Homework3

Yuqian Liu

01 March 2018

Problem 1

Answer:

Verify the validity of the provided algorithm, which means to derive the updating rules in the given algorithm based on the construction of an EM algorithm.

$$
\begin{aligned}
&Q(\Psi|\Psi^{(k)}) = E_z\{l_n^c(\Psi)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}\{log\pi_j + log\phi(y_i - x_i^T\beta_j; 0, \sigma^2)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}\{log\pi_j + log\tfrac{1}{\sqrt{2\pi}\sigma} exp(-\tfrac{(y_i - x_i^T\beta_j)^2}{2\sigma^2})\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)}\{log\pi_j + log\tfrac{1}{\sqrt{2\pi}\sigma} + (-\tfrac{(y_i - x_i^T\beta_j)^2}{2\sigma^2})\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} log\pi_j - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} log2\pi\sigma^2 - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} \tfrac{(y_i - x_i^T\beta_j)^2}{2\sigma^2} \\
&= I_1 - \tfrac{I_2}{2} - \tfrac{I_3}{2}
\end{aligned}
\tag{1}
$$

Thus,

$$
\begin{aligned}
I_1 &= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} log\pi_j \\
I_2 &= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} log2\pi\sigma^2 \\
I_3 &= \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(k+1)} \tfrac{(y_i - x_i^T\beta_j)^2}{2\sigma^2}
\end{aligned}
\tag{2}
$$

Follow the lecture note, only I_3 contains β_j for every certain j in a quadratic form. TO minimize it, from the property of sample mean, β_j must be the mean of a weighted sample $\frac{y_1}{x_1^T}, ..., \frac{y_n}{x_n^T}$, each $\frac{y_i}{x_i^T}$ having weight $x_i x_i^T p_{ij}^{(k+1)}$. So

each xi having weight wik.

$$
\beta_j^{k+1} = \frac{\sum_{i=1}^{n} x_i p_{ij}^{(k+1)} y_i}{\sum_{i=1}^{n} x_i x_i^T p_{ij}^{(k+1)}}
\tag{3}
$$

Next, only I_2 and I_3 contain $\sigma^{2(k+1)}$, to minimize $I_2 + I_3$, σ^2 must be the sample variance of the weighted sample. So

$$
\begin{aligned}
\sigma^{2(k+1)} &= \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} p_{ij}^{(k+1)}(y_i - x_i^T\beta_j^{(k+1)} - 0)^2}{\sum_{j=1}^{m}\sum_{i=1}^{n} p_{ij}^{(k+1)}} \\
&= \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} p_{ij}^{(k+1)}(y_i - x_i^T\beta_j^{(k+1)})^2}{n}
\end{aligned}
\tag{4}
$$

Finally, only I_1 contains π_j in $Q(\Psi|\Psi^{(k)})$, and $\sum\limits_{j=1}^{m} \pi_j = 1$. By using Lagrange method, construct

$$\mathcal{L}(\pi_1, ..., \pi_m, \lambda) = Q(\Psi|\Psi^{(k)}) - \lambda(\sum_{j=1}^{m} \pi_j - 1) = 0 \tag{5}$$

In order to maxmize it, take first derivative of it respect to π_j and λ,

$$\begin{aligned} \mathcal{L}(\pi_1, ..., \pi_m, \lambda)'_{\pi_j} &= 0 \\ \mathcal{L}(\pi_1, ..., \pi_m, \lambda)'_{\lambda} &= 0 \end{aligned} \tag{6}$$

From $\mathcal{L}'_{\pi_j}$,

$$\sum_{i=1}^{n} p_{ij}^{(k+1)}(\frac{1}{\pi_j}) - \lambda = 0 \tag{7}$$

Thus,

$$\begin{aligned} &\pi_j = \frac{\sum_{i=1}^{n} p_{ij}^{(k+1)}}{\lambda} \\ &\because \sum_{j=1}^{m} \pi_j = 1 \\ &\sum_{j=1}^{m} \pi_j = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} p_{ij}^{(k+1)}}{\lambda} \\ &= \frac{n}{\lambda} \\ &\therefore \lambda = n \qquad and \qquad \pi_j = \frac{\sum_{i=1}^{n} p_{ij}^{(k+1)}}{n} \end{aligned} \tag{8}$$

