

Metropolis-Hastings algorithm

Dr. Jarad Niemi

STAT 544 - Iowa State University

March 26, 2024

Outline

- Metropolis-Hastings algorithm
- Independence proposal
- Random-walk proposal
 - Optimal tuning parameter
 - Binomial example
 - Normal example
 - Binomial hierarchical example

Metropolis-Hastings algorithm

Let

- $p(\theta|y)$ be the target distribution and
- $\theta^{(t)}$ be the current draw from $p(\theta|y)$.

The Metropolis-Hastings algorithm performs the following

1. propose $\theta^* \sim g(\theta|\theta^{(t)})$
2. accept $\theta^{(t+1)} = \theta^*$ with probability $\min\{1, r\}$ where

$$r = r(\theta^{(t)}, \theta^*) = \frac{p(\theta^*|y)/g(\theta^*|\theta^{(t)})}{p(\theta^{(t)}|y)/g(\theta^{(t)}|\theta^*)} = \frac{p(\theta^*|y)}{p(\theta^{(t)}|y)} \frac{g(\theta^{(t)}|\theta^*)}{g(\theta^*|\theta^{(t)})}$$

otherwise, set $\theta^{(t+1)} = \theta^{(t)}$.

Metropolis-Hastings algorithm

Suppose we only know the target up to a normalizing constant, i.e.

$$p(\theta|y) = q(\theta|y)/q(y)$$

where we only know $q(\theta|y)$.

The Metropolis-Hastings algorithm performs the following

1. propose $\theta^* \sim g(\theta|\theta^{(t)})$
2. accept $\theta^{(t+1)} = \theta^*$ with probability $\min\{1, r\}$ where

$$r = r(\theta^{(t)}, \theta^*) = \frac{p(\theta^*|y)}{p(\theta^{(t)}|y)} \frac{g(\theta^{(t)}|\theta^*)}{g(\theta^*|\theta^{(t)})} = \frac{q(\theta^*|y)/q(y)}{q(\theta^{(t)}|y)/q(y)} \frac{g(\theta^{(t)}|\theta^*)}{g(\theta^*|\theta^{(t)})} = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \frac{g(\theta^{(t)}|\theta^*)}{g(\theta^*|\theta^{(t)})}$$

otherwise, set $\theta^{(t+1)} = \theta^{(t)}$.

Two standard Metropolis-Hastings algorithms

- Independent Metropolis-Hastings
 - Independent proposal, i.e. $g(\theta|\theta^{(t)}) = g(\theta)$
- Random-walk Metropolis
 - Symmetric proposal, i.e. $g(\theta|\theta^{(t)}) = g(\theta^{(t)}|\theta)$ for all $\theta, \theta^{(t)}$.

Independence Metropolis-Hastings

Let

- $p(\theta|y) \propto q(\theta|y)$ be the target distribution,
- $\theta^{(t)}$ be the current draw from $p(\theta|y)$, and
- $g(\theta|\theta^{(t)}) = g(\theta)$, i.e. the proposal is **independent** of the current value.

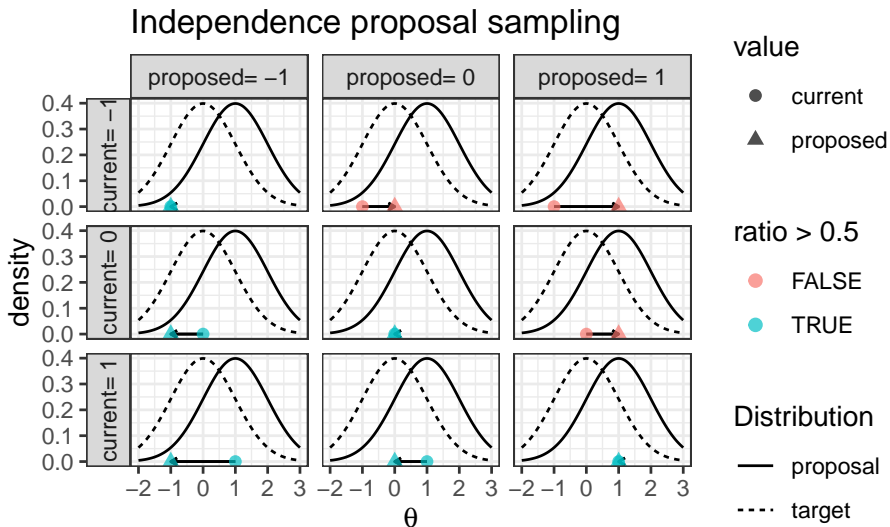
The **independence Metropolis-Hastings algorithm** performs the following

