

STA2102 Homework 3

Kevin Zhang 1002225264

11/21/2020

Question 1

a)

$$E((V^T AV)^2) = \sum_{i=1}^n a_{ii}^2 E(V_i^4) = \sum_{i=1}^n \sum_{j=1}^n a_{ii} a_{jj} E(V_i^2) E(V_j^2) = \sum_{i=1}^n \sum_{j=1}^n a_{ii} a_{jj}$$

Therefore we need to minimize $E(V_i^4)$. Note that $E(V_i^4) = \text{var}(V_i^2) + (E(V_i^2))^2 = \text{var}(V_i^2) + 1$ is minimized if $\text{var}(V_i^2) = 0$ or equivalently V_i^2 is degenerate at some number k . Since $E(V_i^2) = 1$, we have $V_i^2 = 1$ a.s. which implies that $V_i \in \{-1, 1\}$ and $P(-1) = P(1) = 0.5$.

b)

By simple matrix-vector multiplication:

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} V \\ 0 \end{pmatrix} = \begin{pmatrix} H_{11}V \\ H_{21}V \end{pmatrix}$$

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} H_{11}^{k-1}V \\ 0 \end{pmatrix} = \begin{pmatrix} H_{11}H_{11}^{k-1}V \\ H_{21}H_{11}^{k-1}V \end{pmatrix} = \begin{pmatrix} H_{11}^kV \\ H_{21}H_{11}^{k-1}V \end{pmatrix}$$

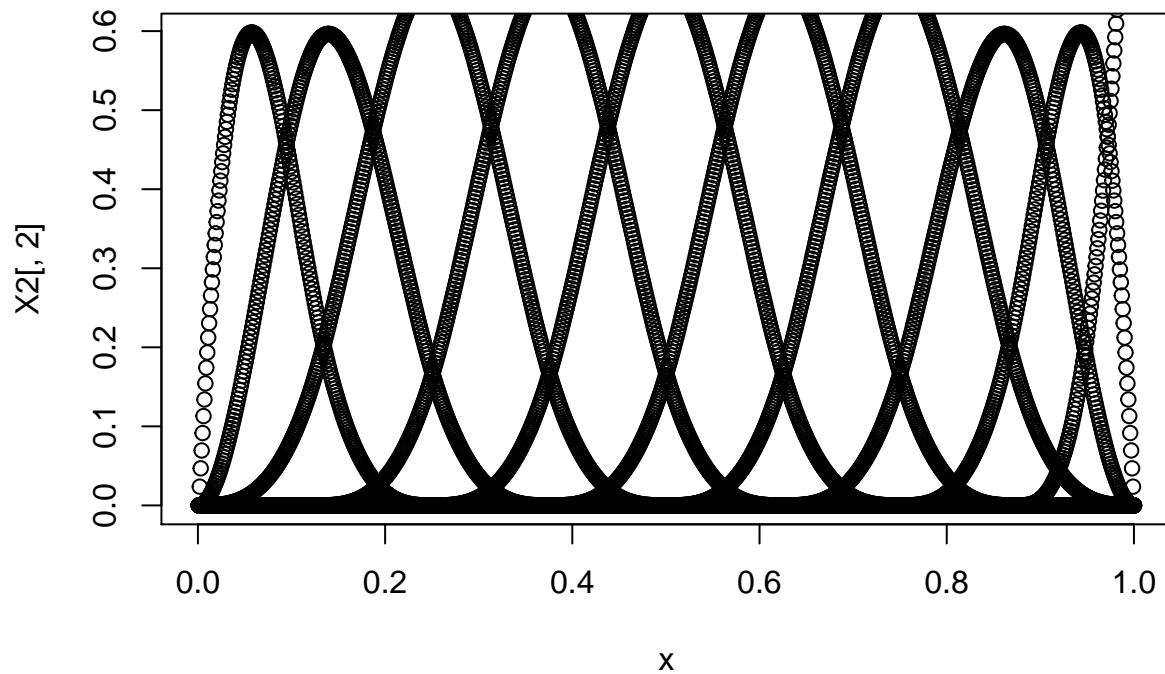
c)

```
leverage <- function(x,w,r=10,m=100){
  set.seed(12345) # modification
  qrx <- qr(x)
  n <- nrow(x)
  lev <- NULL
  for(i in 1:m){
    v <- ifelse(runif(n)>0.5,1,-1)
    v[-w] <- 0
    v0 <- qr.fitted(qrx,v)
    f <- v0
    for(j in 2:r){
      v0[-w] <- 0
      v0 <- qr.fitted(qrx,v0)
      f <- f + v0/j
    }
    lev <- c(lev,sum(v*f))
  }
  std.err <- exp(-mean(lev))*sd(lev)/sqrt(m)
```

```

lev <- 1 - exp(-mean(lev))
r <- list(lev=lev,std.err=std.err)
r
}
x <- c(1:1000)/1000
X1 <- 1
for (k in 1:5) X1 <- cbind(X1,cos(2*k*pi*x),sin(2*k*pi*x))
library(splines) # loads the library of functions to compute B-splines
X2 <- cbind(1,bs(x,df=10))
plot(x,X2[,2])
for (i in 3:11) points(x,X2[,i])

