Statistical Computing Homework 6

Jieying Jiao

13 October 2018

Abstract

This is Jieying Jiao's homework 6 for statistical computing, fall 2018.

Contents

1 Exercise 5.1.1

1.1 Normalize g

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

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

$$
\begin{aligned}
\Rightarrow g(x) &= \frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}\frac{1}{\Gamma(\theta)}x^{\theta-1}e^{-x} + \frac{\Gamma(\theta+1/2)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}\frac{1}{\Gamma(\theta+1/2)}x^{\theta+1/2-1}e^{-x} \\
&= \frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}Gamma(\theta,1) + \frac{\Gamma(\theta+1/2)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}Gamma(\theta+1/2,1)
\end{aligned}
$$

So g is a mixture of $Gamma(\theta, 1)$ and $Gamma(\theta+1/2, 1)$, with weights shown in the above formula.

1.2 Draw sample from g, mixture gamma

Pesudo-Code:

(1)Step1: Sample $U \sim Unif(0,1)$

(2)Step2:

if $U < \frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}\{$

sample $X \sim Gamma(\theta, 1)$

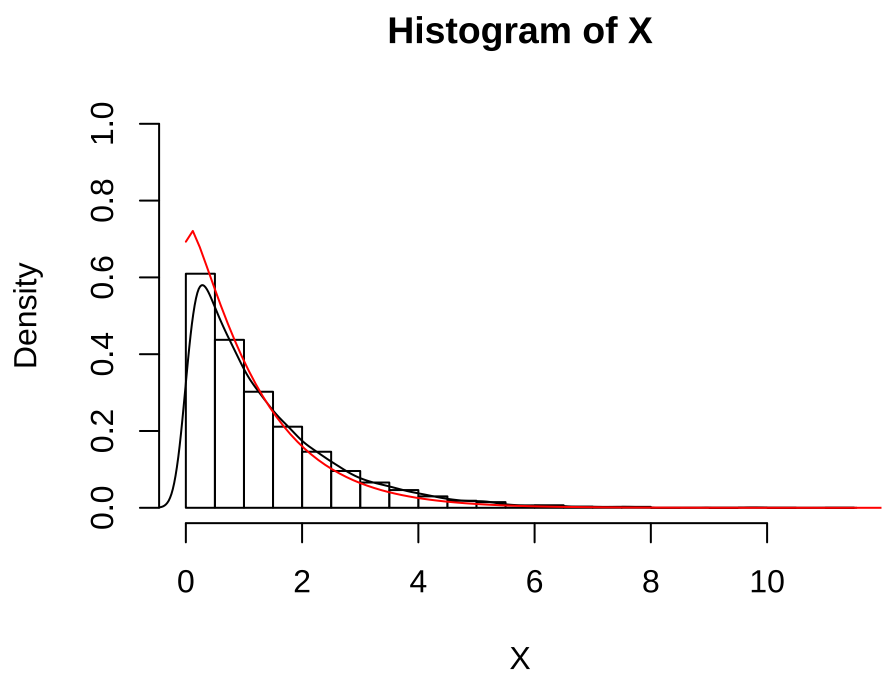
$\}$

else $\{$sample $X \sim Gamma(\theta + 1/2, 1)\}$

(3)Step3: Return X

```
rmixgamma <- function(shape, scale, probs, n) {
  x <- rep(0, n)
  for (i in 1:n) {
    u <- runif(1, 0, 1)
    if(u < probs[1]) {
      x[i] <- rgamma(1, shape = shape[1], scale = scale[1])
    } else x[i] <- rgamma(1, shape = shape[2], scale = scale[2])
  }
  return(x)
}

theta <- 1
shape <- c(theta, theta+1)
scale <- c(1, 1)
probs <- c(2*gamma(theta)/(2*gamma(theta)+gamma(theta+1/2)),
           gamma(theta+1/2)/(2*gamma(theta)+gamma(theta+1/2)))
n <- 10000
X <- rmixgamma(shape = shape, scale = scale, probs = probs, n = n)
hist(X, nclass = 20, probability = TRUE, ylim = c(0, 1))
points(density(X), type = "l")
gfunc <- function(x, para) {
  y <- (2*x^(para-1)+x^(para-0.5))*exp(-x)/(2*gamma(para)+gamma(para+0.5))
  return(y)
}
curve(gfunc(x, para = theta), from = 0, to = 12, add = TRUE, col = "red")
```



1.3 Sample from f, use rejection sampling

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

Pesudo-Code:

(1)Step1: Sample $X \sim g$, $U \sim Unif(0,1)$.

