Metropolis and Metropolis-Hastings II

Dr. Olanrewaju Michael Akande

April 8, 2020

Announcements

- Reminder: let the instructor know if you plan to request a letter grade.

Outline

- Metropolis algorithm
  - Introduction and intuition
  - Algorithm
  - Illustration
  - Application to Poisson regression
- Metropolis-Hastings algorithm

Metropolis algorithm

Introduction

- As a refresher, suppose $Y \sim \pi(y|\theta)$ and suppose we specify a prior $\pi(\theta)$ on θ.
- Then as usual, we are interested in

$$\pi(\theta|y) = \frac{\pi(\theta)L(y;\theta)}{\mathcal{L}(y)}.$$

- As we already know, the challenge is that it is often difficult to compute $\mathcal{L}(y)$.
- Using the Monte Carlo method or Gibbs sampler, we have seen that we don't need to know $\mathcal{L}(y)$. As long as we have conjugate and semi-conjugate priors, we can generate samples directly from $\pi(\theta|y)$.
- What happens if we cannot sample directly from $\pi(\theta|y)$?

Motivating example

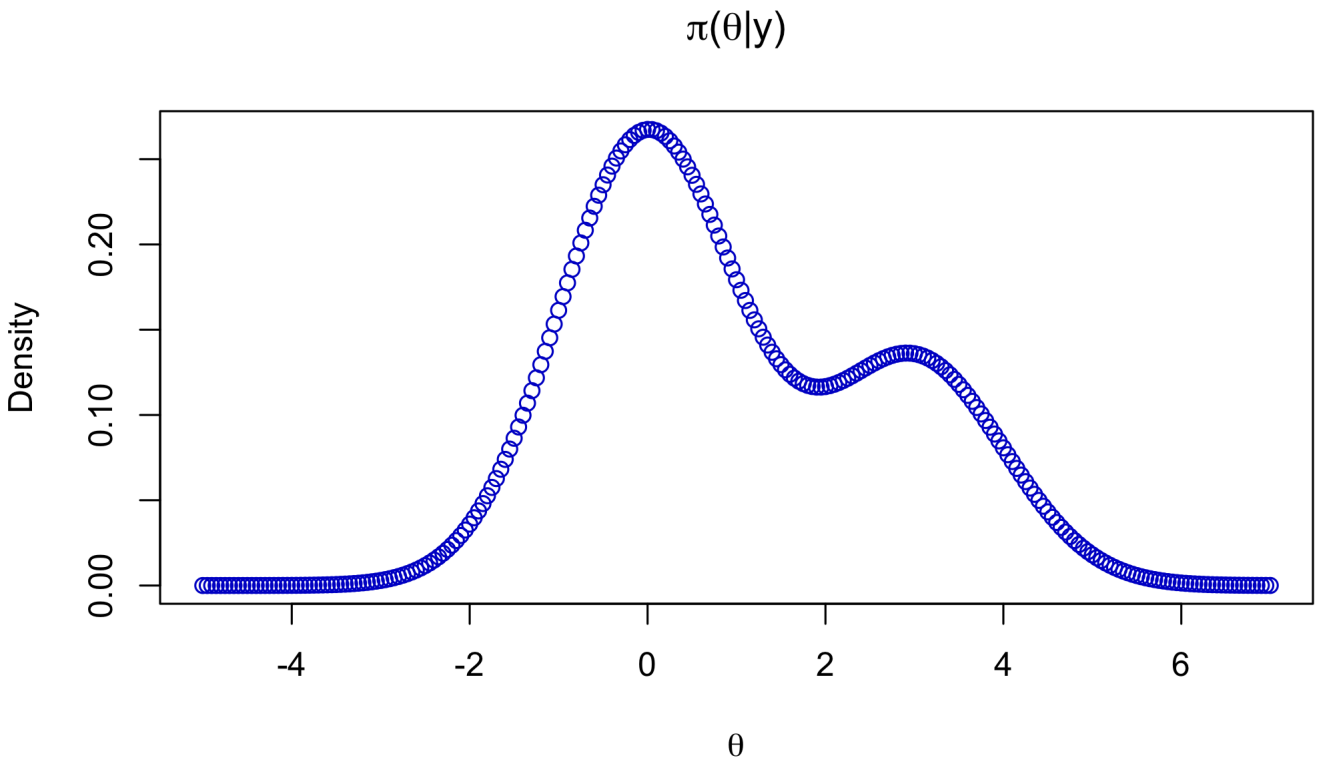
- To motivate our discussions on the Metropolis algorithm, let's explore a simple example.
- Suppose we wish to sample from the following density

