DATA130004: Homework 2

董晴园 14300680173

2017.10.15

1. Exercise 3.6: Prove that the accepted variates generated by the acceptance-rejection sampling algorithm are a random sample from the target density f_X.
   Proof: In the discrete case,

$$\begin{aligned} P(accept|Y=y) &= P(U<\frac{f(y)}{cg(y)}|Y=y) \\ &= P(U<\frac{f(y)}{cg(y)}) = \frac{f(y)}{cg(y)} \\ \Rightarrow P(accept) &= \sum_y P(accept|Y=y)P(Y=y) \\ &= \sum_y \frac{f(y)}{cg(y)} g(y) = \frac{1}{c}\sum_y f(y) = \frac{1}{c} \end{aligned}$$

$$\begin{aligned} \therefore P(X=y|accept) &= \frac{P(accept|Y=y)P(Y=y)}{P(accept)} \\ &= \frac{\frac{f(y)}{cg(y)} g(y)}{\frac{1}{c}} \\ &= f(y) \end{aligned}$$

   In the continuous case,

$$P(accept) = \int_{-\infty}^{\infty} \frac{f(y)}{c(y)} g(y) = \frac{1}{c}\int_{-\infty}^{\infty} f(y) = \frac{1}{c}$$

$$\begin{aligned} P(Y\le y|accept) &= \frac{P(U\le \frac{f(y)}{cg(y)}, Y\le y)}{\frac{1}{c}} \\ &= c\int_{-\infty}^{y} P((U\le \frac{f(y)}{cg(y)}|Y=t\le y) g(t)\mathrm{d}t \\ &= c\int_{-\infty}^{y} \frac{f(t)}{cg(y)} g(t)\mathrm{d}t \\ &= F_x(y) \end{aligned}$$

2. Exercise 3.7: Write a function to generate a random sample of size n from the Beta(a, b) distribution by the acceptance-rejection method. Generate a random sample of size 1000 from the Beta(3,2) distribution. Graph the histogram of the sample with the theoretical Beta(3,2) density superimposed.

```
n<-1000
j<-k<-0
y<-numeric(n)
while(k<n){
  u<-runif(1)
  j<-j+1
  x<-runif(1)
  if(x^2*(1-x)>u)  {
    k<-k+1
    y[k]<-x
    }
  }
```

Here is the histogram of the sample from Beta(3,2) distribution:

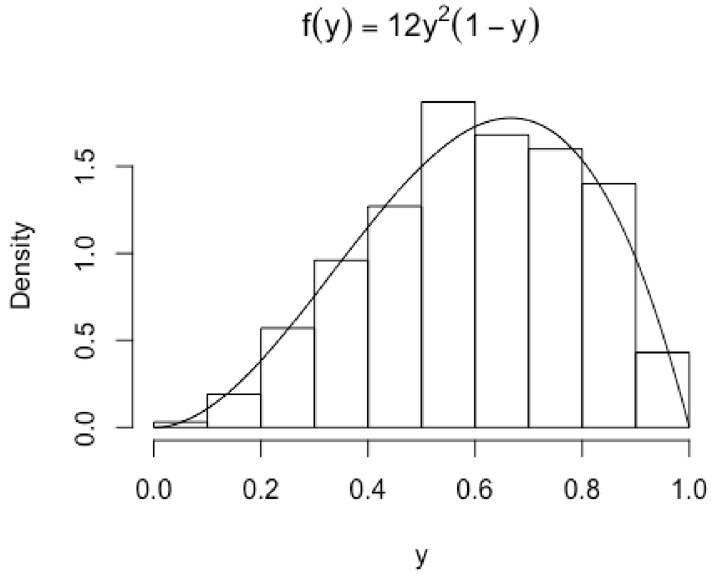

Figure 1: density histogram of sample from Beta(3,2) with theoretical density

3. Exercise 3.9: The rescaled Epanechnikov kernel is a symmetric density function

$$f_e(x) = \frac{3}{4}(1 - x^2), \quad |x| \leq 1.$$

Devroye and Gyorfi give the following algorithm for simulation from this distribution. Generate iid $U_1, U_2, U_3 \sim$ Uniform(−1,1). If $|U3| \geq |U2|$ and $|U3| \geq |U1|$, deliver U_2 ; otherwise deliver U_3 . Write a function to generate random variates from f_e, and construct the histogram density estimate of a large simulated random sample.

```
n <- 1000
j <- 0
x <- numeric(n)
while (j < n) {
```

```
  j <- j + 1
  u1 <- runif(1,min=-1,max=1)
  u2 <- runif(1,min=-1,max=1)
  u3 <- runif(1,min=-1,max=1)
  if (abs(u3)>=abs(u2) & abs(u3)>=abs(u1)) {
    x[j] <- u2
  }
  else {
    x[j] <- u3
  }
}
hist(x, prob = TRUE, main = expression(fe(x) == 3/4*(1-x^2)))
y <- seq(-1,1,0.001)
lines(y, 3/4*(1-y^2))
```

