Metropolis-Hastings

Jonathan Navarrete

July 1, 2017

General Metropolis-Hastings

Given a target density f, we build a Markov kernel K with stationary distribution f and then generate a Markov chain X_t using this kernel so that the limiting distribution of X_t is f and integrals can be approximated according to the Ergodic Theorem.

The **Metropolis-Hastings algorithm** is a general purpose MCMC method for approximating a f. Given the target density f and a conditional density $q(y|x)$ that is easy to simulate from. In addition, q can be almost arbitrary in that the only theoretical requirements are that the ration $\frac{f(y)}{q(y|x)}$ is known up to a constant *independent* of x and that $q(\cdot|x)$ has enough dispertion to lead to an exploration of the entire support of f

We can rely on the feature of Metropolis-Hatings algorithm that for every given q, we can then construct a Metropolis-Hastings kernel such that f is its stationary distribution.

The Metropolis-Hastings algorithm as described Robert & Casella goes as follows Given $x^{(t)}$

1. Generate $Y_t \sim q(y|x_t)$
2. Take

$$X_{t+1} = \begin{cases} Y_t & \text{with probability } \ \rho(x^{(t)}, Y_t) \\ x^{(t)} & \text{with probability } \ 1 - \rho(x^{(t)}, Y_t) \end{cases}$$

   where

$$\rho(x^{(t)}, Y_t) = \min\{\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}\}$$

In simpler terms, as we want to generate $X \sim f$, we first take an initial value $x^{(0)}$ (which can almost be any artibrary value in the support of f).

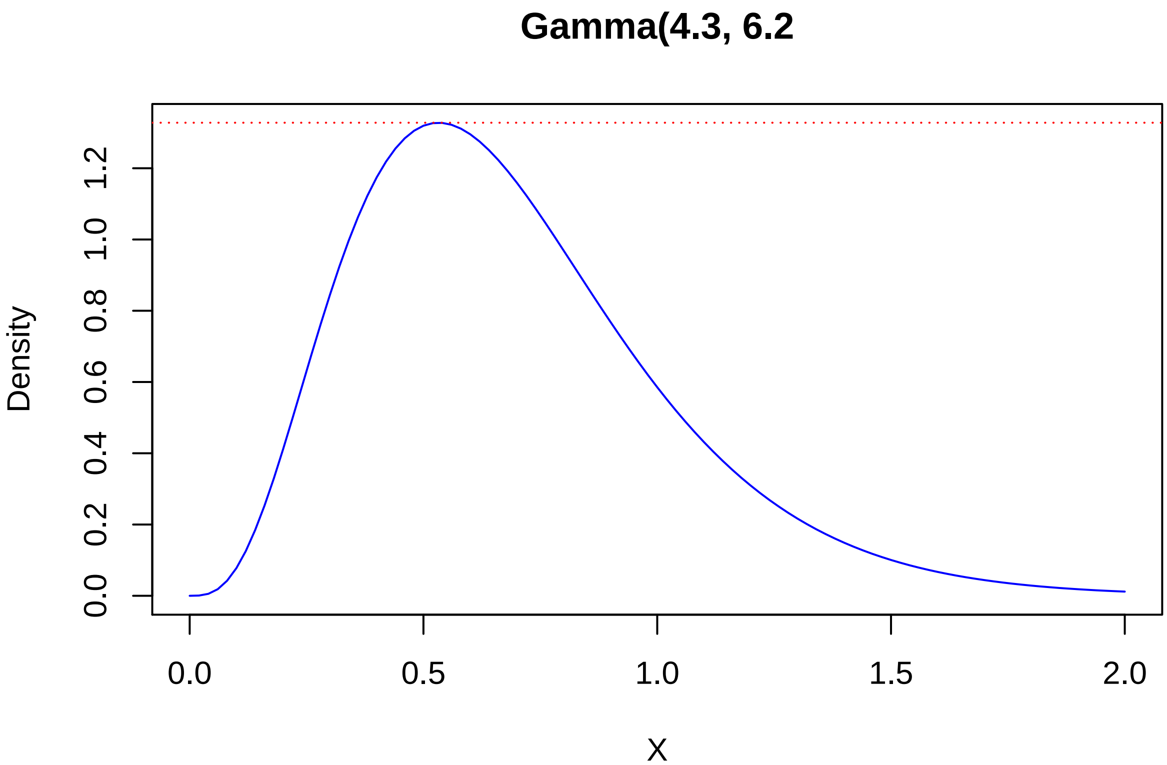
1. We generate a value $Y_0 \sim q(y|x^{(0)})$.
2. We calculate $\rho(x^{(t)}, Y_t)$
3. Generate a random value $U \sim Unif(0, 1)$
4. If $U < \rho(x^{(t)}, Y_t)$, then we accept $X^{(1)} = Y_t$; else we take $X^{(1)} = X^{(0)}$
5. Repeate steps 1-4 until you've satisfied the number of samples needed

