Random Number Generation

HW 6 of STAT 5361 Statistical Computing

Biju Wang[*]

10/17/2018

1 Rejecion Sampling

1.1 Normalizing Constant for g

Since we have the following intergral

$$\begin{aligned}\int_0^\infty (2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x}dx &= 2\int_0^\infty x^{\theta-1}e^{-x}dx + \int_0^\infty x^{\theta-\frac{1}{2}}e^{-x}dx \\ &= 2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})\end{aligned}$$

Obviously

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

Therefore

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

Thus, $g(x)$ is a mixuture of Gamma distributions. The component distributions are Gamma$(\theta, 1)$ and Gamma$(\theta + \frac{1}{2}, 1)$, the corresponding weights are $\frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}$ and $\frac{\Gamma(\theta+\frac{1}{2})}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}$.

1.2 Sampling from $g(x)$

The pseudo-code is as follows

Algorithm 1 Sampling from $g(x)$

1: **procedure** My Procedure
2: Sample U from U$(0, 1)$
3: **if** $U < \frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}$ **then**
4: Sample X from Gamma$(\theta, 1)$
5: **else**
6: Sample X from Gamma$(\theta + \frac{1}{2}, 1)$
7: **end if**
8: **return** X
9: **end procedure**

```
sample.g <- function(n, shape, scale, prob){
  x <- rep(0, n)
  u <- runif(n, 0, 1)
  g1 <- rgamma(n, shape = shape[1], scale = scale[1])
  g2 <- rgamma(n, shape = shape[2], scale = scale[2])
```

[*]bijuwang@uconn.edu

```
  x[u < prob] <- g1[u < prob]
  x[u >= prob] <- g2[u >= prob]

  x
}

theta <- 1
prob <- 2 * gamma(theta)/(2 * gamma(theta) + gamma(theta + 1/2))
shape <- c(theta, theta + 1/2)
scale <- c(1, 1)
n <- 10000

g <- function(x, theta){
    (2 * x^(theta - 1) * exp(-x) + x^(theta - 1/2) * exp(-x))/(2 * gamma(theta) + gamma(theta + 1/2))
}

x <- sample.g(n, shape, scale, prob)

library("ggplot2")
ggplot(data.frame(x = x), aes(x = x)) +
geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
geom_density(aes(x, color = "a")) +
stat_function(aes(x, color = "b"), fun = function(x) g(x, theta = theta)) +
scale_colour_manual(name = "Legend", values = c("a" = "blue", "b" = "red"),
                    labels = c("Kernel", "True")) +
labs(x = expression("Values of"~x), y = expression("Density of"~g(x))) +
theme(plot.title = element_text(hjust = 0.5)) +
ggtitle(expression("Gamma Mixture Distribution with"~theta==1))
```

