Module 6

1 The Overall Goal of the Metropolis-Hastings Algorithm

Remember that the goal is to generate random numbers from an arbitrary density, $f_X(x)$. These numbers will be denoted as $X_1, X_2, \ldots, X_n$, and we hope that they are independent and identically distributed (i.i.d.). The inverse-transform method and the accept-reject method generate each X_i independently from one another, and these methods can only be used if

1. For the inverse-transform method, $F_X^{-1}(x)$ is analytically tractable (easy to calculate).
2. For the accept-reject method, a density $g(x)$ must exist such that for some constant C, $Cg(x) \geq f_X(x)$ for all x. This density $g(x)$ is not always easy to find.

If (a) or (b) can't be satisfied, one might want to do the Metropolis-Hastings Algorithm. It's important to understand, though, that the Metropolis-Hastings Algorithm is a Markov-Chain, meaning that the value of X_{t+1} depends on the value of X_t. This was *not* the case with the inverse-transform method or the accept-reject method.

So how is the value of X_{t+1} generated? The value of X_{t+1} is proposed by what is (not surprisingly) called the proposal distribution. This proposal distribution typically depends (in some way) on X_t and I will denote it as $g(x_{t+1}|x_t)$. After X_{t+1} is generated from $g(x_{t+1}|x_t)$, it will be accepted as the $(t+1)^{st}$ value in the sequence $X_1, X_2, \ldots, X_n$ with probability

$$\rho = \min\left\{1, \frac{f_X(x_{t+1})g(x_t|x_{t+1})}{f_X(x_t)g(x_{t+1}|x_t)})\right\}.$$

If it is not accepted, the $(t+1)^{st}$ value in the sequence is X_t.

Be careful how you pick the candidate density. If the variability in X_{t+1} is too wide, your moves will be too drastic and you won't accept new values of X very often. If the variability of $g(x_{t+1}|x_t)$ is too small, you don't move around enough.

Read, study, and code up Examples 6.1 and Examples 6.2 in the book. In Example 6.1, $f(x)$ is the Beta distribution, and $g(x_{t+1}|x_t)$ is the Uniform. In Example 6.2, $f_X(x)$ is the Cauchy distribution, and $g(x_{t+1}|x_t)$ is the normal distribution with mean 0 and variance 1. For Example 6.2, see what happens when $g(x_{t+1}|x_t)$ is $N(x_t, 1)$. What about when it is $N(x_t, 2)$.

Also take a look at the example I've worked out and posted the code to. The code is entitled 'MyMetropolisExample.R'. In that example, I am sampling from the density

$$f_X(x) = \frac{1}{2} \cdot \left(\frac{x}{\lambda}\right)^{(p-2)/4} \cdot I_{(p-2)/2)}\left(\sqrt{\lambda x}\right) \exp\left\{-\left(\lambda + x\right)/2\right\},$$

where $I()$ is the Bessel function.

2 The Metropolis-Hastings Algorithm and Continuous Markov Chains

2.1 Continuous Markov Chains

For the moment, I would like you to recall the material that we covered in Module 4 of the course, Markov Chains. Recall that a Markov chain is a stochastic process where

$$P\left(X_{t+1} = x_{t+1}|X_t = x_t, X_{t-1} = x_{t-1}, \ldots, X_0 = x_0\right) = P\left(X_{t+1} = x_{t+1}|X_t = x_t, X_{t-1} = x_{t-1}\right).$$

In other words, the entire histoy of the chain is irrelevant when calculating $P\left(X_{t+1}=x_{t+1}|X_t=x_t\right)$. Yet another way to say it is that the conditional distribution of X_{t+1} given $X_t, X_{t-1}, X_{t-2}, \ldots, X_0$ is equal to the conditional probability of X_{t+1} given X_t.

The Metropolis-Hastings algorithm produces a Markov chain because the value of X at time t only depends on the value of X at time $t-1$. The Markov chain produced by the Metropolis-Hastings algorithm is a little different than those studied in Module 4. The Markov chains studied in Module 4 were discrete, and the Metropolis-Hastings algorithm produces a Markov chain over, possibly, a continuous space. In other words, the values of X proposed and accepted can be any number in a particular interval, whereas for the chains studied in Module 4, the states were discrete (or countable) values.