Example: Gamma(4.3, 6.2)

As of now, we've covered multiple ways of generating random samples from a target density. Here we will compare the Accept-Reject algorithm against the Metropolis-Hastings. Generate N random variables $X \sim Gamma(4.3, 6.2)$.

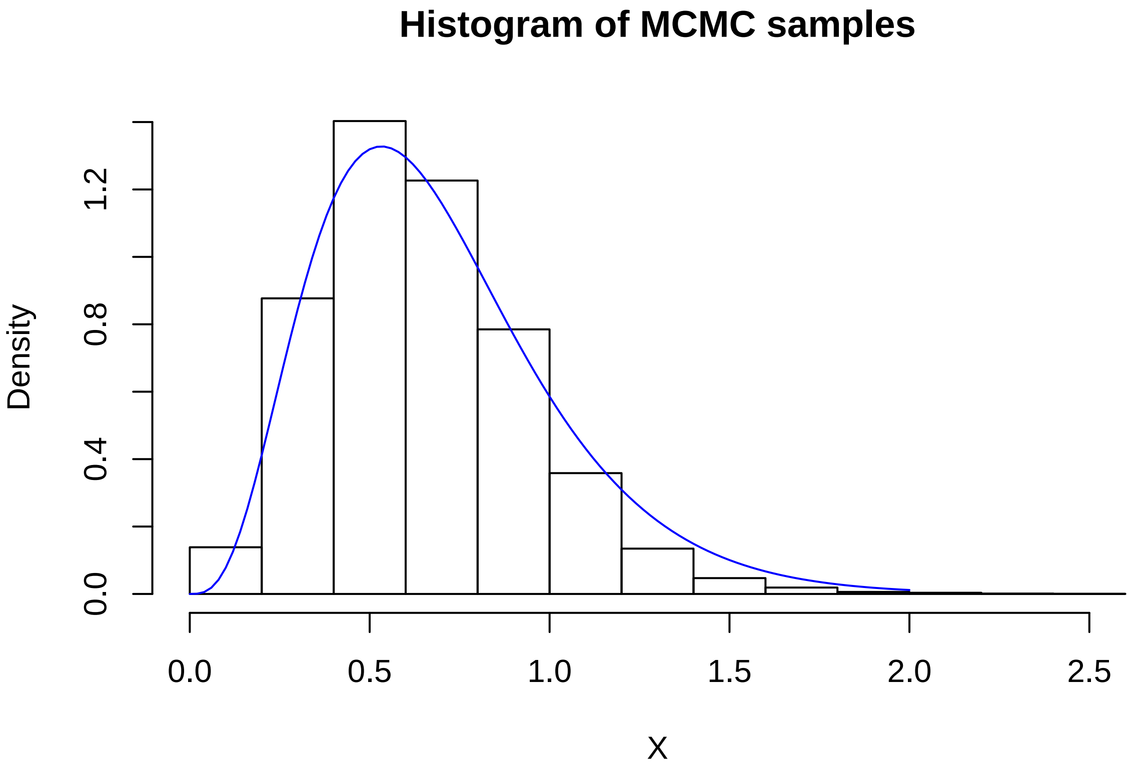
```
N = 10000

## For accept-reject, we need to find a value for M
## we can use `optimize` to find the maximum of our target density
maximum  = optimize(f = function(x){ dgamma(x = x, shape = 4.3, rate = 6.2)},
                    interval = c(0, 2), maximum = TRUE ) ## obtain maximum
```

```r
M = maximum$objective
curve(expr = dgamma(x = x, shape = 4.3, rate = 6.2),
      from = 0, to = 2, col = "blue",
      main = "Gamma(4.3, 6.2", xlab = "X", ylab = "Density")
abline(h = M, lty = 3, col = "red")
```




```r
f = function(x){
  dgamma(x = x, shape = 4.3, rate = 6.2)
}

g = function(x){
  dgamma(x = x, shape = 4, rate = 7)
}


#N = 10
X = numeric(N)
i = 0
while(i < N){
  Y = rgamma(n = 1, shape = 4, rate = 7)
  U = runif(1)
  if(U*M <= f(Y)/g(Y)){
    i = i + 1
    X[i] = Y
  }
}

## see how samples from chain compare to Gamma density
hist(X, main = "Histogram of MCMC samples", prob = TRUE)
```

```r
curve(expr = dgamma(x = x, shape = 4.3, rate = 6.2),
      from = 0, to = 2, add = TRUE, col = "blue")
```

