HW6

Cheng Huang 2658312

19 October 2018

Rejection Sampling

Find C and idenditfy the distribution

$\int_0^\infty 2x^{\theta-1}e^{-x}dx$ This is the kernel for Gamma$(\theta, 1)$. So $\int_0^\infty \frac{1}{\Gamma(\theta)}x^{\theta-1}e^{-x}dx = 1$, therefore $\int_0^\infty 2x^{\theta-1}e^{-x}dx = \frac{\Gamma(\theta)}{2}$. $\int_0^\infty x^{\theta-1/2}e^{-x}dx$ This is the kernel for Gamma$(\theta + 1/2,\ 1)$. So $\int_0^\infty \frac{1}{\Gamma(\theta+1/2)}x^{\theta-1/2}e^{-x}dx = 1$, so $\int_0^\infty x^{\theta-1/2}e^{-x}dx = \Gamma(\theta + 1/2)$.

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

So

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

So the weights are $\frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}$ and $\frac{\Gamma(\theta+1/2)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}$.

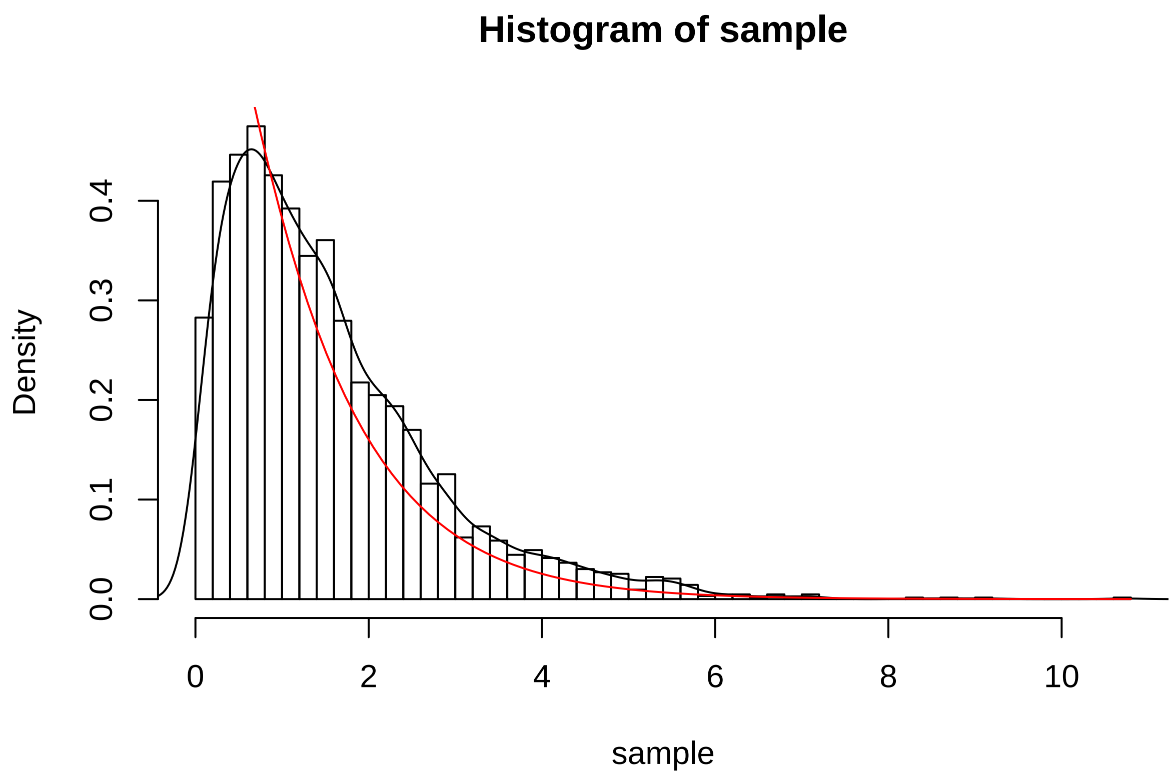
Draw sample from g

Pseudo - code
(1)draw U ~ U(0,1)
(2)if $U < \frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+1/2)}$, draw X ~ Gamma$(\theta, 1)$
else draw X ~ Gamma$(\theta + 1/2, 1)$
(3)return X

```r
