Metropolis Hastings

Rebecca C. Steorts
Bayesian Methods and Modern Statistics: STA 360/601

Module 9

The Metropolis-Hastings algorithm is a general term for a family of Markov chain simulation methods that are useful for drawing samples from Bayesian posterior distributions.

The Gibbs sampler can be viewed as a special case of Metropolis-Hastings (as well will soon see).

Here, we review the basic Metropolis algorithm and its generalization to the Metropolis-Hastings algorithm, which is often useful in applications (and has many extensions).

The setup

Suppose we can sample from $p(\theta|y)$. Then we could generate

$$\theta^{(1)}, \ldots, \theta^{(S)} \overset{iid}{\sim} p(\theta|y)$$

and obtain Monte Carlo approximations of posterior quantities

$$E[g(\theta)|y] \to 1/S \sum_{i=1}^{S} g(\theta^{(i)}).$$

Review of Metropolis

But what if we cannot sample directly from $p(\theta|y)$? The important concept here is that we are able to construct a large collection of θ values (rather than them being iid, since this most certain for most realistic situations will not hold). Thus, for any two different θ values θ_a and θ_b, we need

$$\frac{\#\theta's \text{ in the collection } = \theta_a}{\#\theta's \text{ in the collection } = \theta_b} \approx \frac{p(\theta_a|y)}{p(\theta_b|y)}.$$

How might we intuitively construct such a collection?

Review of Metropolis

- Assume $\{\theta^{(1)}, \ldots, \theta^{(s)}\}$. Suppose adding new value $\theta^{(s+1)}$.
- Consider adding a value θ^* which is nearby $\theta^{(s)}$.
- Should we include θ^* or not?
- If $p(\theta^*|y) > p(\theta^{(s)}|y)$, then we want more θ^*'s in the set than $\theta^{(s)}$'s.
- But if $p(\theta^*|y) < p(\theta^{(s)}|y)$, we shouldn't necessarily include θ^*.

Perhaps our decision to include θ^* or not should be based upon a comparison of $p(\theta^*|y)$ and $p(\theta^{(s)}|y)$. Consider the ratio

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y \mid \theta^*)p(\theta^*)}{p(y \mid \theta^{(s)})p(\theta^{(s)})}.$$

Review of Metropolis

Having computed r, what should we do next?

Review of Metropolis

- If $r > 1$ (intuition): Since $\theta^{(s)}$ is already in our set, we should include θ^* as it has a higher probability than $\theta^{(s)}$.
  (procedure): Accept θ^* into our set and let $\theta^{(s+1)} = \theta^*$.
- If $r < 1$ (intuition): The relative frequency of θ-values in our set equal to θ^* compared to those equal to $\theta^{(s)}$ should be

  $$\frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = r.$$

  This means that for every instance of $\theta^{(s)}$, we should only have a fraction of an instance of a θ^* value.
  (procedure): Set $\theta^{(s+1)}$ equal to either θ^* or $\theta^{(s)}$ with probability r and $1 - r$ respectively.

Review of Metropolis

This is basic intuition behind the Metropolis (1953) algorithm. More formally, it

- It proceeds by sampling a proposal value θ^* nearby the current value $\theta^{(s)}$ using a *symmetric proposal distribution* $J(\theta^* \mid \theta^{(s)})$.
- What does symmetry mean here? It means that $J(\theta_a \mid \theta_b) = J(\theta_b \mid \theta_a)$. That is, the probability of proposing $\theta^* = \theta_a$ given that $\theta^{(s)} = \theta_b$ is equal to the probability of proposing $\theta^* = \theta_b$ given that $\theta^{(s)} = \theta_a$.
- Symmetric proposals include:

$$J(\theta^* \mid \theta^{(s)}) = \text{Uniform}(\theta^{(s)} - \delta, \theta^{(s)} + \delta)$$

and

$$J(\theta^* \mid \theta^{(s)}) = \text{Normal}(\theta^{(s)}, \delta^2).$$

Review of Metropolis

The Metropolis algorithm proceeds as follows:

1. Sample $\theta^* \sim J(\theta \mid \theta^{(s)})$.
2. Compute the acceptance ratio (r):

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y \mid \theta^*)p(\theta^*)}{p(y \mid \theta^{(s)})p(\theta^{(s)})}.$$

3. Let

$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{with prob min(r,1)} \\ \theta^{(s)} & \text{otherwise.} \end{cases}$$

Remark: Step 3 can be accomplished by sampling $u \sim \text{Uniform}(0, 1)$ and setting $\theta^{(s+1)} = \theta^*$ if $u < r$ and setting $\theta^{(s+1)} = \theta^{(s)}$ otherwise.

