STA410 Assignment #2

student number: 1003942326

Yulin WANG

17/10/2019

Question 1

(a)

Since $X = V/U$, so $V = XU$, then, we can get the Jacobian:

$$J = \begin{vmatrix} \frac{\partial u}{\partial u} & \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial u} & \frac{\partial v}{\partial x} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ x & u \end{vmatrix} = u$$

Thus, the joint density of $(U, X)$is

$$g(u, x) = f(u, v) \cdot |u| = \frac{u}{|C_h|} \quad \text{for} \;\; 0 \leq u \leq \sqrt{h(x)}, \;\; \text{since} \;\; h(x) = h(v/u)$$

Then we have the marginal density of X is:

$$f_X(x) = \int_0^{\sqrt{h(x)}} \frac{u}{|C_h|} du = \frac{h(x)}{2|C_h|}$$

Thus, the density of X is $\gamma h(x)$ for $\gamma = \frac{1}{2|C_h|} > 0$

(b)

Since $C_h = \{(u, v) : 0 \leq u \leq \sqrt{h(x)}\}$, so if $(u, v) \in C_h$ then $0 \leq u \leq max_x \sqrt{h(x)} = u_+$
And since u is positive, so we have:

$$\frac{1}{u}\sqrt{h(x)} \geq 1$$

Then we have two cases:
when $v > 0$, we have:

$$v \leq \frac{v}{u}\sqrt{h(x)} \leq max_x x \sqrt{h(x)} = v_+$$

when $v < 0$, we have:

$$v \geq \frac{v}{u}\sqrt{h(x)} \geq min_x x \sqrt{h(x)} = v_-$$

Thus, if $(u, v) \in C_h$, then $0 \leq u \leq u_+$ and $v_- \leq v \leq v_+$, that is $(u, v) \in D_h$

(c)

Since $h(x) = exp(-x^2/2)$, so $0 \leq u \leq \sqrt{h(x)} = exp(-x^2/4) = exp(-(v/u)^2/4)$, since $x = v/u$
Then, we have the accept condition: $u \leq exp(-(v/u)^2/4)$, so we can implement the rejection sampling method and calculate proposal acceptance rate as follows:

```
generator <- function(n) {
             x <- NULL #initial vector containing random variates
             rejections <- 0 #set the initial number of rejections
             bound <- sqrt(2/exp(1)) #the absolute value of the vbound
             for (i in 1:n) {
                     reject <- T
                     while(reject){
                             u <- runif(1,0,1)
                             v <- runif(1,-bound,bound)
                             if (u<=exp(-(v/u)^2/4)){ #accept condition
                                     x <- c(x, v/u) #accept
                                     reject <- F
                             }
                             else rejections <- rejections + 1
                     }
             }
             # x now contains n rvs
             # calculate proposal acceptance rate
             accept.rate <- n/(n+rejections)
             r <- list(x=x,accept.rate=accept.rate)
             r
}
r <- generator(10000) # generate 10000 rvs using the function
r$accept.rate # fraction of accepted proposals
```

```
## [1] 0.7290755
```

Question 2

(a)

Suppose that if $\{y_i\}$ are exactly linear, i.e. $y_i = a \times i + b$ for $i = 1, 2, ..., n$
Then for $\theta_i = y_i$, we have:

$$\begin{aligned}\theta_{i+1} - 2\theta_i + \theta_{i-1} &= a(i+1) + b - 2(ai + b) + a(i-1) + b \\ &= ai + a + b - 2ai - 2b + ai - a + b \\ &= 0\end{aligned}$$

Then, we have our objective function as follows:

$$0 + \lambda \sum_{i=2}^{n-1} (\theta_{i+1} - 2\theta_i + \theta_{i-1})^2 = 0$$

Since the objective function is exactly 0, so $\hat{\theta}_i = y_i$ for all i.

(c)

Firstly, initialize $\hat{\theta}$ and we build a linear equation $y_i = \theta_i + \epsilon_i$, where $i = 1, 2, ..., n$ We choose a randomized modification of the Gauss-Seidel algorithm. Randomly sample a subset w of size p from the integers $1, 2, ..., n$. Define X_w to be the submatrix of X with column indices w and $X_{\bar{w}}$ to be the submatrix of X with column indices in the complement of w; define θw and $\theta_{\bar{w}}$ analogously so that $X\theta = X_w\theta_w + X_{\bar{w}}\theta_{\bar{w}}$.
Let A be the varaibles of X_w and B be the varaibles of $X_{\bar{w}}$. Then we can set a function as follows:

$$g(x_i) = g(A_i, B_i) = ||y - A_i - B_i||$$

After we ran k iterations, B_k became nearly fixed, then we can only minimize over the selected varaibles(i.e.A_k), then we have:

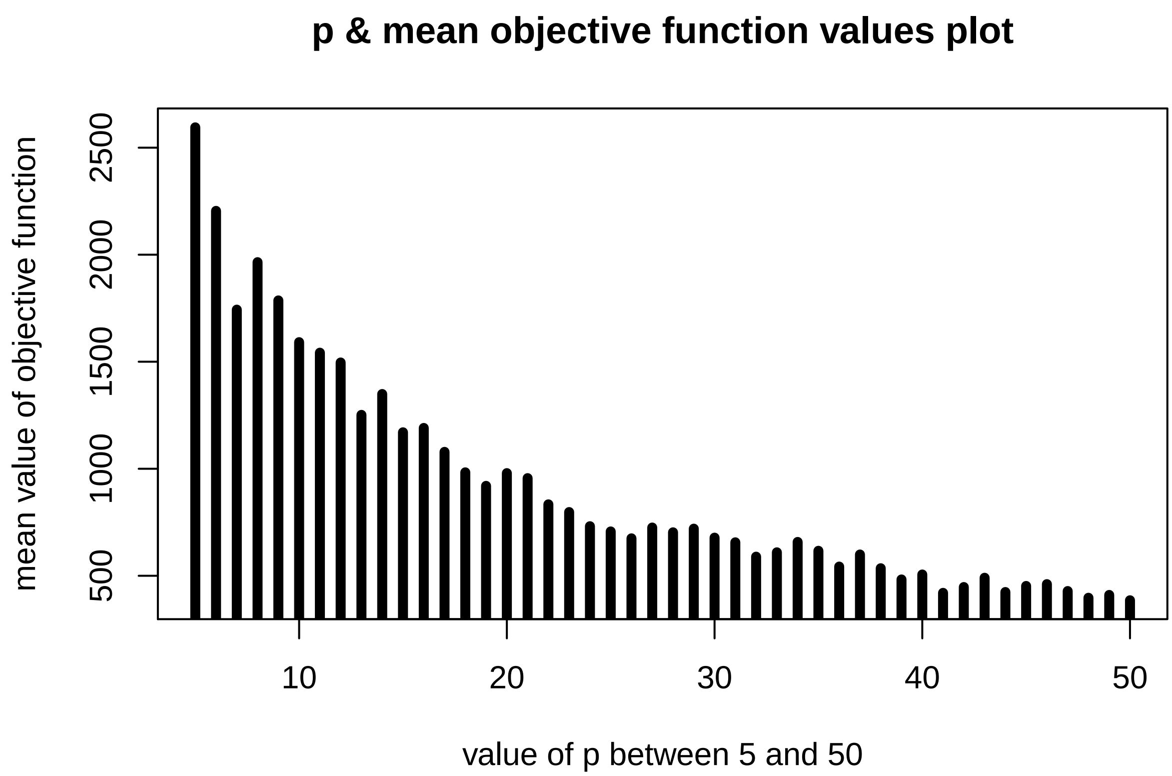
$$\begin{aligned} g(x_{k+1}) = g(A_{k+1}, B_k) &= ||y - A_{k+1} - B_k|| \\ &\leq ||y - A_k - B_k|| \\ &= g(A_k, B_k) \\ &= g(x_k) \end{aligned}$$

Thus, the value of objective function after (k+1)th iteration is less or equal to that after kth iteration. Therefore, the objective function is non-increasing from one iteration to the next.

(d)

```r
yield <- scan("yield.txt") #load the 'yield.txt'
HP <- function(x,lambda,p,niter) {
        n <- length(x)
        a <- c(1,-2,1)
        aa <- c(a,rep(0,n-2))
        aaa <- c(rep(aa,n-3),a)
        mat <- matrix(aaa,ncol=n,byrow=T)
        mat <- rbind(diag(rep(1,n)),sqrt(lambda)*mat)
        xhat <- x
        x <- c(x,rep(0,n-2))
        sumofsquares <- NULL
        for (i in 1:niter) {
           w <- sort(sample(c(1:n),size=p))
           xx <- mat[,w]
           y <- x - mat[,-w]%*%xhat[-w]
           r <- lsfit(xx,y,intercept=F)
           xhat[w] <- r$coef
           sumofsquares <- c(sumofsquares,sum(r$residuals^2))
           }
        r <- list(xhat=xhat,ss=sumofsquares)
        r
}

values <- NULL
for (i in 5:50){ #various values of p between 5 and 50
  r <- HP(yield,lambda=2000,p=i,niter=1000)
  values <- c(values, mean(r$ss)) #save the mean of r$ss for each p
}
result <- list(p=c(5:50),obj=values)
#generate the plot of p and value of objective function
plot(result$p,result$obj,type="h",lwd=5,main="p & mean objective function values plot", xlab="value of p
```



From the plot above, we can conclude that as the p increases between 5 and 50, the value of objective function tends to decrease.