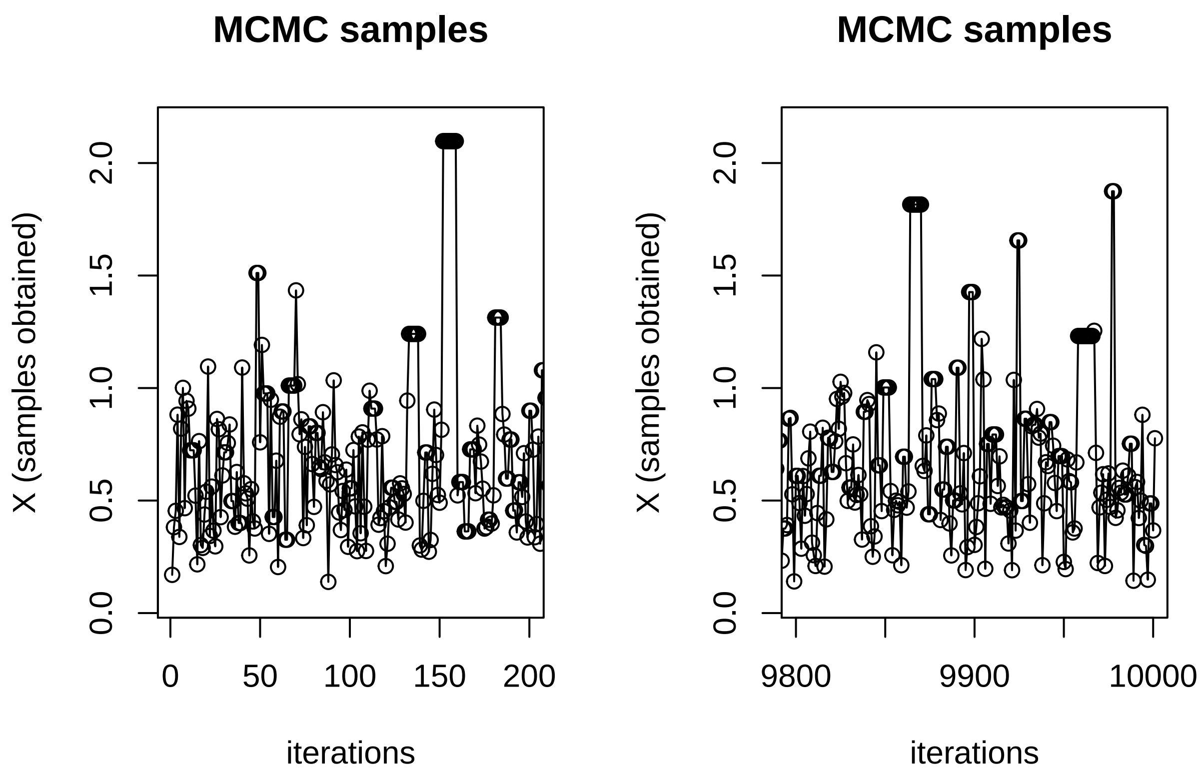



```r
## Metropolis Hastings

N = 10000
X = numeric(N)
X[1] = rgamma(n = 1, shape = 4.3, rate = 6.2)
for(i in 1:N){
  Y = rgamma(n = 1, shape = 4, rate = 7)
  rho = (dgamma(x = Y, shape = 4.3, rate = 6.2) * dgamma(x = X[i], shape = 4, rate = 7)) /
    (dgamma(x = X[i], shape = 4.3, rate = 6.2) * dgamma(x = Y, shape = 4, rate = 7))
  #X[i+1] = X[i] + (Y - X[i])*(runif(1) < rho) ## equivalent to if-else statement below
  if(runif(1) < rho){
    X[i+1] = Y
  } else{
    X[i+1] = X[i]
  }

}



mean(X)
```

