Homework 6

Louis Bensard

March 31, 2018

Problem GH 4.2

(a)

Zero-inflated Poisson mixture:

We have a probability α of belonging to the first group (liars). We have a probability β of belonging to the second group (truthful people). Risky encounters of this group are assumed to follow a $Poisson(\mu)$ distribution. We have a probability $1-\alpha-\beta$ of belonging to the third group (high risk people). Risky encounters of this group are assumed to follow a $Poisson(\lambda)$ distribution. Let y_i be the number of risky sexual encounters reported by subject $i = 1, 2, ..., 1500$. The density of y_i is then :

$$f(y_i) = \alpha \cdot 1_{\{y_i=0\}} + \frac{\beta \cdot e^{-\mu}\mu^{y_i}}{y_i!} + \frac{(1-\alpha-\beta) \cdot e^{-\lambda}\lambda^{y_i}}{y_i!}$$

We want to estimate $\theta = (\alpha, \beta, \mu, \lambda)^T$. To do so, we are going to assume that we already know which individual belongs to which group. Therefore, we are going to consider the complete data $x_i = (y_i, z_i)$, with y_i being the i^{th} observed data and $z_i = (z_{i1}, z_{i2}, z_{i3})$ where:

$$z_{ik} = \begin{cases} 1, \text{ if subject i belongs to group } k = 1, 2, 3 \\ 0, \text{ otherwise} \end{cases}$$

The joint density of (y_i, z_{ik}) is given by:

$$f(y_i, z_i) = f(y_i|z_i)f(z_i) = [\alpha \cdot 1_{\{y_i=0\}}]^{z_{i1}} \cdot [\beta \cdot \frac{e^{-\mu}\mu^{y_i}}{y_i!}]^{z_{i2}} \cdot [(1-\alpha-\beta) \cdot \frac{e^{-\lambda}\lambda^{y_i}}{y_i!}]^{z_{i3}}$$

Thus,

$$l_{c,i}(\theta) = z_{i1} \cdot log[\alpha \cdot 1_{\{y_i=0\}}] + z_{i2} \cdot log[\beta \cdot \frac{e^{-\mu}\mu^{y_i}}{y_i!}] + z_{i3} \cdot log[(1-\alpha-\beta) \cdot \frac{e^{-\lambda}\lambda^{y_i}}{y_i!}]$$

Since $l_c(\theta) = \sum_{i=1}^{N} l_{c,i}(\theta)$ $(N = 1500)$ and $Q(\theta', \theta) = E(l_c(\theta')|y_1, ..., y_N, \theta) = E^*(l_c(\theta'))$ then:

E-step:

$$Q(\theta', \theta) = \sum_{i=1}^{N} z_{i1}^* \cdot (log[\alpha'] \cdot 1_{\{y_i=0\}}) + z_{i2}^* \cdot (log[\beta'] - \mu' + y_i log[\mu']) + z_{i3} \cdot (log[(1-\alpha'-\beta') - \lambda' + y_i log[\lambda']) + C$$

With:

$$z_{i1}^* = E^*(z_{i1}) = P(z_{i1} = 1|\theta, y_i) = \frac{f(y_i|z_{i1}=1) \cdot P(z_{i1}=1)}{f(y_i)} = \frac{\alpha \cdot 1_{\{y_i=0\}}}{f(y_i)}$$

$$z_{i2}^* = \frac{\beta e^{-\mu}\mu^{y_i}}{y_i! f(y_i)}$$

$$z_{i3}^* = \frac{\beta e^{-\lambda}\lambda^{y_i}}{y_i! f(y_i)}$$

M-step:

The maximum likelihood estimates for $\theta = (\alpha, \beta, \mu, \lambda)$ are:

$$\hat{\alpha} = \sum_{i=1}^{N} \frac{z_{i1}^*}{N}$$
$$\hat{\beta} = \sum_{i=1}^{N} \frac{z_{i2}^*}{N}$$

Taking the partial derivative with respect to μ and to λ gives us:

$$\hat{\mu} = \frac{\sum_{i=1}^{N} y_i z_{i2}^*}{\sum_{i=1}^{N} z_{i2}^*} = \frac{\sum_{i=1}^{N} y_i z_{i2}^*}{N\hat{\beta}}$$
$$\hat{\lambda} = \frac{\sum_{i=1}^{N} y_i z_{i3}^*}{\sum_{i=1}^{N} z_{i3}^*}$$

Thus at each iteration, $Q(\theta', \theta)$ is maximized at $\theta' = \hat{\theta} = (\hat{\alpha}, \hat{\beta}, \hat{\mu}, \hat{\lambda})$.