$$\pi(\theta|y) \propto \exp^{-\frac{1}{2}\theta^2} + \frac{1}{2}\exp^{-\frac{1}{2}(\theta-3)^2}$$

- This is a *mixture of two normal densities*, one with mode near 0 and the other with mode near 3.
  FYI: finite mixture models remains the most likely topic we will cover on Friday plus next part of next Wednesday.
- Anyway, let's use this density to explore the main ideas behind the Metropolis sampler.
- By the way, as you will see, we don't actually need to know the normalizing constant for Metropolis sampling but for this example, find it for practice! I will set it up in class.

Motivating example

- Let's take a look at the (normalized) density:



- There are other ways of sampling from this density, but let's focus specifically on the Metropolis algorithm here.

Metropolis algorithm

- From a sampling perspective, we need to have a large group of values, $\theta^{(1)}, \ldots, \theta^{(S)}$ from $\pi(\theta|y)$ whose empirical distribution approximates $\pi(\theta|y)$.
- That means that for any any two values θ_a and θ_b, we want

$$\frac{\#\theta^{(s)} = a}{S} \div \frac{\#\theta^{(s)} = b}{S} = \frac{\#\theta^{(s)} = a}{S} \times \frac{S}{\#\theta^{(s)} = b} = \frac{\#\theta^{(s)} = a}{\#\theta^{(s)} = b} \approx \frac{\pi(\theta_a|y)}{\pi(\theta_b|y)}$$

- Basically, we want to make sure that if θ_a and θ_b are in $\pi(\theta|y)$, the ratio of the number of the $\theta^{(1)}, \ldots, \theta^{(S)}$ values equal to them properly approximates $\frac{\pi(\theta_a|y)}{\pi(\theta_b|y)}$.
- How might we construct a group like this?

Metropolis algorithm

- Suppose we have a working group $\theta^{(1)}, \ldots, \theta^{(s)}$ at iteration s, and need to add a new value $\theta^{(s+1)}$.
- Consider a candidate value $\theta^\star$ that is close to $\theta^{(s)}$ *(we will get to how to generate the candidate value in a minute)*. Should we set $\theta^{(s+1)} = \theta^\star$ or not?
- Well, we should probably compute $\pi(\theta^\star|y)$ and see if $\pi(\theta^\star|y) > \pi(\theta^{(s)}|y)$. Equivalently, look at $r = \dfrac{\pi(\theta^\star|y)}{\pi(\theta^{(s)}|y)}$.
- By the way, notice that

$$r = \frac{\pi(\theta^\star|y)}{\pi(\theta^{(s)}|y)} = \frac{\mathcal{L}(y|\theta^\star)\pi(\theta^\star)}{\mathcal{L}(y)} \div \frac{\mathcal{L}(y|\theta^{(s)})\pi(\theta^{(s)})}{\mathcal{L}(y)}$$

$$= \frac{\mathcal{L}(y|\theta^\star)\pi(\theta^\star)}{\mathcal{L}(y)} \times \frac{\mathcal{L}(y)}{\mathcal{L}(y|\theta^{(s)})\pi(\theta^{(s)})} = \frac{\mathcal{L}(y|\theta^\star)\pi(\theta^\star)}{\mathcal{L}(y|\theta^{(s)})\pi(\theta^{(s)})},$$

which does not depend on the marginal likelihood we don't know!