1. propose $\theta^* \sim g(\theta)$
2. accept $\theta^{(t+1)} = \theta^*$ with probability $\min\{1, r\}$ where

$$r = \frac{q(\theta^*|y)/g(\theta^*)}{q(\theta^{(t)}|y)/g(\theta^{(t)})} = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \frac{g(\theta^{(t)})}{g(\theta^*)}$$

otherwise, set $\theta^{(t+1)} = \theta^{(t)}$.

Intuition through examples



Example: Normal-Cauchy model

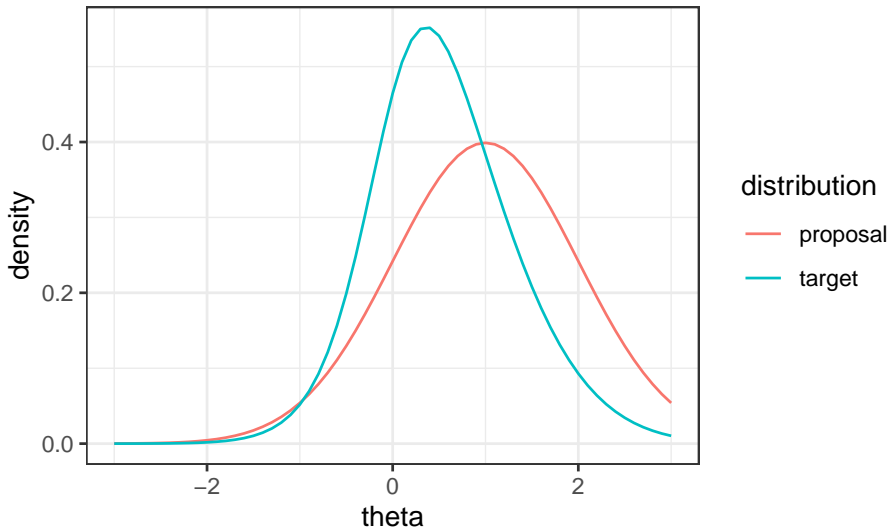
Let $Y \sim N(\theta, 1)$ with $\theta \sim Ca(0, 1)$ such that the posterior is

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

Use $N(y, 1)$ as the proposal, then the Metropolis-Hastings acceptance probability is the $\min\{1, r\}$ with

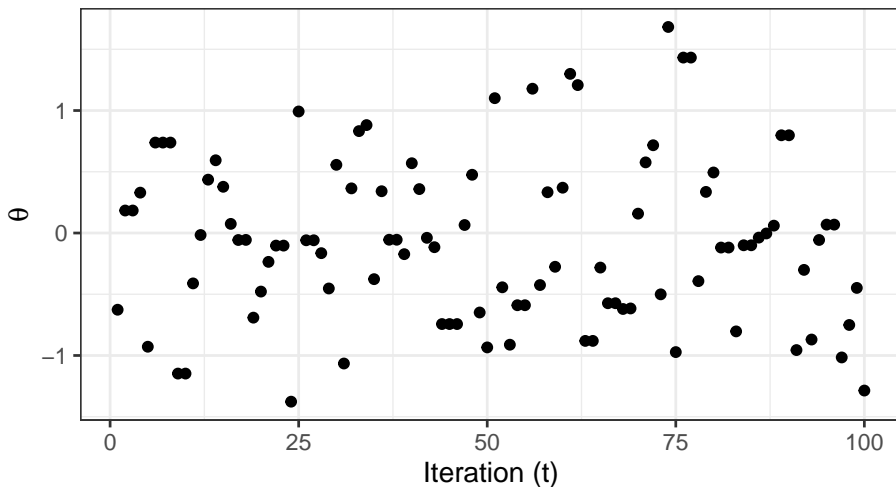
$$\begin{aligned} r &= \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \frac{g(\theta^{(t)})}{g(\theta^*)} \\ &= \frac{\exp(-(y-\theta^*)^2/2)/1+(\theta^*)^2}{\exp(-(y-\theta^{(t)})^2/2)/1+(\theta^{(t)})^2} \frac{\exp(-(\theta^{(t)}-y)^2/2)}{\exp(-(\theta^*-y)^2/2)} \\ &= \frac{1+(\theta^{(t)})^2}{1+(\theta^*)^2} \end{aligned}$$

