Metropolis-Hastings

May 1, 2018

1 Definition

Metropolis-Hastings is an algorithm that allows us to sample from a generic probability distribution, which we'll call our target distribution, even if we don't know the normalizing constant. To do this, we construct and sample from a Markov chain whose stationary distribution is the target distribution that we're looking for.

It consists of picking an arbitrary starting value and then iteratively accepting or rejecting candidate samples drawn from another distribution, one that is easy to sample.

Let's say we want to produce samples from a target distribution. We're going to call it p of theta. But we only know it up to a normalizing constant or up to proportionality. What we have is g of theta.

$$p(\theta) \propto g(\theta)$$

So we don't know the normalizing constant because perhaps this is difficult to integrate. So we only have g of theta to work with.

The Metropolis Hastings Algorithm will proceed as follows: 1. select initial value θ_o 2. for i,...,m repeat:

```
a. draw candidate $\theta^{*} \sim q(\theta^{*}|\theta_{i-1})$

b. $\alpha = \dfrac{\dfrac{g(\theta^{*})}{q(\theta_{i-1}|\theta^{*})}}{\dfrac{g(\theta_{i-1})}{q

c.
* if $\alpha \geq 1$ we accept $\theta^{*}$ and set $\theta_{i} \leftarrow \theta^{*}$.
* if $0<\alpha <1$ :
    * accept  $\theta^{*}$ and set $\theta_{i} <- \theta^{*}$ with probability $\alpha$
    * or we reject  $\theta^{*}$ and set $\theta_{i} <- \theta^{*}$ with probability $1 - \alpha
```

The steps b and c act as a correction since the proposal distribution q is not the target distribution p. At each step in the chain we draw a candidate and decide whether to move the chain there or to remain where we are.

If $a \geq 1$(advantageous) we will move there, if $0 < \alpha < 1$ then we will move there with probability α. Since the decision is based only on where the chain is currently is, it makes it a markov chain.

One careful choice we must make is with the candidate generating distribution q. It may or may not depend on the previous iteration's value of theta. (example: $q(\theta^{*})$ is always the same distribution). If we take this option, q of theta should be similar to p of theta, to approximate it.

When $q(\theta^*)$ depends on the previous iteration, it is called **random walk Metropolis-Hastings**. Here, the proposal distribution is centered on the previous iteration. For instance, it might be a normal distribution where the mean is our previous iteration θ_{i-1}. Because the normal distribution is symmetric around its mean, this q evaluated at the candidate given the mean here, the density value for this q will be equal to this density value of the old value where the mean is the candidate. This causes these two qs to cancel out when we calcualate α.

So in random walk Metropolis-Hastings, where the candidate is drawn from a normal distribution where the mean is the previous iteration's value and we use a constant variance in that normal distribution, the acceptance ratio, alpha, will be: $\alpha = \dfrac{g(\theta^*)}{g(\theta_{i-1})}$

Clearly, not all candidate draws are accepted. So, our Markov chain sometimes stays where it is, possibly for many iterations.

How often you want the chain to accept candidates depends on the type of algorithm you use. If you approximate p with q and always draw candidates from that distribution, accepting candidates often is a good thing. It means that q is approximating p well.

However, you still may want to have q have a larger variance than p, and see some rejection of candidates to be as an assurance that q is covering the space well.

- **a high acceptance rate for random walk Metropolis-Hastings samplers is not a good thing**. If the random walk is taking too small of steps, it will accept candidates often, but will take a very long time to fully explore the posterior distribution.
- If the random walk is taking too large of steps, many of its proposals will have low probability and the acceptance rate will be low. That will cause us to waste many of the draws.
- Ideally, a random walk sampler should accept 23%-50% of the candidates proposed.

2 Example: Markov Chain Monte Carlo

Let assume that your brother has a loaded coin that comes up heads 70% of the time. One day he comes to you with a coin that you are not sure whether it is loaded or not and he wants to bet with you and you need to figure out which coin this is. You know that there is 60% probability of him bringing the loaded coin. He makes 5 tosses, 3 of them are tails and 2 are heads. What is the posterior probability of the coin being loaded?

$$\theta = \{fair, loaded\}$$

$$Prior : P(\theta = loaded) = 0.6$$

$$\text{Likelihood: } f(x|\theta) = \binom{5}{x}(\frac{1}{2})^5 I_{\{\theta=fair\}} + \binom{5}{x}(0.7)^x(0.3)^{5-x} I_{\{\theta=loaded\}}$$

$$\text{Posterior: } f(\theta|x) = \frac{f(x=2|\theta)f(\theta)}{f(x)}$$

$$= \frac{\frac{1}{2}^5(.4)I_{\{\theta=fair\}} + (.7)^2(.3)^3(.6)I_{\{\theta=loaded\}}}{\frac{1}{2}^5(0.4) + (.7)^2(.3)^3(.6)}$$

$$= \frac{0.00125 I_{\{\theta=fair\}} + 0.00794 I_{\{\theta=loaded\}}}{0.0125 + 0.00794}$$