Problem 2

Answer:

a) Because $g(x) \propto (2x^{\theta-1} + x^{\theta-1/2})e^{-x}$, the normalizing constant C for g is the constant such that $\int_0^\infty g(x) = 1$,

$$C\int_0^\infty (2x^{\theta-1} + x^{\theta-1/2})e^{-x}dx = 1 \tag{9}$$

Thus,

$$\int_0^\infty (2x^{\theta-1} + x^{\theta-1/2})e^{-x}dx = \int_0^\infty 2x^{\theta-1}e^{-x}dx + \int_0^\infty x^{\theta-1/2}e^{-x}dx \tag{10}$$

By the definition of gamma function,

$$\Gamma(y) = \int_0^\infty x^{y-1}e^{-x}dx \tag{11}$$

Thus, the integration is equal to,

$$\int_0^\infty (2x^{\theta-1} + x^{\theta-1/2})e^{-x}dx = 2\Gamma(\theta) + \Gamma(\theta + 1/2) \tag{12}$$

and,

$$C = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + 1/2)} \tag{13}$$

$$g = C(2x^{\theta-1} + x^{\theta-1/2})e^{-x} \tag{14}$$

g is a mixture of Gamma distributions, and rewrite it into

$$g = 2C * \Gamma(\theta)\frac{x^{\theta-1}e^{-x}}{\Gamma(\theta)} + C * \Gamma(\theta + 1/2)\frac{x^{\theta-1/2}e^{-x}}{\Gamma(\theta + 1/2)} \tag{15}$$

Thus,

$$g = 2C\Gamma(\theta)Gamma(\theta, 1) + C\Gamma(\theta + 1/2)Gamma(\theta + 1/2, 1) \tag{16}$$

The weights for two terms are $\frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}$ and $\frac{\Gamma(\theta+1/2)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}$ respectively.

b) Using $\theta = 0.5$, thus $\Gamma(\theta) = \Gamma(0.5) = 1.772454$ and $\Gamma(\theta + 1/2) = \Gamma(1) = 1$.

$$C = \frac{1}{2\Gamma(0.5) + \Gamma(1)} = 0.2200265 \tag{17}$$

$$g(x) = C(2x^{-0.5} + x^0)e^{-x} = C(2x^{-0.5} + 1)e^{-x} \tag{18}$$

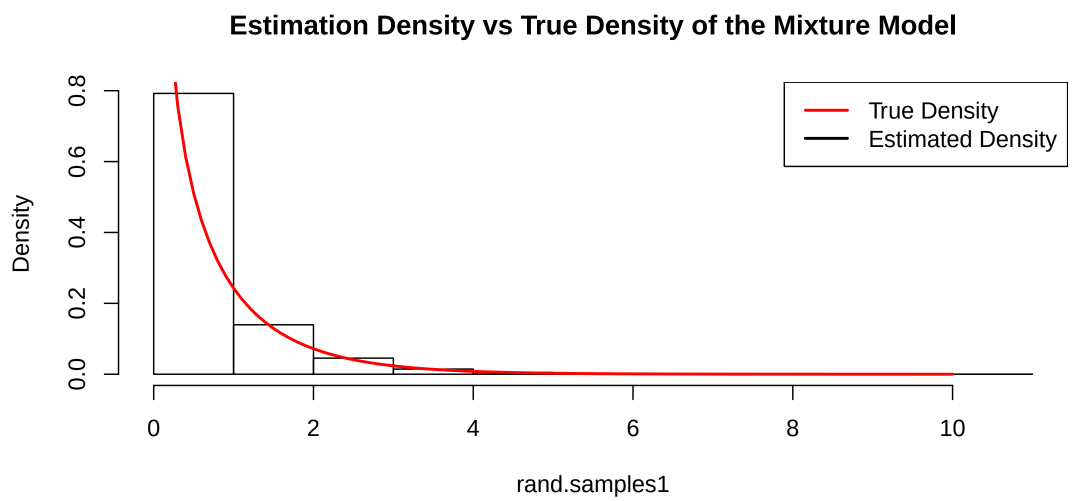
A procedure (pseudo-code) is degined to sample from g with sample size $n = 10,000$. The code chunk implementing Inverse transform method is given as follow,