Metropolis algorithm

- If $r > 1$
  - Intuition: $\theta^{(s)}$ is already a part of the density we desire and the density at $\theta^\star$ is even higher than the density at $\theta^{(s)}$.
  - Action: set $\theta^{(s+1)} = \theta^\star$
- If $r < 1$,
  - Intuition: relative frequency of values on our group $\theta^{(1)}, \ldots, \theta^{(s)}$ equal to $\theta^\star$ should be $\approx r = \dfrac{\pi(\theta^\star|y)}{\pi(\theta^{(s)}|y)}$. For every $\theta^{(s)}$, include only a fraction of an instance of $\theta^\star$.
  - Action: set $\theta^{(s+1)} = \theta^\star$ with probability r and $\theta^{(s+1)} = \theta^{(s)}$ with probability $1 - r$.

Metropolis algorithm

- This is the basic intuition behind the Metropolis algorithm.
- Where should the proposed value $\theta^\star$ come from?
- Sample $\theta^\star$ close to the current value $\theta^{(s)}$ using a symmetric proposal distribution $g[\theta^\star|\theta^{(s)}]$. g is actually a "family of proposal distributions", indexed by the specific value of $\theta^{(s)}$.
- Here, symmetric means that $g[\theta^\star|\theta^{(s)}] = g[\theta^{(s)}|\theta^\star]$.
- The symmetric proposal is usually very simple with density concentrated near $\theta^{(s)}$, for example, $\mathcal{N}(\theta^\star; \theta^{(s)}, \delta^2)$ or $\text{Unif}(\theta^\star; \theta^{(s)} - \delta, \theta^{(s)} + \delta)$.
- After obtaining $\theta^\star$, either add it or add a copy of $\theta^{(s)}$ to our current set of values, depending on the value of r.

Metropolis algorithm

- The algorithm proceeds as follows:

  1. Given $\theta^{(1)}, \ldots, \theta^{(s)}$, generate a *candidate* value $\theta^\star \sim g[\theta^\star | \theta^{(s)}]$.

  2. Compute the acceptance ratio

  $$r = \frac{\pi(\theta^\star | y)}{\pi(\theta^{(s)} | y)} = \frac{\mathcal{L}(y|\theta^\star)\pi(\theta^\star)}{\mathcal{L}(y|\theta^{(s)})\pi(\theta^{(s)})}.$$

  3. Set

  $$\theta^{(s+1)} = \begin{cases} \theta^\star & \text{with probability} \quad \min(r, 1) \\ \theta^{(s)} & \text{with probability} \quad 1 - \min(r, 1) \end{cases}$$

  which can be accomplished by sampling $u \sim U(0, 1)$ independently and setting

  $$\theta^{(s+1)} = \begin{cases} \theta^\star & \text{if} \quad u < r \\ \theta^{(s)} & \text{if} \quad \text{otherwise} \end{cases}.$$

Metropolis algorithm

- Once we obtain the samples, then we are back to using Monte Carlo approximations for quantities of interest.
- That is, we can again approximate posterior means, quantiles, and other quantities of interest using the empirical distribution of our sampled values.
- *Some notes:*
  - The Metropolis chain ALWAYS moves to the proposed $\theta^\star$ at iteration $s+1$ if $\theta^\star$ has higher target density than the current $\theta^{(s)}$.
  - Sometimes, it also moves to a $\theta^\star$ value with lower density in proportion to the density value itself.
  - This leads to a random, Markov process than naturally explores the space according to the probability defined by $\pi(\theta|y)$, and hence generates a sequence that, while dependent, eventually represents draws from $\pi(\theta|y)$.

METROPOLIS ALGORITHM: CONVERGENCE

- We will not cover the convergence theory behind Metropolis chains in detail, but below are a few notes for those interested:
  - The Markov process generated under this condition is ergodic and has a limiting distribution.
  - Here, think of ergodicity as meaning that the chain can move anywhere at each step, which is ensured, for example, if $g[\theta^\star|\theta^{(s)}] > 0$ everywhere!
  - By construction, it turns out that the Metropolis chains are reversible, so that convergence to $\pi(\theta|y)$ is assured.
  - Think of reversibility as being equivalent to symmetry of the joint density of two consecutive $\theta^{(s)}$ and $\theta^{(s+1)}$ in the stationary process, which we do have by using a symmetric proposal distribution.
- If you want to learn more about convergence of MCMC chains, consider taking one of the courses on stochastic processes, or Markov chain theory.

