Random Number Generation in R Markdown

Cosmin Borsa[*]

10/19/2018

Abstract

This document is a homework assignment for the course Statistical Computing at the University of Connecticut.

1 Rejection Sampling

We are given two probability density functions f and g on the interval $(0, \infty)$ such that

$$f(x) \propto \sqrt{(4+x)} x^{\theta-1} e^{-x} \tag{1}$$

$$g(x) \propto (2x^{\theta-1} + x^{\theta-\frac{1}{2}}) e^{-x} \tag{2}$$

1.1 Mixture of Gamma Distributions

At first, we are going to find the value of the normalizing constant C for the density function g. Since g is a density function, we can integrate it over its domain $(0, \infty)$ and we obtain

$$\int_0^\infty g(x) dx = \int_0^\infty C(2x^{\theta-1} + x^{\theta-\frac{1}{2}}) e^{-x} dx = 1 \tag{3}$$

We can spilt the integral into two and we have

$$C \int_0^\infty 2x^{\theta-1} e^{-x} dx + C \int_0^\infty x^{\theta-\frac{1}{2}} e^{-x} dx = 1 \tag{4}$$

Since a continuous random variable X which follows a Gamma distribution with shape parameter $\theta > 0$ and rate parameter 1, has the probability density function given by

$$f_X(\theta) = \begin{cases} \frac{x^{\theta-1} e^{-x}}{\Gamma(\theta)}, \ x > 0 \\ 0, \ x \leq 0 \end{cases}$$

We will now show that the function $g(x)$ is a mixture of Gamma distributions.

$$2C\Gamma(\theta) \int_0^\infty \frac{x^{\theta-1} e^{-x}}{\Gamma(\theta)} dx + C\Gamma(\theta + \frac{1}{2}) \int_0^\infty \frac{x^{(\theta+\frac{1}{2})-1} e^{-x}}{\Gamma(\theta + \frac{1}{2})} dx = 1 \tag{5}$$

[*]cosmin.borsa@uconn.edu; M.S. in Applied Financial Mathematics, Department of Mathematics, University of Connecticut.

In equation (5) it can be seen that we are integrating the density functions of two Gamma distributed random variables over their domain of definition. Thus, the integrals should be equal to 1, and we obtain value of C.

$$2C\Gamma(\theta) + C\Gamma(\theta + \frac{1}{2}) = 1 \quad \Rightarrow \quad C = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})} \tag{6}$$

Using the value of the normalizing constant C we can obtain the probability density function g.

$$g(x) = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})}(2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x} \tag{7}$$

We will now show that g is a mixture of two Gamma distributions.

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

The two Gamma distributions that are included in g are $\Gamma(\text{rate} = 1, \text{shape} = \theta)$ with the weight α_1 and $\Gamma(\text{rate} = 1, \text{shape} = \theta + \frac{1}{2})$ with the weight α_2.

$$\alpha_1 = \frac{2\Gamma(\theta)}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})} \qquad \alpha_2 = \frac{\Gamma(\theta + \frac{1}{2})}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})}$$

Since $\alpha_1 + \alpha_2 = 1$, we have shown that g is a mixture of two Gamma distributions.

1.2 Sample from g

We will now design a procedure to sample from g. To do that we need to take samples from the Gamma distributions $\Gamma(\text{rate} = 1, \text{shape} = \theta)$ and $\Gamma(\text{rate} = 1, \text{shape} = \theta + \frac{1}{2})$. Next, we will briefly explain the procedure using pseudo-code.