Here is the EM-Algorithm that we will use in part (b):

- Start with an initial θ_0
- E-step: Compute z_{i1}^*, z_{i2}^* and z_{i3}^*
- M-step: Compute $\hat{\alpha}$, $\hat{\beta}$, $\hat{\mu}$ and $\hat{\lambda}$ using E-step
- replace θ by $\hat{\theta}$ and if $MRE > 10^{-6}$, go to E-step. Otherwise terminate.

(b)

```
data = c(rep(0, 379), rep(1, 299), rep(2,222), rep(3, 145), rep(4, 109), rep(5, 95), rep(6, 73),
         rep(7, 59), rep(8, 45), rep(9, 30), rep(10, 24), rep(11, 12), rep(12, 4), rep(13, 2),
         rep(14, 0), rep(15, 1), rep(16,1))
y = data

theta_0 = c(1/3,1/3, 4, 8)

maxit = 200 ; tolerr = 1e-6

theta_ME = EM_mixture(y, theta_0, maxit, tolerr)
```

```
##
## It n    alpha_n    beta_n    mu_n    lambda_n      MRE
##  0     0.333333   0.333333 4.000000 8.000000     NA
##  1     0.248040   0.557600 2.569398 6.510116     6.5e-01
##  2     0.215343   0.583702 2.288271 6.779281     1.4e-01
## ...
## ...
## ...
```

```
## 112     0.122169   0.562542 1.467495 5.938914     1.4e-06
## 113     0.122169   0.562542 1.467493 5.938911     1.3e-06
## 114     0.122169   0.562542 1.467491 5.938910     1.2e-06
## 115     0.122169   0.562542 1.467490 5.938908     1.1e-06
## 116     0.122168   0.562542 1.467488 5.938906     9.6e-07
## ------------------------------------------------------------
```

After 116 iterations, the modified relative error finally gets under 10^{-6} and thus the algorithm stops. Thus, the parameter estimates are: $\hat{\alpha} = 0.1222$, $\hat{\beta} = 0.5625$, $\hat{\mu} = 1.4675$ and $\hat{\lambda} = 5.9389$.

