STAT 501 Homework 3

Multinomial

February 21, 2018

1. (a) Let $\boldsymbol{X} = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}^T$ and $\boldsymbol{Y}_1 = \begin{bmatrix} X_1 & X_2 & \dots & X_{n-1} \end{bmatrix}^T$, $Y_2 = \bar{X}$. Since $X_1, \dots, X_n$ are iid $N(\mu, \sigma^2)$, then $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = \mu \mathbf{1}_{n\times 1}$ and $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_n$. We know

$$\begin{bmatrix} \boldsymbol{Y}_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} & \frac{1}{n} \end{bmatrix} \boldsymbol{X}$$

Hence $\begin{bmatrix} \boldsymbol{Y}_1 & Y_1 \end{bmatrix}^T$ also followrs multivariate normal distribution. And we know

$$\mathrm{E}\left(\begin{bmatrix} \boldsymbol{Y}_1 \\ Y_2 \end{bmatrix}\right) = \begin{bmatrix} \mathrm{E}\left(\begin{bmatrix} X_1 & \dots & X_{n-1} \end{bmatrix}^T\right) & \mathrm{E}(\bar{X}) \end{bmatrix}^T = \begin{bmatrix} \mu \mathbf{1}_{(n-1)\times 1} \\ \mu \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$$

$$\mathrm{Var}\left(\begin{bmatrix} \boldsymbol{Y}_1 \\ Y_2 \end{bmatrix}\right) = \begin{bmatrix} \mathrm{Var}(\boldsymbol{Y}_1) & \mathrm{Cov}(\boldsymbol{Y}_1, Y_2) \\ \mathrm{Cov}(Y_2, \boldsymbol{Y}_1^T) & \mathrm{Var}(Y_2) \end{bmatrix} = \begin{bmatrix} \sigma^2 \boldsymbol{I}_{n-1} & \frac{\sigma^2}{n} \mathbf{1}_{(n-1)\times 1} \\ \frac{\sigma^2}{n} \mathbf{1}_{1\times(n-1)} & \frac{\sigma^2}{n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

Thus we have

$$\begin{bmatrix} \boldsymbol{Y}_1 \\ Y_2 \end{bmatrix} \sim N_n\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}\right).$$

Thus we know

$$\boldsymbol{Y}_1 | Y_2 \sim N_{n-1}(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(Y_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})$$

And

$$\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(Y_2 - \boldsymbol{\mu}_2) = \mu \mathbf{1}_{(n-1)\times 1} + \frac{\sigma^2}{n} \mathbf{1}_{(n-1)\times 1} \cdot \frac{n}{\sigma^2}(\bar{X} - \mu) = \bar{X}\mathbf{1}_{(n-1)\times 1}$$

$$\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \sigma^2 \boldsymbol{I}_{n-1} - \frac{\sigma^2}{n}\mathbf{1}_{(n-1)\times 1}\frac{n}{\sigma^2}\frac{\sigma^2}{n}\mathbf{1}_{1\times(n-1)} = \sigma^2 \boldsymbol{I}_{n-1} - \frac{\sigma^2}{n}\mathbf{1}_{(n-1)\times(n-1)}$$

Hence the distribution of $X_1, \dots, X_{n-1}$ given $\bar{X}$ is

$$\begin{bmatrix} X_1 & \cdots & X_{n-1} \end{bmatrix}^T \mid \bar{X} \sim N_{n-1}\left(\bar{X}\mathbf{1}_{(n-1)\times 1}, \sigma^2 \boldsymbol{I}_{n-1} - \frac{\sigma^2}{n}\mathbf{1}_{(n-1)\times(n-1)}\right)$$

(b) Function to generate $\boldsymbol{X}$ given $\bar{X}$.

```r
# 1(b) generate random vector X given X-bar.
library(MASS)
X.gen <- function(xbar, n, sigma){
  mu <- rep(xbar, n-1)
  Sigma <- diag(rep(sigma^2, n-1)) - matrix(rep((sigma^2/n), (n-1)*(n-1)), nrow = n-1)
  Y1 <- mvrnorm(mu = mu, Sigma = Sigma)
  X <- c(Y1, n*xbar - sum(Y1))
  return(X)
}
```

