Homework 6

Xiaokang Liu

19 October 2018

Contents

1 Rejection sampling

1.1 Find the value of the normalizing constant for g.

Since the kernel of a Gamma distribution $Gamma(\alpha, \beta)$ is $x^{(\alpha-1)}\exp(-x/\beta)$, it is easy to find that g is a mixture of Gamma distributions $Gamma(\theta, 1)$ and $Gamma(\theta + 1/2, 1)$. By solving

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

with the help of Gamma distribution function, we have $C = (2\Gamma(\theta) + \Gamma(\theta + 1/2))^{(-1)}$. And the weights for these two Gamma distributions in the mixture are

$$w_1 = \frac{2\Gamma(\theta)}{2\Gamma(\theta) + \Gamma(\theta + 1/2)}$$

and

$$w_2 = \frac{\Gamma(\theta + 1/2)}{2\Gamma(\theta) + \Gamma(\theta + 1/2)},$$

respectively.

1.2 Design a procedure (pseudo-code) to sample from g.

- sample $U \sim U(0, 1)$
- if $U < w_1$, sample $x \sim Gamma(\theta, 1)$, otherwise sample $x \sim Gamma(\theta + 1/2, 1)$.

1.3 Implement it in an R function, and draw a sample of size $n = 10,000$ using the function for at least one θ value.

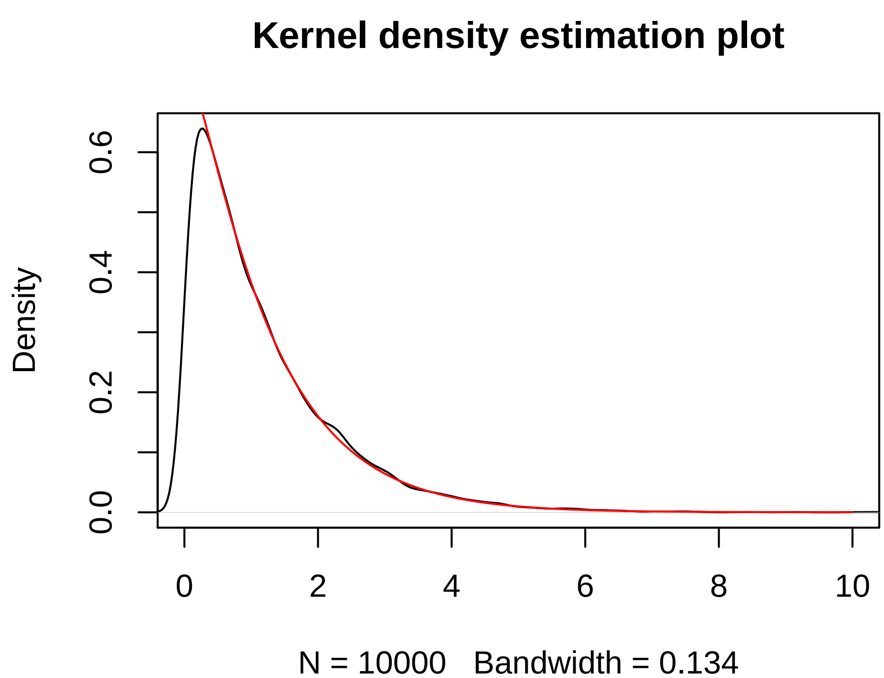
```
mysamp <- function(n,theta){
  rsamp <- rep(0,n)
  w1 <- 2*gamma(theta)/(2*gamma(theta)+gamma(theta+1/2))
  for (i in 1:n){
    u <- runif(1)
    if (u < w1){
      rsamp[i] <- rgamma(1,theta,1)}else{
      rsamp[i] <- rgamma(1,theta+1/2,1)}
  }
  return(list(samp=rsamp, w1=w1, w2=1-w1))
}

res <- mysamp(10000,1) # theta=1
samp1 <-  res$samp
```

1.4 Plot the kernel density estimation of g from your sample and the true density in one figure

The black one is from density estimation, and the red one is for the true density. All plots below have the same description.

```
# Kernel density estimation
plot(density(samp1), main = "Kernel density estimation plot", xlim = c(0,10))
curve(res$w1*dgamma(x,1,1)+res$w2*dgamma(x,3/2,1),from = 0, to = 10, add = TRUE, col = "red")
```



1.5 Design a procedure (pseudo-code) to use rejection sampling to sample from f using g as the instrumental distribution.