knitr::opts_chunk$set(echo = TRUE)
require(data.table)

sample.g <- function(n, theta){
  u <- runif(n)
  g1 <- rgamma(n, shape = theta, rate = 1)
  g2 <- rgamma(n, shape = theta + 0.5, rate = 1)
  weight1 <- 2 * gamma(theta) / (2 * gamma(theta) + gamma(theta + 0.5))
  weight2 <- 1 - weight1
  if(u < weight1){
    return(g1[u < weight1])
  }else{
    return(g2[u >= weight1])
  }
}
sample <- sample.g(10000, 1)
```

```
true.dens <- function(x){
   C <- 1 / (2 * gamma(1) + gamma(1 + 0.5))
   return( C * (2 + x^{1 - 1/2}) * exp(1)^{-x})
}
hist(sample, prob=TRUE, breaks = 50)
lines(density(sample))
curve(true.dens(x), add = T, col = 'red')
legend(5, 0.7, legend= c("true density", "kernel"), col=c("red", "black"), lty = 1)
```



rejection sampling

First find minimal M such that $f(x) \leq Mg(x)$.

$$M = sup_{x>0}\frac{f(x)}{g(x)} = sup_{x>0}\frac{\sqrt{4+x}x^{\theta-1}e^{-x}}{C(2x^{\theta-1}+x^{\theta-1/2})e^{-x}} = sup_{x>0}\frac{\sqrt{4+x}}{C\sqrt{x}} = \frac{1}{C}$$

Therefore the criterion for accepting a sample is

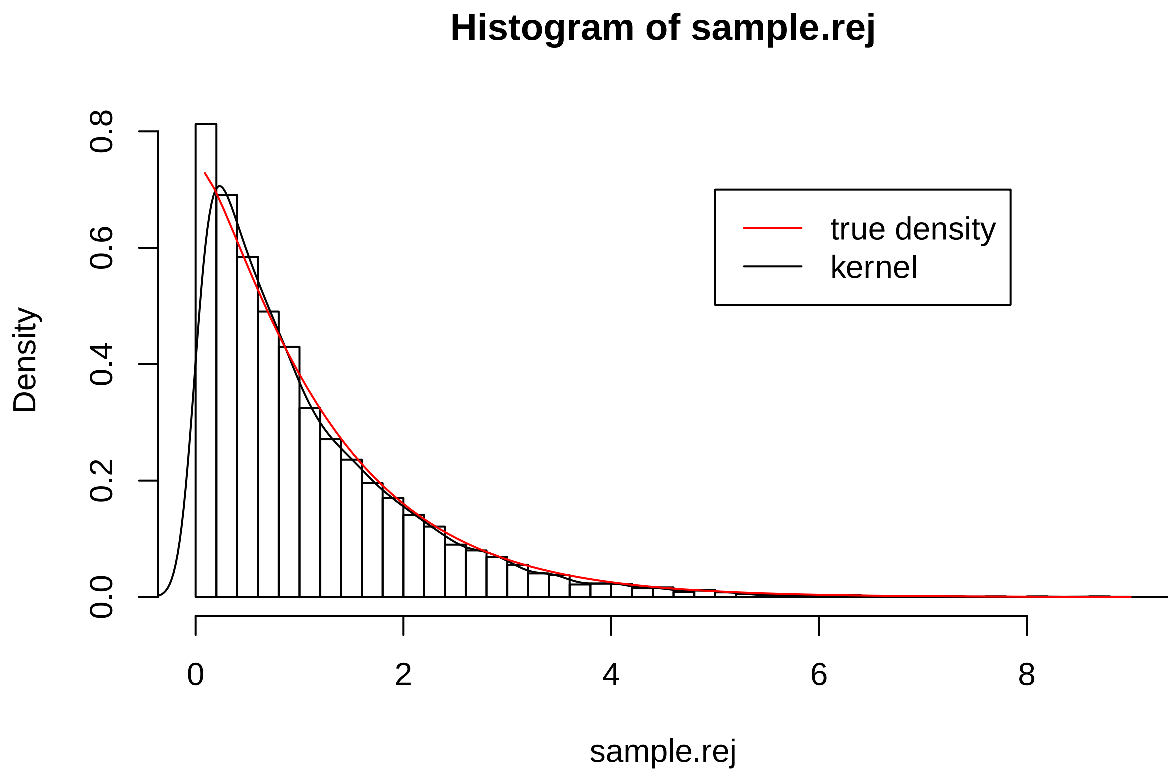
$$U \leq \frac{\sqrt{(x+4)}}{2+\sqrt{(x)}}$$