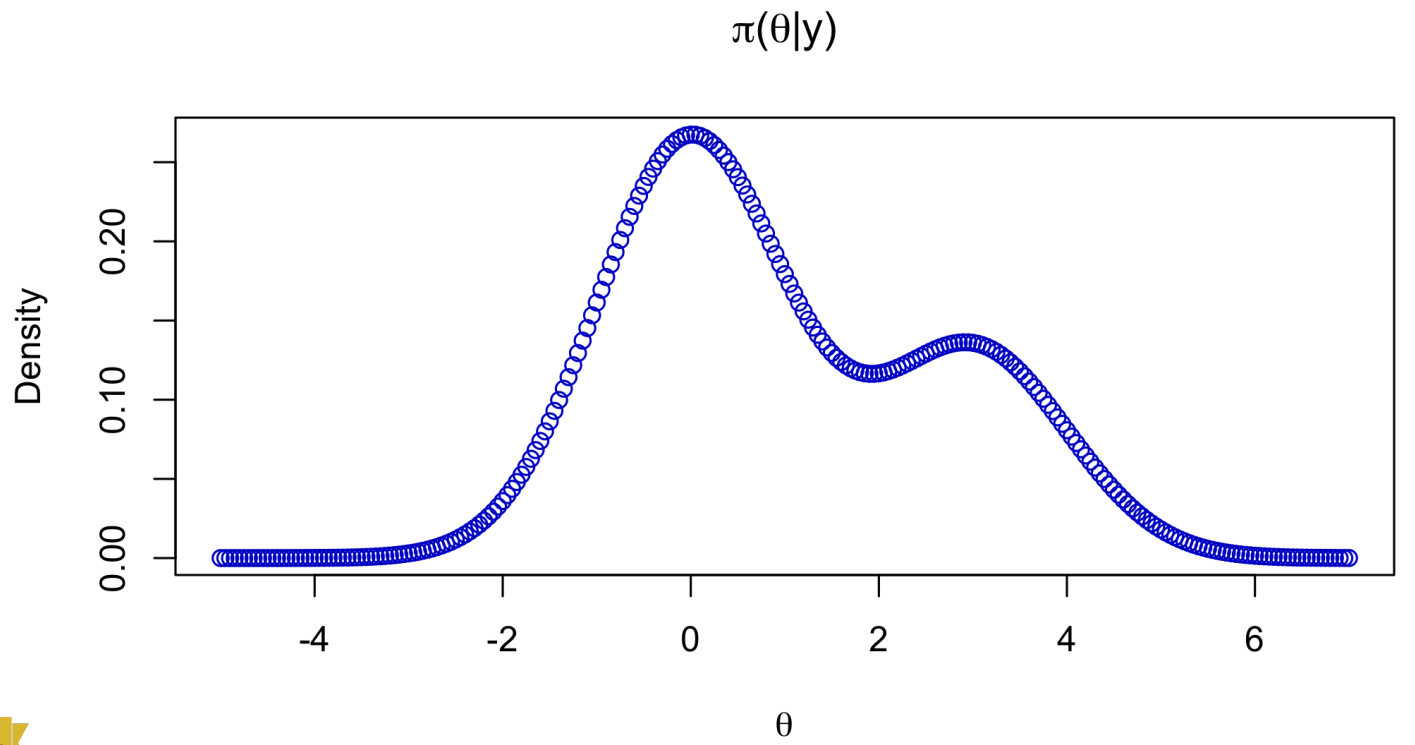
Metropolis algorithm: tuning

- Correlation between samples can be adjusted by selecting optimal δ (i.e., spread of the distribution) in the proposal distribution
- Decreasing correlation increases the effective sample size, increasing rate of convergence, and improving the Monte Carlo approximation to the posterior.
- However,
  - δ too small leads to $r \approx 1$ for most proposed values, a high acceptance rate, but very small moves, leading to highly correlated chain.
  - δ too large can get "stuck" at the posterior mode(s) because $\theta^\star$ can get very far away from the mode, leading to a very low acceptance rate and again high correlation in the Markov chain.
- Thus, good to implement several short runs of the algorithm varying δ and settle on one that yields acceptance rate in the range of 25-50%.
- Burn-in (and thinning) is even more important here!