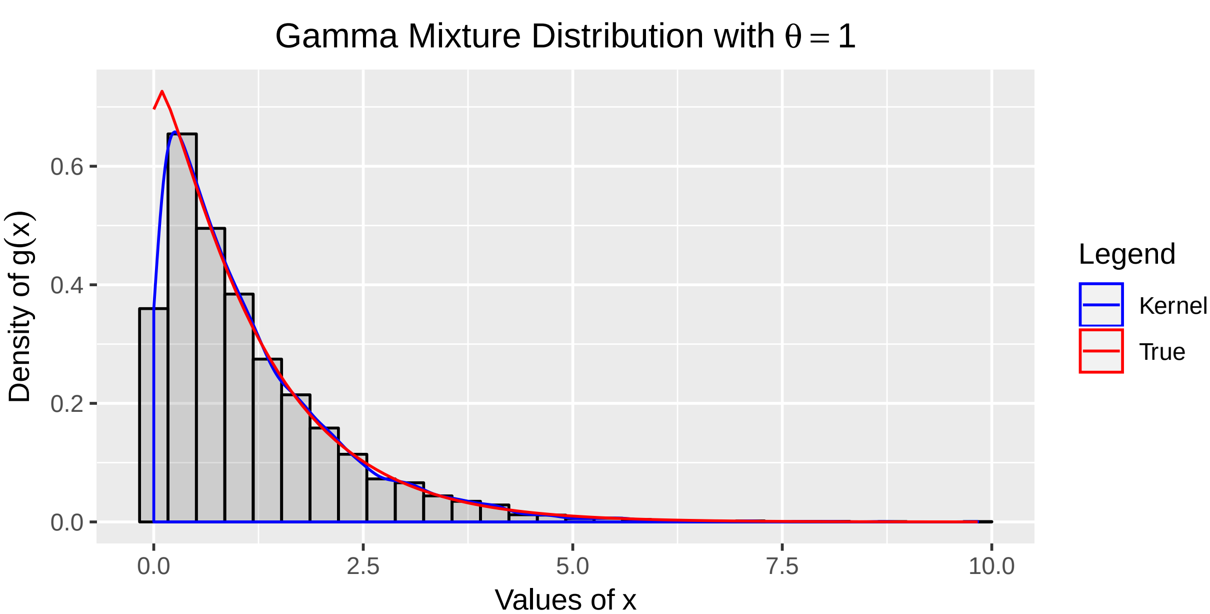

Figure 1: Histogram, Kernel Curve and True Curve for Generated Samples from $g(x)$ with $\theta = 1$

1.3 Sampling from $f(x)$ via Rejection Sampling

Since our target density is $f(x)$ and $f(x) \propto q(x) = \sqrt{4+x}x^{\theta-1}e^{-x}$ and the instrumental density is $g(x)$, we first calculate the minimal α such that $q(x) \leqslant \alpha g(x)$.

$$\begin{aligned}\alpha &= \sup_{x>0} \frac{q(x)}{g(x)} \\ &= \sup_{x>0} \frac{\sqrt{4+x}x^{\theta-1}e^{-x}}{C(2x^{\theta-1}+x^{\theta-\frac{1}{2}})e^{-x}} \\ &= \sup_{x>0} \frac{\sqrt{4+x}}{C\sqrt{x}} \\ &= \frac{1}{C}\end{aligned}$$

Therefore

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

The pseudo-code is as follows

Algorithm 2 Sampling from $f(x)$

```
1: procedure MY PROCEDURE
2:     Sample X from g(x) and U from U(0,1)
3:     if U ⩽ q(X)/(αg(X)) then
4:         return X
5:     else
6:         Go back to 2
7:     end if
8: end procedure
```

```r
sample.f <- function(n, shape, scale, prob){
  sample <- rep(0, n)
  iter <- 1

  while (iter <= n) {
      x <- sample.g(1, shape, scale, prob)
      u <- runif(1, 0, 1)
      if(u <= sqrt(4 + x)/(2 + sqrt(x))){
        sample[iter] <- x
        iter <- iter + 1
      }
  }
  sample
}

theta <- 1
prob <- 2 * gamma(theta)/(2 * gamma(theta) + gamma(theta + 1/2))
shape <- c(theta, theta + 1/2)
scale <- c(1, 1)
n <- 10000

h <- function(x, theta){
  sqrt(4 + x) * x^(theta - 1) * exp(-x)
}
```

```r
integral <- integrate(function(x) h(x, theta = theta), 0, Inf)

f <- function(x, theta){
  sqrt(4 + x) * x^(theta - 1) * exp(-x)/integral$value
}

x <- sample.f(n, shape, scale, prob)

library("ggplot2")
ggplot(data.frame(x = x), aes(x = x)) +
geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
geom_density(aes(x, color = "a")) +
stat_function(aes(x, color = "b"), fun = function(x) f(x, theta = theta)) +
scale_colour_manual(name = "Legend", values = c("a" = "blue", "b" = "red"),
                    labels = c("Kernel", "True")) +
labs(x = expression("Values of"~x), y = expression("Density of"~f(x))) +
theme(plot.title = element_text(hjust = 0.5)) +
ggtitle(expression("Distribution of"~f(x)~"with"~theta==1))
```