Algorithm 1 Sample from g

procedure MY PROCEDURE
    Set α_0
    Construct a binomial distributed vector *indicator*
    loop throughout *sample size*
    if $indicator = 1$ **then**
        Sample X from Gamma$(\theta, 1)$
    else
        Sample X from Gamma$(\theta + \frac{1}{2}, 1)$
    end if
    end loop
    return Vector X
end procedure

We will now implement the pseudo-code in the R function `sample.g`.

```r
sample.g <- function(sample.size, theta)  {

  alpha0 = 2 * gamma(theta) / (2 * gamma(theta) + gamma(theta + (1/2)))
  indicator = rbinom(sample.size, 1, alpha0)
  sample.g = numeric(sample.size)

  for (i in 1:sample.size) {
    if (indicator[i] == 1) {
      sample.g[i] <- rgamma(1, shape = theta, rate = 1)
    }
    else {
      sample.g[i] <- rgamma(1, shape = theta + (1/2), rate = 1)
    }
  }

  return(sample.g)
}
```

Next, we are going to draw a sample of size $10,000$ with $\theta = 2$. After that we will plot the obtained kernel density estimation of g from the sample and the true density of g. It is worth mentioning that the variable `alpha` does not give the probability of acceptance. It returns the probability with which an observation belongs to one of the distributions in the mixture.

```r
n <- 10000
theta.given <- 2
sampling.g <- sample.g(n, theta.given)
alpha = 2 * gamma(theta.given) / (2 * gamma(theta.given) +
        gamma(theta.given + (1/2)))

plot(density(sampling.g), col = "red")
curve(alpha*dgamma(x,shape = 2, rate = 1) + (1-alpha) *
      dgamma(x, shape = 5/2, rate = 1), xlim = c(0, max(sampling.g)),
```

```
      add = TRUE, col = "blue")
legend("topright", legend = c("Kernel Density", "True Density"),
      col = c("red","blue"), lty = c(1,1))
```

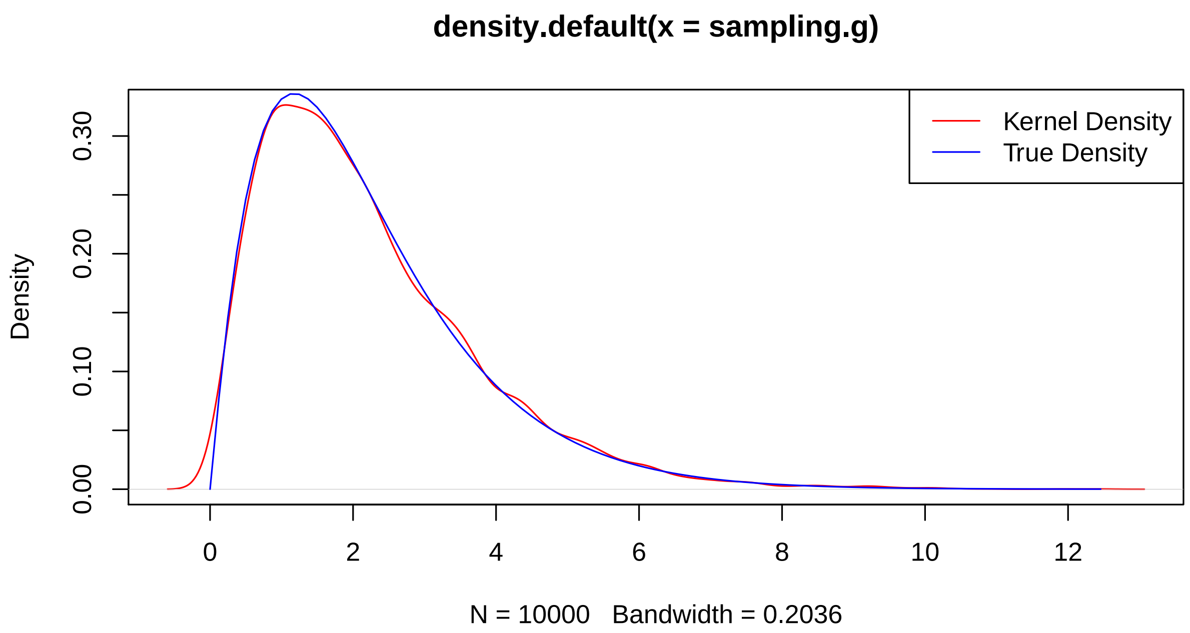

Figure 1: Kernel Density Estimation and the True Density of g

We can see from Figure (1) that the kernel density is a good approximation of the true density when we sample $10,000$ observations from g with $\theta = 2$.