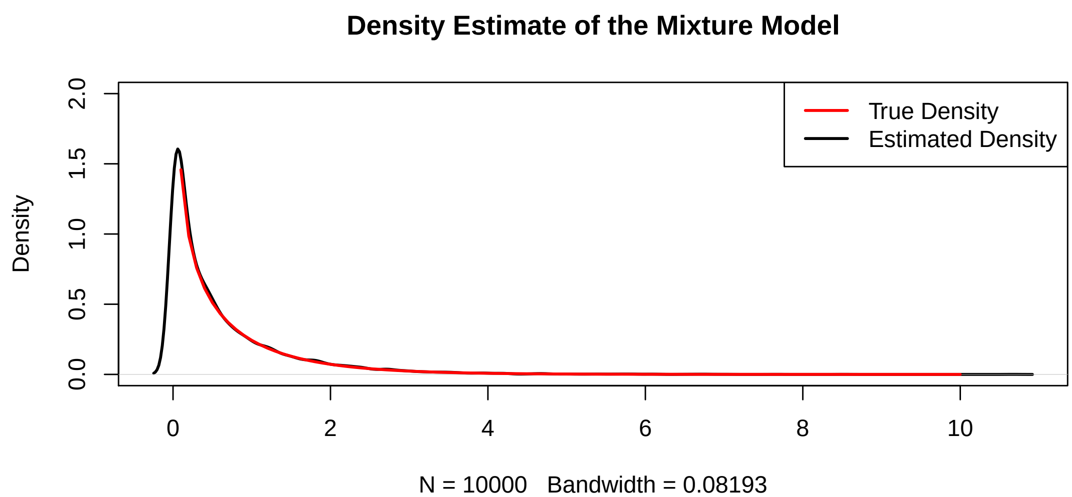
```
#The number of samples from the mixture distribution
N <- 10000

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

#Variable to store the samples from the mixture distribution
rand.samples1 <- rep(NA,N)
rand.samples2 <- rep(NA,N)

#Weights
theta1 <- 0.5
theta2 <- 2
C1 <- 1/(2*gamma(theta1)+gamma(theta1+0.5))
c11 <- 2*C1*gamma(theta1)
c12 <- C1*gamma(theta1+0.5)
C2 <- 1/(2*gamma(theta2)+gamma(theta2+0.5))
c21 <- 2*C2*gamma(theta2)
c22 <- C2*gamma(theta2+0.5)

#Sampling from the mixture
for(i in 1:N){
  if(U[i]<c11){
    rand.samples1[i] <- rgamma(1,theta1,rate = 1)
  }else{
    rand.samples1[i] <- rgamma(1,theta1+0.5,rate = 1)
  }
}
```

```
for(i in 1:N){
  if(U[i]<c21){
    rand.samples2[i] <- rgamma(1,theta2,rate = 1)
  }else{
    rand.samples2[i] <- rgamma(1,theta2+0.5,rate = 1)
  }
}
```

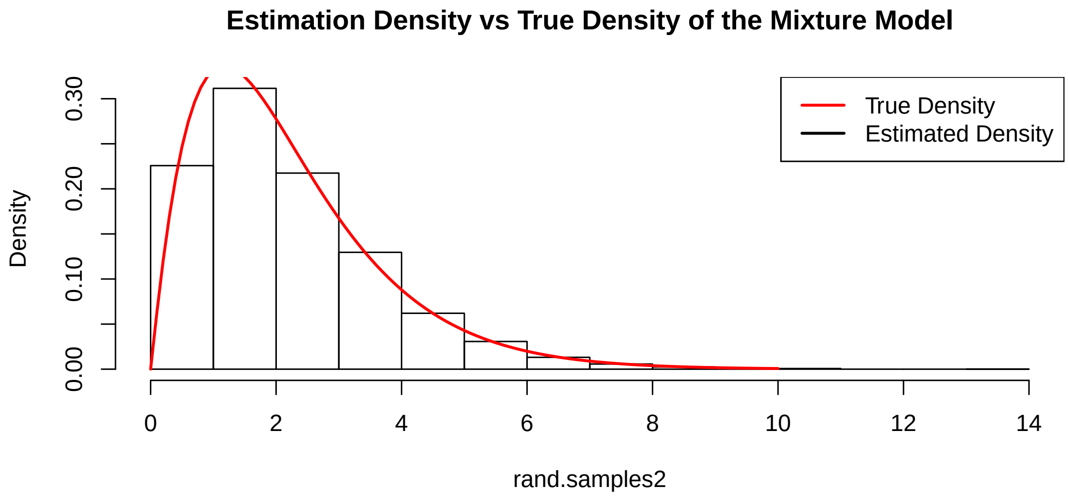
The kernel density estimation of g from my sample and the true density are plotted in the following figure. There are 2 cases with different θ values.

