Assignment 2 - Incomplete Data Analysis

Zhe WANG

Note:The R code for Questions are available via the link (https://github.com/ZheWANG331/IDA-assign2.git) to the repository of Github.

2.

(a) The p.d.f of a non-censored observation is $\phi(x;\mu,\sigma^2)$.

For this left censored observations, we know that $Y < C$ and so its contribution to the likelihood is $\Pr(Y < C;\mu,\sigma^2) = \Phi(C;\mu,\sigma^2) = \phi(x < C;\mu,\sigma^2)$.

Since all ovservations are assumed independent, r_i represents the missing indicators, we can therefore write the likelihood as $\prod_{i=1}^{n}(\phi(x_i;\theta))^{r_i}(\Phi(C;\theta))^{1-r_i}$

Thus the corresponding log likelihood function is:

$$\log L(\mu,\sigma^2;x,r) = \sum_{1}^{n}(r_1 \log\phi(x_i;\mu,\sigma^2) + (1-r_i)\log\Phi(x_i;\mu,\sigma^2))$$

(b) The maximum likelihood estimate of μ is 5.5328.

```r
load("dataex2.Rdata")
require(maxLik)
```

```
## Warning: package 'maxLik' was built under R version 3.6.2
```

```r
#define the lok likelihood function
log_like <- function(param, data){
  x <- data[,1]
  y <- data[,2]
  sum((y*dnorm(x,param,1.5,log = TRUE)) + (1-y)*(pnorm(x,param,1.5,log = TRUE)))
}
#use maxlik
mle <- maxLik(logLik = log_like, data = dataex2, start = 1)
summary(mle)
```

```
## --------------------------------------------
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 3 iterations
## Return code 2: successive function values within tolerance limit
## Log-Likelihood: -336.3821
## 1  free parameters
## Estimates:
##      Estimate Std. error t value Pr(> t)
## [1,]   5.5328     0.1075   51.48  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

3.

(a) MAR, because the missing of y_2 do not depend on y_2 but depends on y_1.

It is ignorable because the missing data are MAR and the parameter of missingness ψ is distinct from the parameter of fata model θ.

(b) MNAR,because the missing of y_2 depends on y_2.

It is not ignorable because the maximum likelihood estimator based on the observed data likelihood can be seriously biased if the data is MNAR.

(c) MAR, because the missing of y_2 do not depend on y_2 but depends on y_1.

It is not ignorable because the parameter of missingness μ_1 is from the parameter of fata model θ.

4. In this case, the complete data log likelihood is

$$L(\beta; y_{\text{obs}}, y_{\text{mis}}) = \prod_1^n (p_i(\beta))^{y_i}(1-p_i(\beta))^{1-y_i}$$

and therefore the corresponding log likelihood is given by

$$\log L(\beta; y_{\text{obs}}, y_{\text{mis}}) = \sum_1^n (y_i(\beta_0 + \beta_1 x_i) - \log(1 + e^{\beta_0 + \beta_1 x_i}))$$

We now proceed to the E-step with assuming that the first m values of Y are observed and the remaining n-m are missing.

$Q(\theta, \theta^{(t)}) = E_Y(\log L(\beta; y_{\text{obs}}, y_{\text{mis}})|y_{\text{obs}}, \beta_0^{(t)}, \beta_1^{(t)})$

$Q(\theta, \theta^{(t)}) = \sum_1^m y_i(\beta_0 + \beta_1 x_i) + \sum_{m+1}^n E(y_i|\beta_0^{(t)}, \beta_1^{(t)})(\beta_0 + \beta_1 x_i) - \sum_1^n \log(1 + e^{\beta_0 + \beta_1 x_i})$

It is difficult to calculate the vertex by differencing Q, therefore, we use the maxlik function to get the mle. The algorithm is as follows:

```r
load("dataex4.Rdata")
#define the EM function
em.missing <- function(data, beta, eps){
  #initialize
  x=data[,1]
  y=data[,2]
  n=sum(x)
  diff =1
  beta0=beta[1]
  beta1=beta[2]
  # get th eindex of missing value
  mis= which(is.na(y))
  obs=which(!is.na(y))

  while(diff>eps){
    beta.old = beta
    ybar=exp(beta.old[1]+beta.old[2]*x[mis])/(1+exp(beta.old[1]+beta.old[2]*x[mis]))
    #E step
    log_like <- function(beta,data){
      x <- data[,1]; y <- data[,2]
      beta0=beta[1]; beta1=beta[2]
      sum(y[obs]*(beta0+beta1*x[obs]))+ sum(ybar*(beta0+beta1*x[mis])) - sum(log(1+exp(beta0+beta1*x)))
```