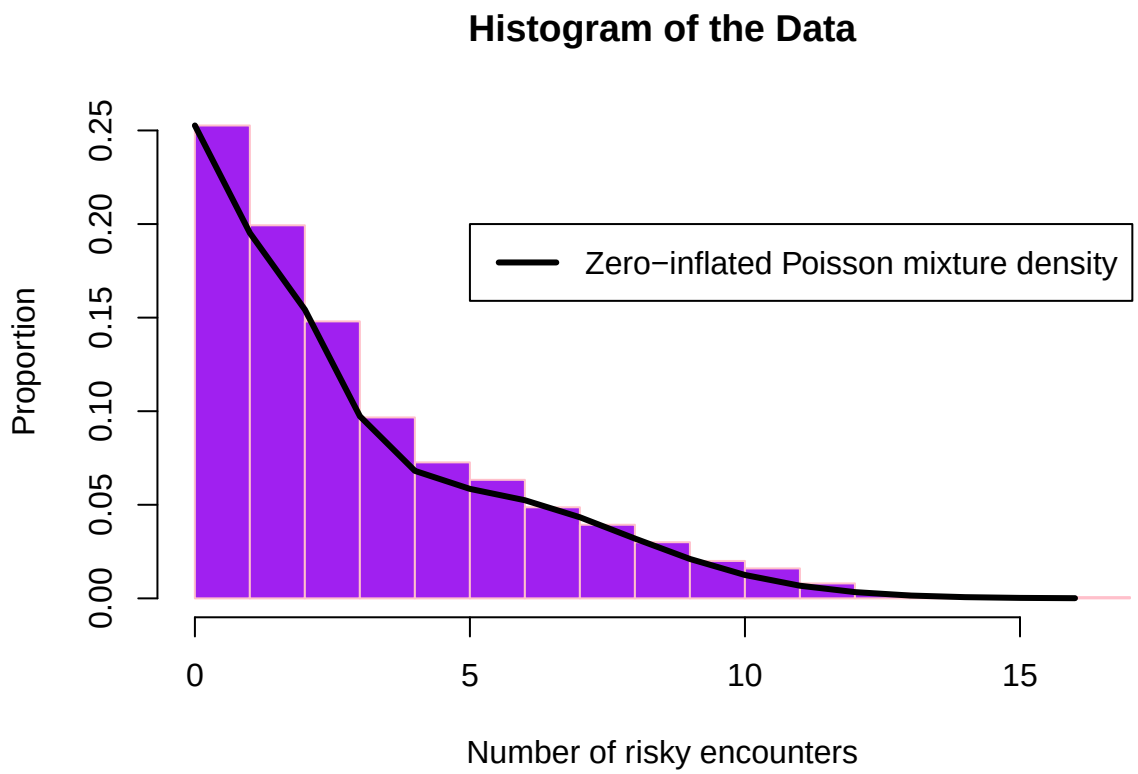
Let's plot the density with those parameter estimates on top of the histogram of the data to check our results:

```
x = seq(0,16, length=17)

z = theta_ME[2]*dpois(x,lambda=theta_ME[3]) + (1-theta_ME[1]-theta_ME[2])*dpois(x,lambda=theta_ME[4])

#we add alpha  when x=0 (zero-inflated mixture)
z[1] = z[1] + theta_ME[1]

hist(data, xlab="Number of risky encounters", ylab='Proportion', main="Histogram of the Data",
     col='purple', border='pink', breaks=seq(-0.001,16.999,by=1), freq=FALSE)
lines(x,z, lwd=3)
legend(5, 0.20, "Zero-inflated Poisson mixture density", cex=1, col="black", lwd=3)
```



We can see that the density follows very well the trend of the histogram. So we are very confident that our above optimization is correct

Problem GH 4.3

(a)

The data set contains missing data drawn from a trivariate normal density with parameters μ and Σ. We are going to use the EM algorithm to estimate those parameters in spite of the missing data. Let's consider the complete data $x = (y_o, y_m)$ with y_o being the observed data and y_m being the missing data. Then, we partition the parameters the following way:

$$\mu = \begin{pmatrix} \mu_o \\ \mu_m \end{pmatrix}, \quad \begin{pmatrix} \Sigma_{oo} & \Sigma_{om} \\ \Sigma_{mo} & \Sigma_{mm} \end{pmatrix}$$

Let $\theta = (\mu, \Sigma)$, then the complete log-likelihood can be written as:

$$l_c(\theta) = \frac{-n}{2}[p \cdot log|\Sigma| + tr\{\Sigma^{-1}\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T\}]$$

E-Step:

At each iteration of the algorithm, we want to find θ' that maximizes
$Q(\theta', \theta) = E(l_c(\theta')|\theta) = E^*(l_c(\theta'))$. Thus, Q can be written as:

$$Q(\theta', \theta) = \frac{-n}{2}[p \cdot log|\Sigma'| + tr\{\Sigma'^{-1}(S^* - \mu'\bar{x^*}^T - \bar{x^*}\mu'^T + \mu'\mu'^T\}]$$