Metropolis in action

Back to our example with

$$\pi(\theta|y) \propto \exp^{-\frac{1}{2}\theta^2} + \frac{1}{2}\exp^{-\frac{1}{2}(\theta-3)^2}$$



In-class analysis: move to the R script here.

Poisson regression

Count data

- We will use the Metropolis sampler on count data with predictors, so let's first do some general review.
- Suppose you have count data as your response variable.
- For example, we may want to explain the number of c-sections carried out in hospitals using potential predictors such as hospital type, (that is, private vs public), location, size of the hospital, etc.
- The models we have covered so far are not (completely) adequate for count data with predictors.
- Of course there are instances where linear regression, with some transformations (especially taking logs) on the response variable, might still work reasonably well for count data.
- That's not the focus here, so we won't cover that.

Poisson regression

- As we have seen so far, a good distribution for modeling count data with no limit on the total number of counts is the Poisson distribution.
- As a reminder, the Poisson pmf is given by

$$\Pr[Y = y] = \frac{\lambda^y e^{-\lambda}}{y!}; \quad y = 0, 1, 2, \ldots; \quad \lambda > 0.$$

- Remember that

$$\mathbb{E}[Y = y] = \mathbb{V}[Y = y] = \lambda.$$

- When our data fails this assumption, we may have what is known as over-dispersion and may want to consider the Negative Binomial distribution instead (actually easy to fit within the Bayesian framework!).
- With predictors, index λ with i, so that each λ_i is a function of $\boldsymbol{X}$. Therefore, the random component of the glm is

$$y_i \sim \text{Poisson}(\lambda_i); \quad i = 1, \ldots, n.$$

Poisson regression

- We must ensure that $\lambda_i > 0$ at any value of $\boldsymbol{X}$, therefore, we need a link function that enforces this. A natural choice is

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}.$$

- Combining these pieces give us our full mathematical representation for the Poisson regression.

- Clearly, λ_i has a natural interpretation as the "expected count", and

$$\lambda_i = e^{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}}$$

  so the e^{β_j}'s are multiplicative effects on the expected counts.

- For the frequentist version, in R, use the `glm` command but set the option `family = "poisson"`.

Analysis of horseshoe crabs

- We have data from a study of nesting horseshoe crabs (J. Brockmann, Ethology, 102: 1–21, 1996). The data has been discussed in Agresti (2002).
- Each female horseshoe crab in the study had a male crab attached to her in her nest.
- The study investigated factors that affect whether the female crab had any other males, called satellites, residing nearby her.
- The response outcome for each female crab is her number of satellites.
- We have several factors (including the female crab's color, spine condition, weight, and carapace width) which may influence the presence/absence of satellite males.
- The data is called `hcrabs` in the R package `rsq`.

Analysis of horseshoe crabs

- Let's fit the Poisson regression model to the data. In vector form, we have

$$y_i \sim \text{Poisson}(\lambda_i); \quad i = 1, \ldots, n;$$

$$\log[\lambda_i] = \boldsymbol{\beta}^T \boldsymbol{x}_i$$

  where y_i is the number of satellites for female crab i, and $\boldsymbol{x}_i$ contains the intercept and female crab i's

  - color;
  - spine condition;
  - weight; and
  - carapace width.

- Suppose we specify a normal prior for $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_{p-1})$, $\pi(\boldsymbol{\beta}) = \mathcal{N}_p(\boldsymbol{\beta}_0, \Sigma_0)$.

- Can you write down the posterior for β? Can you sample directly from it?

