Arif Ali
Math 611 Stochastic Simulation
Nov 17, 2016

Homework 11

Exercise 1

Gibbs Sampler from Example 7.2

From Example 7.2 in the book, we are given the conditional distribution

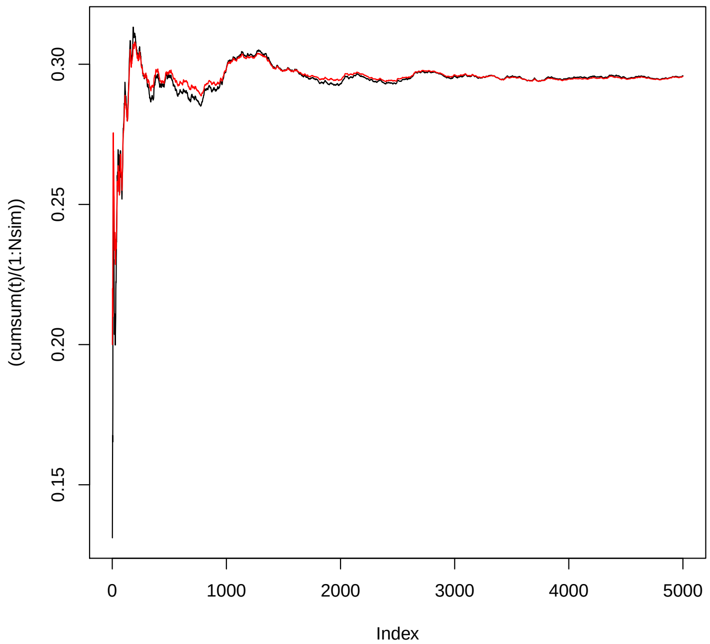
$$\theta|x \sim Beta(x+a, n-x+b) \tag{1}$$

By the properties of the beta distribution:

$$E[\theta|x] = \frac{x+a}{x+a+n-x+b} = \frac{x+a}{a+n+b} \tag{2}$$

```
## code from Example 7.2
## note: rb is the Rao-Blackwellization vectors I created to calculate
Nsim=5000
n=15
a=3
b=7
X=t=rb=array(0,dim=c(Nsim,1))
t[1]=rbeta(1,a,b)
X[1]=rbinom(1,n,t[1])
 for (i in 2:Nsim){
  #rb[i-1] = 1/(i-1)*sum((X+a))/(a+n+b)
  X[i]=rbinom(1,n,t[i-1])
  t[i]=rbeta(1,a+X[i],n-X[i]+b)
 }
  rb = cumsum((X+a)/(a+n+b))/(1:Nsim)

plot((cumsum(t)/(1:Nsim)),type="l")
lines(rb,col="red")
```



Exercise 2

Part A

please notes that:

$$f(y|p) = f(y|p,\theta) \tag{3}$$

By Bayes rule the following is true:

$$\begin{aligned}
f(y|p,\theta) \propto f(p,y,\theta) = f(\theta) * f(p|\theta) * f(y|p,\theta) = \\
\frac{1}{a} * e^{-\frac{\theta}{a}} * \theta * p^{\theta-1} * \left(\begin{array}{c} n \\ y \end{array} \right) * p^y * (1-p)^{n-y} \propto \\
p^{\theta-1} * p^y * (1-p)^{n-y} = \\
p^{\theta+y-1} * (1-p)^{(n-y+1)-1} \propto \\
\frac{1}{\beta(\theta+y, n-y+1)} * p^{\theta+y-1} * (1-p)^{(n-y+1)-1} \sim beta(\theta+y, n-y+1))
\end{aligned} \tag{4}$$

Part B

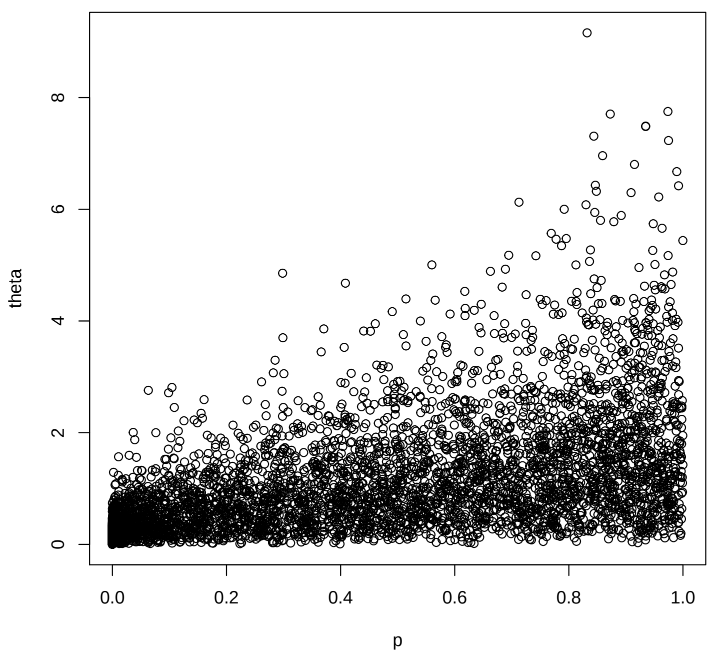
$$\begin{aligned}
f(\theta|y,p) \propto f(p,y,\theta) \propto e^{-\frac{a}{\theta}} * \theta * p^{\theta-1} \propto \\
\theta^{2-1} * e^{-\frac{\theta}{a}} * p^{\theta} = \\
\theta^{2-1} * e^{-\frac{\theta}{a}} * e^{log(p^{\theta})} = \\
\theta^{2-1} * e^{-\frac{\theta}{a}} * e^{(\theta)*log(p)} = \\
\theta^{2-1} * e^{-\frac{\theta}{a}+(\theta)*log(p)} = \\
\theta^{2-1} * e^{-(\frac{1}{a}-log(p))*\theta} = \\
\theta^{2-1} * e^{-\frac{\theta}{(\frac{1}{a}-log(p))^{-1}}} \propto X
\end{aligned} \tag{5}$$