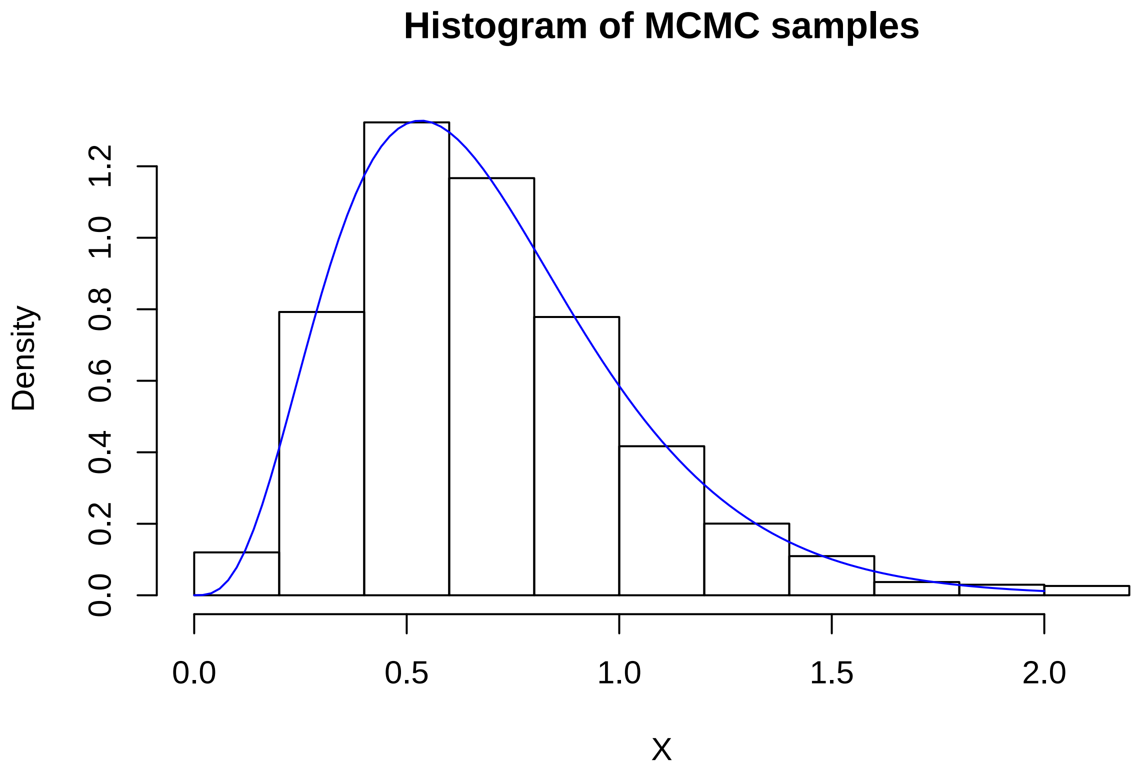
```
## [1] 0.6965631
```

```r
## see chain transitions
par(mfrow = c(1,2))
plot(X, type = "o", main = "MCMC samples",
```

```r
     xlim = c(1,200),
     xlab = "iterations", ylab = "X (samples obtained)")

plot(X, type = "o", main = "MCMC samples",
     xlim = c(N-200,N),
     xlab = "iterations", ylab = "X (samples obtained)")
```