(c) The R code to obtain a higher resolution image is shown below.

```r
# 1(c)
## i. & iii.
## read the image
library(rtiff)
owlet <- readTiff(fn = "Indian_spotted_owlet.tiff")
plot(owlet)



## function super resolution one pixel to 4x4, and truncate at 0 and 1
supres <- function(x){
  y <- X.gen(x, 16, 0.4)
  y1 <- ifelse(y < 0, 0, y)
  y2 <- ifelse(y1 > 1, 1, y1)
  return(matrix(y2, nrow = 4))
}

## high resolution matrix for red channel
size.low <- owlet@size
owletsupr <- apply(X = owlet@red, MARGIN = c(1,2), FUN = supres)
owletsupr1 <- array(dim = c(4,4)*size.low)
for(i in 1:size.low[1]){
  for(j in 1:size.low[2]){
    for(k in 1:4){
      for(l in 1:4){
        owletsupr1[4*(i - 1) + k, 4*(j-1) + l] <- owletsupr[4*(k-1) + l, i, j]
      }
    }
  }
}

## ii.
## high resolution matrix for green and blue

### green channel
owletsupg <- apply(X = owlet@green, MARGIN = c(1,2), FUN = supres)
```

```r
owletsupg1 <- array(dim = c(4,4)*size.low)
for(i in 1:size.low[1]){
  for(j in 1:size.low[2]){
    for(k in 1:4){
      for(l in 1:4){
        owletsupg1[4*(i - 1) + k, 4*(j-1) + l] <- owletsupg[4*(k-1) + l, i, j]
      }
    }
  }
}

### blue channel
owletsupb <- apply(X = owlet@blue, MARGIN = c(1,2), FUN = supres)
owletsupb1 <- array(dim = c(4,4)*size.low)
for(i in 1:size.low[1]){
  for(j in 1:size.low[2]){
    for(k in 1:4){
      for(l in 1:4){
        owletsupb1[4*(i - 1) + k, 4*(j-1) + l] <- owletsupb[4*(k-1) + l, i, j]
      }
    }
  }
}



owlet.high <- array(dim  = c(owlet@size*c(4,4), 3))
owlet.high[,,1] <- owletsupr1
owlet.high[,,2] <- owletsupg1
owlet.high[,,3] <- owletsupb1

## iv.
### create pixmap object
owlet.highres <- pixmapRGB(owlet.high)

## v.
### display the high resolution image, save
par(mar=c(0,0,0,0))
plot(owlet.highres)
writeTiff(owlet.highres, fn = "./owlet_high.tiff")
```

And the higher resolution image is as below in Figure 1. We can see there is more noise compared to the original one in Figure 2.

Figure 1: Higher resolution image

Figure 2: Original image

2. (a) When $\theta = 0$, $Y_i = g(X_i, 0) = X_i \sim N(\mu_0, \sigma_0^2)$. So the cdf of $\boldsymbol{X}_i$ is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{(x-\mu_0)^2}{2\sigma_0^2}\right)$$

when $\theta > 0$, then

$$f(x) = f_Y(g(x,\theta)) \left|\frac{\partial g}{\partial x}\right| = \frac{1}{\sqrt{2\pi}\sigma_\theta} \exp\left(-\frac{(\text{arcsinh}(\theta x)/\theta - \mu_\theta)^2}{2\sigma_\theta^2}\right) \frac{1}{\sqrt{1+\theta^2 x^2}}$$

So in summary,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_\theta} \exp\left(-\frac{(g(x,\theta) - \mu_\theta)^2}{2\sigma_\theta^2}\right) \frac{1}{\sqrt{1+\theta^2 x^2}},\ \theta \geq 0$$

Thus the log likelihood is

$$\ell(\theta, \mu_\theta, \sigma_\theta^2) = \sum_{i=1}^n \log f(x_i) = -\frac{n}{2}\log 2\pi - n\log\sigma_\theta - \frac{1}{2\sigma_\theta^2}\sum_{i=1}^n (g(x_i,\theta) - \mu_\theta)^2 - \frac{1}{2}\sum_{i=1}^n \log(1+\theta^2 x_i^2),\ \theta \geq 0$$