Analysis of horseshoe crabs

- We can use Metropolis to generate samples from the posterior.
- First, we need a "symmetric" proposal density $\boldsymbol{\beta}^\star \sim g[\boldsymbol{\beta}^\star | \boldsymbol{\beta}^{(s)}]$; a reasonable choice is usually a multivariate normal centered on $\boldsymbol{\beta}^{(s)}$.
- What about the variance of the proposal density? We can use the variance of the ols estimate, that is, $\hat{\sigma}^2 \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1}$, which we can scale using δ, to tune the acceptance ratio.
- Here, $\hat{\sigma}^2$ is calculated as the sample variance of $\log[y_i + c]$, for some small constant c, to avoid problems when $y_i = 0$.
- So we have $g[\boldsymbol{\beta}^\star | \boldsymbol{\beta}^{(s)}] = \mathcal{N}_p \left(\boldsymbol{\beta}^{(s)}, \delta \hat{\sigma}^2 \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1}\right)$.
- Finally, since we do not have any information apriori about $\boldsymbol{\beta}$, let's set the prior for it to be $\pi(\boldsymbol{\beta}) = \mathcal{N}_p(\boldsymbol{\beta}_0 = \mathbf{0}, \Sigma_0 = \boldsymbol{I})$.

Analysis of horseshoe crabs

- The Metropolis algorithm for this model is:

1. Given a current $\boldsymbol{\beta}^{(s)}$, generate a *candidate* value $\boldsymbol{\beta}^{\star} \sim g[\boldsymbol{\beta}^{\star}|\boldsymbol{\beta}^{(s)}] = \mathcal{N}_p\left(\boldsymbol{\beta}^{(s)}, \delta\hat{\sigma}^2\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\right)$.

2. Compute the acceptance ratio

$$r = \frac{\pi(\boldsymbol{\beta}^{\star}|Y)}{\pi(\boldsymbol{\beta}^{(s)}|Y)} = \frac{\pi(\boldsymbol{\beta}^{\star}) \cdot \mathcal{L}(Y|\boldsymbol{\beta}^{\star})}{\pi(\boldsymbol{\beta}^{(s)}) \cdot \mathcal{L}(Y|\boldsymbol{\beta}^{(s)})}$$

$$= \frac{\mathcal{N}_p(\boldsymbol{\beta}^{\star}|\boldsymbol{\beta}_0 = \mathbf{0}, \Sigma_0 = \boldsymbol{I}) \cdot \prod_{i=1}^{n} \text{Poisson}\left(Y_i|\lambda_i = \exp\left\{(\boldsymbol{\beta}^{\star})^T\boldsymbol{x}_i\right\}\right)}{\mathcal{N}_p(\boldsymbol{\beta}^{(s)}|\boldsymbol{\beta}_0 = \mathbf{0}, \Sigma_0 = \boldsymbol{I}) \cdot \prod_{i=1}^{n} \text{Poisson}\left(Y_i|\lambda_i = \exp\left\{\left(\boldsymbol{\beta}^{(s)}\right)^T\boldsymbol{x}_i\right\}\right)}.$$

3. Sample $u \sim U(0,1)$ and set

$$\boldsymbol{\beta}^{(s+1)} = \begin{cases} \boldsymbol{\beta}^{\star} & \text{if} \quad u < r \\ \boldsymbol{\beta}^{(s)} & \text{if} \quad \text{otherwise} \end{cases}.$$

In-class analysis: move to the R script here.

Analysis of horseshoe crabs

- Based on the results from the R script, we have that the expected count of male satellites
  - decreases by a multiplicative factor of $e^{-0.49} = 0.6126$ for the group with color=4 (medium dark) in comparison to baseline group with color=2 (medium light). That is, a 39% decrease.
  - increases by a multiplicative factor of $e^{0.08} = 1.0832$ for the group with spine=3 (both worn or broken) in comparison to baseline group with spine=1 (both good). That is, an 8% increase.
- Both width and weight increases the expected count of male satellites.
- Take a look at the CIs to quantify uncertainty.
- We can actually do a better job with model selection but I leave that to you!!

Metropolis-Hastings algorithm

Metropolis-Hastings algorithm

- Gibbs sampling and the Metropolis algorithm are special cases of the Metropolis-Hastings algorithm.
- The Metropolis-Hastings algorithm is more general in that it allows arbitrary proposal distributions.
- These can be symmetric around the current values, full conditionals, or something else entirely as long as they do not depend on values in our sequence that are previous to the most current values.
- That last point is to ensure the sequence is a Markov chain!
- In terms of how this works, the only real change from before is that now, the acceptance probability should also incorporate the proposal density when it is not symmetric.