Example: Normal-Cauchy model

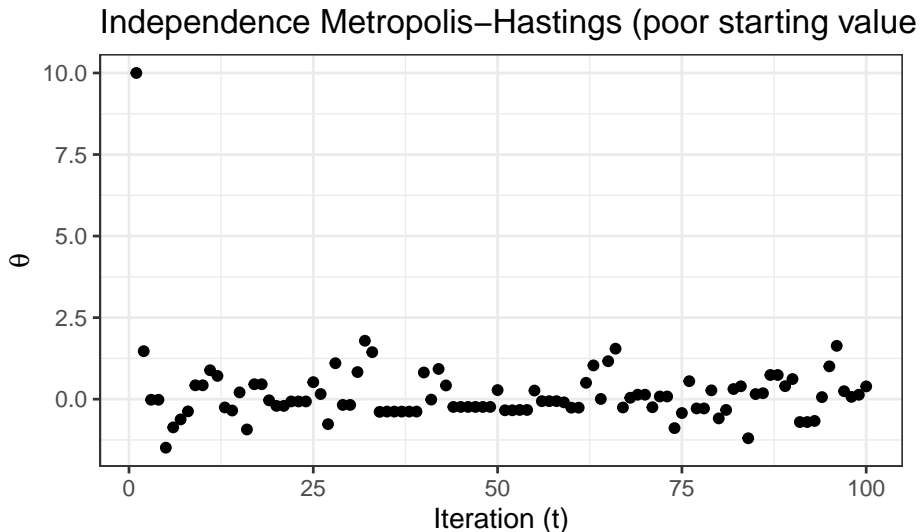


Example: Normal-Cauchy model

Independence Metropolis-Hastings



Example: Normal-Cauchy model - poor starting value



Need heavy tails

Recall that

- rejection sampling requires the proposal to have heavy tails and
- importance sampling is efficient only when the proposal has heavy tails.

Independence Metropolis-Hastings also requires heavy tailed proposals for efficiency since if $\theta^{(t)}$ is

- in a region where $p(\theta^{(t)}|y) \gg g(\theta^{(t)})$, i.e. target has heavier tails than the proposal, then
- any proposal θ^* such that $p(\theta^*|y) \approx g(\theta^*)$, i.e. in the center of the target and proposal,

will result in

$$r = \frac{g(\theta^{(t)})}{p(\theta^{(t)}|y)} \frac{p(\theta^*|y)}{g(\theta^*)} \approx 0$$

and few samples will be accepted.