And we also know for a fixed θ,

$$\hat{\mu}_\theta = \frac{\sum_{i=1}^n g(x_i,\theta)}{n}$$
$$\hat{\sigma}_\theta^2 = \frac{\sum_{i=1}^n (g(x_i,\theta) - \hat{\mu}_\theta)^2}{n}$$

Then the maximized log likelihood for a fixed θ is

$$\begin{aligned}\hat{\ell}(\theta) &= -\frac{n}{2}\log(2\pi\hat{\sigma}_\theta^2) - \frac{1}{2\hat{\sigma}_\theta^2}\sum_{i=1}^n (g(x_i,\theta) - \hat{\mu}_\theta)^2 - \frac{1}{2}\sum_{i=1}^n \log(1+\theta^2 x_i^2) \\ &= -\frac{n}{2}\log(2\pi\hat{\sigma}_\theta^2) - \frac{n}{2} - \frac{1}{2}\sum_{i=1}^n \log(1+\theta^2 x_i^2),\ \theta \geq 0\end{aligned}$$

(b) i. We first look at the summary and histogram of `danish`.

```r
# 2.
## 2 (b) i.
library(SMPracticals)
summary(danish)
hist(danish)
```

Then we have the summary

```
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
  0.3134    1.1572    1.6339    3.0627    2.6455  263.2504 
```

and the histogram as in Figure 3. We can see from the histogram and the summary a lot

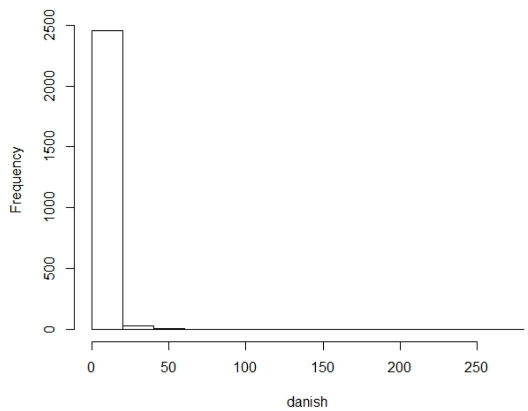


Figure 3: Histogram of `danish`

of mass is close to 0 (and less than 3).
Then we removed the potential outliers and make the histogram again.

```
## remove the potential outlier
danish1 <- danish[!danish %in% boxplot.stats(danish)$out]
hist(danish1)
```

The histogram is shown in Figure 4.

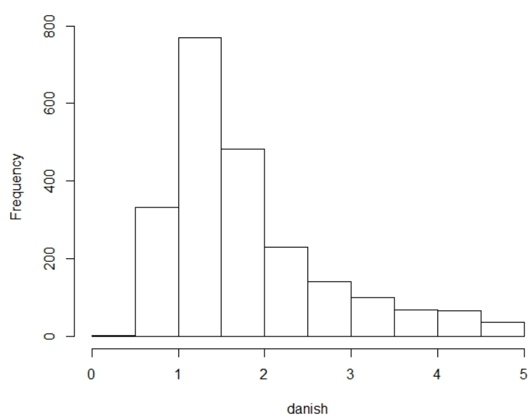


Figure 4: Histogram of `danish` with potential outliers removed

Here we can see the distribution is right skewed and the center is around 1.5. A lot of mass is between 1 and 2.

ii. By varying θ in $[0, 4]$ with step size 0.1, we can obtain the curve of maximized log likelihood against θ. The R codes for the curve are shown below.