(2)Step2: if $U > \frac{q(x)}{\alpha g(x)}$ {

then go to step 1

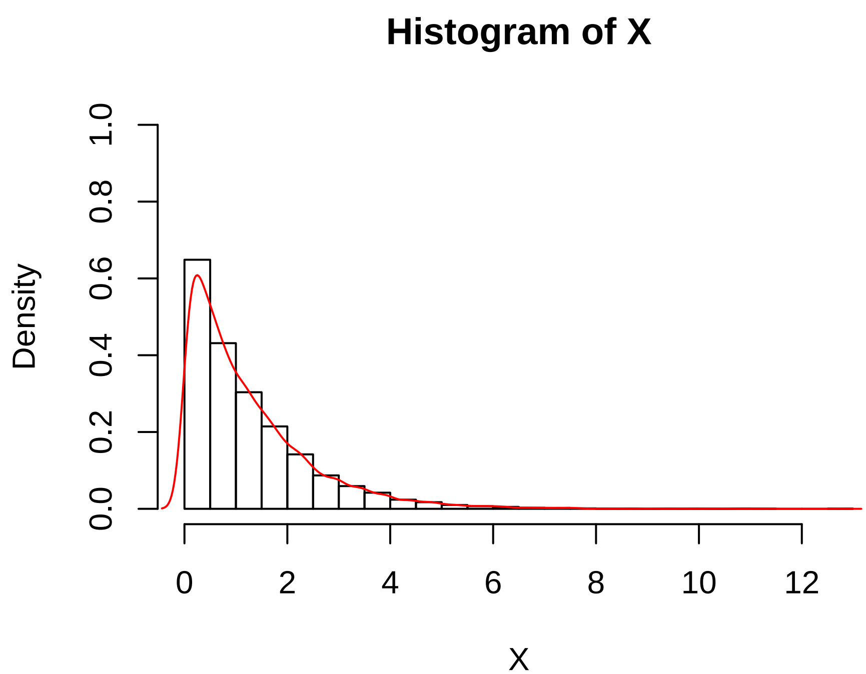
} else return X

(3)Step3: repeat step 1-2, until get desired number of sample.

```
rffunc <- function(n) {
  y <- rep(0, n)
  count <- 0
  while (count < n) {
    u <- runif(1, 0, 1)
    x <- rmixgamma(shape = shape, scale = scale, probs = probs, n = 1)
    if(u <= sqrt(4+x)/(2+sqrt(x))) {
```

```
      count <- count + 1
      y[count] <- x
    }
  }
  return(y)
}

X <- rffunc(10000)
hist(X, nclass = 20, probability = TRUE, ylim = c(0, 1))
points(density(X), type = "l", col = "red")
```



2 Exercise 5.1.2

2.1 Use mixture Beta distribution

I design g to be the mixture of $Beta(\theta, 1)$ and $Beta(1, \beta)$, with equal weights:

$$
\begin{aligned}
g(x) &= \frac{1}{2Beta(\theta,1)}x^{\theta-1} + \frac{1}{2Beta(1,\beta)}(1-x)^{\beta-1} \\
&= C_1 x^{\theta-1} + C_2(1-x)^{\beta-1} \\
\Rightarrow \alpha &= sup_{x\in(0,1)} \frac{\frac{x^{\theta-1}}{1+x^2} + \sqrt{2+x^2}(1-x)^{\beta-1}}{c_1 x^{\theta-1} + C_2(1-x)^{\beta-1}} \\
&= sup_{x\in(0,1)} \frac{x^{\theta-1} + \sqrt{2+x^2}(1+x^2)(1-x)^{\beta-1}}{C_1 x^{\theta-1}(1+x^2) + C_2(1-x)^{\beta-1}(1+x^2)}
\end{aligned}
$$