1.3 Sample from f with Rejection Sampling

We now want to sample from the more complicated probability density function f. To do that we will use Rejection Sampling with instrumental density g. However, before we do that, we will show that f is proportional to a function q which in turn is less than or equal to g. So let $q(x) = \sqrt{(4+x)}x^{\theta-1}e^{-x}$ such that

$$q(x) \leq (\sqrt{4} + \sqrt{x})x^{\theta-1}e^{-x} = (2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x}$$

Thus,

$$q(x) \leq \frac{1}{C} \cdot g(x) \quad \text{with} \quad C = \frac{1}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})}$$

So we are going to define $\alpha := \frac{1}{C} = 2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})$ such that

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

Now, we will design a procedure to sample from f using Rejection Sampling with instrumental density g. We will briefly explain the procedure using pseudo-code.

Algorithm 2 Sample from f

procedure MY PROCEDURE
  Initialize *count* and *counts rejections*
  while *count* < *sample size* **do**
  Call Procedure *Sample from g*
  Sample *observation* from U$(0, 1)$
  Sample X from $g(x)$
  Compute $ratio = \frac{q(x)}{g(x)}$
  if *observation* $\leq$ *ratio* **then**
    Accept sample X
  else
    Reject sample X
  end if
  end while
  return Vector X
end procedure

We will now implement the pseudo-code in the R function `sample.f`.

```r
sample.f <- function(sample.size, theta) {
  count <- 0
  count.rejections <- 0
  sample.f <- numeric(0)
  while(count < sample.size) {
    sampling.g <- sample.g(1, theta)
    uniform  <- runif(1,0,1)
    q.x      <- sqrt(4 + sampling.g)*((sampling.g)^(theta-1))*exp(-1*sampling.g)
    alpha.g.x <- 2*((sampling.g)^(theta-1))*exp(-1*sampling.g) +
                    ((sampling.g)^(theta-(1/2)))*exp(-1*sampling.g)
    ratio    <- (q.x)/(alpha.g.x)
    if (uniform <= ratio) {
      sample.f <- append(sample.f, sampling.g)
      count <- count + 1
    }
    else {
      count.rejections <- count.rejections + 1
    }
  }
  acceptance.rate = count/(count + count.rejections)
  return(sample.f)
}
```

Next we are going to draw a sample of size $10,000$ with $\theta = 2$. Then, we will plot the estimated density of f.

```r
n <- 10000
theta.given <- 2
sampling.f <- sample.f(n, theta.given)

plot(density(sampling.f), col = "blue")
legend("topright", legend = c("Estimated Density of f(x)"), col = c("blue"),
       lty = c(1))
```

