$$x_{n+1} = \sqrt{\left[\frac{1+x_n}{2}\right]} \qquad x_0 = 0$$

which gives a reduction in error by about $1/4$ every iteration. Statistical methods are usually a 'last-ditch' technique.

For our problem, we can generalise:

$$\frac{\pi r^2}{4} = \int_0^1 dx \int_0^1 dy \theta\left[r^2 - x^2 - y^2\right] \equiv p$$

and if we ensure that $p \sim \frac{1}{2}$ by a careful choice of r^2, then we can improve the accuracy in principle. For example, we can use $r^2 = \frac{2}{3}$ and then $p = \frac{\pi}{6}$ which is quite close to a half. The approximation is then

$$\pi \sim \frac{6N_0}{N_{total}}$$

with N_0 being the number of points within $\sqrt{(2/3)}$ distance from the origin. This is likely to be a better approximation.

The area that we intend to investigate using Monte-Carlo is *statistical physics*. You should be aware that expectation values for operators are controlled by:

$$< \hat{O} >= Tr\left(\hat{O}e^{-\beta\hat{H}}\right) Z(\beta)^{-1}$$

$$Z(\beta) = Tr\left(e^{-\beta\hat{H}}\right)$$

and for the case where we have control over an eigenbasis this reduces to:

$$< \hat{O} >= \sum_n < n \mid \hat{O} \mid n > e^{-\beta \epsilon_n} Z(\beta)^{-1}$$

$$Z(\beta) = \sum_n e^{-\beta \epsilon_n}$$

The 'obvious' way to proceed would be to evaluate the sums over eigenstates directly. Unfortunately, for the systems of interest to us, with many identical atoms, N say, each with a finite number of degrees of freedom, m say (for spin-half $m = 2$), the total number of degrees of freedom is m^N, which is immensely large for quite small systems. Even for a system of 10×10 atoms we are involved with $2^{100} \sim 10^{30}$ calculations! Many too many to contemplate.

The next-most 'obvious' method is to try to apply the randomness idea directly to the problem: Approximate the sums by randomly chosen states. Unfortunately, this idea is also not very effective, because most of the states are exponentially irrelevant due to their negligible Boltzmann weights. Most of such a random calculation would consequently be insignificant.

One way to make 'maximal' use of the calculations is via the *Metropolis algorithm*. This algorithm provides a sequence of states which have a probability distribution identical to the Boltzmann distribution, in the limit that the number of states becomes large. For such a sequence of states the expectation values reduce to:

$$< \hat{O} >\sim \frac{1}{N} \sum_{n=1}^{N} < n \mid \hat{O} \mid n >$$

which should converge to the exact answer as $N \mapsto \infty$. The main point is that all chosen states are of equal weight in the sum, and all of the calculations have 'equal' significance.

The metropolis algorithm itself involves choosing states via a sequence of *random* minor alterations combined with a simple acceptance/rejection procedure: The change in energy at the alteration, ΔE, is evaluated and the change is accepted provided that either $\Delta E < 0$, or if $\Delta E > 0$ then the state is only accepted with a probability $e^{-\beta \Delta E}$. This yields a sequence of states with a Boltzmann distribution, as we will now show.

We consider a collection of states with a label n say, and a process that maps each particular state onto the others with a certain probability. These probabilities we denote by $P_{m,n}$, which is the transition probability for state m to transit to state n. Now a transition is required at each step and so:

$$\sum_n P_{m,n} = 1$$

ensures that we end up somewhere. We do not exclude a transition to the same state via $P_{n,n} \neq 0$. If we have converged to equilibrium in such a procedure, then we may set

P_n to be the equilibrium probability of finding the state n. In equilibrium a transition must leave these probabilities invariant, so:

$$P_n = \sum_m P_m P_{m,n}$$

which just adds up all the possible ways that we can arrive at the state n. Using the previous fact that we must end up somewhere, we deduce that:

$$\sum_m \left(P_m P_{m,n} - P_n P_{n,m} \right) = 0$$

is equivalent to equilibrium. For our Boltzmann distribution, we are provided with the relative probabilities, $P_n = e^{-\beta \epsilon_n}$, and we wish to *choose* the transition probabilities to obtain this equilibrium. It is clear that:

$$P_{m,n} = e^{-\beta(\epsilon_n - \epsilon_m)} P_{n,m}$$

for all m and n will generate this distribution. Finally, it is clear that $\epsilon_n > \epsilon_m$ implies $P_m > P_n$ and we must always have a higher probability of accepting the lower-energy state. The highest acceptance rate is achieved by *always* accepting the lower energy state and using this 'detailed balance' equation to provide the probability for accepting the transition to the higher energy state. This choice is precisely the metropolis algorithm.