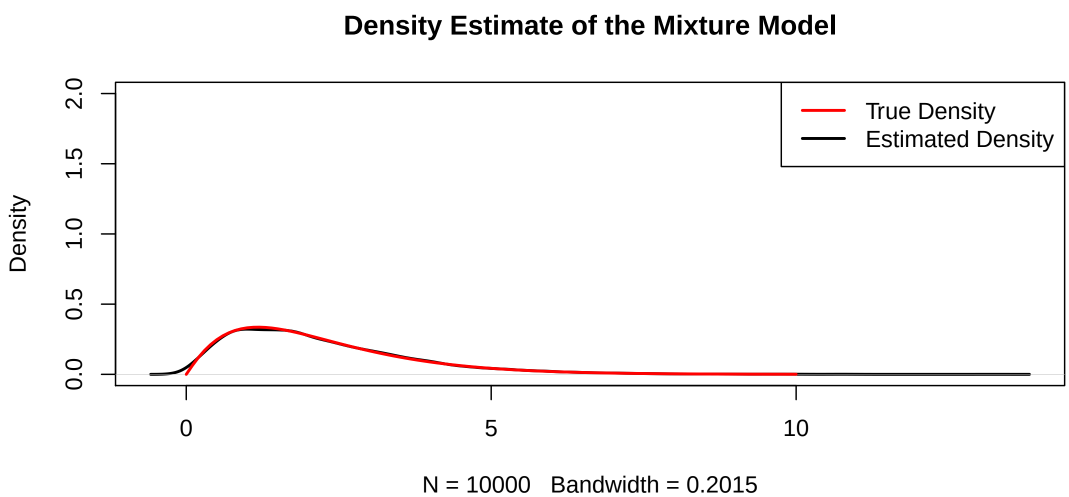
case1: $\theta = 0.5$





case2: $\theta = 2$





c) A procedure (pseudo-code) is degined to use rejection sampling to sample from f using g as the instrumental distribution, which means $f(x) \leq \alpha g(x)$ for all $x \in (0, \infty)$, where $\alpha g(x)$ is the envelope. Since $f(x) \propto \sqrt{4+x}x^{\theta-1}e^{-x}$, the normalizing constant for $f(x)$

$$C' = \frac{1}{\int_0^\infty \sqrt{4+x}x^{\theta-1}e^{-x}dx} \tag{19}$$

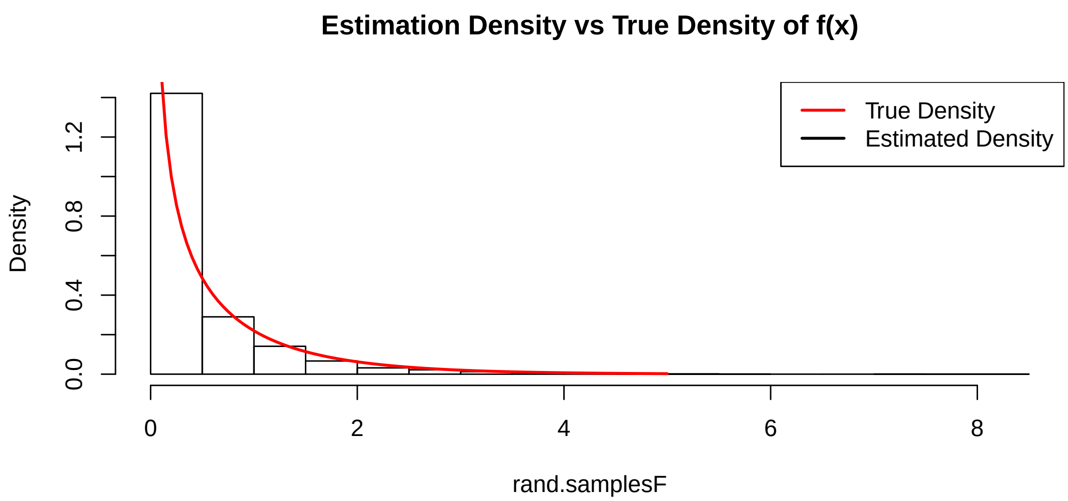
From b), use case 1's setting $\theta = 0.5$. Thus, $C' = 0.26666$. By plotting $f(x)$ and $\alpha g(x)$ with different values of α. $\alpha = 5$ is picked to set up the envelop.