Recall that for discrete Markov chains, there was a transition probability matrix, $\boldsymbol{P}$, where $p_{i,j}=$ the $(i,j)^{\text{th}}$ element of $\boldsymbol{P}$ was the probability of going to state j "given" the chain was at state i. For continuous chains, there is no transition probability matrix, but there is a transition kernel. The textbook we are using denotes this transition kernel as $K(X_t, X_{t+1})$, yet others denote it as $K(X_{t+1}|X_t)$. Either way, it is interpreted as the conditional probability density of X_{t+1} given the chain is currently at X_t. The next subsection gives the details of how the transition kernel for the Metropolis-Hastings algorithm is calculated

2.2 Calculating the Kernel

Recall that

$$X_{t+1}=\begin{cases} X_{\text{cand}} & \text{with proability } \rho(X_t, X_{\text{cand}}) \\ X_t & \text{with probability } 1-\rho(X_t, X_{\text{cand}}) \end{cases}$$

The conditional probability density of X_{t+1} given X_{cand} and X_t is thus

$$f(X_{t+1}|X_{\text{cand}}, X_t)=\delta(X_{\text{cand}}-X_{t+1})\cdot\rho(X_t, X_{\text{cand}})+\delta\left(X_t-X_{t+1}\right)\cdot\left(1-\rho\left(X_t, X_{\text{cand}}\right)\right),$$

where

$$\delta(z)=\begin{cases} 1 & z=0 \\ 0 & \text{otherwise} \end{cases}.$$

The transition kernel just wants the conditional probability of X_{t+1} given X_t, and to obtain this, the distribution of the candidate value has to be integrated out. It is thus the case that

$$K(X_t, X_{t+1})=K(X_{t+1}|X_t)=\int_{\mathcal{R}} f(X_{t+1}|X_{\text{cand}}=x_{\text{cand}}, X_t)f(X_{\text{cand}}=x_{\text{cand}}|X_t)dx_{\text{cand}}.$$

It can be shown that this is equal to

$$\rho(X_t, X_{t+1})g(X_{t+1}|X_t)+\delta(X_{t+1}-X_t)\left(1-r(X_t)\right),$$

where $r(X_t)=\displaystyle\int_{\mathcal{R}}\rho(X_t, X_{\text{cand}}=x_{\text{cand}})g(X_{\text{cand}}=x_{\text{cand}}|X_t)dx_{\text{cand}}$.

2.3 The stationary distribution

Also recall what a stationary distribution is in the discrete sense. The stationary distribution, intuitively, is the long-run probability that a chain will arrive or visit a certain state. If there are k possible states in a discrete chain, then the stationary distribution is the $k-$vector π, and it satisfies

$$\pi^T=\pi^T\boldsymbol{P}. \tag{1}$$

Writing out the first element of the Equation (1), we get

$$\pi(1)=p_{1,1}\pi(1)+p_{2,1}\pi(2)+\cdots+p_{k,1}\pi(k)=\sum_j p_{j,1}\pi(j).$$

For continuous Markov chains, the ideas are similar. The sum is just replaced by an integral. The stationary distribution of a continuous Markov chain is that density f such that

$$f(X_{t+1}) = \int_{\mathcal{R}} K(X_t = x_t, X_{t+1}) f(X_t = x_t) dx_t.$$

2.4 Detailed Balance

A neat thing to note about the Metropolis-Hastings algorithm is that it satisfies detailed balance. Recall that detailed balance is

$$P(X = x_i)P(X_i \to X_{i+1}) = P(X = x_{i+1})P(X_{i+1} \to X_i).$$

Let's prove this:

$$\begin{aligned} P(X = X_i)P(X_i \to X_{i+1}) &= P(X = X_i)P(\text{Moving to } X_{i+1} \text{ given at } X_i) \\ &= f(x_i) \quad \underbrace{g(x_{i+1}|x_i)}_{\text{Prob. of proposing } x_{i+1}} \quad \cdot \underbrace{\left\{ \min\left(1, \frac{f(x_{i+1})g(x_{i+1}|x_i)}{f(x_i)g(x_i|x_{i+1})}\right)\right\}}_{\text{Prob of accepting proposal}}. \end{aligned}$$

If we assume that the ratio in the above expression is less than 1, this becomes

$$\begin{aligned} &= f(x_i)g(x_{i+1}|x_i)\left\{\frac{f(x_{i+1})g(x_{i+1}|x_i)}{f(x_i)g(x_i|x_{i+1})}\right\} \\ &= f(x_{i+1})g(x_{i+1}|x_i) \\ &= f(x_{i+1})g(x_{i+1}|x_i)\left\{\min\left(1, \frac{f(x_i)g(x_i|x_{i+1})}{f(x_{i+1})g(x_{i+1}|x_i)}\right)\right\} \\ &= P(X = X_{i+1})P(X_{i+1} \to X_i) \end{aligned}$$