HW3

Xueying Zhou, Zimin Zhuang

2/23/2018

Problem1

(1)

For $P_{ij}^{(k+1)}$, based on the lecture notes, we have the following:

$$Q(\Psi|\Psi(k)) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(Z_{ij} = 1|y_i, \Psi(k)) \times lnP(Z_{ij} = 1, y_i|\Psi(k))$$

denote $p_{ij} = P(Z_{ij} = 1|y_i, \Psi(k))$, then by Bayes Rule, we can compute

$$p_{ij}^{(k+1)} = \frac{P(Z_{ij} = 1, y_i|\Psi(k))}{P(y_i|\Psi(k))} = \frac{P(Z_{ij} = 1, y_i|\Psi(k))}{\sum_{s=1}^{m} P(Z_{is} = 1, y_i|\Psi(k))}$$

And

$$P(Z_{ij} = 1, y_i|\Psi(k)) = P(Z_{ij} = 1|\Psi(k)) \times P(y_i|Z_{ij} = 1, \Psi(k))$$

$$= \pi_j^{(k)}\phi(y_i - \vec{x_i}^T\vec{\beta_j}^{(k)}; 0, \sigma^2)$$

Thus we have

$$p_{ij}^{(k+1)} = \frac{\pi_j^{(k)}\phi(y_i - \vec{x_i}^T\vec{\beta_j}^{(k)}; 0, \sigma^{2k})}{\sum_{j=1}^{m}\pi_j^{(k)}\phi(y_i - \vec{x_i}^T\vec{\beta_j}^{(k)}; 0, \sigma^{2k})}$$

(2)

From "(1)", we have

$$Q(\Psi|\Psi(k)) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} \times lnP(Z_{ij} = 1, y_i|\Psi(k))$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} \times ln[\pi_j^{(k)}\phi(y_i - \vec{x}_i^T\vec{\beta}_j^{(k)}; 0, \sigma^{2k})]$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}ln\pi_j^{(k)} + \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}ln(\frac{1}{\sqrt{2\pi}\sigma}) + \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}(-\frac{(y_i - \vec{x}_i^T\vec{\beta}_j^{(k)})^2}{2\sigma^2})$$

Only the third part contains $\vec{\beta}_j^{(k)}$, so in order to maximize this part, we can minimize $\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}(y_i - \vec{x}_i^T\vec{\beta}_j^{(k)})^2$, from the property of the sample mean, $\vec{x}_i^T\vec{\beta}_j^{(k)}$ must be the mean of the weighted sample $y_i(i = 1, 2, \ldots, n)$, each y_i has weight $p_{ij}^{(k+1)}$. So

$$\vec{x}_i^T\vec{\beta}_j^{(k+1)} = \frac{\sum_{i=1}^{n} p_{ij}^{(k+1)}y_{ij}}{\sum_{i=1}^{n} p_{ij}^{(k+1)}}$$

$$\vec{x}_i\vec{x}_i^T\vec{\beta}_j^{(k+1)} = \frac{(\sum_{i=1}^{n} \vec{x}_i p_{ij}^{(k+1)}y_i)}{\sum_{i=1}^{n} p_{ij}^{(k+1)}}$$

$$\vec{\beta}_j^{(k+1)} = (\sum_{i=1}^{n} \vec{x}_i\vec{x}_i^T p_{ij}^{(k+1)})^{-1} \times (\sum_{i=1}^{n} \vec{x}_i p_{ij}^{(k+1)}y_i), y = 1, 2, \ldots, m$$

which is $(2.b)$

Recall that in $Q(\Psi|\Psi(k))$ only the 2nd and 3rd parts, which we call I_2, I_3 contain σ^2, and if we denote

$$I_2^{(\star)} = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}ln(\sigma^2)$$

, and

$$I_3^{(\star)} = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}(y_i - \vec{x}_i^T\vec{\beta}_j^{(k)})^2/\sigma^2$$

then $I_2^{(\star)} + I_3^{(\star)}$ is the sum of m terms of the following form,

$$S_j = \sum_{i=1}^{n} p_{ij}^{(k+1)}ln(\sigma^2) + \sum_{i=1}^{n} p_{ij}^{(k+1)}(y_i - \vec{x}_i^T\vec{\beta}_j^{(k)})^2/\sigma^2$$

