RNG

Yaqiong Yao

10/19/2018

5.2.1 Rejection sampling

1

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

Thus $C = \left(2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})\right)^{-1}$. The components are two gamma distribution $G(\theta,1)$ and $G(\theta+\frac{1}{2},1)$ with weights $2\Gamma(\theta)C$ and $\Gamma(\theta+\frac{1}{2})C$.

2

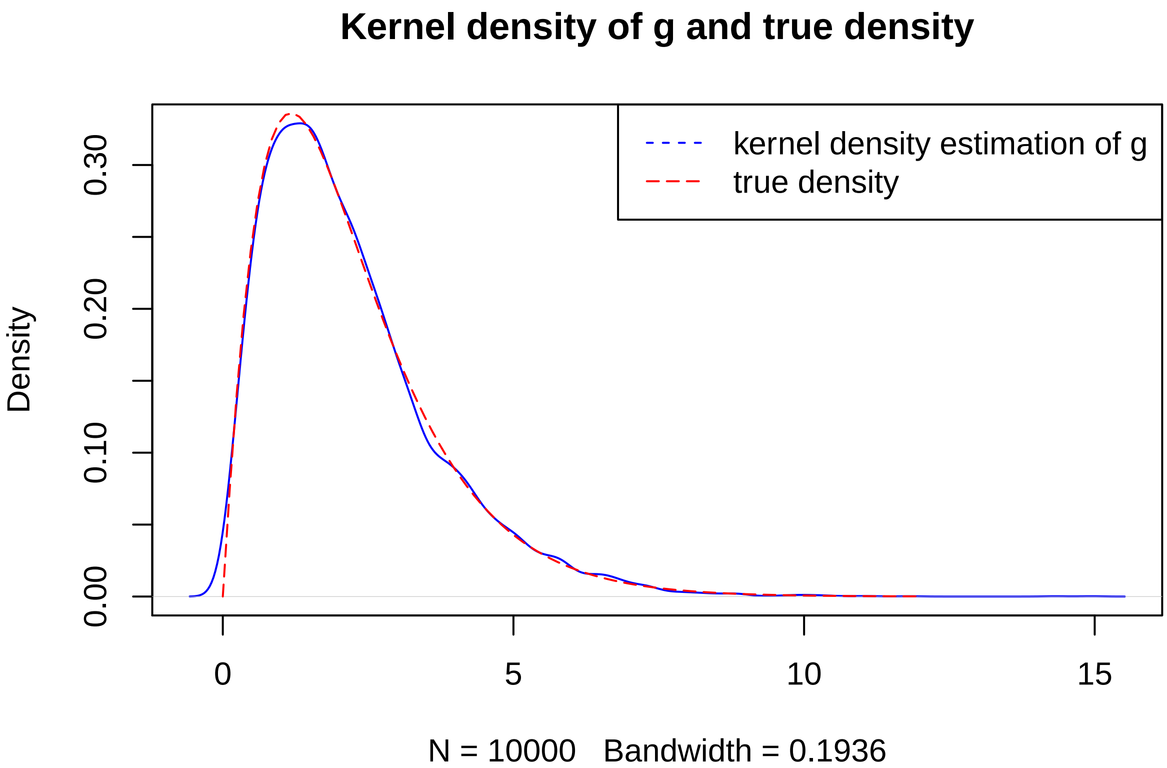
Algorithm 1 Sample from g(x)

1. Sample $u \sim \mathcal{U}(0,1)$
2. Calculate

$$X = I(u < 2\Gamma(\theta)C)Gamma(\theta,1)+I(u > 2\Gamma(\theta)C)Gamma(\theta+\frac{1}{2},1)$$

```r
g.sample <- function(n, theta){
  C <- 1/(2*gamma(theta)+gamma(theta + 1/2))
  ind <- runif(n, 0, 1)
  g.sample <- as.numeric(ind < 2*gamma(theta)*C) * rgamma(n, shape = theta, scale = 1) +
    (1 - as.numeric(ind < 2*gamma(theta)*C)) * rgamma(n, shape = (theta + 1/2), scale = 1)
  return(g.sample)
}
n <- 10000
theta <- 2
set.seed(123)
g.s <- g.sample(n, theta = theta)


gx <- function(x){
  C <- 1/(2*gamma(theta)+gamma(theta + 1/2))
  2*gamma(theta)*C*dgamma(x, shape = theta, scale = 1) +
    gamma(theta + 1/2)*C*dgamma(x, shape = (theta + 1/2), scale = 1)
}

plot(density(g.s), main = "Kernel density of g and true density", col = "blue", lty = 1)
plot(gx, 0, 12, add = TRUE, col = "red", lty = 5)
```

```r
legend("topright", col = c("blue", "red"), c("kernel density estimation of g", "true density"),
       lty = c(2, 5))
```



Here, I set $\theta = 2$.