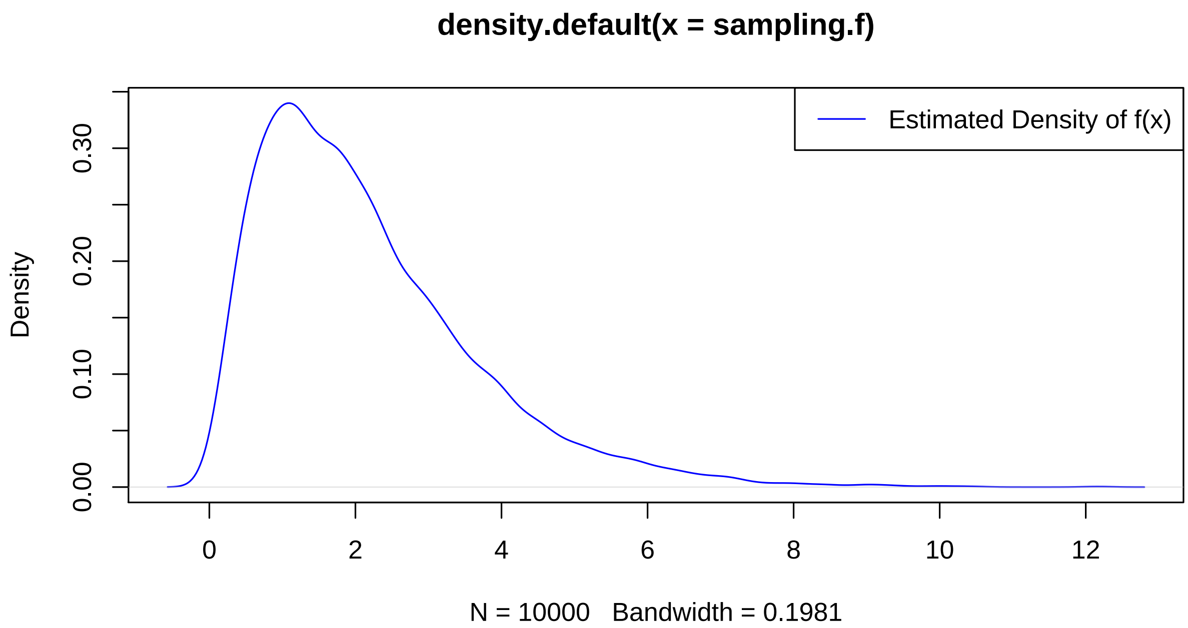

Figure 2: Estimated density of f

Even though it is difficult to graph the actual density of f without evaluating the normalizing constant, it seems that a sample size of $10,000$ observations gives us a fairly good estimation of density curve. Comparing the plot of f with that of g we can observe that the curves look similar. This implies that we get a relatively high acceptance probability. After running the code a few times with the parameter $\theta = 2$ and sample size $10,000$ we obtain an acceptance rate of approximately 73%.

2 Mixture Proposal

We are given a probability density function f on the interval $(0, 1)$ such that

$$f(x) \propto \frac{x^{\theta-1}}{1+x^2} + \sqrt{2+x^2}(1-x)^{\beta-1} \tag{9}$$

2.1 Mixture of Beta Distributions

We now want to sample from the probability density function f and use a mixture of Beta distributions as the instrumental density. However, before we do that, we will construct a function q such that for $x \in (0,1)$

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

Since $1+x^2 \geq 1$ and $\sqrt{2+x^2} \leq \sqrt{2}+x$ we will show that q is less than a mixture of Beta distributions.

$$q(x) \leq x^{\theta-1} + (\sqrt{2}+x)(1-x)^{\beta-1} = x^{\theta-1} + \sqrt{2}(1-x)^{\beta-1} + \sqrt{x^2}(1-x)^{\beta-1}$$

So let g be a function such that

$$g(x) \propto x^{\theta-1} + \sqrt{2}(1-x)^{\beta-1} + x(1-x)^{\beta-1}$$

We will now find the value of the normalizing constant C to make g a density function.

$$g(x) = C\Big(x^{\theta-1} + \sqrt{2}(1-x)^{\beta-1} + x(1-x)^{\beta-1}\Big)$$

A continuous random variable X which follows a Beta distribution with shape parameters $\alpha > 0$ and $\beta > 0$ has the following probability density function on the interval $(0,1)$

$$f_X(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1} = \frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

We will now show that the function $g(x)$ is a mixture of Beta distributions.

$$g(x) = C\Bigg(B(\theta,1)\frac{x^{\theta-1}}{B(\theta,1)} + \sqrt{2}B(1,\beta)\frac{(1-x)^{\beta-1}}{B(1,\beta)} + B(2,\beta)\frac{x(1-x)^{\beta-1}}{B(2,\beta)}\Bigg)$$

Therefore, g is a mixture of three Beta distributions: $B(\theta,1)$, $B(1,\beta)$, and $B(2,\beta)$. To obtain C we will integrate the density function $g(x)$ over $(0,1)$

$$\int_0^1 g(x)\,dx = C\int_0^1 B(\theta,1)\frac{x^{\theta-1}}{B(\theta,1)} + \sqrt{2}B(1,\beta)\frac{(1-x)^{\beta-1}}{B(1,\beta)} + B(2,\beta)\frac{x(1-x)^{\beta-1}}{B(2,\beta)}\,dx = 1$$