Thus we only need to find σ^2 to minimize each S_j. Now that $\vec{x_i}^T \vec{\beta_j}^{(k+1)}$ is equal to the weighted mean of y_i, to minimize S_j, also from the property of sample variance, σ^2 must be the sample variance of the weighted sample $y_1, y_2, \ldots, y_n$. Therefore,

$$\sigma_j^{2(k+1)} = \frac{\sum_{i=1}^n p_{ij}^{(k+1)}(y_i - \vec{x_i}^T \vec{\beta_j}^{(k)})^2}{\sum_{i=1}^n p_{ij}^{(k+1)}}$$

According to the given condition,

$$\sigma_1^{2(k+1)} = \sigma_2^{2(k+1)} = \cdots = \sigma_m^{2(k+1)} = \sigma^{2(k+1)}$$

So we have

$$\sigma^{2(k+1)} \sum_{i=1}^n p_{ij}^{(k+1)} = \sum_{i=1}^n p_{ij}^{(k+1)}(y_i - \vec{x_i}^T \vec{\beta_j}^{(k)})^2$$

$$\sigma^{2(k+1)} = \frac{\sum_{j=1}^m \sum_{i=1}^n p_{ij}^{(k+1)}(y_i - \vec{x_i}^T \vec{\beta_j}^{(k+1)})^2}{n}$$

which is $(2.c)$.

Finally, as $\sum_{j=1}^m \pi_j = 1$, we can construct the Lagrangian Function:

$$L(\pi_1^{(k)}, \ldots, \pi_m^{(k)}; \lambda) = Q(\Psi|\Psi(k)) - \lambda(\sum_{j=1}^m \pi_j^{(k)} - 1)$$

Set $\frac{\partial L}{\partial \pi_j^k} = 0,$(j=1, 2, $\ldots$, m), we have

$$\sum_{i=1}^n p_{ij}^{(k+1)} \frac{1}{\pi_j^{(k+1)}} - \lambda = 0, (j = 1, 2, \ldots, m)$$

$$\pi_j = \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{\lambda}$$

$$\sum_{j=1}^m \pi_j = \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{\lambda}$$

$$= \frac{n}{\lambda} = 1$$

$$\lambda = n$$

Thus we have $\pi_j^{(k+1)} = \frac{\sum_{i=1}^n p_{ij}^{(k+1)}}{n}$, which is $(2.a)$.

Problem2

(a)

First, note that the value of C can be obtained by calculating the reciprocal of the value of the integral. Thus, we first calculate the value of integral. Also note that, the integral is in the form of a mixture of Gamma distributions with $\alpha = \theta, \theta + 1/2$ and $\beta = 1, 1$ correspondingly. Recall that, according to the property of the Gamma distribution,

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

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

$$= 2\Gamma(\theta) + \Gamma(\theta + 1/2)$$

Recall that, according to the property of the Gamma distribution,

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

Therefore, the value of the constant C should be

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

Since $g(x)$ is a probability density on $(0, \infty)$, according to our work above, it should be

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

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

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

(b)

