More on random numbers and the Metropolis algorithm

We examined a number of approaches to generate random numbers sampled from different distributions. The basic distribution that we usually start with is of random numbers uniformly distributed between zero and one. We wish the change this distribution to something else. For example we talked on the use of the Central Limit Theorem to obtain “Gaussian” random numbers. Here is quick review of that argument: Let $X_i\, i = 1,...,n$ be random numbers sampled from $n$ (potentially different) distribution functions for which the first and the second moments of the random variable exist, i.e., the average of $X_i$, $\langle X_i \rangle$ and the average of its square, $X_i^2$, $\langle X_i^2 \rangle$ are finite.

For example, consider the probability density $P(x) = \lambda \exp(-\lambda x) \quad x > 0$, It is normalized such that $\int_0^\infty p(x)\, dx = 1$, the first moment is

$$\langle x \rangle = \int_0^\infty x (\lambda \exp(-\lambda x))\, dx = -\lambda \frac{\partial}{\partial \lambda} \int_0^\infty \exp(-\lambda x)\, dx = -\lambda \frac{\partial}{\partial \lambda} \left(\frac{1}{\lambda}\right) = \lambda^{-1}$$

and the second

$$\langle x^2 \rangle = \int_0^\infty x^2 (\lambda \exp(-\lambda x))\, dx = \lambda \frac{\partial^2}{\partial \lambda^2} \left(\frac{1}{\lambda}\right) = \frac{2}{\lambda^2}$$

Both moments exist and hence the central limit theorem is applicable for random number sampled from the above distribution.

Note that for the probability density $P(x) = \frac{a}{1+x^2}$ \quad $-\infty < x < \infty$ a second moment does not exist.

Using the central limit theorem we define a new random number, $Y$

$$Y = \sum_{i=1}^n X_i, \quad \text{at the limit of } n \to \infty \text{ we have } P(Y) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left[ -\frac{(Y-\langle Y \rangle)^2}{2\sigma^2} \right]$$

where

$$\sigma = \sqrt{\langle Y^2 \rangle - \langle Y \rangle^2}$$

In practice the $X_i$ are sampled from the same uniform distribution between 0 and 1 that is the fastest to create on the computer. In that case $\langle Y \rangle = n/2$ and $\langle Y^2 \rangle = n/3$. Why? In practical application $n$ is always set below 100.

Yet another way of generating one distribution using sampling from another probability density is the reject/accept procedure (the Metropolis algorithm is a variant of this algorithm).
We wish to sample from a probability density, \( p(x) \), between 0 and \( a \). We know how to generate uniform random numbers between 0 and \( a \) (just call the usual computer random number generator and multiply the result by \( a \)).

The reject/accept algorithm is

1. Select a random number from a uniform distribution between zero and one, say \( x_i \).
2. Compute \( p(x_i) \).
3. Select a second uniform random number, say \( q_i \), between zero and the maximal value of \( p(x) \).
4. If \( q_i > p(x_i) \) reject \( x_i \), otherwise accept the step.
5. Return to 1.

We call the sequence of configurations so generated a chain.

The Metropolis algorithm is very similar. It generates configuration from the distribution \( p(x) \propto \exp\left(-\frac{U(x)}{T}\right) \equiv w(x) \). Direct application of the reject/accept algorithm for that distribution will be inefficient, since we will have many rejections near the end of the interval (the exponential function decays rapidly). We are also unsure about the maximum of the probability density since this depends on the minimal value of the potential energy \( U(x) \) (which is actually what we are looking for when performing a minimization). So here is the Metropolis adjustment for configuration sampling with respect to the reject/accept procedure.

0. Initiate the process selecting a coordinate set, \( x_0 \). Compute \( U(x_0) \) and \( w(x_0) \).
1. Select a random number, \( x_i \), appropriate for coordinate displacement.
2. Compute \( w(x_i) \).
3. Select a uniform random number \( q_i \) between zero and one.
4. If \( q_i > w(x_i)/w(x_{i-1}) \) reject \( x_i \), otherwise accept the step.
5. Return to 1.

What is the probability density created by the Metropolis algorithm? Assume that we are running in parallel a large number of chains of structures, using the procedure described above. Assume further that the process reaches equilibrium (a non trivial assumption, that nevertheless seems to hold for a great number of examples). By equilibrium we mean that probability density estimated from the large number of parallel chains, is not changing from set \( i \) step to step \( i+1 \).

More formally, let \( x_{ij} \) be a configuration sampled for the \( j \) chain at the \( i \) step. Histogramming over all chains at a specific step \( i \) means estimating the probability density \( p_i(x) \). Equilibrium is assumed when \( p_i(x) = p_{i+1}(x) \). Consider for example transitions between \( x_k \) and \( x_k^* \) during step \( i \) and \( i+1 \). Let \( n_i(x_k) \) be the number density
of chains that are at $x_k$ in step $i$. Similarly $n_{i+1}(x_k)$ is the number density of chains at $x_k$ in step $i+1$. One-way of reaching equilibrium is when the number of transitions from $x_k$ to $x_{k'}$ is equal to the number of transitions from $x_{k'}$ to $x_k$. Or using the Metropolis criterion

$$n_i(x_k) \cdot \left( \min \left[ w(x_{k'}) / w(x_k), 1 \right] \right) = n_i(x_{k'}) \cdot \left( \min \left[ w(x_k) / w(x_{k'}), 1 \right] \right) \quad \text{(at equilibrium)}$$

Therefore, at equilibrium

$$\frac{n_{eq}(x_k)}{n_{eq}(x_{k'})} = \exp \left[ \frac{-U(x_k) - U(x_{k'})}{T} \right]$$

The exponential distribution emphasizes lower potential energies and is therefore useful for the detection of low minima. Of course, the sampling depends also on the value of the “temperature” $T$. The lower is the temperature the more emphasis is made on low-lying energy values. Hence, it seems reasonable to generate equilibrium distributions at ever-lower temperatures to recover the global minimum.

The problem is that the system equilibrates very slowly at low temperatures and therefore it is useful to use the temperature as a parameter and to cool the system slowly. Better sampling will be obtained at higher temperature, and as the temperature decreases some guidance to low energy minima should be obtained.