Since for $x > 0$ we have $\sqrt{4+x} < 2 + \sqrt{x}$, we know that

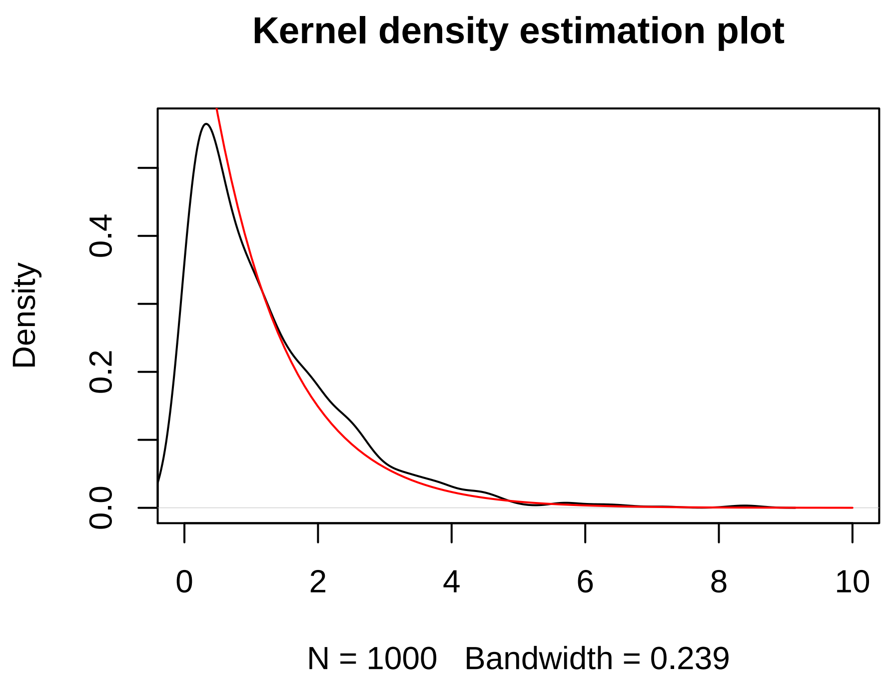
$$q(x) = \sqrt{4+x}x^{(\theta-1)}e^{(-x)} \leq \alpha g(x),\ x > 0.$$

Thus $\alpha g(x)$ is an instrumental distribution where $\alpha = 1/C$. The pseudo-code to use rejection sampling to sample from f using g is like this:

- sample $U \sim U(0,1)$ and $x \sim g(x)$
- if $U < \frac{q(x)}{\alpha g(x)}$, return x; otherwise go to the first step.

```
n <- 1000
theta <- 1
rsamp1 <- rep(0,n)
c0 <- 2*gamma(theta)+gamma(theta+1/2)
for (i in 1:n){
  repeat{
      u <- runif(1)
      g <- mysamp(1,theta)
      x <- g$samp
      rsamp1[i] <- x
      if (u < sqrt(4+x)*x^(theta-1)*exp(-x)*c0/(g$w1*dgamma(x,theta,1)+g$w2*dgamma(x,theta+1/2
    }
}

# for the true distribution f(x)
f1 <- function(x) {sqrt(4+x)*x^(theta-1)*exp(-x)}
cons <- integrate(f1, lower = 0, upper = Inf)$value
plot(density(rsamp1), main = "Kernel density estimation plot", xlim = c(0,10))
curve(sqrt(4+x)*x^(theta-1)*exp(-x)/cons,from = 0, to = 10, add = TRUE, col = "red")
```



2 Mixture Proposal

Let f be a probability density on (0,1) such that

$$f(x) = C(\frac{x^{(\theta-1)}}{1+x^2} + \sqrt{2+x^2}(1-x)^{(\beta-1)}),\ 0 < x < 1.$$