Pseudo - code
(1)draw U ~ U(0,1), X ~ g(x)

(2)if $U \leq \frac{f(x)}{Mg(x)}$, return X else go back to (1)
(3)return X

```
knitr::opts_chunk$set(echo = TRUE)
require(data.table)

rejection.g <- function(n, theta){
  iter <- 1
  sample <- rep(0,n)
  while (iter <= n) {
    u <- runif(1)
    x <- sample.g(1, theta)
    if(u <= sqrt(4 + x)/(2 + sqrt(x))){
      sample[iter] <- x
      iter <- iter +1
    }
  }
  sample
}
sample.rej <- rejection.g(10000, 1)
true.dens <- function(x){
   C <- 1 / (2 * gamma(1) + gamma(1 + 0.5))
   return( C * (2 + x^{1 - 1/2}) * exp(1)^{-x})
}
hist(sample.rej, prob = TRUE, breaks = 50)
lines(density(sample.rej))
curve(true.dens(x), add = T, col = 'red')
legend(5, 0.7, legend= c("true density", "kernel"), col=c("red", "black"), lty = 1)
```



Mixture Proposal

Mixture of Beta

$f(x) \propto \frac{x^{\theta-1}}{1+x^2} + \sqrt{2+x^2}(1-x)^{\beta-1} \leq x^{\theta-1} + \sqrt{3}(1-x)^{\beta-1}$, therefore choose $g(x) = \lambda Beta(\theta, 1) + (1-\lambda)Beta(1, \beta) = \lambda\theta x^{\theta-1} + (1-\lambda)\beta(1-x)^{\beta-1}$.
Choose $\lambda = 1/2$,

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

Therefore,

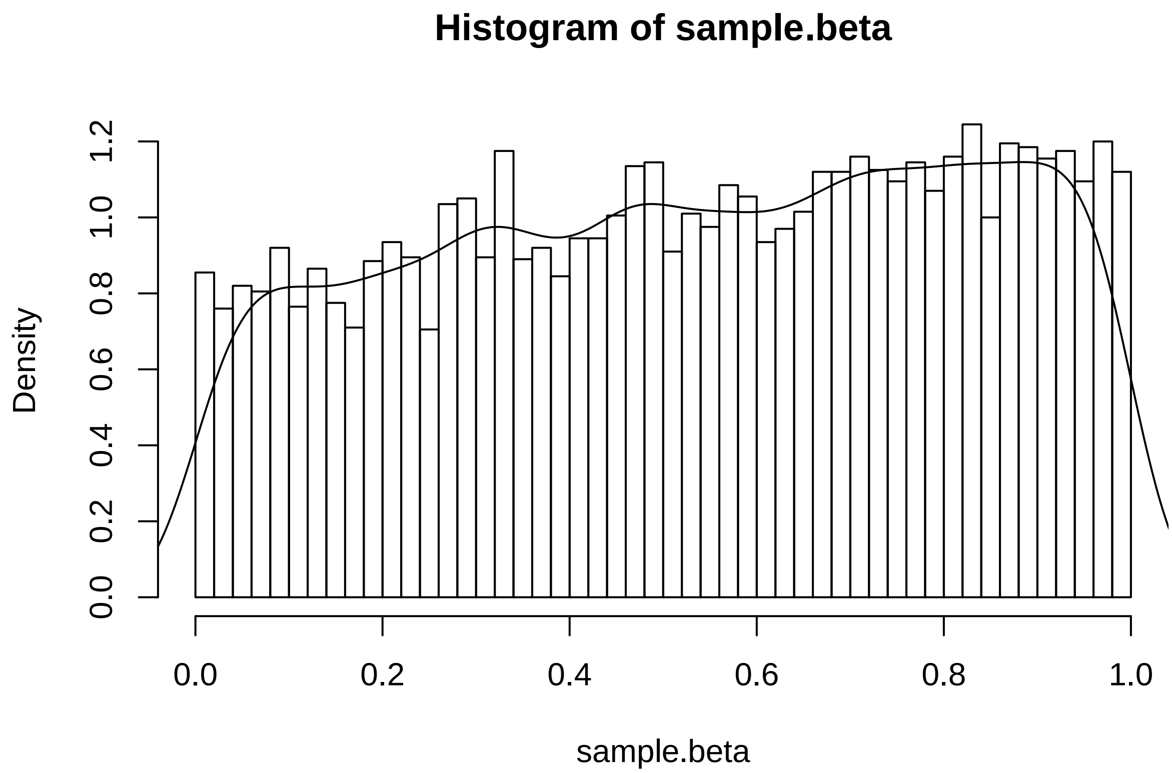
$$f(x) \leq x^{\theta-1} + \sqrt{3}(1-x)^{\beta-1} \leq \alpha g(x) = \frac{M\theta}{2}x^{\theta-1} + \frac{M\beta}{2}(1-x)^{\beta-1}$$