Thus,

$$CB(\theta,1)\int_0^1 \frac{x^{\theta-1}}{B(\theta,1)}dx + C\sqrt{2}B(1,\beta)\int_0^1 \frac{(1-x)^{\beta-1}}{B(1,\beta)}dx + CB(2,\beta)\int_0^1 \frac{x(1-x)^{\beta-1}}{B(2,\beta)}\,dx = 1 \quad (10)$$

As it can be seen from equation (10) we are integrating the density functions of Beta distributed random variables over their domain of definition. Thus, the integrals should be equal to 1, and we obtain C

$$C\Big(B(\theta,1) + \sqrt{2}B(1,\beta) + B(2,\beta)\Big) = 1 \Rightarrow C = \frac{1}{B(\theta,1) + \sqrt{2}B(1,\beta) + B(2,\beta)}$$

Therefore, g is a mixture of three Beta distributions: $B(\theta, 1)$ with the weight p_1, $B(1, \beta)$ with the weight p_2, and $B(2, \beta)$ with the weight p_3.

$$g(x) = p_1 \frac{x^{\theta-1}}{B(\theta,1)} + p_2 \frac{(1-x)^{\beta-1}}{B(1,\beta)} + p_3 \frac{x(1-x)^{\beta-1}}{B(2,\beta)}$$

The weights are given as follows

$$p_1 = \frac{B(\theta,1)}{B(\theta,1) + \sqrt{2}B(1,\beta) + B(2,\beta)}$$

$$p_2 = \frac{\sqrt{2}B(1,\beta)}{B(\theta,1) + \sqrt{2}B(1,\beta) + B(2,\beta)}$$

$$p_3 = \frac{B(2,\beta)}{B(\theta,1) + \sqrt{2}B(1,\beta) + B(2,\beta)}$$

Next, we are going to define $\alpha := \frac{1}{C}$ such that

$$q(x) \leq \frac{1}{C} \cdot C\Big(x^{\theta-1} + \sqrt{2}(1-x)^{\beta-1} + \sqrt{x^2}(1-x)^{\beta-1}\Big) = \alpha g(x)$$

Since we've shown that $q(x) \leq \alpha g(x)$, where $\alpha = B(\theta, 1) + \sqrt{2}B(1, \beta) + B(2, \beta)$ we can now design a procedure to sample from f. To do that we need to take samples from the Beta distributions $B(\theta, 1)$, $B(1, \beta)$ and $B(2, \beta)$. Next, we will briefly explain the procedure using pseudo-code.

Algorithm 3 Sampling from f using a mixture of Beta Distributions

procedure MY PROCEDURE
  Initialize *count* and *counts rejections*
  while *count* $<$ *sample size* **do**
  Sample vector x of Beta Distributions
  Sample *observation* from $\mathrm{U}(0, 1)$
  Comute $q(x)$ and $\alpha g(x)$
  Compute *ratio* $= \frac{q(x)}{\alpha g(x)}$
  if *observation* $\leq$ *ratio* **then**
    Accept sample x
  else
    Reject sample x
  end if
  end while
  return Vector of sampled $f(x)$
end procedure

Before we implement the pseudo-code, we would like to construct the function `sample.beta.mixture` that samples Beta distributions.

```r
sample.beta.mixture <- function(sample.size2, theta.given, beta.given)  {
  alpha.0 <- beta(theta.given,1) / (beta(theta.given,1) + sqrt(2) *
             beta(1, beta.given) + beta(2,beta.given))
  alpha.1 <- sqrt(2) * beta(1, beta.given) / (beta(theta.given,1) + sqrt(2) *
             beta(1, beta.given) + beta(2,beta.given))
  indicator <- runif(sample.size2, 0, 1)
  sample.beta.mixture <- numeric(sample.size2)

  for (i in 1:sample.size2) {
    if (indicator[i] <= alpha.0) {
      sample.beta.mixture[i] <- rbeta(1, shape1 = theta.given, shape2 = 1)
    }
    else if (indicator[i] <= alpha.0 + alpha.1) {
      sample.beta.mixture[i] <- rbeta(1, shape1 = 1, shape2 = beta.given)
    }
    else {
      sample.beta.mixture[i] <- rbeta(1, shape1 = 2, shape2 = beta.given)
    }
  }
  return(sample.beta.mixture)
}
```