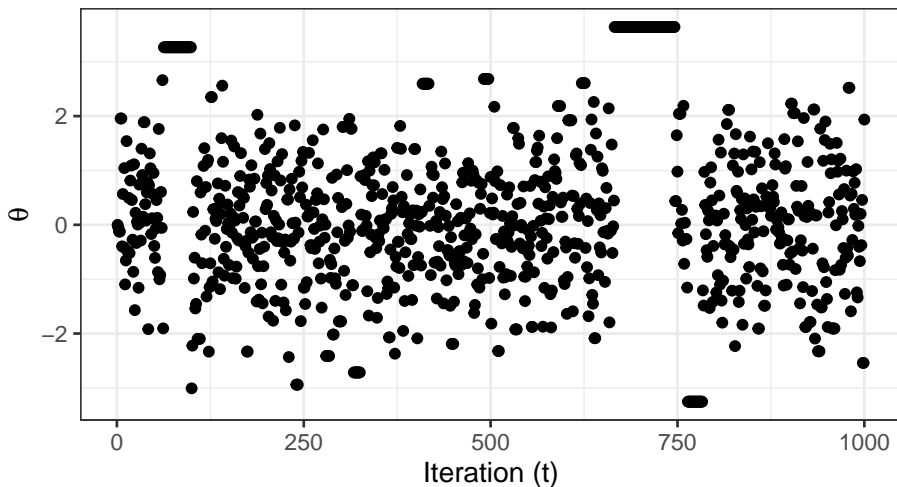
Need heavy tails - example

Suppose $\theta|y \sim Ca(0,1)$ and we use a standard normal as a proposal. Then

```
Error in melt(d[, c("x", "target", "proposal")], id.var = "x", variable.name = "density"): could not find function "melt"  
Error in eval(expr, envir, enclos): object 'g1' not found
```

Need heavy tails

Independence Metropolis-Hastings (light-tailed proposal)



Random-walk Metropolis

Let

- $p(\theta|y) \propto q(\theta|y)$ be the target distribution,
- $\theta^{(t)}$ be the current draw from $p(\theta|y)$, and
- $g(\theta^*|\theta^{(t)}) = g(\theta^{(t)}|\theta^*)$, i.e. the proposal is **symmetric**.

The **Metropolis algorithm** performs the following

1. propose $\theta^* \sim g(\theta|\theta^{(t)})$
2. accept $\theta^{(t+1)} = \theta^*$ with probability $\min\{1, r\}$ where

$$r = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \frac{g(\theta^{(t)}|\theta^*)}{g(\theta^*|\theta^{(t)})} = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)}$$

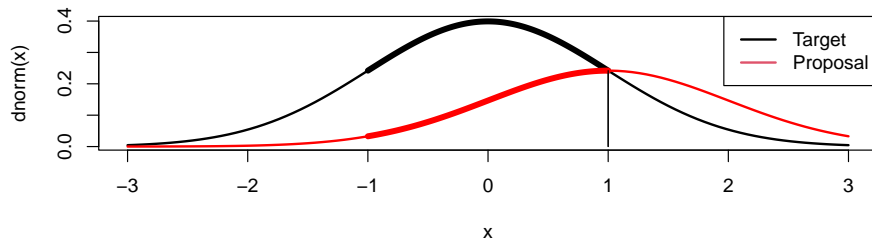
otherwise, set $\theta^{(t+1)} = \theta^{(t)}$.

This is also referred to as **random-walk Metropolis**.

Stochastic hill climbing

Notice that $r = q(\theta^*|y)/q(\theta^{(t)}|y)$ and thus will accept whenever the target density is larger when evaluated at the proposed value than it is when evaluated at the current value.

Suppose $\theta|y \sim N(0, 1)$, $\theta^{(t)} = 1$, and $\theta^* \sim N(\theta^{(t)}, 1)$.



Example: Normal-Cauchy model

Let $Y \sim N(\theta, 1)$ with $\theta \sim Ca(0, 1)$ such that the posterior is

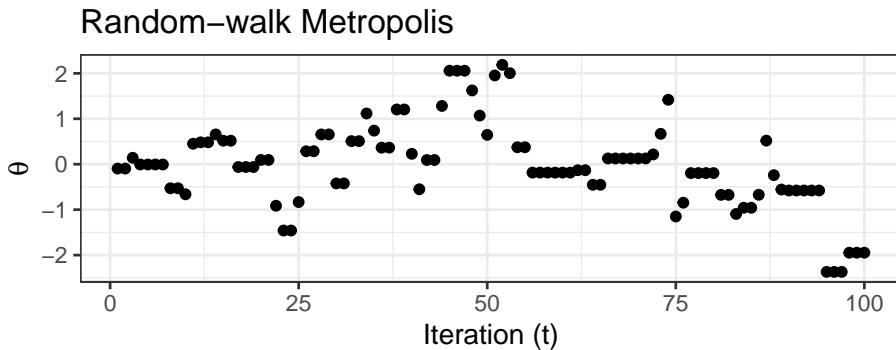
$$p(\theta|y) \propto p(y|\theta)p(\theta) \propto \frac{\exp(-(y - \theta)^2/2)}{1 + \theta^2}$$

