Random Num Generation HW6

Yichu Li.

10/16/2018

Abstract

In the project, we want to create a process that simulates observations from a complex density function, and practice rejection sampling method. Also, apply the method to Beta Mixture Model.

5.2.1 Rejection sampling

Show that g is a mixture of Gamma distributions.

First calculate the value of integral. The integral is in the form of a mixture of Gamma distributions with $\alpha = \theta, \theta + 1/2$ and $\beta = 1, 1$ correspondingly. According to the property of the Gamma distribution,

$$\int_0^\infty (2x^{\theta-1} + x^{\theta-1/2})e^{-x}dx = 2\int_0^\infty x^{\theta-1}e^{-x}dx + \int_0^\infty x^{\theta-1/2}e^{-x}dx \tag{1}$$

$$\tag{2}$$

$$= 2\Gamma(\theta) + \Gamma(\theta + 1/2) \tag{3}$$

According to the property of the Gamma distribution,

$$\int_0^\infty x^{\alpha-1}e^{-\beta/\alpha}dx = \beta^\alpha\Gamma(\alpha)$$

The value of the constant C is

$$C = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + 1/2)}$$

Since $g(x)$ is a probability density on $(0, \infty)$

$$g(x) = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + 1/2)}(2x^{\theta-1} + x^{\theta-1/2})e^{-x} \tag{4}$$

$$\tag{5}$$

$$g(x) = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + 1/2)}2x^{\theta-1}e^{-x} + x^{\theta-1/2}e^{-x} \tag{6}$$

$$\tag{7}$$

$$g(x) = \frac{2\Gamma(\theta)}{2\Gamma(\theta) + \Gamma(\theta + 1/2)}\frac{2x^{\theta-1}e^{-x}}{\Gamma(\theta)} + \frac{\Gamma(\theta + 1/2)}{2\Gamma(\theta) + \Gamma(\theta + 1/2)}\frac{x^{\theta-1/2}e^{-x}}{\Gamma(\theta + 1/2)} \tag{8}$$