Next, we implement the pseudo-code in the R function `sample.f.new`

```r
sample.f.new <- function(sample.size2, theta.given, beta.given) {
  count <- 0
  count.rejections <- 0
  sample.f.new <- numeric(0)
  while(count < sample.size2) {
    obs <- sample.beta.mixture(1, theta.given, beta.given)
    unif <- runif(1,0,1)
    q.obs <- (obs^(theta.given - 1)) / (1 + obs^2) + (sqrt(2 + obs^2)) *
             ((1 - obs)^(beta.given - 1))
    alpha.g.obs <- obs^(theta.given - 1) + (sqrt(2)) * (1 - obs)^(beta.given -
                   1) + obs*((1 - obs)^(beta.given - 1))
    ratio.obs <- q.obs/alpha.g.obs
    if(unif <= ratio.obs) {
      sample.f.new <- append(sample.f.new, obs)
      count = count + 1
    }
    else {
      count.rejections = count.rejections + 1
    }
  }
  acceptance.rate = count/(count + count.rejections)
  return(sample.f.new)
}
```

We will now estimate the density of f by drawing a sample of size $10,000$ with the parameters $\theta = 2$, and $\beta = 4$. After that we will plot the kernel density of f from the sample against the true density.

```
q <- function(x) {
       return((x/(1 + x^2)) + sqrt(2 + x^2) * ((1 - x)^3))
     }

plot(density(sample.f.new(10000, 2, 4)), col = "red")
curve(q(x)/0.7058734, xlim = c(0,1), add = TRUE, col = "blue")
legend("topright", legend = c("Kernel Density", "True Density"),
       col = c("red","blue"), lty = c(1,1))
```

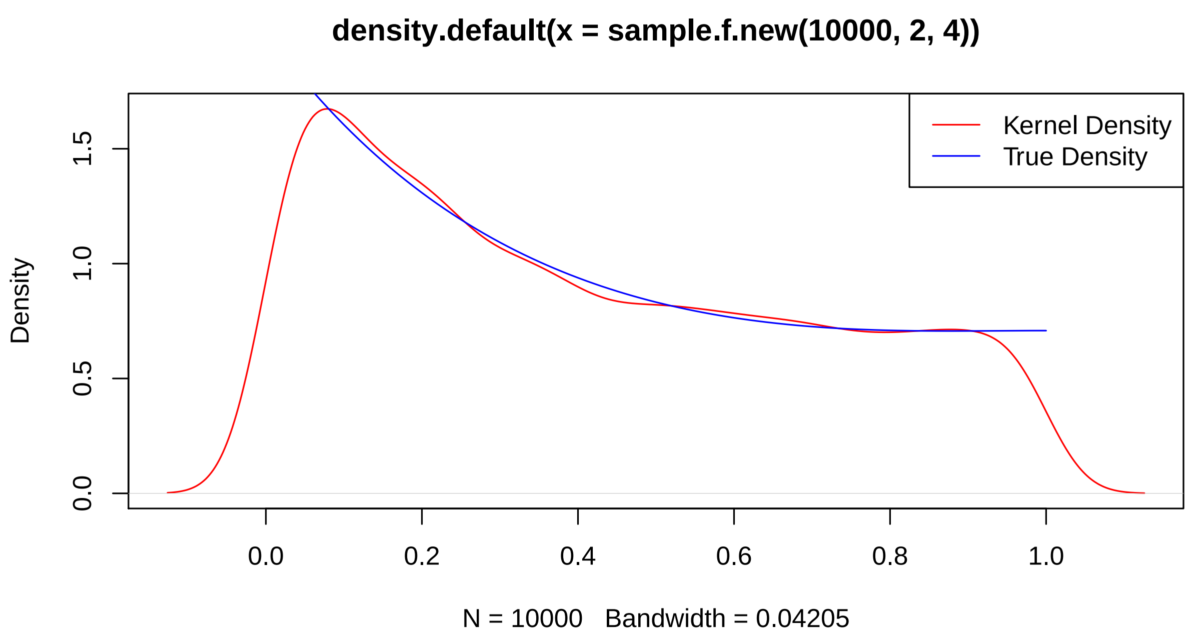

Figure 3: Estimated density of f using a Mixture of Beta distributions

As we can see from Figure (3) the estimated density of $f(x)$ is fairly good throughout the interval $(0, 1)$. However, when x gets close to 0 or 1, the estimated density drifts away from the actual density of f.