Use $N(\theta^{(t)}, v^2)$ as the proposal, then the acceptance probability is the $\min\{1, r\}$ with

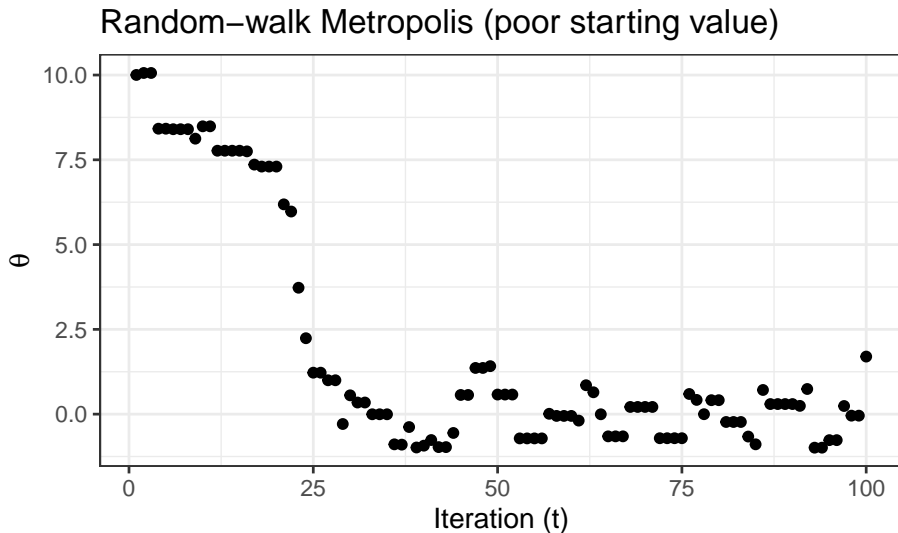
$$r = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(t)})p(\theta^{(t)})}.$$

For this example, let $v^2 = 1$.

Example: Normal-Cauchy model



Example: Normal-Cauchy model - poor starting value



Random-walk tuning parameter

Let $p(\theta|y)$ be the target distribution, the proposal is symmetric with scale v^2 , and $\theta^{(t)}$ is (approximately) distributed according to $p(\theta|y)$.

- If $v^2 \approx 0$, then $\theta^* \approx \theta^{(t)}$ and

$$r = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \approx 1$$

and all proposals are accepted, but $\theta^* \approx \theta^{(t)}$.

- As $v^2 \rightarrow \infty$, then $q(\theta^*|y) \approx 0$ since θ^* will be far from the mass of the target distribution and

$$r = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \approx 0$$

so all proposed values are rejected.

So there is an optimal v^2 somewhere. For normal targets, the optimal random-walk proposal variance is $2.4^2 \text{Var}(\theta|y)/d$ where d is the dimension of θ which results in an acceptance rate of 40% for $d = 1$ down to 20% as $d \rightarrow \infty$.

Random-walk with tuning parameter that is too big and too small

Let $y|\theta \sim N(\theta, 1)$, $\theta \sim Ca(0, 1)$, and $y = 1$.

```
Error in .(v): could not find function "."  
Error in eval(expr, envir, enclos): object 'r' not found
```

Binomial model

Let $Y \sim \text{Bin}(n, \theta)$ and $\theta \sim \text{Be}(1/2, 1/2)$, thus the posterior is

$$p(\theta|y) \propto \theta^{y-0.5}(1-\theta)^{n-y-0.5}\mathbf{I}(0 < \theta < 1).$$

To construct a random-walk Metropolis algorithm, we choose the proposal

$$\theta^* \sim N(\theta^{(t)}, 0.4^2)$$

and accept, i.e. $\theta^{(t+1)} = \theta^*$ with probability $\min\{1, r\}$ where

$$r = \frac{p(\theta^*|y)}{p(\theta^{(t)}|y)} = \frac{(\theta^*)^{y-0.5}(1-\theta^*)^{n-y-0.5}\mathbf{I}(0 < \theta^* < 1)}{(\theta^{(t)})^{y-0.5}(1-\theta^{(t)})^{n-y-0.5}\mathbf{I}(0 < \theta^{(t)} < 1)}$$

otherwise, set $\theta^{(t+1)} = \theta^{(t)}$.