Easy to observe that the above fraction is bounded on $x \in (0,1)$, so α exists. For simplicity of calculation, we make $\theta = 2, \beta = 1$, then:

$$
\begin{aligned}
C_1 &= \frac{1}{2Beta(\theta,1)} = 1 \\
C_2 &= \frac{1}{2Beta(1,\beta)} = \frac{1}{2} \\
\alpha &= sup_{x\in(0,1)} \frac{x+\sqrt{2+x^2}(1+x^2)}{x(1+x^2)+\frac{1}{2}(1+x^2)} = 2\sqrt{2} \\
\alpha g(x) &= 2\sqrt{2}x + \sqrt{2}
\end{aligned}
$$

Pesudo-Code:

(1)Step1: Sample $X \sim g$, $U \sim Unif(0,1)$.

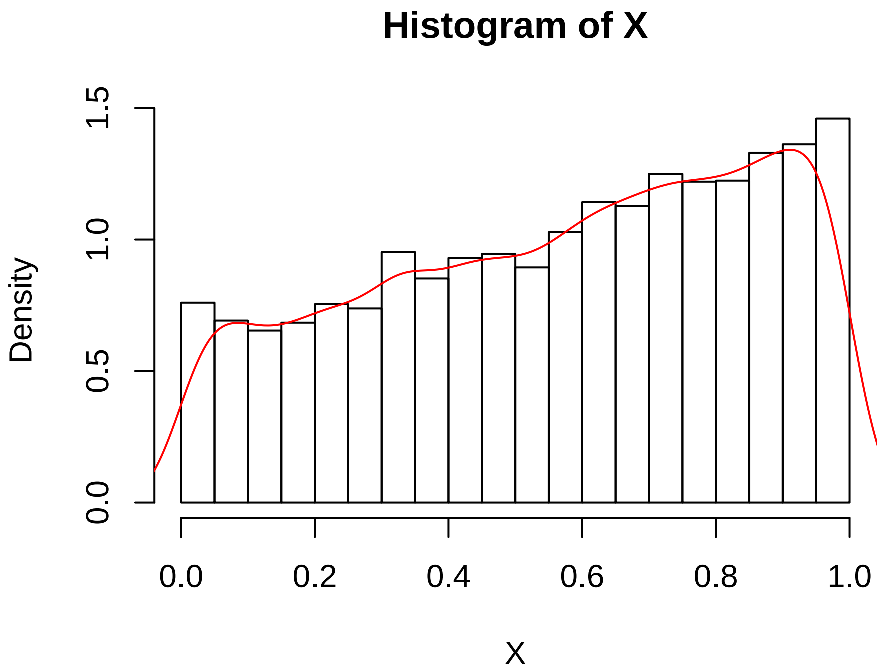
(2)Step2: if $U > \frac{g(x)}{\alpha g(x)}$ {

then go to step 1

} else return X

(3)Step3: repeat step 1-2, until get desired number of sample.

```r
rffunc2 <- function(n) {
  y <- rep(0, n)
  count <- 0
  while (count < n) {
    u <- runif(1, 0, 1)
    if (u < 0.5) {
      x <- rbeta(1, shape1 = 2, shape2 = 1)
    } else x <- rbeta(1, shape1 = 1, shape2 = 1)
    if (u <= (x/(1+x^2)+sqrt(2+x^2))/sqrt(2)/(2*x+1)) {
      count <- count+1
      y[count] <- x
    }
  }
  return(y)
}
X <- rffunc2(10000)
hist(X, nclass = 20, probability = TRUE)
points(density(X), type = "l", col = "red")
```