Metropolis for Normal-Normal (review)

Let's test out the Metropolis algorithm for the conjugate Normal-Normal model with a known variance situation. That is let

$$
\begin{aligned}
X_1, \ldots, X_n \mid \theta &\overset{iid}{\sim} \text{Normal}(\theta, \sigma^2) \\
\theta &\sim \text{Normal}(\mu, \tau^2).
\end{aligned}
$$

Recall that the posterior of θ is $\text{Normal}(\mu_n, \tau_n^2)$, where

$$
\mu_n = \bar{x}\frac{n/\sigma^2}{n/\sigma^2 + 1/\tau^2} + \mu\frac{1/\tau^2}{n/\sigma^2 + 1/\tau^2}
$$

and

$$
\tau_n^2 = \frac{1}{n/\sigma^2 + 1/\tau^2}.
$$

Metropolis for Normal-Normal (review)

Suppose (taken from Hoff, 2009), $\sigma^2 = 1, \tau^2 = 10$, $\mu = 5$, $n = 5$, and $y = (9.37, 10.18, 9.16, 11.60, 10.33)$. For these data, $\mu_n = 10.03$ and $\tau_n^2 = 0.20$.

Let's use metropolis to estimate the posterior (just as an illustration).

Based on this model and prior, we need to compute the acceptance ratio r

$$
\begin{aligned}
r &= \frac{p(\theta^*|x)}{p(\theta^{(s)}|x)} = \frac{p(x|\theta^*)p(\theta^*)}{p(x|\theta^{(s)})p(\theta^{(s)})} && (1) \\
&= \left(\frac{\prod_i \text{dnorm}(x_i, \theta^*, \sigma)}{\prod_i \text{dnorm}(x_i, \theta^{(s)}, \sigma)} \right) \left(\frac{\text{dnorm}(\theta^*, \mu, \tau)}{\text{dnorm}(\theta^{(s)}, \mu, \tau)} \right) && (2)
\end{aligned}
$$

Metropolis for Normal-Normal (review)

In many cases, computing the ratio r directly can be numerically unstable, however, this can be modified by taking $\log r$.
This results in

$$\log r = \sum_i \left[\log \mathsf{dnorm}(x_i, \theta^*, \sigma) - \log \mathsf{dnorm}(x_i, \theta^{(s)}, \sigma)\right] + \sum_i \left[\log \mathsf{dnorm}(\theta^*, \mu, \tau) - \log \mathsf{dnorm}(\theta^{(s)}, \mu, \tau)\right].$$

Then a proposal is accepted if $\log u < \log r$, where u is sampled from the Uniform(0,1).

Metropolis for Normal-Normal (review)

We run 10,000 iterations of the Metropolis algorithm stating at $\theta^{(0)} = 0$. and using a normal proposal distribution, where

$$\theta^{(s+1)} \sim \text{Normal}(\theta^{(s)}, 2).$$

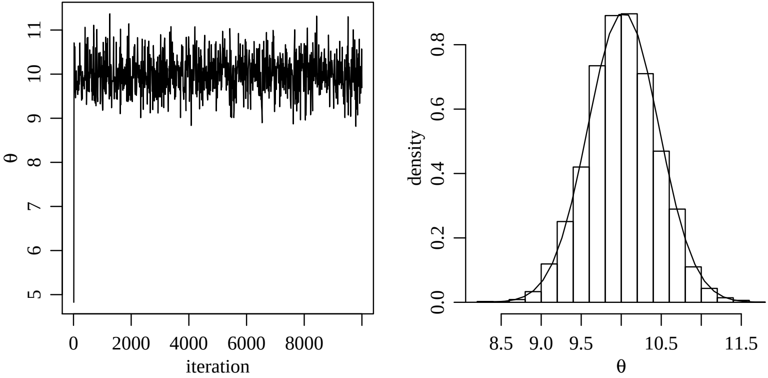

Figure 1: Results from the Metropolis sampler for the normal model.