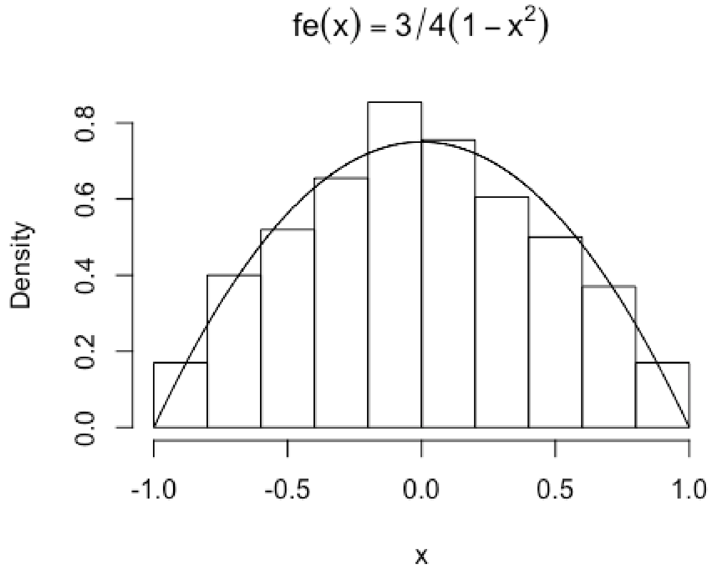

Figure 2: density histogram of sample from f_e with theoretical density

4. Exercise 3.10: Prove that the algorithm given in Exercise 3.9 generates variates from the density f_e .
   不会证 T.T

5. Exercise 5.3: Compute a Monte Carlo estimate $\hat{\theta}$ of

$$\theta = \int_0^{0.5} e^{-x} \mathrm{d}x$$

   by sampling from Uniform(0, 0.5), and estimate the variance of $\hat{\theta}$. Find another Monte Carlo estimator θ^* by sampling from the exponential distribution. Which of the variances (of $\hat{\theta}$ and θ^*) is smaller, and why?

```
m <- 10000
x <- runif(m, min=0, max=0.5)
theta.hat <- mean(exp(-x)) * 0.5
var <- 0.5^2/m*var(exp(-x))
print(theta.hat)
print(var)
```

```
[1] 0.3930249
[1] 3.216936e-07
```

The estimated variance of $\hat{\theta}$ is 3.22e-07.

```
m <- 10000
j <- 0
x <- numeric(m)
while (j < m) {
  u <- rexp(1)
  if (u<=0.5 & u>=0) {
    x[j] <- u
    j <- j + 1
  }
}
theta.hat <- mean(exp(-x)) * 0.5
var <- 0.5^2/m* var(exp(-x))
print(theta.hat)
print(var)
```

```
[1] 0.4015045
[1] 3.23621e-07
```

The estimated variance of θ^* is 3.24e-07. Thus, the variance of $\hat{\theta}$ is small. Because the variance of sample from exponential distribution between (0,0.5) is much larger than sample from uniform(0,0.5).

6. Exercise 5.5: Compute (empirically) the efficiency of the sample mean Monte Carlo method of estimation of the definite integral in Example 5.3 relative to the "hit or miss"method in Example 5.4.

```
m <- 10000
j <- 0         #iterations
x <- numeric(m)

while (j < m) {
  u <- rexp(1)
  j <- j + 1
  if (u<=0.5 & u>=0) {
    x[j] <- 1
  }else{
    x[j] <- 0
```

```
  }
}
fx.hat <- mean(x)
fx.var <- (fx.hat*(1-fx.hat))/m
print(fx.hat)
print(fx.var)
```

```
[1] 0.3949
[1] 2.38954e-05
```

The variance of the estimator using ”hit or miss” method is 2.39e-05, which is much larger than the variance computed in Exercise 5.3. Thus, the sample mean Monte Carlo method is more efficient.