3

First, we need to determine a α that satisfies $\alpha = \sup \frac{q(x)}{g(x)}$.

$$
\begin{aligned}
\alpha &= \sup \frac{q(x)}{g(x)} \\
&= \sup \frac{\sqrt{4+x}}{C(2+x^{\frac{1}{2}})}
\end{aligned}
$$

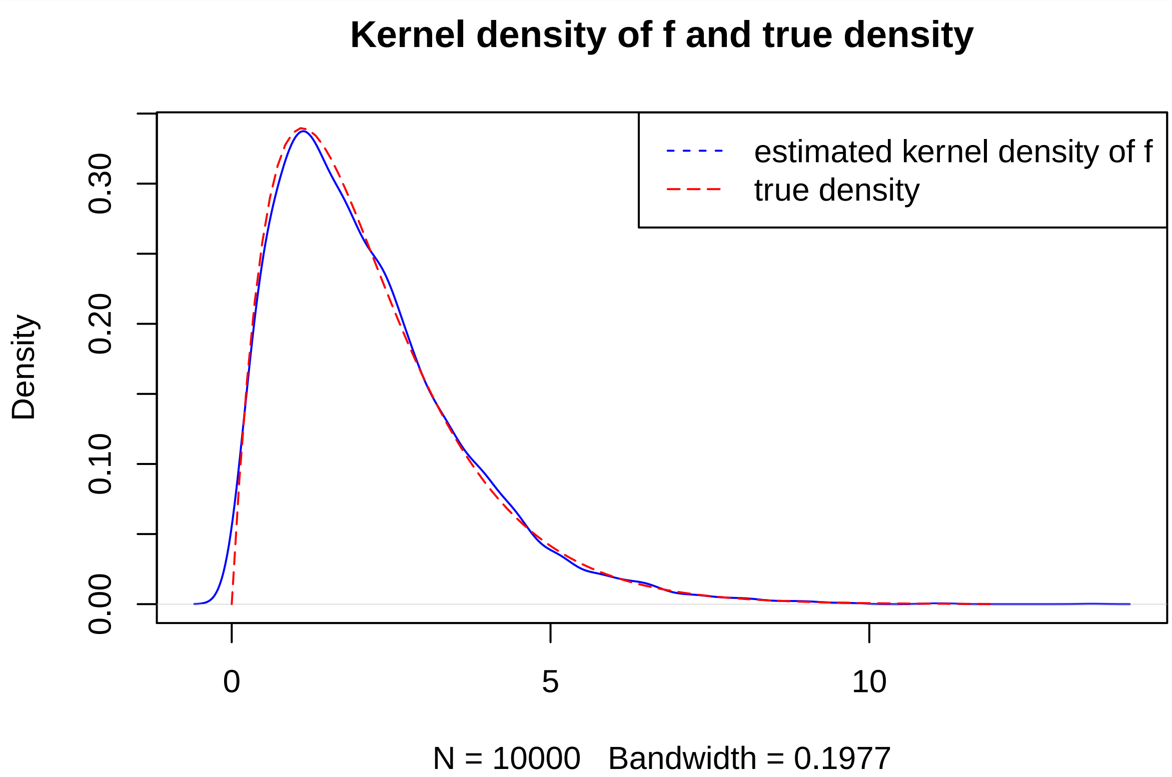
where $q(x) = \sqrt{4+x}x^{\theta-1}e^{-x}$.

Since $f(x) = \sqrt{x}$ is a convace function, $\alpha = \sup \frac{q(x)}{g(x)} = \frac{1}{C}$. So, clearly, if we substitue $g(x)$ with $h(x) = (2x^{\theta-1} + x^{\theta-\frac{1}{2}})e^{-x}$. Then we should use $\beta = 1$ instead of α.