Metropolis-Hastings algorithm

- Suppose our target distribution is $p_0(\theta)$. The algorithm proceeds as follows:

  1. Given a current draw $\theta^{(s)}$, propose a new value $\theta^\star \sim g_\theta[\theta^\star|\theta^{(s)}]$.

  2. Compute the acceptance ratio

$$r = \frac{p_0(\theta^\star)}{p_0(\theta^{(s)})} \times \frac{g_\theta[\theta^{(s)}|\theta^\star]}{g_\theta[\theta^\star|\theta^{(s)}]}.$$

  3. Sample $u \sim U(0,1)$ and set

$$\theta^{(s+1)} = \begin{cases} \theta^\star & \text{if} \quad u < r \\ \theta^{(s)} & \text{if} \quad \text{otherwise} \end{cases}.$$

Metropolis-Hastings algorithm

- If $p_0(\theta)$ corresponds to a posterior distribution $\pi(\theta|y)$ as is often the case for us, then we have

1. Propose a new value $\theta^\star \sim g_\theta[\theta^\star|\theta^{(s)}]$.
2. Compute the acceptance ratio

$$r = \frac{\pi(\theta^\star|y)}{\pi(\theta^{(s)}|y)} = \frac{\mathcal{L}(y|\theta^\star)\pi(\theta^\star)}{\mathcal{L}(y|\theta^{(s)})\pi(\theta^{(s)})} \times \frac{g_\theta[\theta^{(s)}|\theta^\star]}{g_\theta[\theta^\star|\theta^{(s)}]}.$$

3. Sample $u \sim U(0,1)$ and set

$$\theta^{(s+1)} = \begin{cases} \theta^\star & \text{if} \quad u < r \\ \theta^{(s)} & \text{if} \quad \text{otherwise} \end{cases}.$$

Metropolis-Hastings algorithm

- Suppose our target distribution is $p_0(u, v)$, a bivariate distribution for random variables U and V.
- For example, $p_0(u, v)$ could be the joint posterior distribution for U and V.
- Two options:
  - Define one joint proposal density $g_{u,v}[u^\star, v^\star | u^{(s)}, v^{(s)}]$ for U and V if possible; or
  - Define two proposal densities, one for U and the other for V. That is, $g_u[u^\star | u^{(s)}, v^{(s)}]$ and $g_u[v^\star | u^{(s)}, v^{(s)}]$.
- First option follows directly from the main algorithm and often works very well when possible. Second option needs a little modification.

Metropolis-Hastings algorithm

1. Update U: first, sample $u^\star \sim g_u[u^\star|u^{(s)}, v^{(s)}]$. Then,
   - Compute the acceptance ratio
   $$r = \frac{p_0(u^\star, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} \times \frac{g_u[u^{(s)}|u^\star, v^{(s)}]}{g_u[u^\star|u^{(s)}, v^{(s)}]}.$$
   - Sample $w \sim U(0,1)$. Set $u^{(s+1)}$ to $u^\star$ if $w < r$, or set $u^{(s+1)}$ to $u^\star$ otherwise.
2. Update V: first sample $v^\star \sim g_v[v^\star|u^{(s+1)}, v^{(s)}]$. Then,
   - Compute the acceptance ratio
   $$r = \frac{p_0(u^{(s+1)}, v^\star)}{p_0(u^{(s+1)}, v^{(s)})} \times \frac{g_v[v^{(s)}|u^{(s+1)}, v^\star]}{g_v[v^\star|u^{(s+1)}, v^{(s)}]}.$$
   - Sample $w \sim U(0,1)$. Set $v^{(s+1)}$ to $v^\star$ if $w < r$, or set $v^{(s+1)}$ to $v^\star$ otherwise.

Metropolis-Hastings algorithm

- The acceptance ratio looks like what we had before except with an additional factor.
- That factor is the ratio of the probability of generating the current value from the proposed to the probability of generating the proposed value from the current (ratio is equal to one for symmetric proposal – see homework!).
- Also, it is often the case that full conditionals are available for some parameters but not all.
- Very useful trick is to combine Gibbs and Metropolis. We may get to that very briefly next time if we have time.