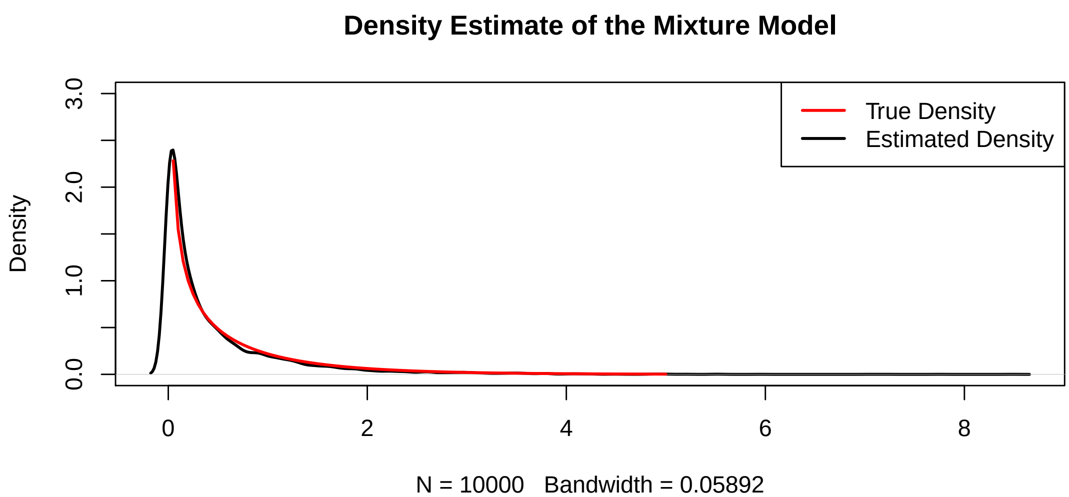
The code chunk implementing Rejection sampling is given as follow,

```
# The number of samples from the mixture distribution
N2 <- 10000

# Variable to store the samples from the mixture distribution
rand.samplesF <- rep(NA,N2)

# Check for the normalizing constant
```

```r
integrandF <- function(x) {sqrt(4+x)*x^(-0.5)*exp(-x)}
C_f <- integrate(integrandF, lower = 0, upper = Inf)
C_f <- C_f$value
C_f <- 1/C_f

# Envelope setting: alpha \in (10,20)
alpha <- 5
theta <- 0.5
C <- 1/(2*gamma(theta)+gamma(theta+0.5))
c1 <- 2*C*gamma(theta)
c2 <- C*gamma(theta+0.5)
g_x <- function(x) {c1*dgamma(x,theta,1) + c2*dgamma(x,theta+0.5,1)}

# Plot f and envelop
x_value <- seq(0,5,.05)
F <- function(x) {sqrt(4+x)*x^(0.5-1)*exp(-x)}
truthF <- C_f*F(x_value)
#plot(x_value,truthF,col="red",lwd=2)
truthG <- c1*dgamma(x_value,theta,1) + c2*dgamma(x_value,theta+0.5,1)
Envelop <- alpha*truthG
#lines(x_value,Envelop,col="blue",lwd=2)

#Rejection Sampling from the envelop
for (i in 1:N2) {
  while (TRUE) {
    U <- runif(1)
    if(U < c1){
      cand <- rgamma(1,theta,rate = 1)
    }else{
      cand <- rgamma(1,theta+0.5,rate = 1)
    }
    ratio <- F(cand)/(alpha*g_x(cand))
    if (U < ratio) break
  }
  rand.samplesF[i] <- cand
}
```

The the estimated density of a random sample generated by my procedure and f are plotted in the following figure.





Problem 3

Answer:

a) Design a mixture of Beta distributions $g(x)$ as the instrumental density to draw sample from $f(x)$, where $f(x)$ is a pbability density on $(0,1)$ such that

$$f(x) \propto q(x) = \frac{x^{\theta-1}}{1+x^2} + \sqrt{(2+x^2)}(1-x)^{\beta-1}, 0 < x < 1 \tag{20}$$

.

The instrumental density $g(x)$ have the form $\sum_{k=1}^{m} p_k g_k(x)$, where p_k are weights and g_k are densities of Beta distribution. Comparing to Beta(θ, β) density definition,

$$z(x) = \frac{x^{\theta-1}(1-x)^{\beta-1}}{B(\theta, \beta)} \tag{21}$$

Rewite $q(x)$,

$$q(x) = \frac{1}{1+x^2} x^{\theta-1}(1-x)^0 + \sqrt{(2+x^2)} x^0 (1-x)^{\beta-1} \tag{22}$$

Since $x \in (0,1)$

$$q(x) \leq x^{\theta-1}(1-x)^0 + \sqrt{3} x^0 (1-x)^{\beta-1} = x^{\theta-1} + \sqrt{3}(1-x)^{\beta-1} \tag{23}$$