With:

$$\bar{x^*} = \frac{1}{n}\sum_{i=1}^{n} E^*(x_i)$$
$$E^*(x_i) = \begin{pmatrix} y_{o,i} \\ y^*_{m,i} \end{pmatrix}$$
$$y^*_{m,i} = \mu_{m,i} + \Sigma_{mo,i}\Sigma^{-1}_{mo,i}(y_{o,i} - \mu_{o,i})$$
$$S^* = \frac{1}{n}\sum_{i=1}^{n} E^*(x_i x^T)$$
$$E^*(x_i x^T) = \begin{pmatrix} y_{o,i}y_{o,i}^T & y_{o,i}(y^*_{m,i})^T \\ y^*_{m,i}y_{o,i}^T & E^*(y_{m,i}, y_{m,i}) \end{pmatrix}$$
$$E^*(y_{m,i}, y^T_{m,i}) = \Sigma_{mm,i} - \Sigma_{mo,i}\Sigma^{-1}_{oo,i}\Sigma_{om,i} + y^*_{m,i}(y^*_{m,i})^T$$

M-Step:

We know that the parameter estimates that maximize the likelihood are: $\tilde{\mu} = \bar{x}$ and
$\tilde{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \tilde{\mu})(x_i - \tilde{\mu})^T$. Therefore, the parameter estimates that maximize Q are going to be:

$$\hat{\mu} = \bar{x^*}$$
$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} E^*[(x_i - \tilde{\mu})(x_i - \tilde{\mu})^T] = S^* - \hat{\mu}\hat{\mu}^T$$

Thus at each iteration, $Q(\theta', \theta)$ is maximized at $\theta' = \hat{\theta} = (\hat{\mu}, \hat{\Sigma})$.

Here is the EM-Algorithm that we will use in part (b):

- Start with an initial θ_0
- E-step: Compute $\bar{x^*}$ and S^*
- M-step: Compute $\hat{\mu}$ and $\hat{\Sigma}$ using E-step
- replace θ by $\hat{\theta}$ and if $MRE > 10^{-6}$, go to E-step. Otherwise terminate.

(b)

```
data = read.table("C:/Users/Louis/Documents/UPMC/M1/Spring 2018/MATH 534/Homework 6/data.txt",
                  header=T)

x = data
p = dim(x)[2]

#we get rid of any row with no data
for(i in 1:(dim(x)[1])){

    if(length(which(is.na(x[i,]))) == p) x = x[-i,]
}

n = dim(x)[1]

mu_0 = rep(0,p)
Sigma_0 = diag(p)
theta_0 = vectorize(mu_0, Sigma_0, p)

maxit = 200 ; tolerr = 1e-6

theta_ME = EM_N3(x, theta_0, maxit, tolerr)
```

```
##
## It n                                        theta_n                                      MRE     CV
##  0   0.00000 0.00000 0.00000 1.00000 0.00000 1.00000 0.00000 0.00000 1.00000   NA  NA
##  1   0.59646 2.48354 7.68604 1.41448 0.71097 1.74241 0.51496 1.64071 12.50237   2.3e+00  1.0e+00
##  2   0.80061 2.81206 8.83479 1.35374 0.82365 0.92309 0.93348 0.86147 4.10560   2.4e+00  1.6e-01
## ...
## ...
## ...
## 18   0.87860 2.85016 9.02574 1.41313 1.00158 0.77793 1.31847 0.70433 2.52243   1.2e-05  4.7e-01
## 19   0.87860 2.85016 9.02574 1.41313 1.00158 0.77793 1.31847 0.70433 2.52244   5.7e-06  4.5e-01
## 20   0.87860 2.85016 9.02574 1.41313 1.00158 0.77793 1.31847 0.70432 2.52244   2.8e-06  4.2e-01
## 21   0.87860 2.85016 9.02574 1.41313 1.00158 0.77793 1.31847 0.70432 2.52244   1.3e-06  3.7e-01
## 22   0.87860 2.85016 9.02574 1.41313 1.00158 0.77793 1.31847 0.70432 2.52244   6.3e-07  5.7e-01
## ------------------------------------------------------------
## $mu
## [1] 0.8786011 2.8501565 9.0257446
##
## $Sigma
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.413132 1.0015797 1.3184714
## [2,] 1.001580 0.7779349 0.7043234
## [3,] 1.318471 0.7043234 2.5224412
```

Note that θ here is the vectorized version of μ and Σ (ie θ contains μ and the lower triangle of Σ as a vector).