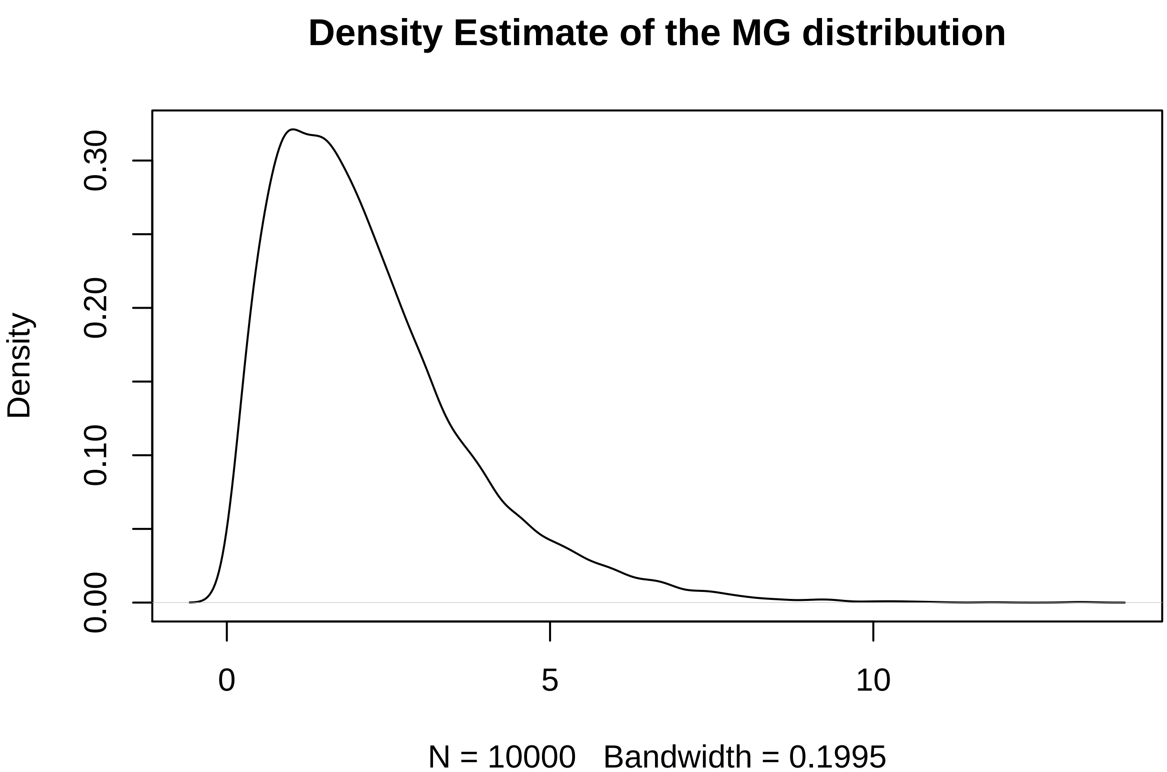
$$\tag{9}$$

$$\tag{10}$$

Design a procedure (pseudo-code) to sample from g(x)

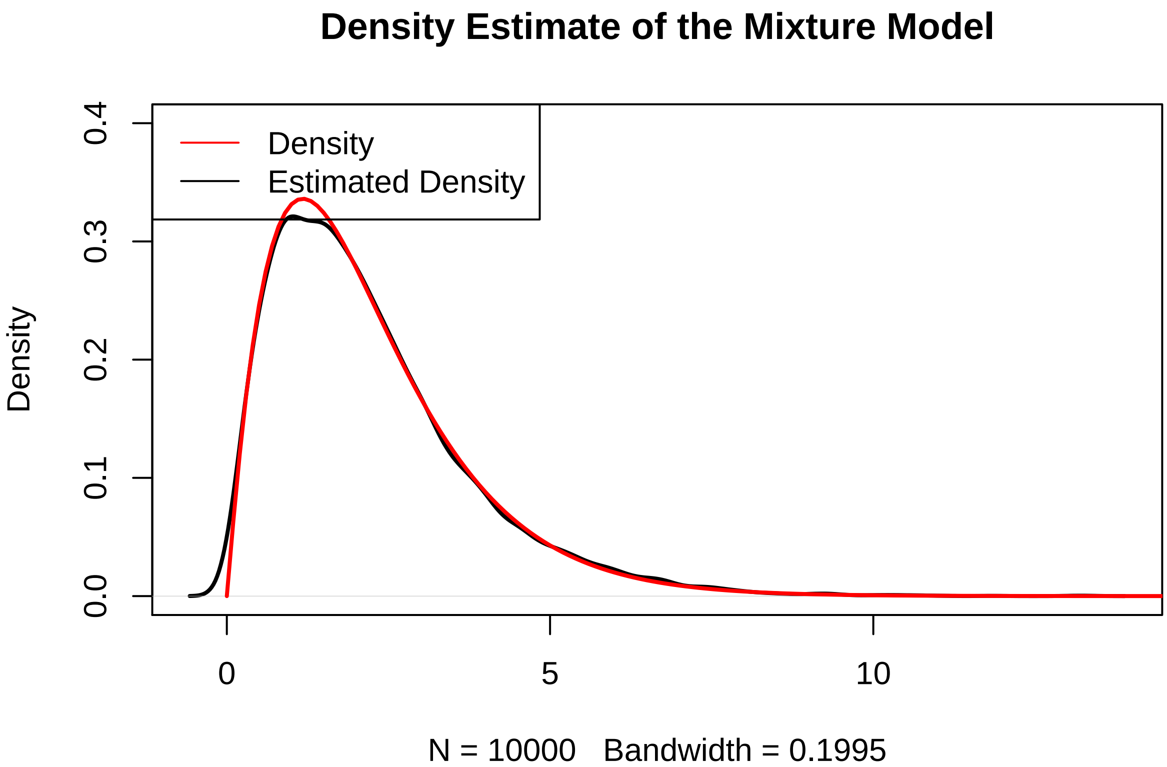
To sample from $g(x)$, we first generate random numbers U for weights from the standard uniform distribution $U(0,1)$ for $10,000$ times as required. Then, compare the values with w_1. If $U < w_1$, return $X \sim Gamma(\theta, 1)$; otherwise, return $X \sim Gamma(\theta + 1/2, 1)$.
We chose $\theta = 2$

```
N <- 10000
U <- runif(N)
rand.samples <- rep(NA,N)

theta <- 2
w1 <- 2*gamma(theta) / (2*gamma(theta) + gamma(theta + 0.5))
C <- 1/(2*gamma(theta) + gamma(theta + 0.5))

for(i in 1:N){
  if(U[i] < w1){
    rand.samples[i] <- rgamma(1,theta,1)
  }
  else{
    rand.samples[i] <- rgamma(1,theta + 0.5,1)
  }
}
plot(density(rand.samples), main = "Density Estimate of the MG distribution")
```




```
x = seq(0, 40, .1)
truth = w1*dgamma(x, theta, 1) + (1-w1) * dgamma(x,theta + 0.5, 1)
```

```r
plot(density(rand.samples),
     main = "Density Estimate of the Mixture Model",ylim = c(0,.4),lwd = 2)
lines(x, truth, col = "red",lwd = 2)

legend("topleft", c("Density","Estimated Density"),
        col = c("red","black"),cex = 1,lwd = 1)
```