Metropolis Hastings

Let's recall what a Markov chain is.

The Gibbs sampler and the Metropolis algorithm are both ways of generating Markov chains that approximate a target probability distribution.

Consider a simple example where our target probability distribution is $p_o(u, v)$, a bivariate distribution for two random variables U and V.

In the one-sample normal problem, $U = \theta$, $V = \sigma^2$ and

$$p_o(u, v) = p(\theta, \sigma^2 | y).$$

Gibbs: iteratively sample values of U and V from their conditional distributions. That is,

1. update U : sample $u^{(s+1)} \sim p_o(u \mid v^{(s)})$
2. update V : sample $v^{(s+1)} \sim p_o(v \mid u^{(s+1)})$.

Metropolis: proposes changes to $X = (U, V)$ and then accepts or rejects those changes based on p_o.

An alternative way to implement the Metropolis algorithm is to propose and then accept or reject change to one element at a time:

1. update U :
   1.1 sample $u^* \sim J_u(u \mid u^{(s)})$
   1.2 compute $r = \dfrac{p_o(u^*, v^{(s)})}{p_o(u^{(s)}, v^{(s)})}$
   1.3 set $u^{(s+1)}$ equal to u^* or $u^{(s+1)}$ with prob min(1,r) and max(0,1-r).
2. update V : sample $v^{(s+1)} \sim p_o(v \mid u^{(s+1)})$.
   2.1 sample $v^* \sim J_u(v \mid v^{(s)})$
   2.2 compute $r = \dfrac{p_o(u^{(s+1)}, v^*)}{p_o(u^{(s+1)}, v^{(s)})}$
   2.3 set $v^{(s+1)}$ equal to v^* or $v^{(s)}$ with prob min(1,r) and max(0,1-r).

Here, J_u and J_v are separate symmetric proposal distributions for U and V.

- The Metropolis algorithm generates proposals from J_u and J_v
- It accepts them with some probability min(1,r).
- Similarly, each step of Gibbs can be seen as generating a proposal from a full conditional and then accepting it with probability 1.
- The Metropolis-Hastings (MH) algorithm generalizes both of these approaches by allowing arbitrary proposal distributions.
- The proposal distributions can be symmetric around the current values, full conditionals, or something else entirely.

A MH algorithm for approximating $p_o(u, v)$ runs as follows:

1. update U :
   1.1 sample $u^* \sim J_u(u \mid u^{(s)}, v^{(s)})$
   1.2 compute
   $$r = \frac{p_o(u^*, v^{(s)})}{p_o(u^{(s)}, v^{(s)})} \times \frac{J_u(u^{(s)} \mid u^*, v^{(s)})}{J_u(u^* \mid u^{(s)}, v^{(s)})}$$
   1.3 set $u^{(s+1)}$ equal to u^* or $u^{(s+1)}$ with prob min(1,r) and max(0,1-r).
2. update V :
   2.1 sample $v^* \sim J_v(u \mid u^{(s+1)}, v^{(s)})$
   2.2 compute
   $$r = \frac{p_o(u^{(s+1)}, v^*)}{p_o(u^{(s+1)}, v^{(s)})} \times \frac{J_u(v^{(s+1)} \mid u^{(s+1)}, v^*)}{J_u(v^* \mid u^{(s+1)}, v^{(s)})}$$
   2.3 set $v^{(s+1)}$ equal to v^* or $v^{(s+1)}$ with prob min(1,r) and max(0,1-r).

Above: J_u and J_v are not required to be symmetric. They cannot depend on U or V values in our sequence previous to the most current values. This requirement ensures that the sequence is a Markov chain.

Doesn't the algorithm above look familiar? Yes, it looks a lot like Metropolis, except the acceptance ratio r contains an extra factor:

- It contains the ratio of the prob of generating the current value from the proposed to the prob of generating the proposed from the current.
- This can be viewed as a correction factor.
- If a value u^* is much more likely to be proposed than the current value $u^{(s)}$ then we must down-weight the probability of accepting u.
- Otherwise, such a value u^* will be overrepresented in the chain.