2.1 Design a procedure to sample from f using a mixture of Beta distributions as the instrumental density.

Since $\frac{x^{(\theta-1)}}{1+x^2} < x^{(\theta-1)}$ and $\sqrt{2+x^2}(1-x)^{(\beta-1)} < \sqrt{3}(1-x)^{(\beta-1)}$ when $0 < x < 1$, if we take

$$g(x) = w_1 Beta(\theta, 1) + w_2 Beta(1, \beta)$$

where $w_1 = \frac{B(\theta,1)}{B(\theta,1)+\sqrt{3}B(1,\beta)}$, and $w_2 = 1 - w_1$, $B(,)$ is a beta function, we have

$$q(x) \leq \alpha g(x)$$

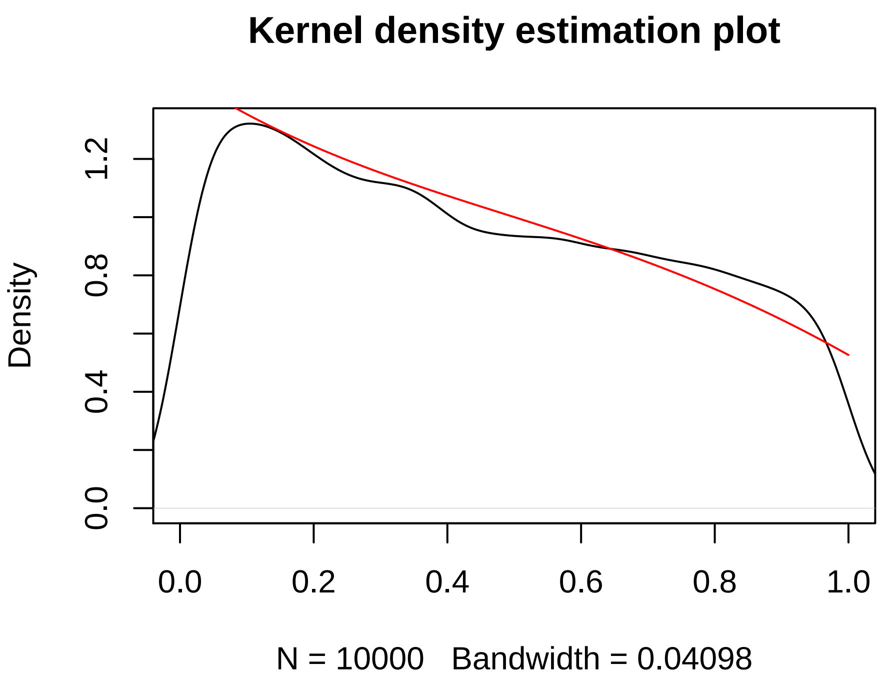
with $\alpha = B(\theta, 1) + \sqrt{3}B(1, \beta)$. The pseudo-code is

- sample $U \sim U(0, 1)$ and $x \sim g(x)$
- if $U < \frac{q(x)}{\alpha g(x)}$, return x; otherwise go to the first step.

```
mysamp2 <- function(n,theta,bet){
  rsamp <- rep(0,n)
  w1 <- beta(theta,1)/(beta(theta,1)+sqrt(3)*beta(1,bet))
 for (i in 1:n){
  u <- runif(1)
```

```r
  if (u < w1){rsamp[i] <- rbeta(1,theta,1)}else{rsamp[i] <- rbeta(1,1,bet)}
  }
  return(list(samp=rsamp, w1=w1, w2=1-w1))
}

n <- 10000
theta <- 3
bet <- 2
rsamp2 <- rep(0,n)
c2 <- 1/theta+sqrt(3)/bet
for (i in 1:n){
  u <- 100000
  upp <- 1
  low <- 1
  while (u > upp*c2/low){
      u <- runif(1)
      g <- mysamp2(1,theta,bet)
      x <- g$samp
      rsamp2[i] <- x
      upp <- (x^(theta-1)/(1+x^2)+sqrt(2+x^2)*(1-x)^(bet-1))
      low <- g$w1*dbeta(x,theta,1)+g$w2*dbeta(x,1,bet)
  }
}
f2 <- function(x) {x^(theta-1)/(1+x^2)+sqrt(2+x^2)*(1-x)^(bet-1)}
cons2 <- integrate(f2, lower = 0, upper = 1)$value
plot(density(rsamp2), main = "Kernel density estimation plot", xlim=c(0,1))
curve((x^(theta-1)/(1+x^2)+sqrt(2+x^2)*(1-x)^(bet-1))/cons2,
      from = 0, to = 1, add = TRUE, col = "red")
```