Binomial model

```

n <- 10000
log_q <- function(theta, y = 3, n = 10) {
  if (theta < 0 | theta > 1) return(-Inf)
  (y-0.5) * log(theta) + (n-y-0.5) * log(1-theta)
}
current <- 0.5      # Initial value
samps <- rep(NA,n)
for (i in 1:n) {
  proposed <- rnorm(1, current, 0.4) # tuning parameter is 0.4

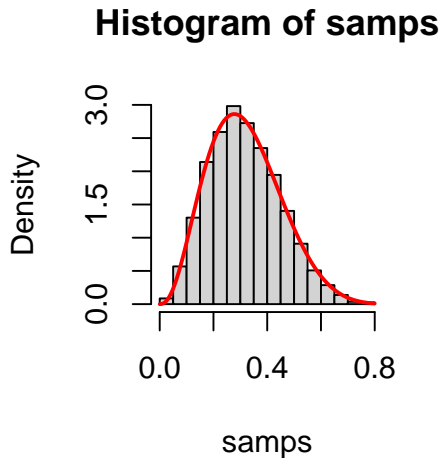
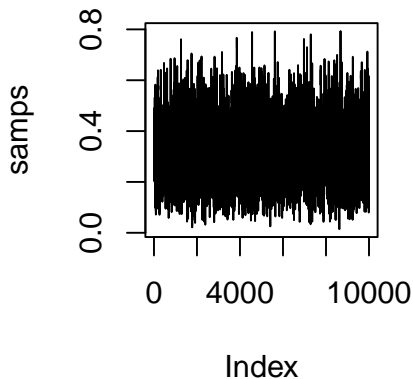
  logr <- log_q(proposed)-log_q(current)
  if (log(runif(1)) < logr) current <- proposed

  samps[i] <- current
}
length(unique(samps))/n # acceptance rate

[1] 0.3731

```

Binomial



Normal model

Assume

$$Y_i \stackrel{\text{ind}}{\sim} N(\mu, \sigma^2) \quad \text{and} \quad p(\mu, \sigma) \propto Ca^+(\sigma; 0, 1)$$

and thus

$$\begin{aligned} p(\mu, \sigma | y) &\propto \left[\prod_{i=1}^n \sigma^{-1} \exp \left(-\frac{1}{2\sigma^2} (y_i - \mu)^2 \right) \right] \frac{1}{1+\sigma^2} \mathbf{I}(\sigma > 0) \\ &= \sigma^{-n} \exp \left(-\frac{1}{2\sigma^2} \left[\sum_{i=1}^n y_i^2 - 2\mu n\bar{y} + \mu^2 \right] \right) \frac{1}{1+\sigma^2} \mathbf{I}(\sigma > 0) \end{aligned}$$

Perform a random-walk Metropolis using a normal proposal, i.e. if $\mu^{(t)}$ and $\sigma^{(t)}$ are the current values for μ and σ , then

$$\begin{pmatrix} \mu^* \\ \sigma^* \end{pmatrix} \sim N \left(\begin{bmatrix} \mu^{(t)} \\ \sigma^{(t)} \end{bmatrix}, S \right)$$

where S is the tuning parameter.

Adapting the tuning parameter

Recall that the optimal random-walk tuning parameter (if the target is normal) is $2.4^2 \text{Var}(\theta|y)/d$ where $\text{Var}(\theta|y)$ is the unknown posterior covariance matrix. We can estimate $\text{Var}(\theta|y)$ using the sample covariance matrix of draws from the posterior.

