Last month’s posting introduced the Metropolis-Hastings (MH) method for generating a sample distribution from just about any functional form. This method can in handy particularly for methods that are analytically intractable so that the inverse-transform method would be useless and for which the accept-reject method would be too inefficient.

The Metropolis-Hastings algorithm, which is particular instance of the broader Markov Chain Monte Carlo (MCMC) methods, brings us the desired flexibility at the expense of being computationally more involved and expensive. As a reminder, the algorithm works as follows:

* Setup
  + Select the distribution $f(x)$ from which a set random numbers are desired
  + Arbitrarily assign a value to a random realization $x\_{start}$ of the desired distribution consistent with its range
  + Select a simple distribution $g(x|x\_{cur})$, centered on the current value of the realization from which to draw candidate changes
* Execution
  + Draw a change from $g(x|x\_{cur})$
  + Create a candidate new realization $x\_{cand}$
  + Form the ratio $R\_f = f(x\_{cand})/f(x\_{cur})$
  + Make a decision as the keeping or rejecting $x\_{cand}$ as the new state to transition to according to the Metropolis-Hastings rule
    - If $R\_f \geq 1$ then keep the change and let $x\_{cur} = x\_{cand}$
    - If $R\_f < 1$ then draw a random number $q$ from the unit uniform distribution and make the transition if $q < R\_f$

and the explicit promise of this month’s installment was the explanation as to why this set of rules actually works.

Technically speaking, the algorithm, as written above, is strictly the Metropolis-Hastings algorithm for the case where the simple distribution $g(x|x\_{cur})$ is symmetric; the modification for the case of non-symmetric distributions will emerge naturally in the course of justification for the rules.

The arguments presented here closely follow the excellent online presentation by ritvikmath:

<iframe width="560" height="315" src="https://www.youtube.com/embed/yCv2N7wGDCw" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

To start, we relate our distribution $f(x)$ to the corresponding probability by

\[ p(x) = \frac{f(x)}{N\_C} \; , \]

where $N\_C$ is the normalization constant that is usually impossible to calculate analytically either because the functional form of $f(x)$ possess no analytic anti-derivative or the distribution is defined in high dimensions or both.

The MH algorithm pairs $p(x)$ with a known probability distribution $g(x)$ but in a way such that the algorithm ‘learns’ where to concentrate its effort by having $g(x)$ follow the current state around (in contrast to the accept-reject method where $g(x)$ is stationary). At each step, the transition from the current state $a$ to the proposed or candidate next state $b$ happens according to the transition probability

\[ T(a \rightarrow b) = g(b|a) A(a \rightarrow b) \; , \]

where $g(b|a)$ is conditional draw from $g(x)$ centered on state $a$ (i.e., $g$ follows the current state around) and the transition probability $A(a \rightarrow b)$ is the MH decision step. The right-hand-side of the transition probability directly maps $g(b|a)$ and $A(a \rightarrow b)$ to the first two and the last two sub-bullets of the execution step above, respectively.

To derive the MH decision step correctly describes $A(a \rightarrow b)$, we impose detailed balance,

\[ p(a) T(a \rightarrow b) = p(b) T(b \rightarrow a) \; \]

guaranteeing that the MCMC will eventually result in a stationary distribution. That detailed balance give stationary distributions of the Markov chain is most easily seen by summing both sides over $a$ to arrive at a matrix analog of the stationary requirement ${\bar S}^\* = M {\bar S}^\*$ used in the [Sunny Day Markov](http://aristotle2digital.blogwyrm.com/?p=1455) chain post.

The next step involve a straightforward plugging-in of earlier relations to yield

\[ \frac{f(a)}{N\_C} g(b|a) A(a \rightarrow b) = \frac{f(b)}{N\_C} g(a|b) A(b \rightarrow a) \; . \]

Collecting all of the $A$’s over on one side gives

\[ \frac{A(a \rightarrow b)}{A(b \rightarrow a)} = \frac{f(b)}{f(a)} \frac{g(a|b)}{g(b|a)} \; . \]

Next define, for convenience, the two ratios

\[ R\_f = \frac{f(b)}{f(a)} \; \]

and

\[ R\_g = \frac{ g(a|b) }{ g(b|a) } \; . \]

Note that $R\_f$ is the ratio defined earlier and that $R\_g$ is exactly 1 when the $g(x)$ is a symmetric distribution.

Now, we analyze two separate cases to make sure that the transitions $A(a \rightarrow b)$ and $A(b \rightarrow a)$ can actually satisfy detailed-balance.

First, if $R\_f R\_g < 1$ then detailed balance will hold if we assign $A(a \rightarrow b) = R\_f R\_g$ and $A(b \rightarrow a) = 1$. These conditions mean that we can interpret $A(a \rightarrow b)$ as a probability for making the transition and we can then draw a number $q$ from a uniform distribution to see if we beat the odds $q \leq R\_f R\_g$, in which case we transition to the state $b$.

Second, if $R\_f R\_g \geq 1$ then detailed balance will hold is we assign $A(a \rightarrow b) = 1$ and $A(b \rightarrow a) = R\_f R\_g$. These conditions mean that we can interpret $A(a \rightarrow b)$ is being a demand, with 100% certainty, that we should make the change.

These two cases, which comprise the MH decision rule, are often summarized as

\[ A(a \rightarrow b) = Min(1,R\_f R\_g) \; \]

or, in the case when $g(x)$ is symmetric

\[ A(a \rightarrow b) = Min\left(1,\frac{f(b)}{f(a)}\right) \; , \]

which is the form used in the algorithm listing above.