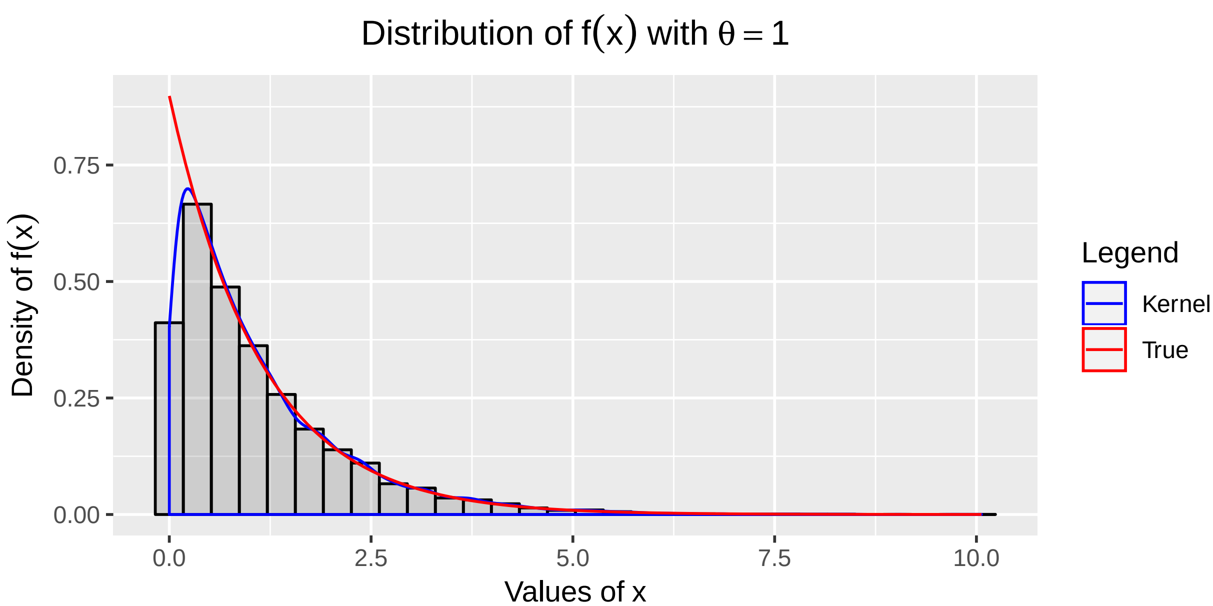

Figure 2: Histogram, Kernel Curve and True Curve for Generated Samples from $f(x)$ with $\theta = 1$

2 Mixture of Beta

2.1 Treating $f(x)$ as a Whole and Using Rejection Sampling

It's easy to see

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

One intuitive way to choose mixed Beta distributions(here we choose $\lambda = \frac{1}{2}$) is

$$g(x) = \lambda \text{Beta}(\theta, 1) + (1-\lambda)\text{Beta}(1, \beta) = \frac{\lambda}{\theta}x^{\theta-1} + \frac{1-\lambda}{\beta}(1-x)^{\beta-1} = \frac{1}{2\theta}x^{\theta-1} + \frac{1}{2\beta}(1-x)^{\beta-1}$$

And select $\alpha = \max\{2\theta, 4\beta\}$. The following inequality is straightforward

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

The pseudo-code is as follows

Algorithm 3 Sampling from $f(x)$ with treating $f(x)$ as a whole

```
1:  procedure MY PROCEDURE
2:      Sample U from U(0,1)
3:      if U < λ then
4:          Sample X from Beta(θ,1)
5:      else
6:          Sample X from Beta(1,β)
7:      end if
8:      Sample U_1 from U(0,1)
9:      if U_1 ⩽ q(X)/(αg(X)) then
10:         return X
11:     else
12:         Go back to 2
13:     end if
14: end procedure
```

```r
sample.g <- function(n, shape1, shape2, prob){
  x <- rep(0, n)
  u <- runif(n, 0, 1)
  g1 <- rbeta(n, shape1 = shape1[1], shape2 = shape2[1])
  g2 <- rbeta(n, shape1 = shape1[2], shape2 = shape2[2])

  x[u < prob] <- g1[u < prob]
  x[u >= prob] <- g2[u >= prob]

  x
}

sample.f <- function(n, shape1, shape2, prob){
  sample <- rep(0, n)
  iter <- 1
  alpha <- max(shape1[1] * 2, shape2[2] * 4)

  while (iter <= n) {
      x <- sample.g(1, shape1, shape2, prob)
      u <- runif(1, 0, 1)

      q <- x ^ (shape1[1] - 1)/(1 + x^2) + sqrt(2 + x^2) * (1 - x)^(shape2[2] - 1)
      g <- (1/2) * dbeta(x, shape1 = shape1[1], shape2 = shape2[1]) +
        (1/2) * dbeta(x, shape1 = shape1[2], shape2 = shape2[2])

      if(u <= q/(alpha * g)){
        sample[iter] <- x
        iter <- iter + 1
      }
  }
  sample
```

```
}

theta <- 2
beta <- 2
prob <- 1/2

shape1 <- c(theta, 1)
shape2 <- c(1, beta)
n <- 10000


h <- function(x, theta, beta){
  x ^ (theta - 1)/(1 + x^2) + sqrt(2 + x^2) * (1 - x)^(beta - 1)
}
integral <- integrate(function(x) h(x, theta = theta, beta = beta), 0, 1)

f <- function(x, theta, beta){
  (x ^ (theta - 1)/(1 + x^2) + sqrt(2 + x^2) * (1 - x)^(beta - 1))/integral$value
}

x <- sample.f(n, shape1, shape2, prob)

library("ggplot2")
ggplot(data.frame(x = x), aes(x = x)) +
geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
geom_density(aes(x, color = "a")) +
stat_function(aes(x, color = "b"), fun = function(x) f(x, theta = theta, beta = beta)) +
scale_colour_manual(name = "Legend", values = c("a" = "blue", "b" = "red"),
                    labels = c("Kernel", "True")) +
labs(x = expression("Values of"~x), y = expression("Density of"~f(x))) +
theme(plot.title = element_text(hjust = 0.5)) +
ggtitle(expression(paste(" Distribution of ", f(x), " with ", theta==2, ", ", beta==2 )))
```