```



```

lev1 <- NULL
err1 <- NULL
lev2 <- NULL
err2 <- NULL
for(k in 1:20){
  w <- (1:1000)[x>(k-1)/20 & x<=k/20]
  r1 <- leverage(X1, w)
  r2 <- leverage(X2, w)
  lev1 <- c(lev1, r1$lev)
  err1 <- c(err1, r1$std.err)
  lev2 <- c(lev2, r2$lev)
  err2 <- c(err2, r2$std.err)
}
data.frame(k=1:20, lev1=lev1, err1=err1, lev2=lev2, err2=err2)

```

| ## | k | lev1 | err1 | lev2 | err2 |
|-------|----|-----------|------------|-----------|------------|
| ## 1 | 1 | 0.4557024 | 0.03512688 | 0.9053758 | 0.02397850 |
| ## 2 | 2 | 0.6255833 | 0.04694883 | 0.7383969 | 0.04520032 |
| ## 3 | 3 | 0.5466491 | 0.05074565 | 0.5297106 | 0.04988424 |
| ## 4 | 4 | 0.6075882 | 0.04494969 | 0.5343172 | 0.04370109 |
| ## 5 | 5 | 0.4886954 | 0.04142627 | 0.3843179 | 0.03714324 |
| ## 6 | 6 | 0.5386923 | 0.04842888 | 0.4631653 | 0.04637430 |
| ## 7 | 7 | 0.5661153 | 0.04418983 | 0.3870280 | 0.03653007 |
| ## 8 | 8 | 0.5504269 | 0.04154915 | 0.4886912 | 0.04113533 |
| ## 9 | 9 | 0.5311669 | 0.04251794 | 0.3542561 | 0.03366749 |
| ## 10 | 10 | 0.4996194 | 0.04430449 | 0.3906994 | 0.03982617 |
| ## 11 | 11 | 0.5472358 | 0.04278452 | 0.4380903 | 0.03997192 |
| ## 12 | 12 | 0.5968979 | 0.04247119 | 0.4113550 | 0.03634399 |
| ## 13 | 13 | 0.5333564 | 0.04110098 | 0.4698133 | 0.04037578 |
| ## 14 | 14 | 0.5214683 | 0.04524945 | 0.3461899 | 0.03505435 |
| ## 15 | 15 | 0.5797373 | 0.04701736 | 0.5003101 | 0.04637395 |
| ## 16 | 16 | 0.5031542 | 0.04407215 | 0.3992349 | 0.03893502 |
| ## 17 | 17 | 0.5023650 | 0.04708470 | 0.4392638 | 0.04615961 |
| ## 18 | 18 | 0.5070157 | 0.05292635 | 0.4937906 | 0.05141883 |
| ## 19 | 19 | 0.4929380 | 0.04013808 | 0.5942051 | 0.04291982 |
| ## 20 | 20 | 0.4767470 | 0.04558771 | 0.9405694 | 0.02090240 |

The estimates for the two models are not exactly close to each other. The standard error for model 2 seems to be smaller.