Design a procedure (pseudo-code) to use rejection sampling to sample from f using g as the instrumental distribution.

We let $f(x) = \frac{q(x)}{C_1}$, and $q(x) = \sqrt{x+4}x^{\theta-1}e^{-x}$
First, find the value of α

$$\alpha = sup\frac{q(x)}{g(x)} = sup\frac{\sqrt{x+4}x^{\theta-1}e^{-x}}{C(2x^{\theta-1}+x^{\theta-1/2})e^{-x}} = sup\frac{\sqrt{x+4}}{C(2+x^{1/2})}$$

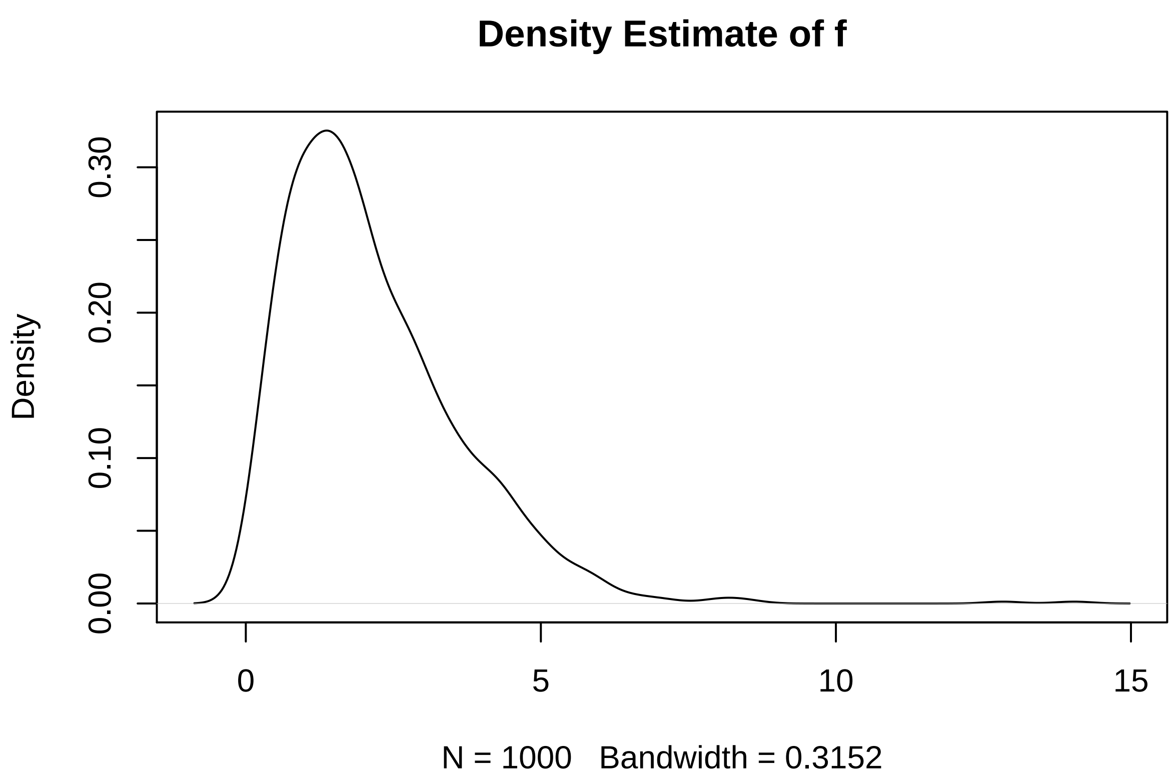
Let $y = \frac{\sqrt{x+4}}{C(2+x^{1/2})}$ and $y' = 0$. Solve it and we get $x = 4$ Thus,

