The Metropolis-Hastings Algorithm

The Metropolis-Hasting (MH) algorithm is historically probably the most important and most well-studied sampling algorithm.

There is some controversy regarding who should be credited for the development of the algorithm, but it is named after Metropolis *et al.* (1953) and Hastings (1970)[1] [2].

The most important ingredient in the MH algorithm is the *proposal distribution* $Q(y|x)$ on $\mathcal{X}$. If the chain is currently at position x_i, this is used to propose possibilities for the next chain position, $y \sim Q(y|x_i)$, which can either be accepted (i.e. $x_{i+1} = y$) or rejected (i.e. $x_{i+1} = x_i$).

The most basic version of the MH algorithm proceeds as follows.

Algorithm 0.1 Metropolis Hastings

$x_0 \sim \alpha$ ▷ Initialise
$i \leftarrow 0$
while $i \geq 0$ **do** ▷ Iterate $i = 0, 1, 2, \ldots$
  $y \sim Q(y|x_i)$ ▷ Proposal
  $a \leftarrow (P(y)Q(x_i|y)) \,/\, (P(x_i)Q(y|x_i))$ ▷ MH acceptance probability
  $u \sim \mathcal{U}(0, 1)$
  if $u < a$ **then**
    $x_{i+1} \leftarrow y$ ▷ Accept
  else
    $x_{i+1} \leftarrow x_i$ ▷ Reject
  end if
  $i \leftarrow i + 1$
end while

The output of the MH algorithm (Alg. 0.1) is the Markov chain $x_0, x_1, x_2, \ldots$

The MH algorithm clearly defines a time-homogeneous Markov chain. For any sensible choice of proposal distribution Q this Markov chain will also generally be irreducible. The key step in the algorithm, namely the probabilistic rule for accepting/rejecting proposed

[1] Metropolis *et al.* (1953) "Equation of State Calculations by Fast Computing Machines", Journal of Chemical Physics, **21** 6 1087–1092, doi:10.1063/1.1699114

[2] Hastings (1970) "Monte Carlo Sampling Methods Using Markov Chains and Their Applications", Biometrika, **57** 1, 97–109, doi:10.1093/biomet/57.1.97

points, is designed such that the resulting Markov chain satisfies detailed balance with $\pi = P$. Therefore, from the results in the previous section, we know that chain will have the target distribution P as its unique stationary, limiting distribution.

Theorem 0.0.1. *The MH algorithm (Alg. 0.1) produces a Markov chain $x_0, x_1, \ldots$ that satisfies the detailed balance condition with $\pi = P$.*

Proof. From the MH algorithm, the Markov chain transition probabilities consist of the sum of two terms corresponding to mixture of accepting or rejecting the proposed point;

$$\rho(y,x) = a(y,x)Q(y|x) + \delta^{(d)}(y-x)\int \mathrm{d}y' \left[1 - a(y',x)\right] Q(y'|x), \tag{1}$$

where $a(y,x) = \min(1, \frac{P(y)Q(x|y)}{P(x)Q(y|x)})$ and the d-dimensional Dirac delta function $\delta^{(d)}(y-x)$ corresponds to the finite probability that the chain doesn't move (i.e. the proposed point is rejected). A similar expression holds for $\rho(x,y)$ involving $a(x,y)$. Substituting into the detailed balance condition with $\pi = P$ gives

$$P(x)a(y,x)Q(y|x) + \delta^{(d)}(y-x)P(x)\int \mathrm{d}y' \ \left[1 - a(y',x)\right] Q(y'|x) \stackrel{?}{=} \tag{2}$$

$$P(y)a(x,y)Q(x|y) + \delta^{(d)}(x-y)P(y)\int \mathrm{d}y' \ \left[1 - a(y',y)\right] Q(y'|y). \tag{3}$$

The terms involving the delta functions on both sides are clearly equal. Therefore, checking the detailed balance condition reduces to checking

$$P(x)a(y,x)Q(y|x) \stackrel{?}{=} P(y)a(x,y)Q(x|y). \tag{4}$$

Note, in the expressions for $a(y,x)$ and $a(x,y)$, exactly one of the two min conditions evaluates to 1. We consider each case individually. First, if $a(y,x) = 1$ and $a(x,y) = \frac{P(x)Q(y|x)}{P(y)Q(x|y)}$, then Eq. 5 becomes

$$P(x)Q(y|x) \stackrel{?}{=} P(y)\frac{P(x)Q(y|x)}{P(y)Q(x|y)}Q(x|y), \tag{5}$$

which after cancelling terms is true. The second case where $a(x,y) = 1$ and $a(y,x) = \frac{P(y)Q(x|y)}{P(x)Q(y|x)}$ proceeds similarly. $\square$

In the special case where a *symmetric proposal* is used, i.e. $Q(y|x) = Q(x|y)$, this is called the *Metropolis algorithm*. This means the probability of proposing a move to y from x is the same as proposing x from y. In this case the expression for the acceptance ratio a used on line 5 of Algorithm 0.1 simplifies to

$$a = \frac{P(y)}{P(x_i)}. \tag{6}$$

Note that the key step in the MH algorithm depends only on the ratio $P(y)/P(x_i)$. This means that the Metropolis–Hastings algorithm doesn't require us to know the normalised PDF of the target distribution; all that is needed is a function $\hat{P}(x) \propto P(x)$. This is useful because the normalisation constant is often difficult to calculate (in a Bayesian context, it is equivalent to finding the evidence, Z).

Note that a MCMC chain produced by the Metropolis-Hasting algorithm can have repeated entries; i.e. we might have $x_i = x_{i+1}$. This occurs when the algorithm rejects a proposed point, setting $x_{i+1} = x_i$. This can lead to the chain getting temporarily stuck in one place. This behaviour is in stark contrast to the Gibbs algorithm (which will be described next) which moves to a new point at every iteration. We define the *acceptance fraction* $0 < g < 1$ as the number of accepted proposals divided by the number of iterations.

Implementing the MH algorithm requires the user to make a choice for the proposal distribution, Q. Essentially any choice will give a Markov chain that eventually converges to P eventually, but the rate of convergence can differ wildly. A badly chosen proposal can lead to a chain that takes a long time to explore $\mathcal{X}$. The MH algorithm generally works best if the proposal closely matches the target distribution. For best performance, we can tune the proposal distribution so that $g \sim 30\%$. (But be careful, changing the proposal mid evolution breaks the time-homogeneity of the Markov chain!) The choice of proposal can strongly influences the efficiency of the MH algorithm.

Many variations and extensions of the MH algorithm have been proposed. One of the most widely used variants is *Hamiltonian Monte Carlo* which will be discussed in a coming lecture.