My choice of the mixture model is,

$$g(x) = p_1 g_1(x) + p_2 g_2(x), \tag{24}$$

where

$$\begin{array}{l} g_1(x) = \frac{x^{\theta-1}(1-x)^0}{B(\theta,1)} = \frac{x^{\theta-1}}{B(\theta,1)} = Beta(\theta, 1) \\ g_2(x) = \frac{x^0(1-x)^{\beta-1}}{B(1,\beta)} = \frac{(1-x)^{\beta-1}}{B(1,\beta)} = Beta(1, \beta) \\ p_1 = p_2 = 0.5 \qquad and \qquad \sum p_k = 1 \end{array} \tag{25}$$

In order to satisfy

$$q(x) \leq x^{\theta-1} + \sqrt{3}(1-x)^{\beta-1} \leq \alpha g(x), \tag{26}$$

the choice of α is shown as follow.

$$\begin{array}{l} \alpha g(x) = 0.5\alpha(g_1(x) + g_2(x)) \\ = 0.5\alpha(\frac{x^{\theta-1}}{B(\theta,1)} + \frac{(1-x)^{\beta-1}}{B(1,\beta)}) \\ = 0.5\alpha\theta x^{\theta-1} + 0.5\alpha\beta(1-x)^{\beta-1}, \end{array} \tag{27}$$

where

$$1 \leq 0.5\alpha\theta \qquad and \qquad \sqrt{3} \leq 0.5\alpha\beta \tag{28}$$

Solve for α,

$$\frac{2}{\theta} \leq \alpha \qquad and \qquad \frac{2\sqrt{3}}{\beta} \leq \alpha \tag{29}$$

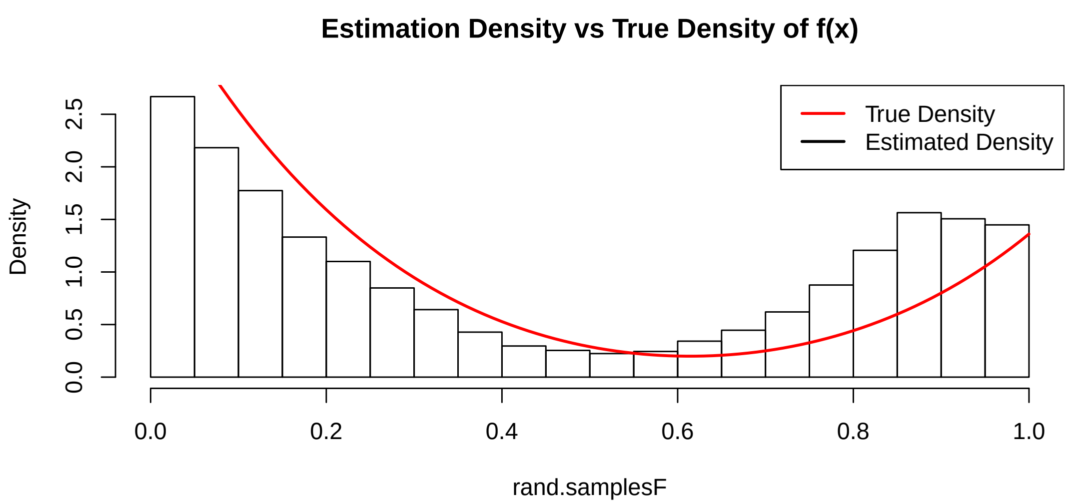
This means $\alpha = max(\frac{2}{\theta}, \frac{2\sqrt{3}}{\beta})$ based on target function's parameters.