$$\alpha = \frac{\sqrt{2}}{2C}$$

First sample $X \sim g(x)$ and $U \sim U(0,1)$. Then compare the values of U and $\frac{q(x)}{\alpha g(x)}$

```r
N <- 1000
samplef <- rep(NA,N)
theta <- 2
```

```r
w1 <- 2 * gamma(theta) / (2 * gamma(theta) + gamma(theta + 0.5))
C <- 1 / (2 * gamma(theta) + gamma(theta + 0.5))
alpha <- sqrt(2) / (2 * C)

for(k in 1:N){
  V <- runif(1)
  if(V < w1){
    X <- rgamma(1,theta,1)
  }
  else{
    X <- rgamma(1,theta+0.5,1)
  }
  U <- runif(1)
  q_x <- sqrt(4 + X) * X^(theta - 1) * exp(-1 * X)
  g_x <- C * (2 * X ^ (theta - 1) + X ^ (theta - 0.5)) * exp(-1 * X)
  m <- q_x/(alpha * g_x)
  if (U > m){cat("one more iteration"); k <- k-1; next}
  else samplef[k] <- X
  }

plot(density(samplef),main = "Density Estimate of f")
```



5.2.2 Mixture Proposal

Design a procedure (pseudo-code) to sample from f using a mixture of Beta distributions

Define $g(x)$ as a mixture of 2 Beta distribution $beta(\theta, 1)$ and $beta(1, \beta)$. Assume they have the weights of p_1 and p_2, $p_1 + p_2 = 1$. Thus,

$$g(x) = p_1 \frac{x^{\theta-1}}{beta(\theta, 1)} + p_2 \frac{(1-x)^{\beta-1}}{beta(1, \beta)} \tag{11}$$

$$\tag{12}$$

$$\alpha = sup \frac{q(x)}{g(x)} \tag{13}$$

The coefficient of $x^{2\theta-3}$ gives the equation

$$\frac{(\theta-1)(1+x^2) - 2x^2}{1+x^2} \frac{p_1}{beta(\theta, 1)} = \frac{1}{1+x^2}(\theta-1)\frac{p_1}{beta(\theta, 1)} \tag{14}$$

$$\tag{15}$$

$$(\theta-1) - \frac{2x^2}{1+x^2} = \theta - 1 \tag{16}$$

which has $x = 0$ as the only solution.

The coefficient of $x^{2\beta-3}$ gives the equation

$$[\frac{x(1-x)}{\sqrt{2+x^2}} - (\beta-1)\sqrt{2+x^2}] \frac{p_2}{beta(1, \beta)} = -\sqrt{2+x^2}(\beta-1)\frac{p_2}{beta(1, \beta)} \tag{17}$$

$$\tag{18}$$

$$x(1-x) = 0 \tag{19}$$

which has $x = 0, 1$ as the solutions.
Now take $x = 0$ back into the original numerator, it becomes