```
    }
    #M step
    mle <- maxLik(logLik = log_like, data = data, start = c(0, 0))
    beta=mle[[2]]
    diff <- sum((beta - beta.old)**2)
    }
  return(beta)
}
#choose a starting point and eps
em.missing(data = dataex4, beta=c(1,-2), eps=0.0000001)
```

```
## [1]  0.9755768 -2.4800822
```

5.

(a) In this case, let $z_i = 1$ if y_i is observed, $z_i = 0$ if y_i is missing. Note that $\Pr(Z_i = 1) = p$

The likelihood function is

$$L(\beta; y_{\text{obs}}, y_{\text{mis}}) = \prod_1^n (p f_{\text{LogNormal}})^{z_i} ((1-p)(f_{\exp}))^{1-z_i}$$

and therefore the corresponding log likelihood is given by

$$\log L = \sum_1^n z_i(-\frac{(\log y_i - \mu)^2}{2\sigma^2} + \log p - \log y_i - \log \sigma - 0.5\log(2\pi)) + \sum_1^n (1-z_i)(-\lambda y_i + \log\lambda + \log(1-p))$$

Therefore, for the E-step we would need to compute $Q(\theta|\theta^{(t)}) = E_z(\log L|y, \theta^{(t)})$

$$= \sum_1^n E_z(z_i|y,\theta^{(t)})(-\frac{(\log y_i - \mu)^2}{2\sigma^2} + \log p - \log y_i - \log\sigma - 0.5\log(2\pi)) + \sum_1^n (1 - E_z(z_i|y,\theta^{(t)}))(-\lambda y_i + \log\lambda + \log(1-p))$$

Here, let

$$E(Z_i|y,\theta^{(t)}) = 1 \times Pr(z_i = 1) + 0 \times Pr(z_i = 0) = \frac{p^{(t)} f_{\text{LogNormal}}}{p^{(t)} f_{\text{LogNormal}} + (1-p^{(t)}) f_{\text{Exp}}} = \tilde{p}_i^{(t)}$$

Therefore,

$$Q(\theta|\theta^{(t)}) = \sum_1^n \tilde{p}_i^{(t)}(-\frac{(\log y_i - \mu)^2}{2\sigma^2} + \log p - \log y_i - \log\sigma - 0.5\log(2\pi)) + \sum_1^n (1-\tilde{p}_i^{(t)})(-\lambda y_i + \log\lambda + \log(1-p))$$

For the M step, the updating equations are as follows:

$$\frac{\partial}{\partial p} Q(\theta|\theta^{(t)}) = 0 \Rightarrow p^{(t+1)} = \frac{\sum_1^n \tilde{p}_i^{(t)}}{n}$$

$$\frac{\partial}{\partial \mu} Q(\theta|\theta^{(t)}) = 0 \Rightarrow \mu^{(t+1)} = \frac{\sum_1^n \tilde{p}_i^{(t)} \log y_i}{\sum_1^n \tilde{p}_i^{(t)}}$$