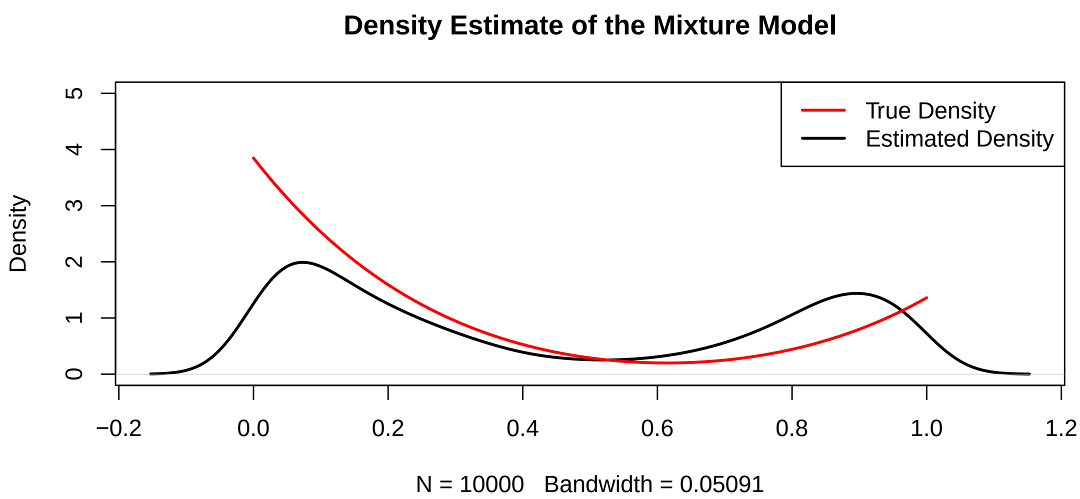
The code chunk implementing Rejection sampling is given as follow,

```
#The number of samples from the mixture distribution
N <- 10000

#Variable to store the samples from the mixture distribution
rand.samplesF <- rep(NA,N)

#Setup parameters
theta <- 7
beta <- 5

#check for the normalizing constant
integrandq <- function(x) {((x^(theta-1))/(1+x^2))+sqrt(2+x^2)*(1-x)^(beta-1)}
C_f <- integrate(integrandq, lower = 0, upper = 1)
C_f <- C_f$value
C_f <- 1/C_f
```

```
#envelope setting (10,20)
alpha <- max(2/theta,2*sqrt(3)/beta)
c1 <- 0.5
c2 <- 0.5
g_x <- function(x) {c1*dbeta(x,theta,1) + c2*dbeta(x,1,beta)}

#plot f and envelope
x_value3 <- seq(0,1,.01)
truthF <- C_f*integrandq(x_value3)
#plot(x_value3,truthF,col="red",lwd=2)
truthG <- g_x(x_value3)
Envelop <- alpha*truthG
#lines(x_value3,Envelop,col="blue",lwd=2)

#Rejection Sampling from the envelop
for (i in 1:N) {
  while (TRUE) {
    U <- runif(1)
    if(U < c1){
      cand <- rbeta(1,theta,1)
    }else{
      cand <- rbeta(1,1,beta)
    }
    ratio <- F(cand)/(alpha*g_x(cand))
    rejection <- runif(1)
    if (rejection < ratio) break
  }
  rand.samplesF[i] <- cand
}
```

The the estimated density of a random sample generated by my procedure and f are plotted in the following figure.





b) $f(x)$ can also be sampled using rejection sampling, by dealing with two components $\frac{x^{\theta-1}}{1+x^2}$ and $\sqrt{(2+x^2)}(1-x)^{\beta-1}$ separately using individual Beta distributions.

From a), two Beta distribution have been picked up.

$$\begin{aligned} & g(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) \\ & q(x) \le x^{\theta-1} + \sqrt{3}(1-x)^{\beta-1} \le \alpha g(x) \end{aligned} \tag{30}$$

Thus,

$$\begin{aligned} & x^{\theta-1} \le \alpha_1 g_1(x) \qquad and \qquad \sqrt{3}(1-x)^{\beta-1} \le \alpha_2 g_2(x) \\ & \frac{1}{\theta} \le \alpha_1 \qquad and \qquad \frac{\sqrt{3}}{\beta} \le \alpha_2 \end{aligned} \tag{31}$$

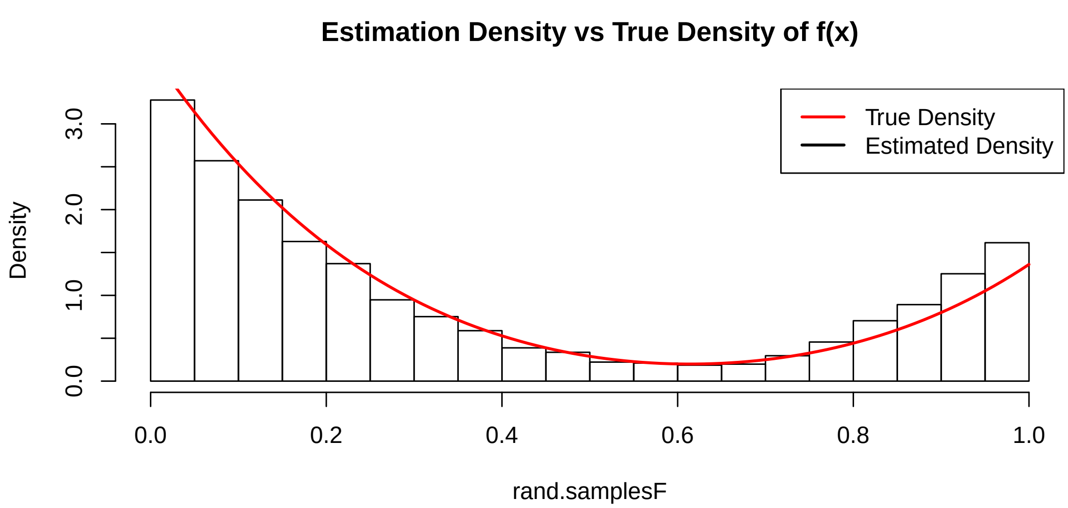
Set weighted $\lambda = \frac{\frac{1}{\theta}}{\frac{1}{\theta}+\frac{\sqrt{3}}{\beta}}$.