To sample from $g(x)$, we first generate random numbers U for weights from the standard uniform distribution $U(0, 1)$ for $100,000$ 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 $\theta = 2$

```r
#The number of samples from the mixture distribution
N <- 100000

#Sample N random uniforms U
U <- runif(N)

#Variable to store the samples from the mixture distribution
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))

#Sampling from the mixture
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)
  }
}

#Density plot of the random samples
plot(density(rand.samples), main = "Density Estimate of the Mixture Gamma distribution")
```
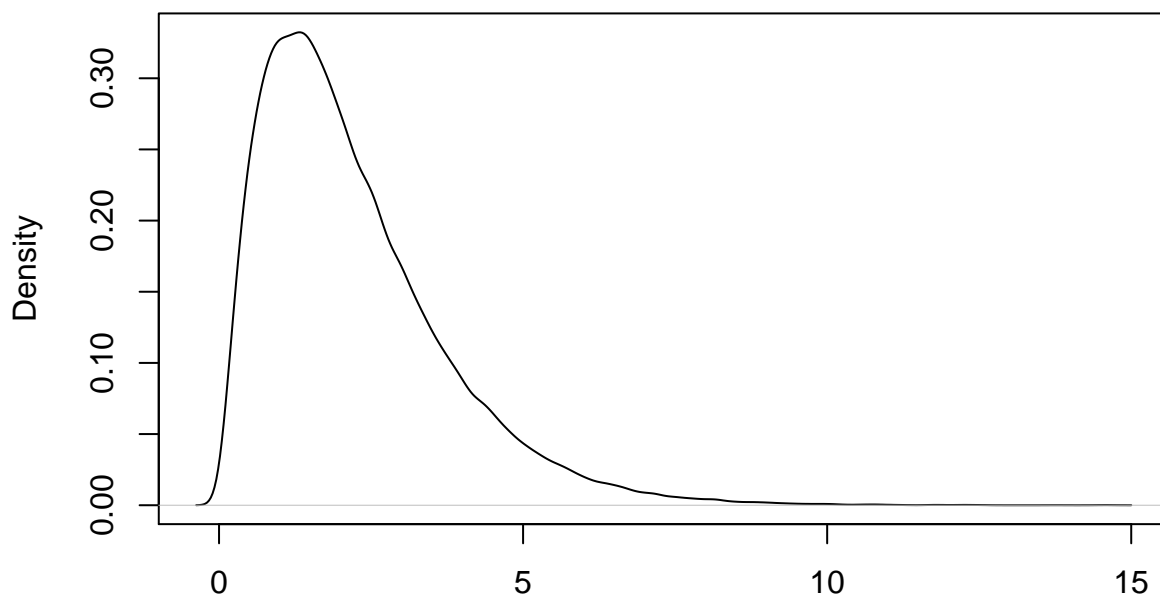
Density Estimate of the Mixture Gamma distribution
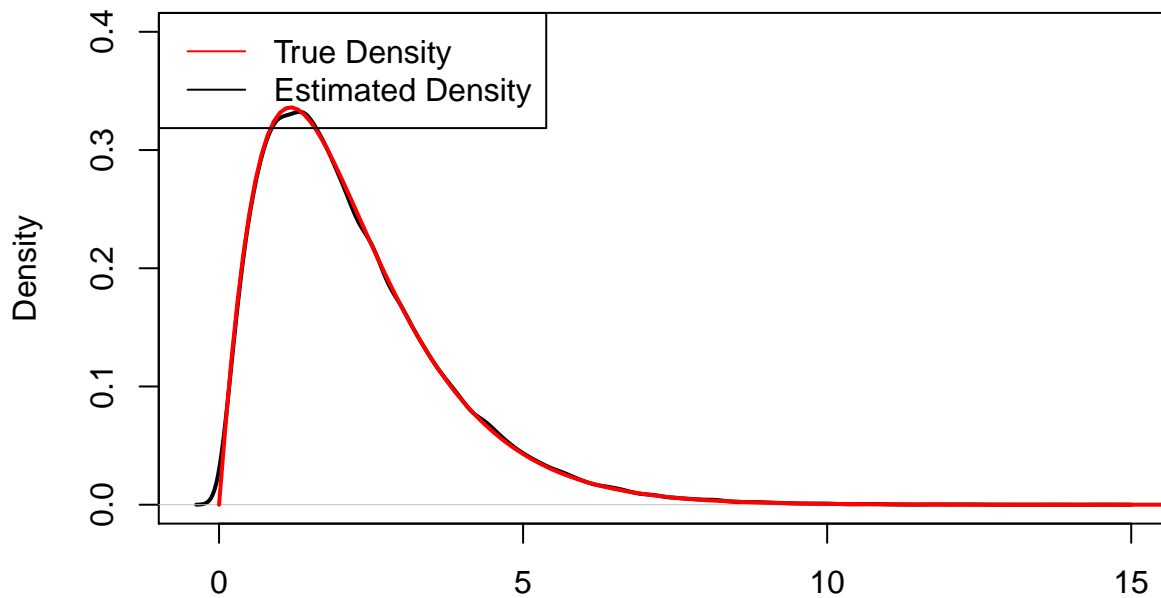

N = 100000   Bandwidth = 0.1259

```
#Plotting the true density as a sanity check
x = seq(0, 40, .1)
truth = w1*dgamma(x, theta, 1) + (1-w1) * dgamma(x,theta + 0.5, 1)
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("True Density","Estimated Density"),
        col = c("red","black"),cex = 1,lwd = 1)
```

Density Estimate of the Mixture Model

N = 100000   Bandwidth = 0.1259

(c)

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,

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

For the pseudo-code, we first sample $X \sim g(x)$ and $U \sim U(0,1)$. Then compare the values of U and
$\frac{q(x)}{\alpha g(x)}$: If U is bigger, return to the first step; return X otherwise.
The returned value is a random sample from the density function $f(x)$.