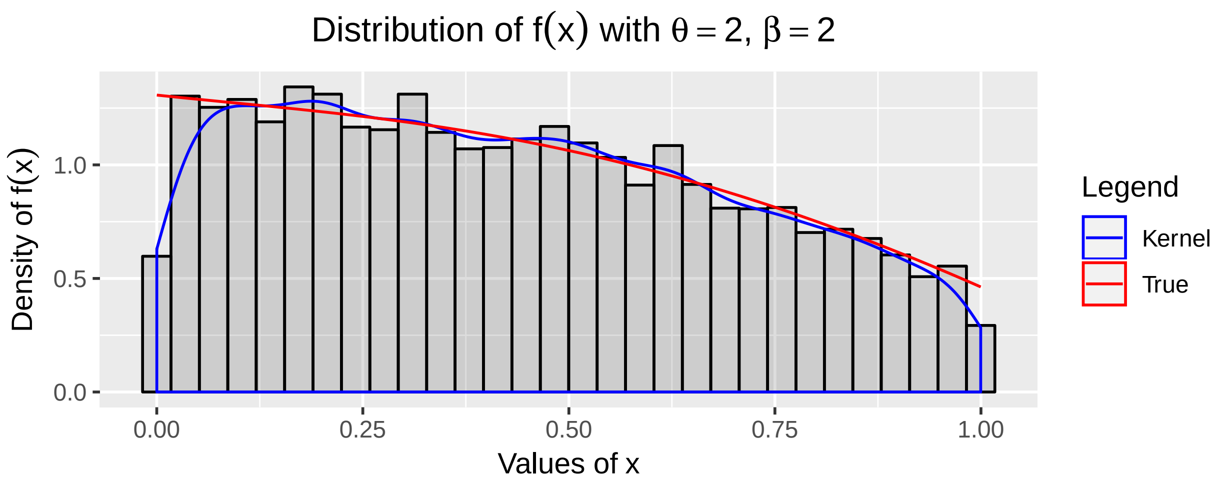

Figure 3: Histogram, Kernel Curve and True Curve for Generated Samples from $f(x)$ with $\theta = 2$, $\beta = 2$

2.2 Treating $f(x)$ as the Sum of Two Functions and Using Rejection Sampling

Let's define

$$q_1(x) = \frac{x^{\theta-1}}{1+x^2} \qquad q_2(x) = \sqrt{2+x^2}(1-x)^{\beta-1}$$
$$g_1(x) = \frac{1}{\theta}x^{\theta-1} \qquad g_2(x) = \frac{1}{\beta}(1-x)^{\beta-1}$$
$$\alpha_1 = \theta \qquad \alpha_2 = \sqrt{3}\beta$$

Thus

$$f(x) \propto q_1(x) + q_2(x)$$
$$q_1(x) \leqslant \alpha_1 g_1(x) \qquad q_2 \leqslant \alpha_2 g_2(x)$$