$$X \sim Gamma(2, (\frac{1}{a} + *log(p))^{-1}) \quad (6)$$

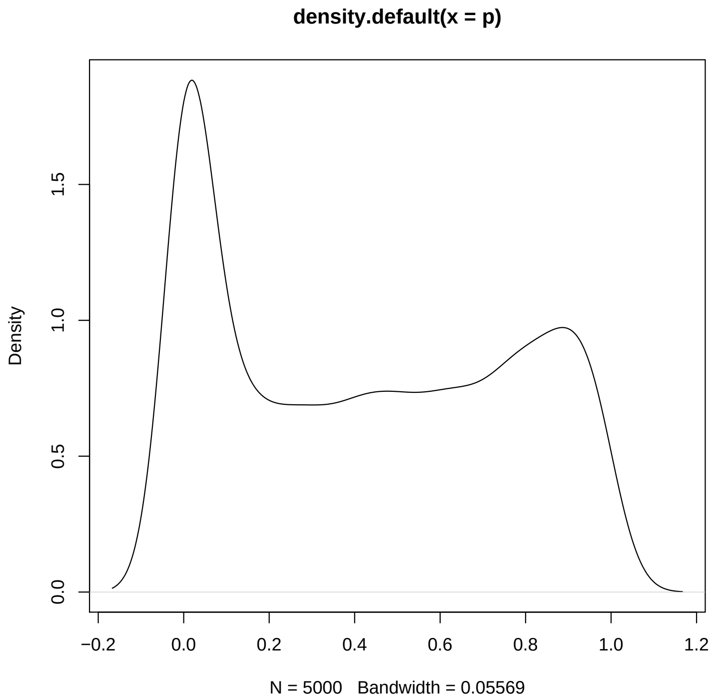
Part C

```
a = 1
p = 0.5
n = 10
theta = rgamma(1, a)
y = rbinom(1,n,p)
Nsim = 5e3
 for (i in 2:Nsim){
  theta[i] = rgamma(1, 2, scale = (1/a-log(p[i-1]))^-1)
  y[i] = rbinom(1, n, p[i-1])
  p[i] = rbeta(1, y[i]+theta[i],n-y[i]+1)

 }
plot(p, theta)
```




```
plot(density(p))
```



Exercise 3

Note that $gamma(1, \beta) = exponential(\beta)$

$$
\begin{aligned}
\int_0^y \beta * e^{-\beta * x} dx = 0.5 = \\
1 - e^{-\beta * y} = 0.5 \implies \\
-exp(-\beta * y) = -0.5 \implies \\
exp(-\beta * y) = 0.5 \implies \\
-\beta * y = ln(0.5) \implies \\
-\beta * y = -ln(2) \implies \\
y = \frac{ln(2)}{\beta}
\end{aligned}
\tag{7}
$$

Using bcanon we obtain the following:

```
median_expo = function(beta){log(2)/beta}
library(bootstrap)
bootstrap_exponential = function(x, beta, N = 1e4, alpha_int = c(0.05, 0.95)){
  aa = bcanon(x, N, theta = median, alpha = alpha_int)
}
x = rexp(1e4, rate = 2)
bs_medians = bootstrap_exponential(x, 2)
bs_medians$confpoints[1,2]<median_expo(2)

## bca point
##      TRUE
```

```
bs_medians$confpoints[2,2]>median_expo(2)

## bca point
##      TRUE
```

The median of $gamma(1, \beta)$ is 0.3465736, which is within the 90% BCa confidence Interval of (0.3399802,0.3567573). Instead of using a function from the the R package, we adapt the R function build in class.

```
boot.BCa <-
  function(x, th0, th, Stat, conf = .90) {
    # BCa confidence interval
    # th0: observed statistic
    # th: vector of bootstrap distribution
    # stat is the function to compute the statistic

    x <- as.matrix(x)
    n <- nrow(x) #observations in rows
    N <- 1:n
    alpha <- (1 + c(-conf, conf))/2
    zalpha <- qnorm(alpha)

    # the bias correction factor
    z0 <- qnorm(sum(th < th0) / length(th))

    # the acceleration factor (jackknife est.)
    th.jack <- numeric(n)
    for (i in 1:n) {
      th.jack[i] <- Stat(x[-i, ]) #unlike the class example, we are calculating the median of a functio
    }
    L <- mean(th.jack) - th.jack
    a <- sum(L^3)/(6 * sum(L^2)^1.5)

    # BCa conf. limits
    adj.alpha <- pnorm(z0 + (z0+zalpha)/(1-a*(z0+zalpha)))
    limits <- quantile(th, adj.alpha, type=6)
    list("est"=th0, "BCa"=limits)
  }
x = rexp(1e4, rate = 2)
boot.median = c()
for(i in 1:1e4){
  boot.median[i] = median(sample(x, size = 1e4, replace = T))
  }

bca_int = boot.BCa(x = rexp(1e4, rate = 2), th0 = median(x), th = boot.median ,Stat = median)$BCa
bca_int[1]<median_expo(2)

## 4.736841%
##      TRUE

bca_int[2]>median_expo(2)

## 94.72531%
##      TRUE
```

The median of $gamma(1, \beta)$ is 0.3465736, which is within the 90% BCa confidence Interval of (0.3429105,0.3589501).

If instead of BCa confidence intervals we did Percentile Confidence Intervals, that 90% Confidence interval would be (0.3431754,0.3589769).