6

```r
#rejection sampling

N <- 1000
samplef <- 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))
alpha <- sqrt(2) / (2 * C)

for(k in 1:N){
  #sampling from g
  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 iter since this value is rejected"); k <- k-1; next}
  else samplef[k] <- X

}

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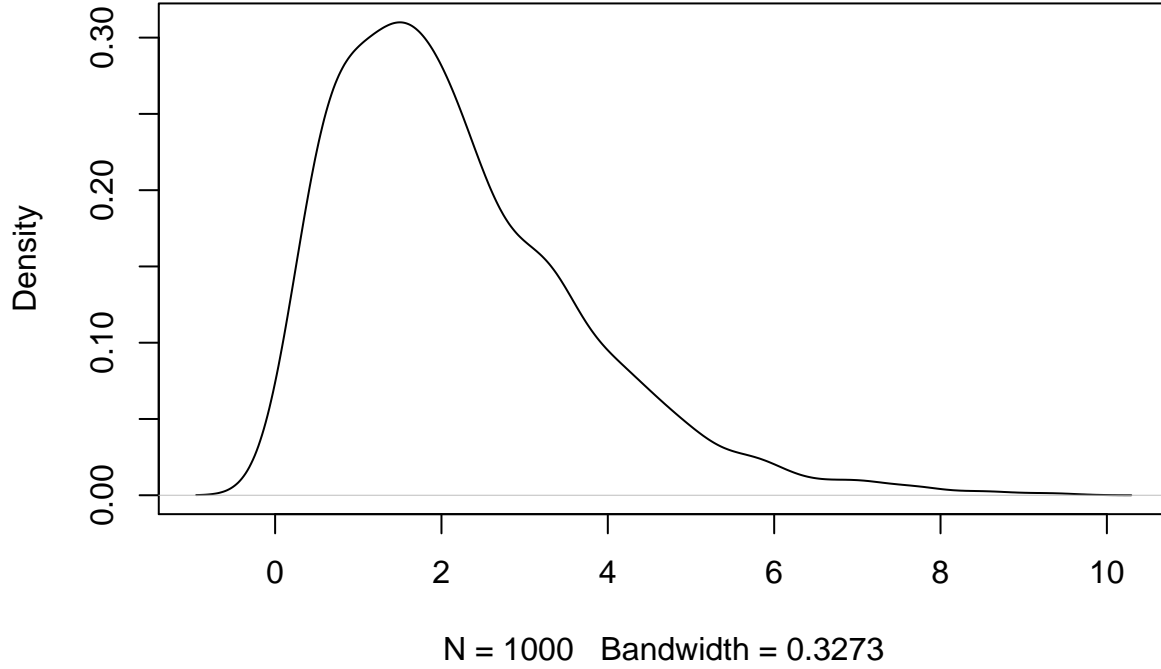
Density Estimate of f generated from rejection sampling

N = 1000   Bandwidth = 0.3273

Problem 3

(a)

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,

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

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

To calculate the value of α, we take the first derivative of $\frac{q(x)}{g(x)}$ and find the x when it is 0, which is equivalent to have the value of the numerator equal to zero in this case. Then we find the x to make the coefficients of $x^{2\theta-3}$ and $(1-x)^{2\beta-3}$ equal to 0.
The coefficient of $x^{2\theta-3}$ gives the equation

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

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

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

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

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

$$x(1-x)=0$$

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,

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

For the pseudo-code, we first sample $X\sim g(x)$ and $U\sim U(0,1)$. Then compare the values of U and $\frac{q(x)}{\alpha g(x)}$: If U is bigger, return to the first step; return X otherwise. The returned value is a random sample from the density function $f(x)$.