Exercise 1: Show that Metropolis is a special case of MH. Hint: Think about the jumps J.
Exercise 2: Show that Gibbs is a special case of MH. Hint: Show that r = 1.

We implement the Metropolis algorithm for a Poisson regression model.

- We have a sample from a population of 52 song sparrows that was studied over the course of a summer and their reproductive activities were recorded.
- In particular, their age and number of new offspring were recorded for each sparrow (Arcese et al., 1992).
- A simple probability model to fit the data would be a Poisson regression where, Y = number of offspring conditional on x = age.

Thus, we assume that

$$Y|\theta_x \sim \text{Poisson}(\theta_x).$$

For stability of the model, we assume that the mean number of offspring θ_x is a smooth function of age. Thus, we express $\theta_x = \beta_1 + \beta_2 x_+ \beta_3 x^2$.

Remark: This parameterization allows some values of θ_x to be negative, so as an alternative we reparameterize and model the log-mean of Y, so that

$$\log E(Y|x) = \log \theta_x = \log(\beta_1 + \beta_2 x_+ \beta_3 x^2)$$

which implies that

$$\theta_x = \exp(\beta_1 + \beta_2 x_+ \beta_3 x^2) = \exp(\boldsymbol{\beta}^T \boldsymbol{x}).$$

Now back to the problem of implementing Metropolis. For this problem, we will write

$$\log E(Y_i|x_i) = \log(\beta_1 + \beta_2 x_i + \beta_3 x_i^2) = \boldsymbol{\beta}^T \boldsymbol{x_i},$$

where x_i is the age of sparrow i. We will abuse notation slightly and write $\boldsymbol{x_i} = (1, x_i, x_i^2)$.

- We will assume the prior on the regression coefficients is iid Normal(0,100).
- Given a current value $\boldsymbol{\beta^{(s)}}$ and a value $\boldsymbol{\beta}^*$ generated from $J(\boldsymbol{\beta}^*, \boldsymbol{\beta^{(s)}})$ the acceptance ration for the Metropolis algorithm is:

$$r = \frac{p(\boldsymbol{\beta}^*|\boldsymbol{X}, \boldsymbol{y})}{p(\boldsymbol{\beta^{(s)}}|\boldsymbol{X}, \boldsymbol{y})} = \frac{\prod_{i=1}^{n} \mathsf{dpois}(y_i, x_i^T \beta^*)}{\prod_{i=1}^{n} \mathsf{dpois}(y_i, x_i^T \beta^{(s)})} \times \frac{\prod_{j=1}^{3} \mathsf{dnorm}(\beta_j^*, 0, 10)}{\prod_{j=1}^{3} \mathsf{dnorm}(\beta_j^{(s)}, 0, 10)}.$$

- We just need to specify the proposal distribution for θ^*
- A convenient choice is a multivariate normal distribution with mean $\boldsymbol{\beta}^{(s)}$.
- In many problems, the posterior variance can be an efficient choice of a proposal variance. But we don't know it here.
- However, it's often sufficient to use a rough approximation. In a normal regression problem, the posterior variance will be close to $\sigma^2(X^TX)^{-1}$ where σ^2 is the variance of Y.

In our problem: $E\log Y = \beta^T x$ so we can try a proposal variance of $\hat{\sigma}^2(X^TX)^{-1}$ where $\hat{\sigma}^2$ is the sample variance of $\log(y+1/2)$.
Remark: Note we add $1/2$ because otherwise $\log 0$ is undefined.
The code of implementing the algorithm will be done in the corresponding lab.

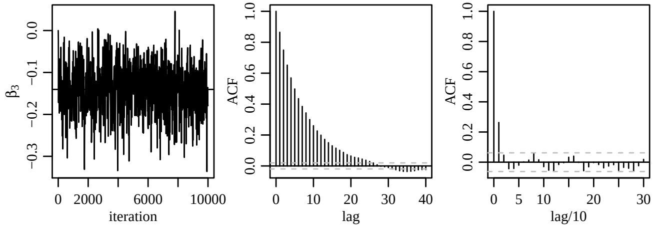

Figure 2: Plot of the Markov chain in β_3 along with autocorrelations functions

More details of this example will be done in lab.