The pseudo-code is as follows

Algorithm 4 Sampling from $f(x)$ with treating $f(x)$ as the sum of two functions

```
procedure MY PROCEDURE
    Sample U from U(0,1)
    if U < α1/(α1+α2) then
        Sample X from g1(x) and U1 from U(0,1)
        if U1 ⩽ q1(X)/(α1 g1(X)) then
            return X
        else
            Go back to 2
        end if
    else
        Sample X from g2(x) and U2 from U(0,1)
        if U2 ⩽ q2(X)/(α2 g2(X)) then
            return X
        else
            Go back to 2
        end if
    end if
end procedure
```

```r
sample.f <- function(n, shape1, shape2){
  sample <- rep(0, n)
  iter <- 1
  alpha1 <- shape1[1]
  alpha2 <- sqrt(3) * shape2[2]
  prob <- alpha1/(alpha1 + alpha2)

  while (iter <= n) {
    u <- runif(1, 0, 1)

    if(u < prob){
      x <- rbeta(1, shape1 = shape1[1], shape2 = shape2[1])
      u1 <- runif(1, 0, 1)
```

```r
      if(u1 <= 1/(1 + x^2)){
        sample[iter] <- x
        iter <- iter + 1
      }
    }
    else{
      x <- rbeta(1, shape1 = shape1[2], shape2 = shape2[2])
      u2 <- runif(1, 0, 1)

      if(u2 <= sqrt(2 + x^2)/sqrt(3)){
        sample[iter] <- x
        iter <- iter + 1
      }
    }
  }
  sample
}

theta <- 2
beta <- 2

shape1 <- c(theta, 1)
shape2 <- c(1, beta)
n <- 10000


h <- function(x, theta, beta){
  x ^ (theta - 1)/(1 + x^2) + sqrt(2 + x^2) * (1 - x)^(beta - 1)
}
integral <- integrate(function(x) h(x, theta = theta, beta = beta), 0, 1)

f <- function(x, theta, beta){
  (x ^ (theta - 1)/(1 + x^2) + sqrt(2 + x^2) * (1 - x)^(beta - 1))/integral$value
}

x <- sample.f(n, shape1, shape2)

library("ggplot2")
ggplot(data.frame(x = x), aes(x = x)) +
geom_histogram(aes(y=..density..), color = "black", alpha = 0.2) +
geom_density(aes(x, color = "a")) +
stat_function(aes(x, color = "b"), fun = function(x) f(x, theta = theta, beta = beta)) +
scale_colour_manual(name = "Legend", values = c("a" = "blue", "b" = "red"),
                    labels = c("Kernel", "True")) +
labs(x = expression("Values of"~x), y = expression("Density of"~f(x))) +
theme(plot.title = element_text(hjust = 0.5)) +
ggtitle(expression(paste(" Distribution of ", f(x), " with ", theta==2, ", ", beta==2 )))
```

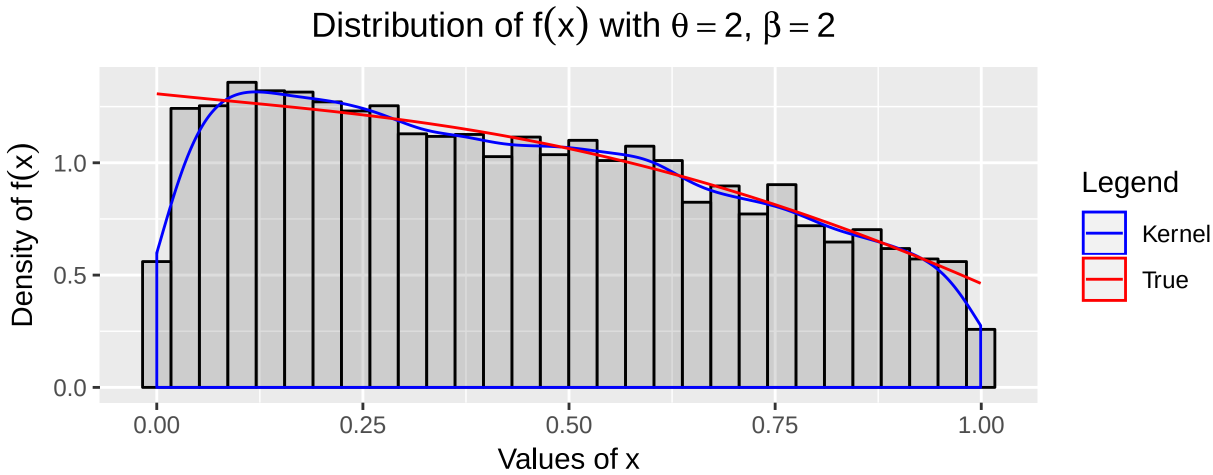

Figure 4: Histogram, Kernel Curve and True Curve for Generated Samples from $f(x)$ with $\theta = 2$, $\beta = 2$