```
#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
#a
#find the upper limit alpha
plot(x, test, main = "find the upper limit", lwd = 2)
```

find the upper limit
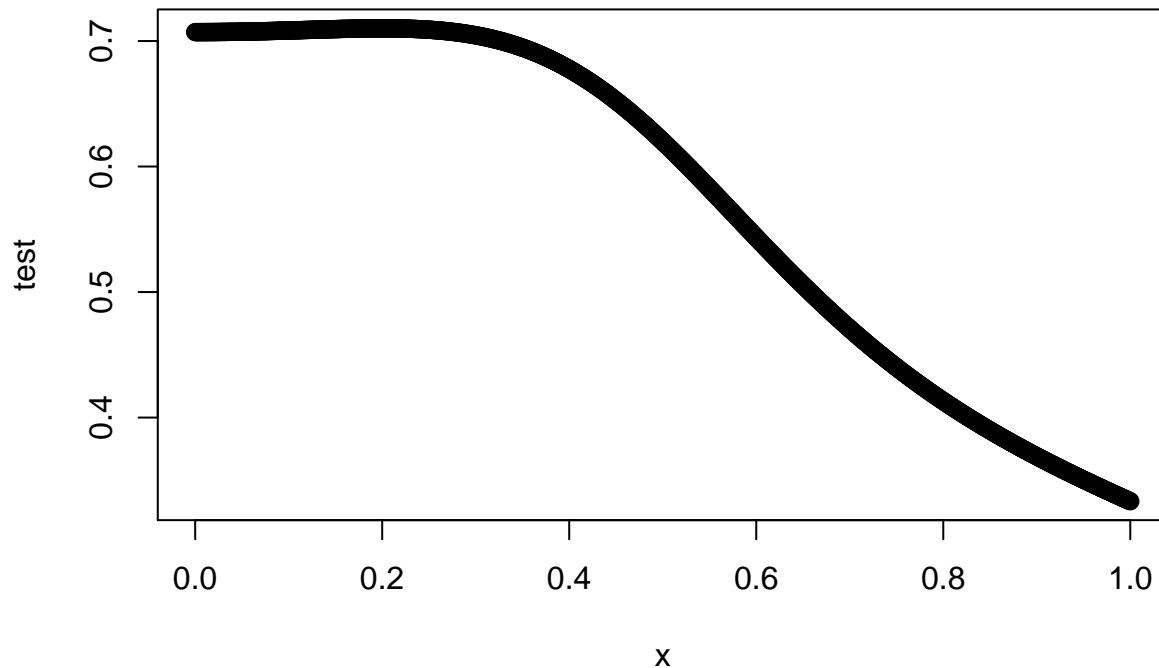

```r
X <- 0
denomina <- p1 * (1 / beta(theta,1)) * X^(theta-1) +
  (1-p1) * (1/beta(1, beta)) * ((1-X)^(beta-1))
nomina <- X^(theta-1) / (1 + X^2) + sqrt(2 + X^2) * (1 - X)^(beta - 1)
alpha <- nomina/denomina
alpha
```