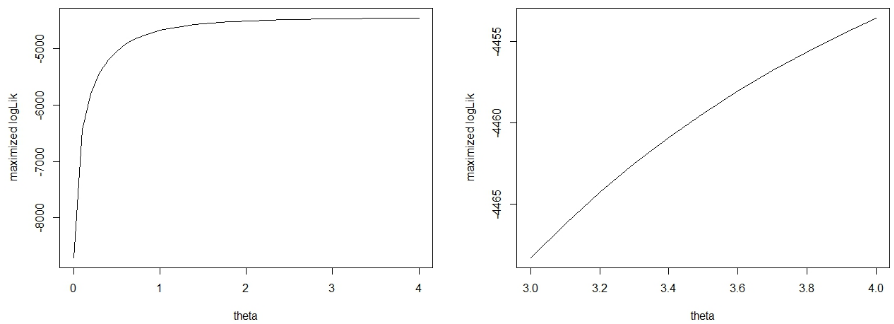
```r
## 2 (b) ii.
### define g function
g <- function(x, theta){
  if(theta == 0)
    return(x)
  else
    return(asinh(theta*x)/theta)
}

### define the maximized log likelihood
ell <- function(x, theta){
  n <- length(x)
  y <- g(x, theta)
  muhat <- sum(y)/n
  sigmahat2 <- sum((y - muhat)^2)/n
  return(-(n/2)*log(2*pi* sigmahat2) - (n/2) - sum(log(1 + theta^2 * x^2))/2)
}

### find maxmized log likelihood for each theta and plot the curve
thetas <- seq(0, 4, 0.1)
ells <- sapply(thetas, ell, x = danish)
plot(x = thetas, y = ells, 'n',xlab = "theta", ylab = "maximized logLik")
lines(x = thetas, y = ells)

### plot between theta from 3 to 4
thetas1 <- thetas[thetas >= 3]
ells1 <- ells[thetas >= 3]
plot(x = thetas1, y = ells1, 'n',xlab = "theta", ylab = "maximized logLik")
lines(x = thetas1, y = ells1)
```

In Figure 5(a), we can see the curve is increasing and becoming flat after 3. Then we plot the curve for $\theta \in [3, 4]$ in Figure 5(b) to have a better idea of the trend. Hence the maximized log likelihood for $\theta \in [0, 4]$ attains the maximum when $\theta = 4$.



(a) $\hat{\ell}(\theta)$ for $\theta \in [0, 4]$ (b) $\hat{\ell}(\theta)$ for $\theta \in [3, 4]$

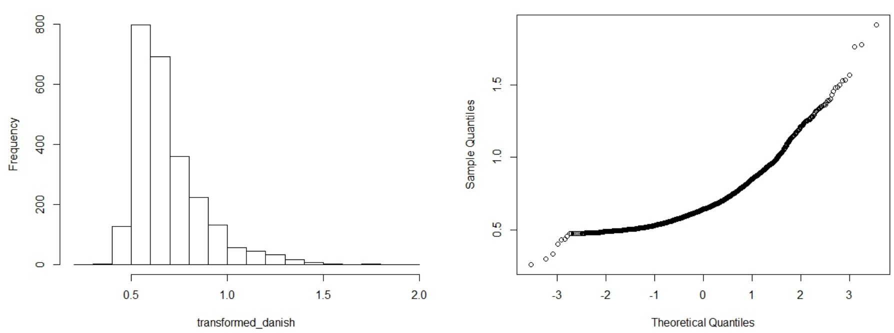
Figure 5: Maximized log likelihood $\hat{\ell}(\theta)$ against θ

Thus we use $g(x, 4)$ to transform `danish`. And we plot the histogram in Figure 6(a) and Q-Q plot of transformed data in Figure 6(b).

```
### transform the data with theta = 4
transformed_danish <- g(danish, 4)

hist(transformed_danish, main = "")

qqnorm(transformed_danish, main = "")
```



(a) Histogram of transformed `danish` (b) Q-Q plot of transform `danish`

Figure 6: Graphical display of transformed `danish`

We can see the histogram is not quite symmetric and the Q-Q plot is apparently a curve, not a straight line. So the transformed data is not likely to be normally distributed.

We then did a Shapiro-Wilk's test, and the p-value is small, thus the transformed data are not normal.

```
### shapiro-wilk's test
shapiro.test(transformed_danish)
```

The result is

```
        Shapiro-Wilk normality test

data:  transformed_danish
W = 0.85989, p-value < 2.2e-16
```