2.2 Two Components

Breaking f(x) into pieces that can be simulated using Beta distributions

Now, we are going to estimate the denisty of f from Equation (9) by splitting the function into two components. Since f is proportional to

$$f(x) \propto \left(\frac{x^{\theta-1}}{1+x^2}\right) + \left(\sqrt{2+x^2}(1-x)^{\beta-1}\right)$$

We will break f up into the functions q_1 and q_2 such that

$$f(x) \propto q_1(x) + q_2(x) \quad \text{where} \quad q_1(x) = \frac{x^{\theta-1}}{1+x^2} \quad \text{and} \quad q_2(x) = \sqrt{2+x^2}(1-x)^{\beta-1}$$

Since $1 + x^2 \geq 1$ we can easily show that q_1 is less than a Beta distributions with shape parameters θ and 1.

$$q_1(x) = \frac{x^{\theta-1}}{1+x^2} \leq B(\theta, 1) \cdot \frac{x^{\theta-1}}{B(\theta, 1)} = \alpha_1 \cdot g_1(x)$$

Where

$$\alpha_1 = B(\theta, 1) \quad \text{and} \quad g_1(x) = \frac{x^{\theta-1}}{B(\theta, 1)}$$

Since $\sqrt{2+x^2} \leq \sqrt{3}$ when $x \in (0, 1)$ we have

$$q_2(x) = \sqrt{2+x^2}(1-x)^{\beta-1} \leq \sqrt{3}(1-x)^{\beta-1} = \sqrt{3}B(1, \beta) \cdot \frac{(1-x)^{\beta-1}}{B(1, \beta)} = \alpha_2 \cdot g_2(x)$$

Where

$$\alpha_2 = \sqrt{3}B(1, \beta) \quad \text{and} \quad g_2(x) = \frac{(1-x)^{\beta-1}}{B(1, \beta)}$$

Since we've shown that $q_1(x) \leq \alpha_1 g_1(x)$ and $q_2(x) \leq \alpha_2 g_2(x)$, we can now design a procedure to sample from f. To do that we need to take samples from the Beta distributions $B(\theta, 1)$, $B(1, \beta)$. Next, we will briefly explain the procedure using pseudo-code.

Algorithm 4 Sampling from f using two components

```
procedure My Procedure
    Initialize count and counts rejections
    while count < sample size do
    Sample trial from Bernoulli Distribution
    if trial = success then
        Sample x from Beta(θ, 1)
        Compute ratio = q_1(x) / (α_1 g_1(x))
    else
        Sample x from Beta(1, β)
        compute ratio = q_2(x) / (α_2 g_2(x))
    end if
    Sample observation from U(0, 1)
    if observation ≤ ratio then
        Accept sample x
    else
        Reject sample x
    end if
    end while
    return Vector of sampled f(x)
end procedure
```

We will now implement the pseudo-code in R by constructing the function `sample.f.2`.

```
sample.f.2 <- function(sample.size2, theta.given, beta.given){
  alpha.1 <- beta(theta.given,1)
  alpha.2 <- sqrt(3)*beta(1, beta.given)
  prob <- alpha.1/(alpha.1 + alpha.2)
  count <- 0
  count.rejections <- 0
  sample.f.2 <- numeric(0)

  while(count < sample.size2) {
    trial <- rbinom(1, 1, prob)
    if (trial == 1) {
      x <- rbeta(1, theta.given, 1)
      q.1.x <- (x^(theta.given - 1))/(1 + x^2)
      alpha.g.1.x <-  (x^(theta.given - 1))
      ratio.x <-  q.1.x/alpha.g.1.x
    }
    else {
      x <- rbeta(1,1,beta.given)
      q.2.x <- (sqrt(2 + x^2))*((1-x)^(beta.given - 1))
      alpha.g.2.x <- sqrt(2)*((1-x)^(beta.given - 1)) + x*(1-x)^(beta.given - 1)
      ratio.x <- q.2.x/alpha.g.2.x
    }
    u <- runif(1, 0, 1)
```

```
    if(u <= ratio.x) {
      sample.f.2 <- append(sample.f.2, x)
      count <- count  + 1
    }
    else {
      count.rejections <- count.rejections + 1
    }
  }
  acceptance.rate <- count/(count + count.rejections)
  return(sample.f.2)
}
```