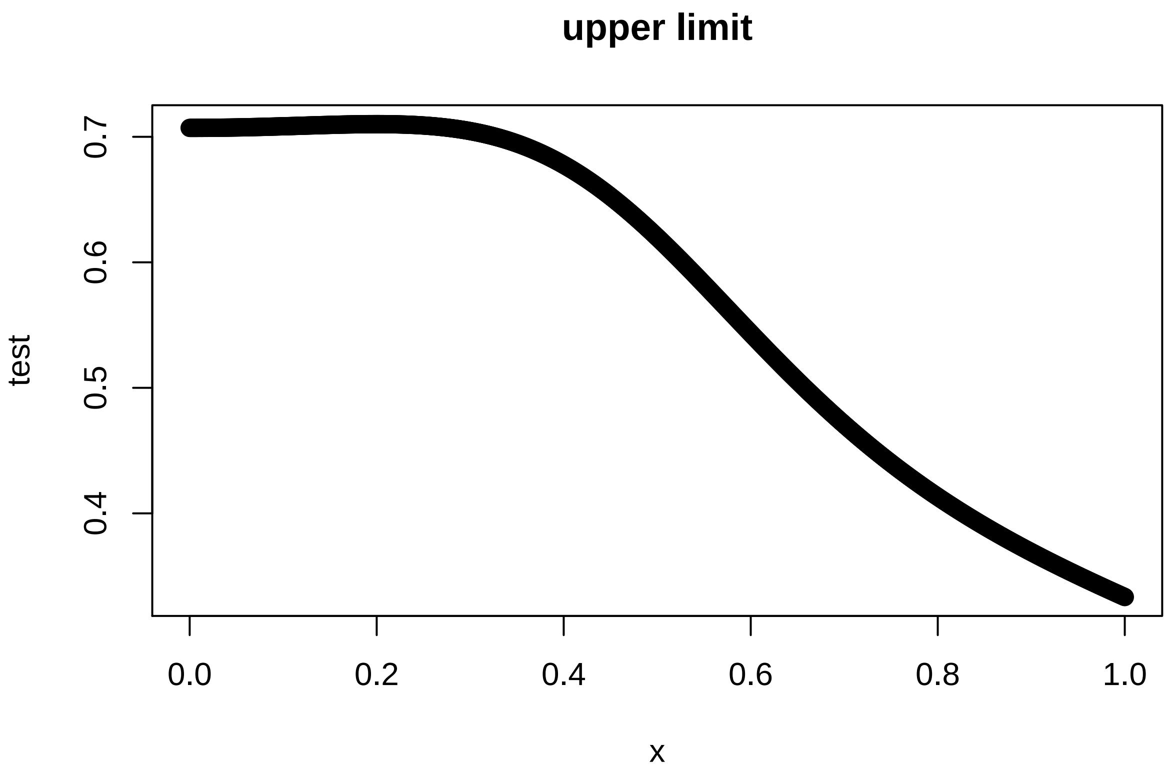
$$[-(\beta-1)\sqrt{2}]\frac{p_2}{beta(1, \beta)} - \sqrt{2}[-(\beta-1)\frac{p_2}{beta(1, \beta)}] = 0$$

Therefore, $x = 0$ is the point when the supremum is reached.
In our model, we let $(\theta, \beta) = (3, 4)$, which gives $p_1 = p_2 = 0.5$. In this way,