2.2 Use seperate Beta distribution

Let $g_1(x) \sim Beta(\theta, 1)$, $g_2(x) \sim Beta(1, \beta)$:

$$g_1(x) = \frac{1}{Beta(\theta,1)} x^{\theta-1}$$
$$g_2(x) = \frac{1}{Beta(1,\beta)} (1-x)^{\beta-1}$$
$$\alpha_1 = sup_{(0,1)} \frac{q_1(x)}{g_1(x)} = sup_{(0,1)} \frac{Beta(\theta,1)}{1+x^2} = Beta(\theta,1)$$
$$\alpha_2 = sup_{(0,1)} \frac{q_2(x)}{g_2(x)} = sum_{(0,1)} Beta(1,\beta)\sqrt{2+x^2} = \sqrt{3}Beta(1,\beta)$$
$$p_1 = \frac{\alpha_1}{\alpha_1+\alpha_2}$$
$$p_2 = \frac{\alpha_2}{\alpha_1+\alpha_2}$$

Pesudo-Code:

(1)Step1: Sample k from $\{1, 2\}$ with probs p_1, p_2

(2)Step2: Sample $X \sim g_k$, $U \sim Unif(0, 1)$

(3)Step3: if $U > \frac{q_k(X)}{\alpha_k g_k(X)}$ {

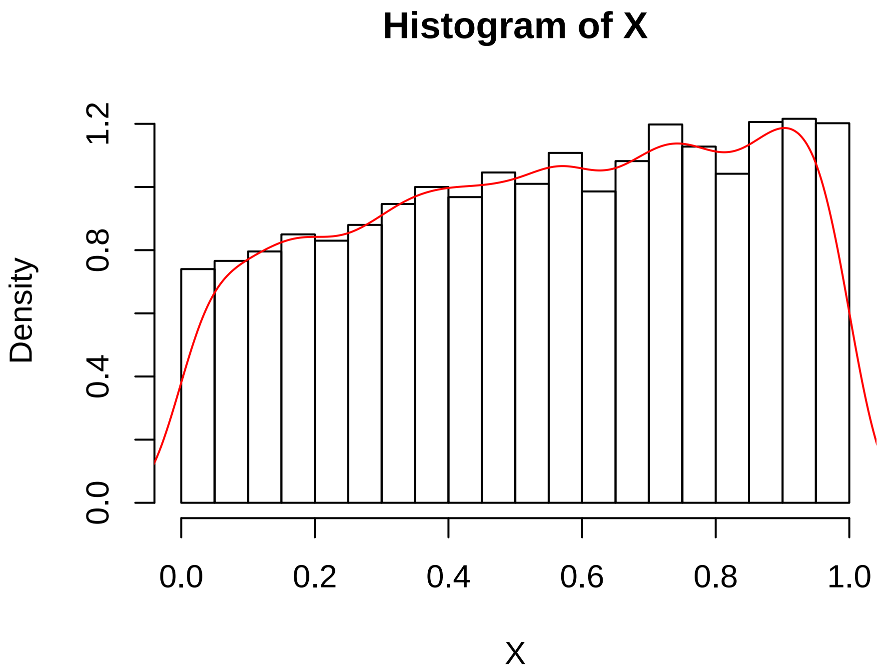
then go to step 1

} else return X

(4)Step4: Repeat Step 1-3 until get the desired number of sample.

```r
rffunc3 <- function(n, theta, beta_par) {
  alpha1 <- beta(theta, 1)
  alpha2 <- beta(1, beta_par) * sqrt(3)
  p1 <- alpha1/(alpha1+alpha2)
  p2 <- alpha2/(alpha1+alpha2)
  y <- rep(0, n)
  count <- 0
  while (count < n) {
    u <- runif(1, 0, 1)
    k <- sample(c(1, 2), size = 1, replace = 1, prob = c(p1, p2))
    if (k == 1) {
      x <- rbeta(1, shape1 = theta, shape2 = 1)
      if (u <= 1/(1+x^2)) {
        count <- count + 1
        y[count] <- x
      }
    } else {
     x <- rbeta(1, shape1 = 1, shape2 = beta_par)
     if (u <= sqrt((2+x^2)/3)) {
       count <- count + 1
       y[count] <- x
     }
    }
  }
  return(y)
}

X <- rffunc3(10000, theta = 2, beta_par = 1)
hist(X, nclass = 20, probability = TRUE)
points(density(X), type = "l", col = "red")
```


Histogram of X
Density
0.0
0.4
0.8
1.2
0.0
0.2
0.4
0.6
0.8
1.0
X