2.2 Dealing with the two components separatlely using individual Beta distributions.

For the component $x^{(\theta-1)}/(1+x^2)$, we can use $Beta(\theta,1)$ as its instrumental density. For another component $\sqrt{2+x^2}(1-x)^{\beta-1}$, we can use $Beta(1,\beta)$ as its instrumental density. Also we have,

$$\alpha_1 = \sup_{0<x<1} \frac{x^{(\theta-1)}/(1+x^2)}{Beta(\theta,1)} = B(\theta,1)$$

and

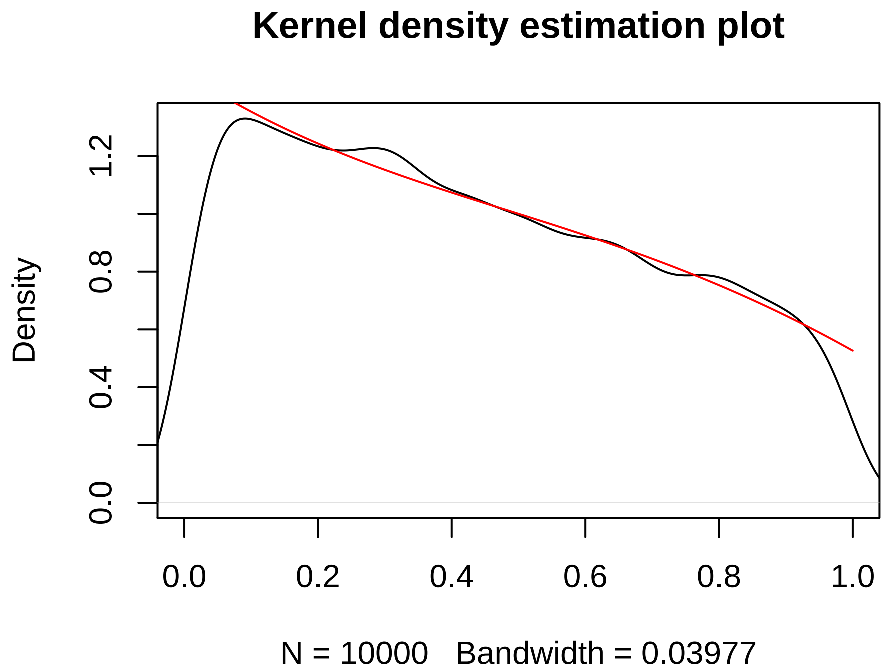
$$\alpha_2 = \sup_{0<x<1} \frac{\sqrt{2+x^2}(1-x)^{\beta-1}}{Beta(1,\beta)} = \sqrt{3}B(1,\beta).$$

Thus, the pseudo-code is:

- sample k from $\{1,2\}$ with probabilities $p_k = \alpha_k/(\alpha_1+\alpha_2)$.
- sample $x \sim g_k$ and $U \sim unif(0,1)$.
- if $U > \frac{q_k(x)}{\alpha_k g_k(x)}$ then go to step 1; other return x.

The r code should be:

```
alpha1 <- 1/theta
alpha2 <- sqrt(3)/bet
p1 <- alpha1/(alpha1+alpha2)
p2 <- 1-p1
rsamp3 <- rep(0,10000)
for (i in 1:10000){
     u1 <- u2 <- 10000000
     x <- 0.9
     while ((u1 > 1/(1+x^2)) & (u2 > sqrt(2+x^2)/sqrt(3))) {
       u1 <- u2 <- 10000000
       u <- runif(1)
       if (u < p1) {k <- 1} else {k <- 2}
       if (k==1){
         x <- rbeta(1,theta,1)
         u1 <- runif(1)
         rsamp3[i] <- x
       }else{
         x <- rbeta(1,1,bet)
         u2 <- runif(1)
         rsamp3[i] <- x
       }
     }
}
plot(density(rsamp3), main = "Kernel density estimation plot", xlim=c(0,1))
curve((x^(theta-1)/(1+x^2)+sqrt(2+x^2)*(1-x)^(bet-1))/cons2,
      from = 0, to = 1, add = TRUE, col = "red")
```


Kernel density estimation plot
Density
0.0
0.4
0.8
1.2
0.0
0.2
0.4
0.6
0.8
1.0
N = 10000 Bandwidth = 0.03977