$$\alpha = \frac{q(0)}{g(0)} = \frac{\sqrt{2} beta(1, \beta)}{p_2} = 0.7071$$

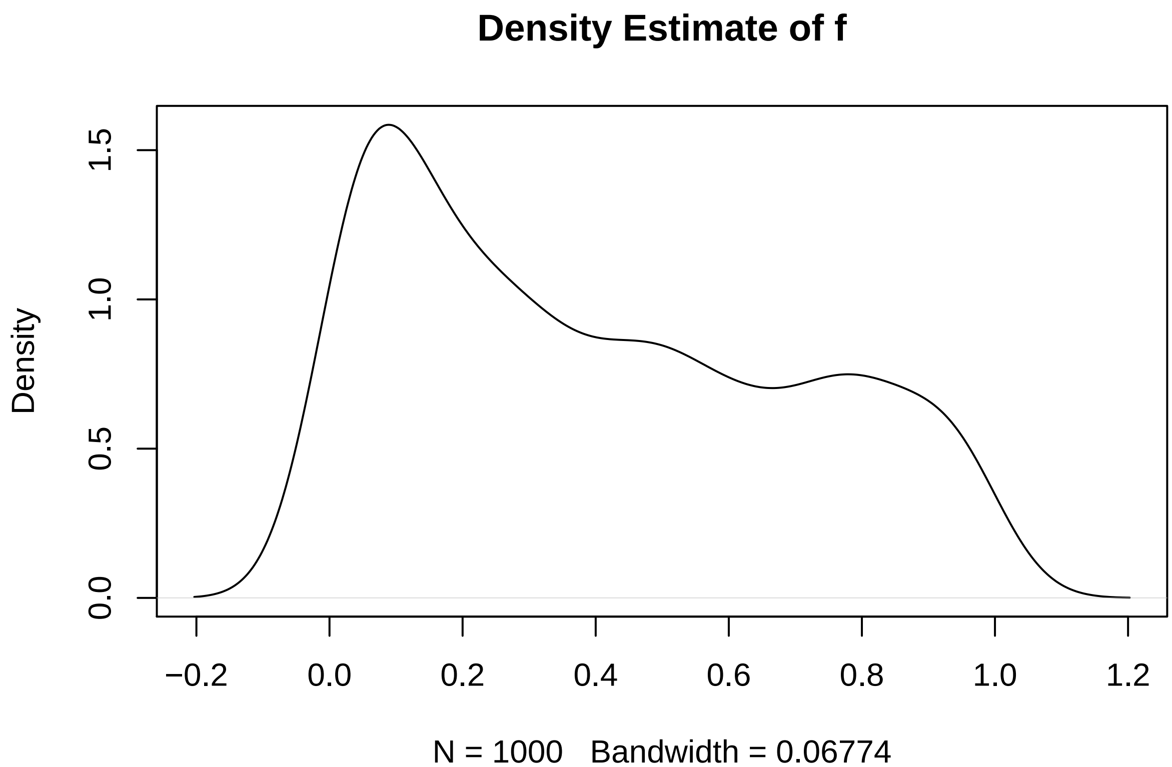
Compare the values of U and $\frac{q(x)}{\alpha g(x)}$: If U is bigger, return to the first step; return X otherwise. The returned value is a random sample from the density function $f(x)$.

```r
#Plotting
x <- seq(0,1,.001)
p1 <- 0.5
theta <- 3
beta <- 4
denomina <- p1*(1 / beta(theta,1)) * x ^ (theta-1) +
  (1-p1) * (1/beta(1,beta)) * ((1-x)^(beta-1))

nomina <- x^(theta - 1) / (1 + x^2) + sqrt(2 + x^2)*(1 - x)^(beta-1)
test <- nomina/denomina

plot(x, test, main = "upper limit", lwd = 2)
```




```r
X <- 0
denomina <- p1 * (1 / beta(theta,1)) * X^(theta-1) +
  (1-p1) * (1/beta(1, beta)) * ((1-X)^(beta-1))
nomina <- X^(theta-1) / (1 + X^2) + sqrt(2 + X^2) * (1 - X)^(beta - 1)
alpha <- nomina/denomina
N <- 1000
samplef <- rep(NA,N)
k <- 1
while(k <= N){
  V <- runif(1)
  if(V < p1){
    X <- rbeta(1, theta, 1)
  }
```

```
  else{
    X <- rbeta(1, 1, beta)
  }
  U <- runif(1)
  q_x <- X^(theta - 1) / (1 + X^2) + sqrt(2 + X^2) * (1-X)^(beta-1)
  g_x <- p1 * (1/beta(theta,1)) * X^(theta-1)
         + (1-p1) * (1/beta(1,beta)) * ((1-X)^(beta-1))
  m <- q_x / (alpha * g_x)
  if ( U > m ) {next}
  samplef[k] <- X
  k <- k + 1

}

plot(density(samplef), main = "Density Estimate of f")
```



Design a procedure dealing with the two components separately using individual Beta distributions.

We let $q_1(x) = \frac{x^{\theta-1}}{1+x^2}$, $g_1(x) \sim beta(\theta, 1)$, then we have