This Inequality holds when $M = max(2/\theta, 2\sqrt{3}/\beta)$.

Select $\theta = 2, \beta = 1, M = 2\sqrt{3}$. Then $Mg(x) = 2\sqrt{3}x + \sqrt{3}$,The sample is accepted when $U \leq \frac{f(X)}{Mg(X)} = \frac{\frac{x}{1+x^2}+\sqrt{2+x^2}}{2\sqrt{3}x+\sqrt{3}}$.
Pseudo - code
(1)draw U ~ U(0,1), X ~ g(x)
(2)if $U \leq \frac{f(x)}{Mg(x)}$, return X else go back to (1)
(3)return X

```
knitr::opts_chunk$set(echo = TRUE)
require(data.table)
sample.beta <- function(n){
  u <- runif(n)
  g1 <- rbeta(n, shape1 = 2, shape2 = 1)
  g2 <- rbeta(n, shape1 = 1, shape2 = 1)
  weight1 <- 1/2
  weight2 <- 1 - weight1
  if(u < weight1){
    return(g1[u < weight1])
  }else{
    return(g2[u >= weight1])
  }
}
rejection.beta <- function(n, theta){
  iter <- 1
  sample <- rep(0,n)
  while (iter <= n) {
    u <- runif(1)
    x <- sample.beta(1)
    if(u <= (x/(1 + x^2) + sqrt(2 + x^2))/(2 * sqrt(3) * x + sqrt(3))){
      sample[iter] <- x
      iter <- iter +1
    }
  }
  sample
}
sample.beta <- rejection.beta(10000)


hist(sample.beta, prob = TRUE, breaks = 50)
lines(density(sample.beta))
```



Dealing with two components using Beta

It is shown from above that for $x \in [0, 1]$

$$f_1(x) = \frac{x^{\theta-1}}{1+x^2} \leq x^{\theta-1}$$

So define $g_1(x) = \theta x^{\theta-1}$, such that

$$f_1(x) = \frac{x^{\theta-1}}{1+x^2} \leq x^{\theta-1} \leq M_1 g_1(x) = M_1 \theta x^{\theta-1}$$

So, the minimal of M_1 is $1/\theta$. So Criterion for accepting a sample is $U \leq \frac{f_1(x)}{M_1 g_1(x)} = \frac{1}{1+x^2}$
For the second component,

$$f_2(x) = \sqrt{2+x^2}(1-x)^{\beta-1} \leq \sqrt{3}(1-x)^{\beta-1}$$

So define $g_2(x) = \beta(1-x)^{\beta-1}$, such that

$$f_2(x) = \sqrt{2+x^2}(1-x)^{\beta-1} \leq \sqrt{3}(1-x)^{\beta-1} \leq M_2 g_2(x) = M_2 \beta (1-x)^{\beta-1}$$