```
## [1] 0.7071068
```

```r
N <- 1000
samplef <- rep(NA,N)
k <- 1
while(k <= N){
  #sampling from g
  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)
```
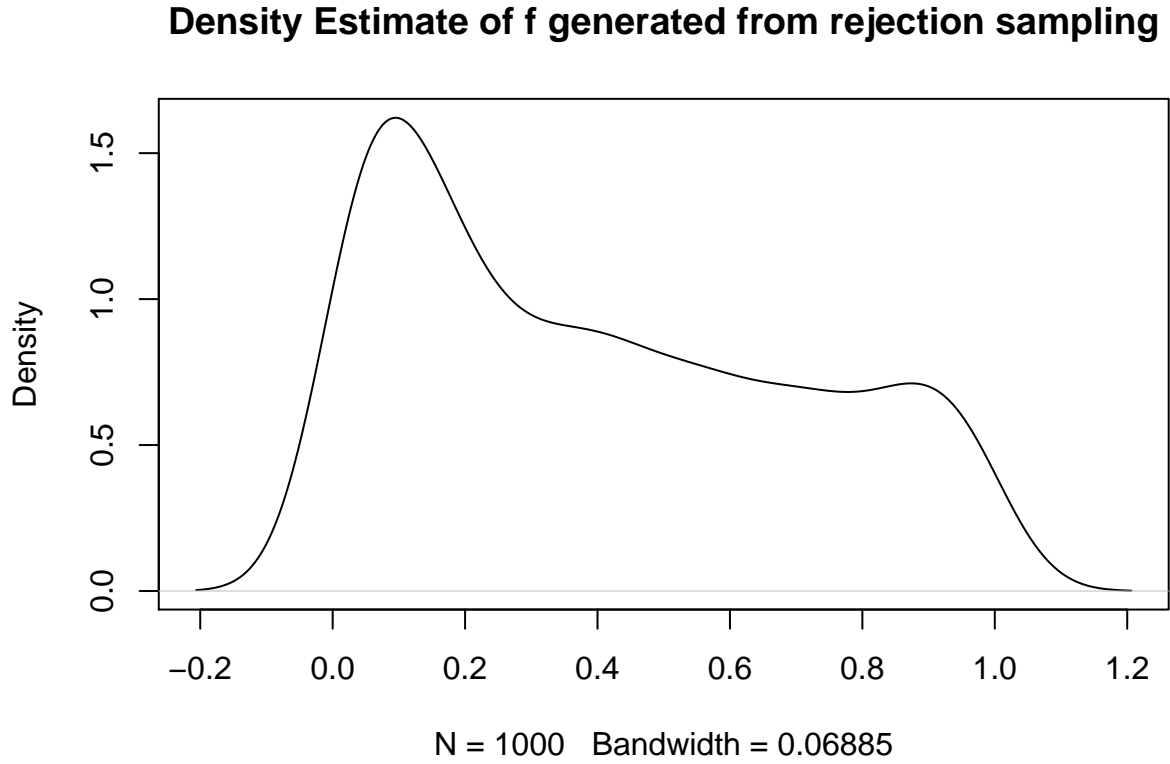
```
  if ( U > m ) {next}
  samplef[k] <- X
  k <- k + 1

}
```

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

Density Estimate of f generated from rejection sampling

N = 1000   Bandwidth = 0.06885

(b)

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; eccept otherwise.
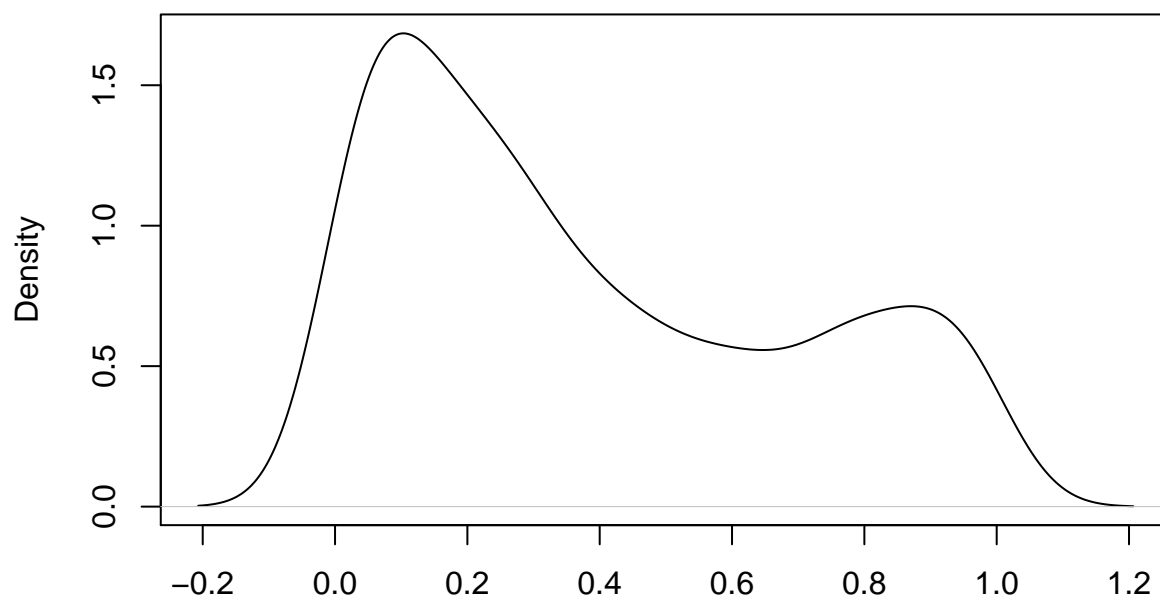
11

```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

N = 1000   Bandwidth = 0.06903