Algorithm 2 Sample f(x) from g(x)

1. Sample $u \sim \mathcal{U}(0,1)$
2. Sample X from $g(x)$
3. Calculate
$$rate = \frac{\sqrt{4+X}}{2+X^{\frac{1}{2}}}$$
4. Keep samples let $rate > u$

```r
f.rs <- function(n, theta){
  u <- runif(n, 0, 1)
  g_s <- g.sample(n, theta)
  rate <- sqrt(4 + g_s)/(2 + g_s^(1/2))
  f.s <- g_s[u < rate]
  return(f.s)
}

n <- 15000
theta <- 2
f.s <- f.rs(n, theta)[1:10000]


fx <- function(x){
  qx <- function(x) sqrt(4 + x) * (x^(theta-1)) * exp(-x)
  G <- integrate(qx, 0, Inf)
  (sqrt(4 + x) * (x^(theta-1)) * exp(-x))/G$value
}

plot(density(f.s), main = "Kernel density of f and true density", col = "blue", lty = 1)
plot(fx, 0, 12, add = TRUE, col = "red", lty = 5)
legend("topright", col = c("blue", "red"), c("estimated kernel density of f", "true density"),
       lty = c(2, 5))
```



5.2.2 Mixture proposal

1

Use the mixture beta distribution of $\frac{1}{3}I(\theta \leq \beta)Beta(\theta, 1) + \frac{2}{3}I(\theta \leq \beta)Beta(1, \theta) + \frac{1}{3}I(\theta > \beta)Beta(\beta, 1) + \frac{2}{3}I(\theta > \beta)Beta(1, \beta)$.

Since

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

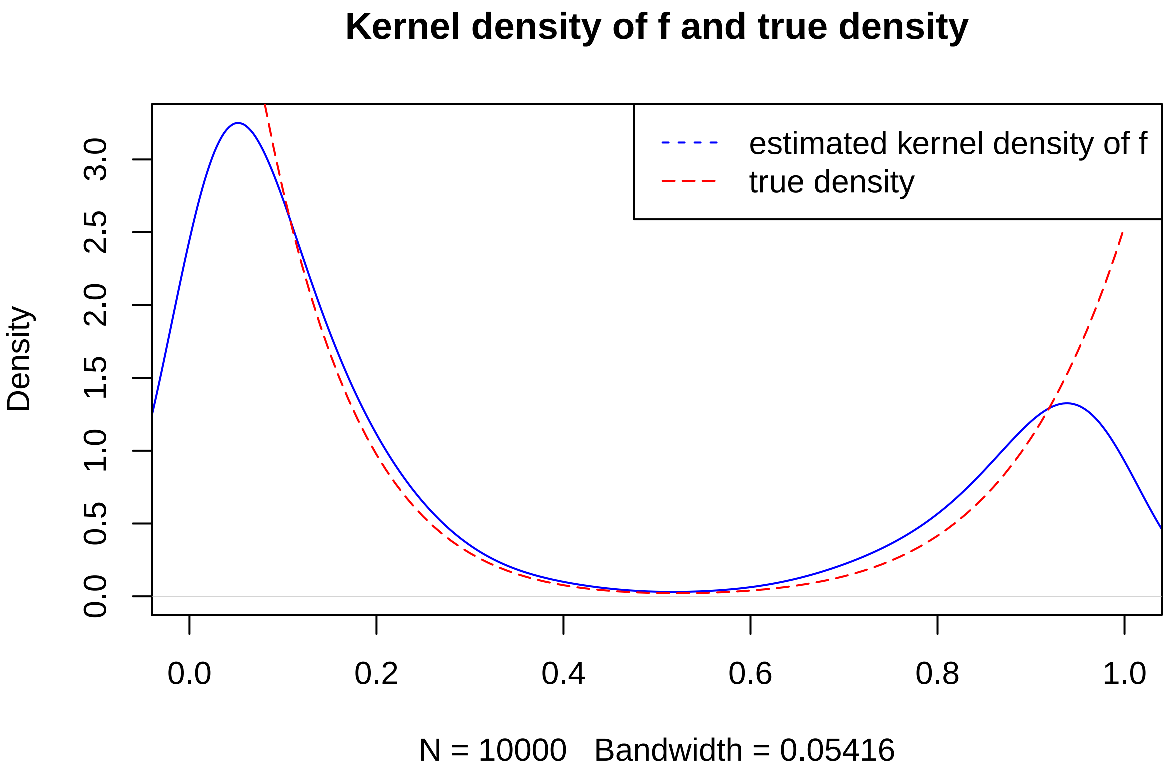
where $a = min(\theta, \beta)$.