Proposed automatic adapting of the Metropolis-Hastings tuning parameter:

1. Start with S_0 . Set $b = 0$.
2. Run M iterations of the MCMC using $2.4^2 S_b/d$.
3. Set S_{b+1} to the sample covariance matrix of all previous draws.
4. If $b < B$, set $b = b + 1$ and return to step 2. Otherwise, throw away all previous draws and go to step 5.
5. Run K iterations of the MCMC using $2.4^2 S_B/d$.

R code for Metropolis-Hastings

```
n      <- 20
y      <- rnorm(n)
sum_y2 <- sum(y^2)
nybar  <- mean(y)

log_q  <- function(x) {
  if (x[2]<0) return(-Inf)
  -n*log(x[2])-(sum_y2-2*nybar*x[1]+n*x[1]^2)/(2*x[2]^2)-log(1+x[2]^2)
}

B <- 10
M <- 100

samps <- matrix(NA, nrow=B*M, ncol=2)
```

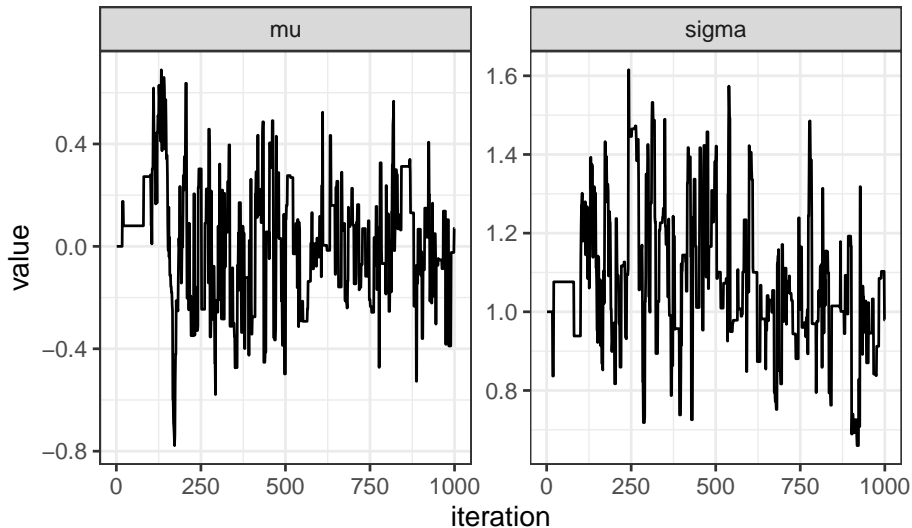
R code for Metropolis-Hastings - Adapting

```
# Adapt
for (b in 1:B) {
  for (m in 1:M) {
    i <- (b-1)*M+m

    proposed <- mvrnorm(1, current, 2.4^2*S/2)

    logr <- log_q(proposed) - log_q(current)
    if (log(runif(1)) < logr) current <- proposed
    samps[i,] <- current
  }
  a_rate[b] <- length(unique(samps[1:i,1]))/length(samps[1:i,1])
  S <- var(samps[1:i,])
}
```

Metropolis-Hastings - Adapting Traceplots



R code for Metropolis-Hastings - Inference

```
# Final run
K      <- 10000
samps <- matrix(NA, nrow = K, ncol = 2)

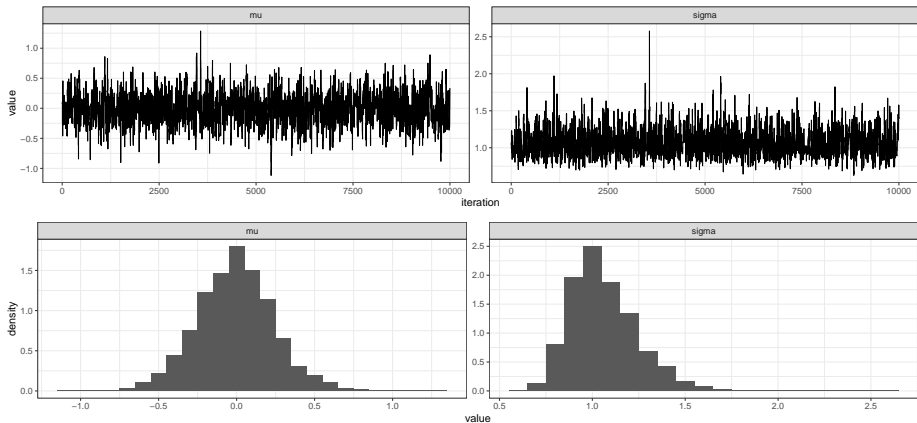
for (k in 1:K) {
  proposed <- mvrnorm(1, current, 2.4^2*S/2)

  logr <- log_q(proposed) - log_q(current)
  if (log(runif(1)) < logr) current <- proposed
  samps[k,] <- current
}

length(unique(na.omit(samps[,1]))) / length(na.omit(samps[,1])) # acceptance ra

[1] 0.3265
```