We will now estimate the density of f by drawing a sample of size $10,000$ with the parameters $\theta = 2$, and $\beta = 4$. After that we will plot the kernel density of f from the sample against the true density.

```
q <- function(x) {
        return((x/(1 + x^2)) + sqrt(2 + x^2)*((1 - x)^3))
     }

plot(density(sample.f.2(10000, 2, 4)), col = "red")
curve(q(x)/0.7058734, xlim = c(0,1), add = TRUE, col = "blue")
legend("topright", legend = c("Kernel Density", "True Density"),
       col = c("red","blue"), lty = c(1,1))
```

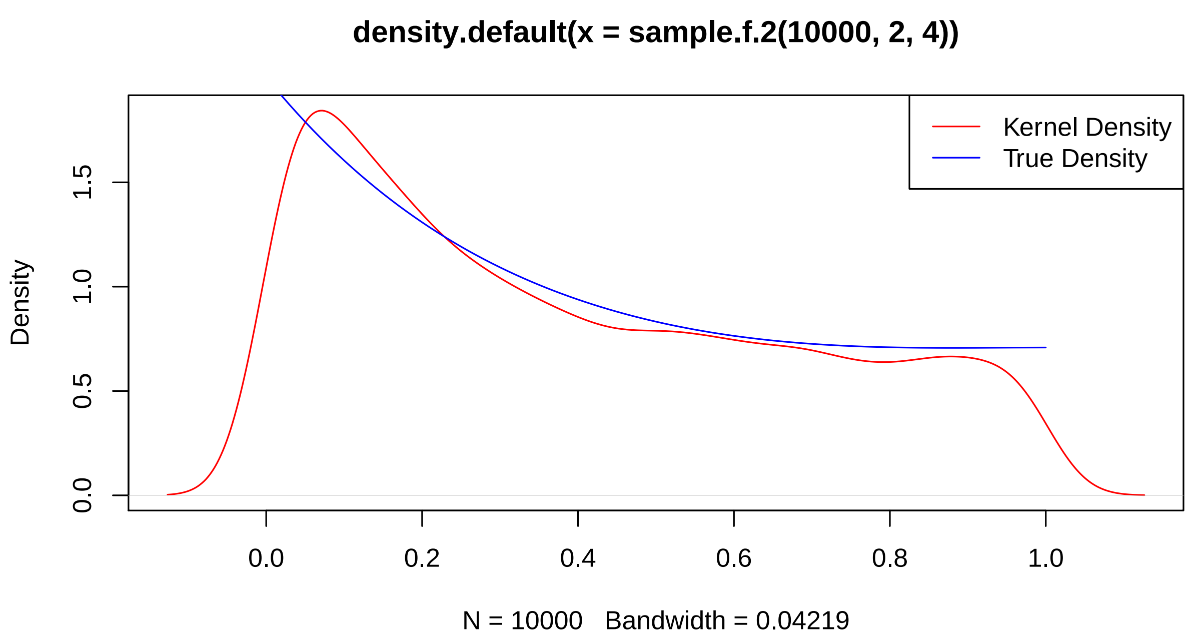

Figure 4: Estimated density of f using two component functions

As we can see from Figure (4), the estimated density of $f(x)$ is fairly good throughout the interval $(0, 1)$. The graph is similar to Figure (3) since the estimated density drifts away from the actual density of f when x gets close to 0 or 1.