$$\frac{\partial}{\partial \sigma^2} Q(\theta|\theta^{(t)}) = 0 \Rightarrow (\sigma^2)^{(t+1)} = \frac{\sum_1^n \tilde{p}_i^{(t)} (\log y_i - \mu^{(t+1)})^2}{\sum_1^n \tilde{p}_i^{(t)}}$$

$$\frac{\partial}{\partial \lambda} Q(\theta|\theta^{(t)}) = 0 \Rightarrow (\lambda)^{(t+1)} = \frac{\sum_1^n (1 - \tilde{p}_i^{(t)})}{\sum_1^n (1 - \tilde{p}_i^{(t)}) y_i}$$

which can be solved iteratively.

(b) The mle for each component $\hat{\theta} = (\hat{p}, \hat{\mu}, \hat{\sigma}^2, \hat{\lambda}) = (0.4795500, 2.0132615, 0.9293769^2, 1.0330191)$

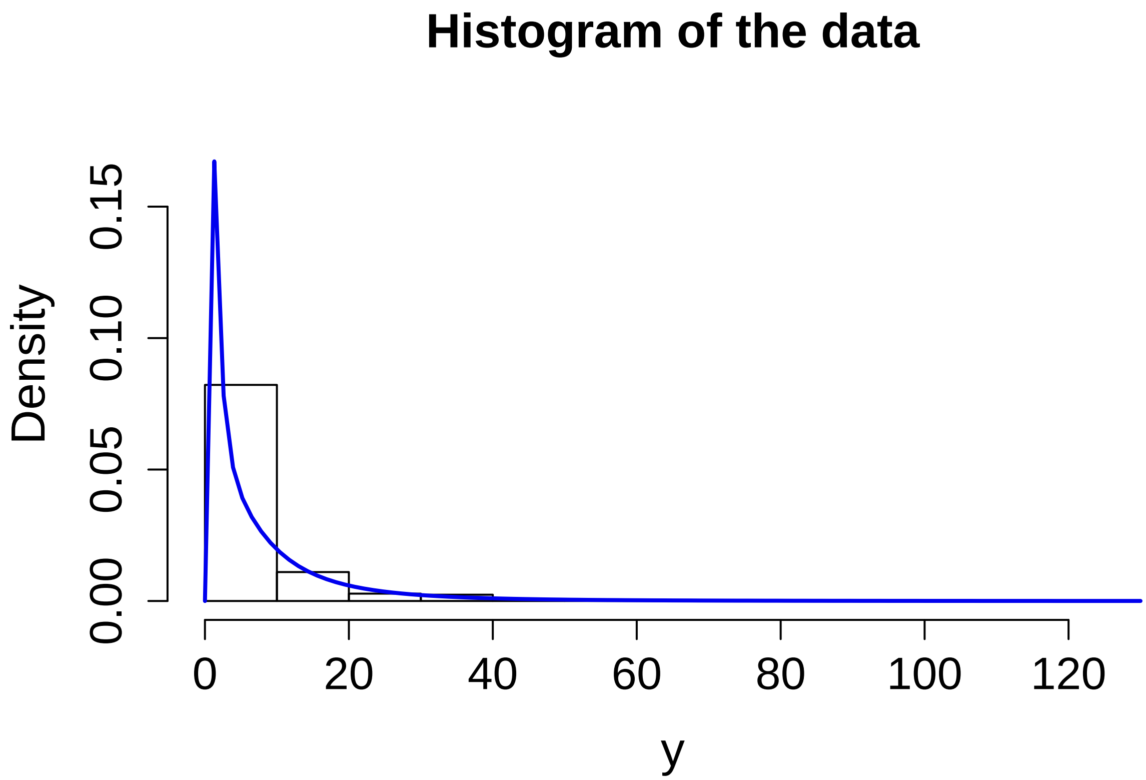
```
load("dataex5.Rdata")
#EM function
em.mixture.two <- function(y, theta0, eps){
  n <- length(y)
  theta <- theta0
  p <- theta[1]
  mu <- theta[2]; sigma <- theta[3]; lambda <- theta[4]
  diff <- 1
  while(diff > eps){
    theta.old=theta
    # E step
    ptilde1 <- p*dlnorm(y, meanlog = mu, sdlog = sigma)
    ptilde2 <- (1 - p)*dexp(y, rate = lambda)
    ptilde <- ptilde1/(ptilde1 + ptilde2)
    #M-step
    p <- mean(ptilde)
    mu <- sum(ptilde * log(y))/sum(ptilde)
    sigma <- sqrt(sum(ptilde * (log(y)-theta[2])**2)/sum(ptilde))
    lambda <- sum(1-ptilde)/sum((1-ptilde)*y)

    theta <- c(p, mu, sigma, lambda)
    diff <- sum(abs(theta - theta.old))
    }
  return(theta)
  }

y = dataex5
theta0 = c(0.1, 1, 0.25, 2)
eps = 0.000001
res <- em.mixture.two(y = dataex5, theta0 = c(0.1, 1, 0.5, 2), eps = 0.000001)
#the theta we get is
res
```

```
## [1] 0.4795500 2.0132615 0.9293769 1.0330191
```

```
# plotting
p <- res[1]; mu <- res[2]; sigma <- res[3]; lambda <- res[4]
hist(y, main = "Histogram of the data",
     xlab = "y",
     ylab = "Density",
     cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.4,freq = F, ylim = c(0,0.18))
curve(p*dlnorm(x, mu, sigma) + (1 - p)*dexp(x, lambda), add = TRUE, lwd = 2, col = "blue2")
```


Histogram of the data
Density
0.00
0.05
0.10
0.15
0
20
40
60
80
100
120
y