Algorithm 3 Sample f(x) from mixture beta distribution

1. Sample $u \sim \mathcal{U}(0, 1)$
2. Set $a = min(\theta, \beta)$
3. Sample X from mixture beta distribution
4. Calculate rate
$$\frac{\frac{X^{\theta-1}}{1+X^2} + \sqrt{2+X^2}(1-X)^{\beta-1}}{X^a + 2(1-X)^{a-1}}$$
5. Keep samples let rate $> u$

```
mix.beta <- function(n, beta, theta){
  a <- min(theta, beta)
  u <- runif(n, 0, 1)
  s <- as.numeric(u > 1/3)*rbeta(n, shape1 = 1, shape2 = a) +
    (1-as.numeric(u > 1/3))*rbeta(n, shape1 = a, shape2 = 1)
  return(s)
}

mix.rs <- function(n, beta, theta){
  u <- runif(n, 0, 1)
  a <- min(beta, theta)
  mbs <- mix.beta(n, beta, theta)
  rate <- (mbs^(theta-1)/(1+mbs^2) + sqrt(2 + mbs^2)*(1-mbs)^(beta-1))/(mbs^a+2*(1-mbs)^(a-1))
  m.sample <- mbs[u < rate]
  return(m.sample)
}

theta <- 10
beta <- 10
n <- 22000
set.seed(123)
sp <- mix.rs(n, beta, theta)[1:10000]

fx <- function(x){
  qx <- function(x) x^(theta-1)/(1+x^2) + sqrt(2 + x^2)*(1-x)^(beta-1)
  (x^(theta-1)/(1+x^2) + sqrt(2 + x^2)*(1-x)^(beta-1))/integrate(qx, 0, 1)$value
}

plot(density(sp), xlim = c(0,1), main = "Kernel density of f and true density", col = "blue", lty = 1)
plot(fx, 0, 1, add = TRUE, col = "red", lty = 5)
```

```
legend("topright", col = c("blue", "red"), c("estimated kernel density of f", "true density"),
       lty = c(2, 5))
```



2

Since

$$\frac{x^{\theta-1}}{1+x^2} \leq x^{\theta-1}$$

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

Thus, I use two beta distribution $Beta(\theta, 1)$ and $Beta(1, \beta)$. Here $\alpha_1 = \frac{1}{\theta}$ and $\alpha_2 = \frac{\sqrt{3}}{\beta}$.

Algorithm 4 Sample f(x) seperately

1. Sample $u_1 \sim \mathcal{U}(0,1)$
2. If $u_1 > \frac{\frac{1}{\theta}}{\frac{1}{\theta}+\frac{\sqrt{3}}{\beta}}$, sample X from $Beta(1, \beta)$ and calculate rate using $\frac{1}{1+X^2}$; otherwise, sample X from $Beta(\theta, 1)$ and calculate rate using $\sqrt{\frac{2+X^2}{3}}$
3. Sample $u_2 \sim \mathcal{U}(0,1)$
4. Keep samples let rate $> u_2$

```
sep.beta <- function(n, beta, theta){
  X <- rep(NA, n)
  rate <- rep(NA, n)
  u1 <- runif(n, 0, 1)
  ind <- 1/theta + sqrt(3)/beta
```

```
  I1 <- u1 > (1/theta)/ind; n1 <- sum(u1 > (1/theta)/ind)
  I2 <- u1 <= (1/theta)/ind; n2 <- n-n1
  X[I1] <- rbeta(n1, shape1 = 1, shape2 = beta)
  X[I2] <- rbeta(n2, shape1 = theta, shape2 = 1)

  u2 <- runif(n, 0, 1)
  rate[I1] <- 1/(1 + X[I1]^2)
  rate[I2] <- sqrt((2+X[I2]^2)/3)

  s <- X[u2 < rate]
  return(s)
}

theta <- 10
beta <- 10
n <- 12000
set.seed(123)
sp <- sep.beta(n, beta, theta)[1:10000]

plot(density(sp), xlim = c(0,1), main = "Kernel density of f and true density", col = "blue", lty = 1)
plot(fx, 0, 1, add = TRUE, col = "red", lty = 5)
legend("topright", col = c("blue", "red"), c("estimated kernel density of f", "true density"),
       lty = c(2, 5))
```