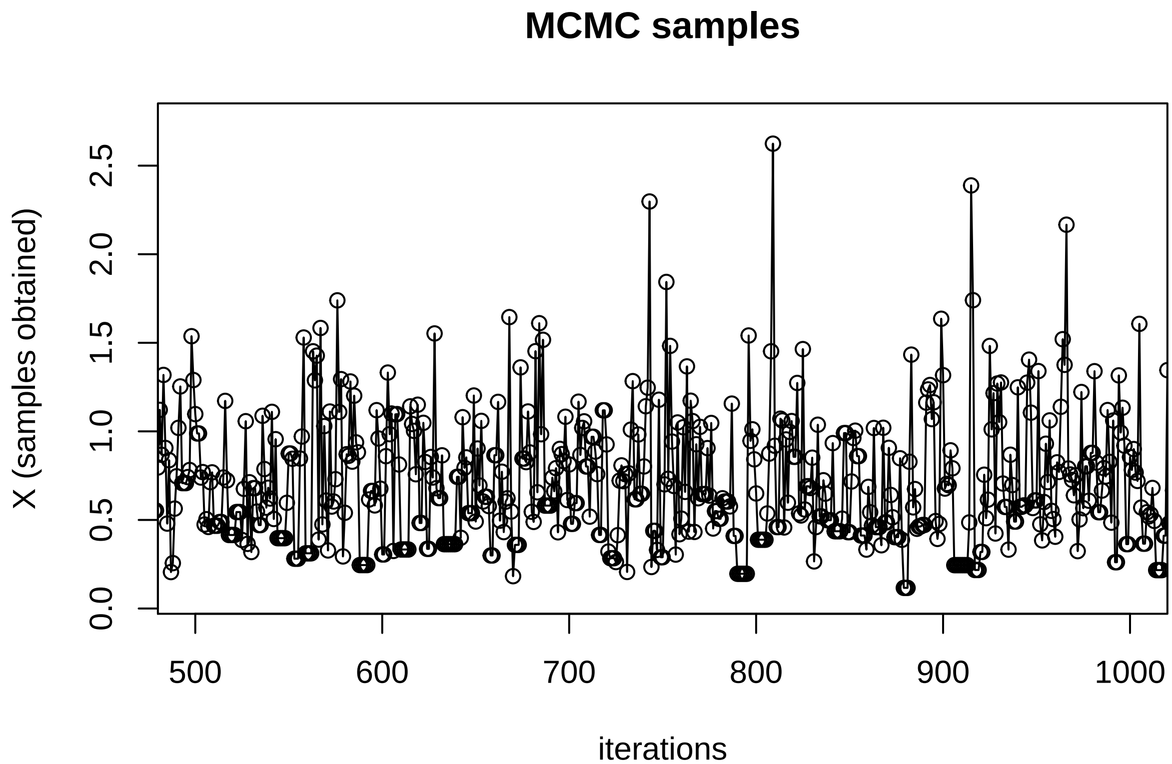

```r
par(mfrow = c(1,1))


## see how samples from chain compare to Gamma density
hist(X, main = "Histogram of MCMC samples", prob = TRUE)
curve(expr = dgamma(x = x, shape = 4.3, rate = 6.2),
      from = 0, to = 2, add = TRUE, col = "blue")
```




```r
## Metropolis Hastings
## now compare results with Gamma(5,6)

N = 10000
X = numeric(N)
X[1] = rgamma(n = 1, shape = 4.3, rate = 6.2)
for(i in 1:N){
  Y = rgamma(n = 1, shape = 5, rate = 6)
  rho = (dgamma(x = Y, shape = 4.3, rate = 6.2) * dgamma(x = X[i], shape = 5, rate = 6)) /
    (dgamma(x = X[i], shape = 4.3, rate = 6.2) * dgamma(x = Y, shape = 5, rate = 6))
  #X[i+1] = X[i] + (Y - X[i])*(runif(1) < rho) ## equivalent to if-else statement below
  if(runif(1) < rho){
    X[i+1] = Y
  } else{
    X[i+1] = X[i]
  }

}



mean(X)
```

```
## [1] 0.6974466
```

```r
## see chain transitions
plot(X, type = "o", main = "MCMC samples",
     xlim = c(500,1000),
     xlab = "iterations", ylab = "X (samples obtained)")
```




```r
## see how samples from chain compare to Gamma density
hist(X, main = "Histogram of MCMC samples", prob = TRUE)
curve(expr = dgamma(x = x, shape = 4.3, rate = 6.2),
      from = 0, to = 2, add = TRUE, col = "blue")
```