R code for Metropolis-Hastings - Inference



Hierarchical binomial model

Recall the hierarchical binomial model

$$Y_i \stackrel{\text{ind}}{\sim} \text{Bin}(n_i, \theta_i), \quad \theta_i \stackrel{\text{ind}}{\sim} \text{Be}(\alpha, \beta), \quad p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2}$$

and after marginalizing out the θ_i

$$Y_i \stackrel{\text{ind}}{\sim} \text{Beta-binomial}(n_i, \alpha, \beta), \quad p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2} \text{I}(a > 0) \text{I}(b > 0)$$

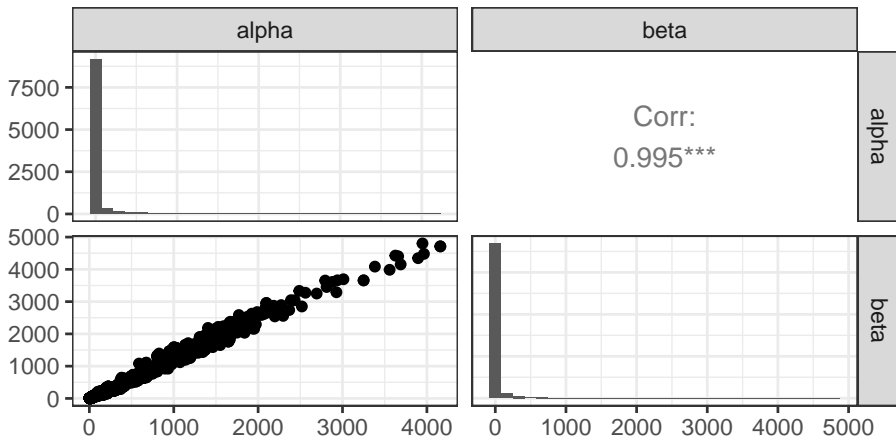
Thus the posterior is

$$p(\alpha, \beta | y) \propto \left[\prod_{i=1}^n \frac{B(\alpha + y_i, \beta + n_i - y_i)}{B(\alpha, \beta)} \right] (\alpha + \beta)^{-5/2} \text{I}(a > 0) \text{I}(b > 0)$$

where $B(\cdot)$ is the beta function.

We can perform exactly the same adapting procedure, but now using this posterior as the target distribution.

Beta-binomial hyperparameter posterior



Metropolis-Hastings summary

- The Metropolis-Hastings algorithm, samples $\theta^* \sim g(\cdot|\theta^{(t)})$ and sets $\theta^{(t+1)} = \theta^*$ with probability equal to $\min\{1, r\}$ where

$$r = \frac{q(\theta^*|y)}{q(\theta^{(t)}|y)} \frac{g(\theta^{(t)}|\theta^*)}{g(\theta^*|\theta^{(t)})}$$

and otherwise sets $\theta^{(t+1)} = \theta^{(t)}$.

- There are two common Metropolis-Hastings proposals
 - independent proposal: $g(\theta^*|\theta^{(t)}) = g(\theta^*)$
 - random-walk proposal: $g(\theta^*|\theta^{(t)}) = g(\theta^{(t)}|\theta^*)$
- Independent proposals suffer from the same heavy-tail problems as rejection sampling proposals.
- Random-walk proposals require tuning of the random walk parameter.