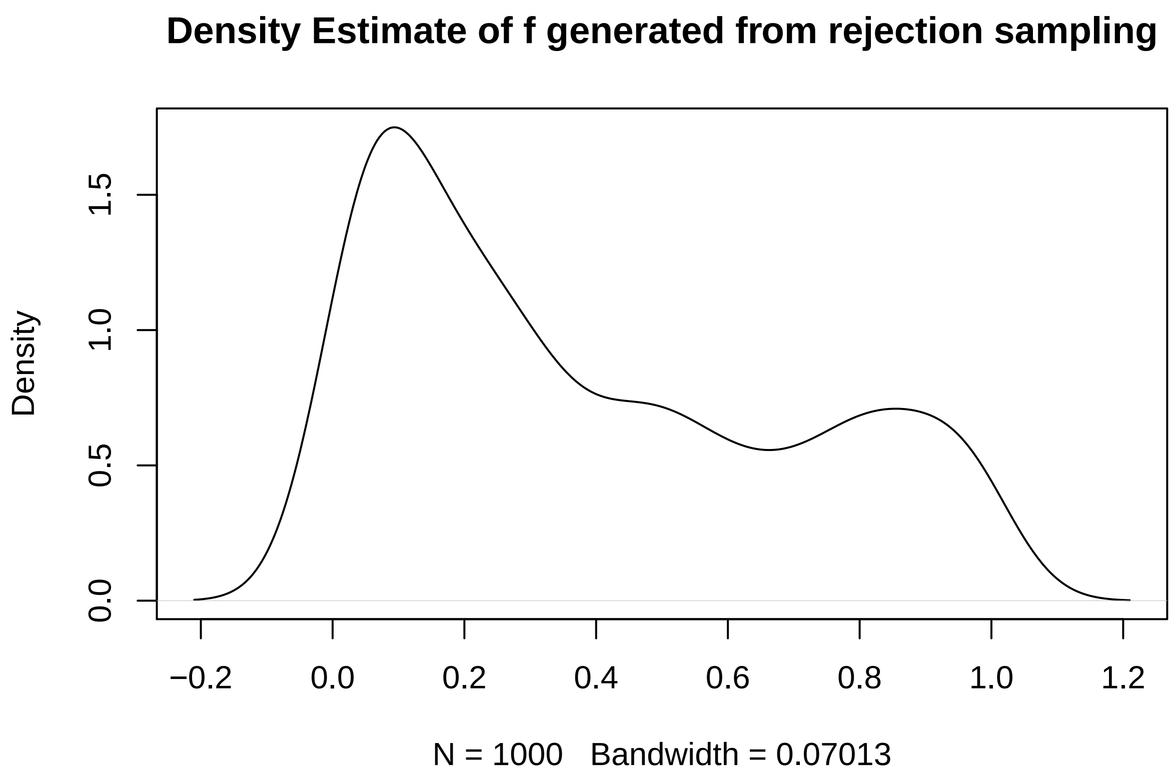
$$\alpha_1 = sup\frac{q_1(x)}{g_1(x)} = \frac{beta(\theta, 1)}{1 + x^2}$$

at $x = 0$, $\alpha_1 = beta(\theta, 1)$.
Similarly for $q_2(x) = \sqrt{2 + x^2}(1 - x)^{\beta-1}$, $g_2(x) \sim beta(1, \beta)$

$$\alpha_2 = sup\frac{q_2(x)}{g_2(x)} = beta(1, \beta)\sqrt{2 + x^2}$$

at $x = 1$, $\alpha_2 = \sqrt{3}beta(1, \beta)$
For the pseudo-code, we first sample k from $\{1, 2\}$ with probabilities $\{\frac{\alpha_1}{\alpha_1+\alpha_2}, \frac{\alpha_2}{\alpha_1+\alpha_2}\}$. Then, we generate V from $U(0, 1)$. Third, compare the values of V and $\frac{q_k(x)}{\alpha g_k(x)}$. If V is bigger, reject.

```r
N <- 1000
samplef <- rep(NA,N)
theta <- 3
beta <- 4
alpha_1 <- beta(theta,1)
alpha_2 <- sqrt(3)*beta(1,beta)
w1 <- alpha_1 / (alpha_1 + alpha_2)
k <- 1
while(k <= N){
  #sampling from g
  V <- runif(1)
  if(V < w1){
    #k=1
    X <- rbeta(1,theta,1)
    U <- runif(1)
    q_x <- X^(theta-1) / (1 + X^2)
    g_x <- X^(theta-1) / beta(theta,1)
    m <- q_x / (alpha_1 * g_x)
  }
  else{
    #k=2
    X <- rbeta(1,1,beta)
    U <- runif(1)
    q_x <- sqrt(2 + X^2)*(1 - X)^(beta - 1)
    g_x <- (1 - X)^(beta - 1) / beta(1, beta)
    m <- q_x / (alpha_2 * g_x)
  }
  if ( U > m ) { next }
  samplef[k] <- X
  k <- k+1
  }

plot(density(samplef), main = "Density Estimate of f generated from rejection sampling")
```


Density Estimate of f generated from rejection sampling
Density
0.0
0.5
1.0
1.5
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
N = 1000   Bandwidth = 0.07013