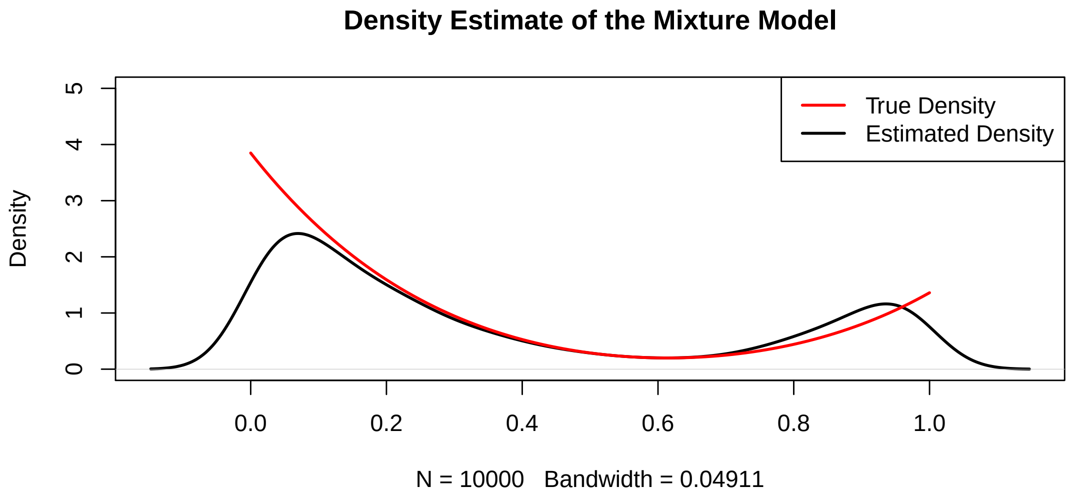
The code chunk implementing Rejection sampling is given as follow,

```
#The number of samples from the mixture distribution
N <- 10000

#Variable to store the samples from the mixture distribution
rand.samplesF <- rep(NA,N)

#Setup parameters
theta <- 7
beta <- 5
#theta <- 9
#beta <- 5

#check for the normalizing constant
integrandq <- function(x) {((x^(theta-1))/(1+x^2))+sqrt(2+x^2)*(1-x)^(beta-1)}
C_f <- integrate(integrandq, lower = 0, upper = 1)
C_f <- C_f$value
C_f <- 1/C_f

#envelope setting (10,20)
alpha1 <- 1/theta
```

```r
alpha2 <- sqrt(3)/beta
g_x <- function(x) {dbeta(x,theta,1) + dbeta(x,1,beta)}
Envelop <- function(x) {alpha1*dbeta(x,theta,1) + alpha2*dbeta(x,1,beta)}

#plot f
x_value3 <- seq(0,1,.01)
truthF <- C_f*integrandq(x_value3)
#plot(x_value3,truthF,col="red",lwd=2)
truthG <- g_x(x_value3)
EnvelopG <- Envelop(x_value3)
#lines(x_value3,EnvelopG,col="blue",lwd=2)

#Rejection Sampling from the envelop
for (i in 1:N) {
  while (TRUE) {
    U <- runif(1)
    if(U < alpha1/(alpha1+alpha2)){
      cand <- rbeta(1,theta,1)
    }else{
      cand <- rbeta(1,1,beta)
    }
    ratio <- F(cand)/(Envelop(cand))
    rejection <- runif(1)
    if (rejection < ratio) break
  }
  rand.samplesF[i] <- cand
}
```

The the estimated density of a random sample generated by my procedure and f are plotted in the following figure.




Density Estimate of the Mixture Model
True Density
Estimated Density
Density
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
N = 10000 Bandwidth = 0.04911