Question 2

a)

Let x_m denote the median then

$$\begin{aligned} \int_{-\infty}^{x_m} \frac{1}{\pi\sigma} \left(\frac{\sigma^2}{(x-\theta)^2 + \sigma^2} \right) dx &= \int_{-\infty}^{x_m-\theta} \frac{\sigma}{\pi} \left(\frac{1}{s^2 + \sigma^2} \right) ds = \frac{1}{\pi} \arctan \frac{s}{\sigma} \Big|_{-\infty}^{x_m-\theta} = 0.5 \\ \implies \frac{1}{\pi} \arctan \frac{x_m - \theta}{\sigma} + 0.5 &= 0.5 \implies x_m = \theta \end{aligned}$$

Similarly, let x_q denote the third quartile, then

$$\frac{1}{\pi} \arctan \frac{s}{\sigma} \Big|_0^{x_r-\theta} = \frac{1}{4} \implies \arctan \frac{x_r - \theta}{\sigma} = \frac{\pi}{4} \implies x_r = \theta + \sigma$$

By symmetry around θ , the IQR is 2σ .

b)

$$\begin{aligned} l(x, \theta, \sigma) &= n \log \sigma - n \log \pi - \sum_{i=1}^n \log[(x_i - \theta)^2 + \sigma^2] \\ \frac{\partial l}{\partial \theta} &= \sum_{i=1}^n \frac{2(x_i - \theta)}{(x_i - \theta)^2 + \sigma^2} \\ \frac{\partial l}{\partial \sigma} &= \frac{n}{\sigma} - \sum_{i=1}^n \frac{2\sigma}{(x_i - \theta)^2 + \sigma^2} \end{aligned}$$

$$\frac{\partial^2 l}{\partial \theta^2} = \sum_{i=1}^n \frac{2(x_i - \theta)^2 - 2\sigma^2}{((x_i - \theta)^2 + \sigma^2)^2}$$

$$\frac{\partial^2 l}{\partial \sigma^2} = \frac{-n}{\sigma^2} - \sum_{i=1}^n \frac{2((x_i - \theta)^2 + \sigma^2) - 4\sigma^2}{((x_i - \theta)^2 + \sigma^2)^2}$$

$$\frac{\partial^2 l}{\partial \theta \partial \sigma} = \sum_{i=1}^n \frac{4\sigma(x_i - \theta)}{-((x_i - \theta)^2 + \sigma^2)^2}$$

```
newton_raph <- function(x, n_iter=1000){
  theta <- median(x)
  sigma <- IQR(x) / 2
  res <- c(theta, sigma)
  h1 <- function(x, theta, sigma){
    temp1 <- 0
    for(x_temp in x){
      temp1 <- temp1 + 2*(x_temp-theta)/((x_temp-theta)^2+sigma^2)
    }
    temp2 <- length(x)/sigma
    for(x_temp in x){
      temp2 <- temp2 - 2*sigma/((x_temp-theta)^2+sigma^2)
    }
    return(c(temp1, temp2))
  }
  h2 <- function(x, theta, sigma){
    temp11 <- 0
    for(x_temp in x){
      temp11 <- temp11 + (2*(x_temp-theta)^2-2*sigma^2)/((x_temp-theta)^2+sigma^2)^2
    }
    temp22 <- -length(x)/sigma^2
    for(x_temp in x){
      temp22 <- temp22 - (2*(x_temp-theta)^2-2*sigma^2)/((x_temp-theta)^2+sigma^2)^2
    }
    temp12 <- 0
    for(x_temp in x){
      temp12 <- temp12 - 4*sigma*(x_temp-theta)/((x_temp-theta)^2+sigma^2)^2
    }
    return(matrix(c(temp11, temp12, temp12, temp22), nrow=2))
  }
  for(i in 1:n_iter){
    h1_temp <- h1(x, res[1], res[2])
    h2_temp <- h2(x, res[1], res[2])
    res <- res - solve(h2_temp) %*% h1_temp
  }
  return(list(res=res, err=-solve((h2(x, res[1], res[2])))))
}
x <- rcauchy(n=1000, location=0, scale=1)
newton_raph(x)
```

```
## $res
##           [,1]
## [1,] -0.03210368
## [2,]  1.00834939
##
```

```
## $err
##           [,1]           [,2]
## [1,]  2.104992e-03 -1.502122e-06
## [2,] -1.502122e-06  1.966776e-03
```