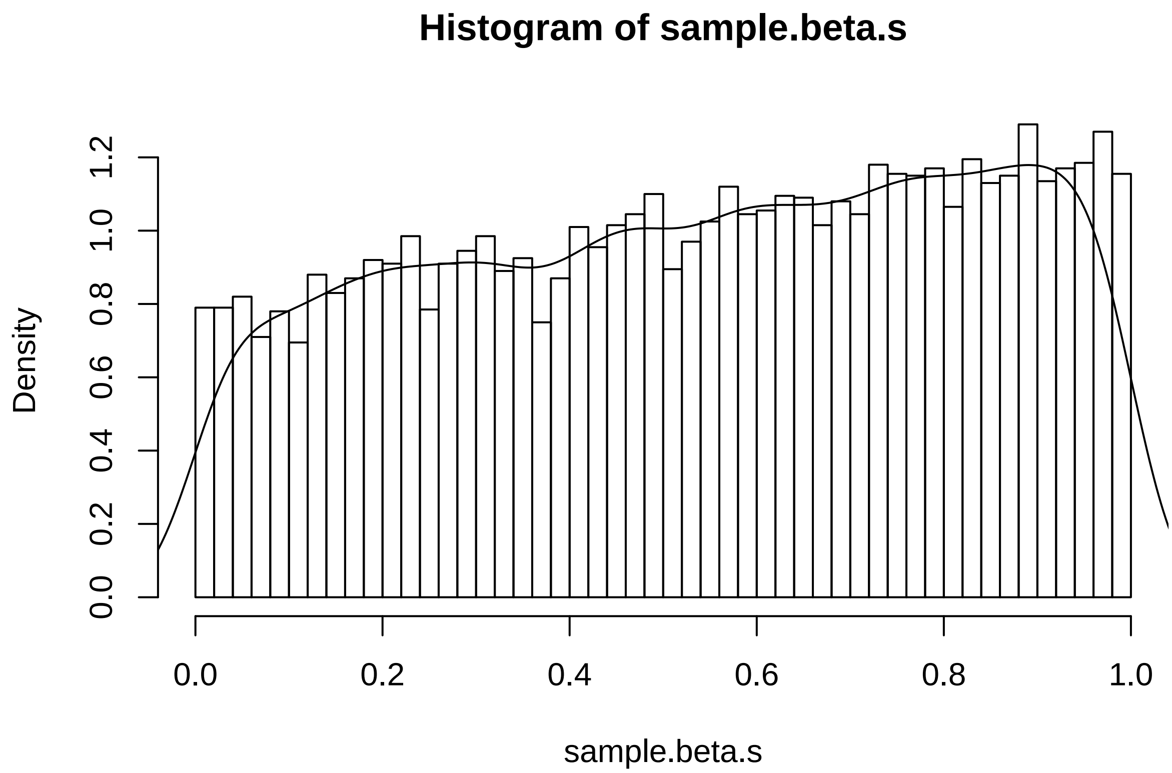
So, the minimal of M_1 is $\sqrt{3}/\beta$. So Criterion for accepting a sample is $U \leq \frac{f_2(x)}{M_2 g_2(x)} = \sqrt{\frac{2+x^2}{3}}$

Pseudo - code
(1)draw $U \sim U(0,1)$
(2)if $U \leq \frac{M_1}{M_1+M_2}$, draw $U_1 \sim U(0,1), X_1 \sim g_1(x)$
(3)if $U_1 \leq \frac{f_1(x)}{M_1 g_1(x)}$, return X_1 else go back to (2)
(4)if $U > \frac{M_1}{M_1+M_2}$, draw $U_2 \sim U(0,1), X_2 \sim g_2(x)$
(5)if $U_2 \leq \frac{f_2(x)}{M_2 g_2(x)}$, return X_2 else go back to (4)

Select $\theta = 2, \beta = 1$.

```
knitr::opts_chunk$set(echo = TRUE)
require(data.table)
sample.beta.s <- function(n){
  iter <- 1
  sample <- rep(0,n)
  u <- runif(n)
  M1 <- 1/2
  M2 <- sqrt(3)
  weight1 <- M1/(M1 + M2)
  while (iter <= n) {
    if(u[iter] < weight1){
      u1 <- runif(1)
      g1 <- rbeta(1, shape1 = 2, shape2 = 1)
      if(u1 <= 1/(1+g1^2)){
        sample[iter] <- g1
        iter <- iter +1
       }
    }else{
      u2 <- runif(1)
      g2 <- rbeta(n, shape1 = 1, shape2 = 1)
      if(u2 <= sqrt((2+g2^2)/3)){
        sample[iter] <- g2
        iter <- iter +1
       }
    }
  }
  sample
}

sample.beta.s <- sample.beta.s(10000)
hist(sample.beta.s, prob = TRUE, breaks = 50)
lines(density(sample.beta.s))
```



Reference

[jun-yan/stat-5361]https://github.com/jun-yan/stat-5361