Therefore, the parameter estimates considering the observed data and the missing data are:

$$\hat{\mu} = \begin{pmatrix} \hat{\mu_1} \\ \hat{\mu_2} \\ \hat{\mu_3} \end{pmatrix} = \begin{pmatrix} 0.8786 \\ 2.8502 \\ 9.0257 \end{pmatrix}$$

$$\hat{\Sigma} = \begin{pmatrix} \hat{\sigma_{11}} & \hat{\sigma_{12}} & \hat{\sigma_{13}} \\ \hat{\sigma_{21}} & \hat{\sigma_{22}} & \hat{\sigma_{23}} \\ \hat{\sigma_{31}} & \hat{\sigma_{32}} & \hat{\sigma_{33}} \end{pmatrix} = \begin{pmatrix} 1.4131 & 1.0016 & 1.3185 \\ 1.0016 & 0.7779 & 0.7043 \\ 1.3185 & 0.7043 & 2.5224 \end{pmatrix}$$

Let's compute the MLE of μ and Σ from the data if we omit the missing values. We should get something different but still close from the MLE's obtained above.

```
cat("mle_mu = (",mean(na.omit(x[,1])), mean(na.omit(x[,2])), mean(na.omit(x[,3])),")\n")
```

```
## mle_mu = ( 0.818 2.838333 8.998293 )
```

```
print("mle_Sigma=")
```

```
## [1] "mle_Sigma="
```

```
print(((n-1)/n)*cov(na.omit(x)))
```

```
##           x1        x2        x3
## x1 1.730230 1.2543543 1.4457899
## x2 1.254354 0.9731887 0.8282001
## x3 1.445790 0.8282001 2.4864903
```

We do get MLE's of the same magnitude if we omit the missing data, so we are confident that our above optimization is correct.

With trials and error, the convergence ratio is kind of stabilized only by $\beta = 1$. Here is my matrix of convergence ratios for each θ_i for a power $\beta = 1$ (linear). Each column correspond to a different θ_i and each row correspond to an iteration of the algorithm. We can clearly observe the stabilization of the convergence ratio for each θ_i. Therefore, the EM is linerarly convergent for this problem.

[,1] 3.2e-01 1.3e-01 1.5e-01 3.3e-03 2.9e-01 4.3e+00 6.1e-01 1.3e+00 6.6e+00
[,2] 2.8e-01 1.0e-01 1.4e-01 4.4e+01 6.1e-01 1.5e-01 4.8e-01 1.7e-01 1.6e-01
[,3] 3.8e-01 3.3e-01 2.6e-01 1.2e+00 4.7e-01 2.7e-01 4.7e-01 6.4e-01 1.7e-01
[,4] 4.2e-01 6.7e-01 5.5e-01 8.0e-01 4.6e-01 3.9e-01 4.5e-01 8.1e-01 1.7e-01
[,5] 3.7e-01 6.1e-01 7.0e-01 7.2e-01 5.3e-01 3.9e-01 4.8e-01 7.1e-01 3.1e-01
[,6] 3.0e-01 5.6e-01 6.6e-01 6.6e-01 5.7e-01 3.8e-01 5.2e-01 6.1e-01 2.0e+00
[,7] 1.2e-01 5.2e-01 6.1e-01 6.0e-01 5.6e-01 3.9e-01 5.2e-01 5.6e-01 8.2e-01
[,8] 1.2e+00 5.0e-01 5.7e-01 5.5e-01 5.4e-01 4.0e-01 5.0e-01 5.2e-01 6.3e-01
[,9] 1.1e+00 4.9e-01 5.4e-01 5.2e-01 5.1e-01 4.1e-01 4.8e-01 5.1e-01 5.6e-01
[,10] 7.4e-01 4.8e-01 5.2e-01 5.0e-01 5.0e-01 4.3e-01 4.7e-01 5.0e-01 5.3e-01
[,11] 6.3e-01 4.8e-01 5.1e-01 4.9e-01 4.9e-01 4.4e-01 4.6e-01 4.9e-01 5.1e-01
[,12] 5.8e-01 4.8e-01 5.0e-01 4.8e-01 4.8e-01 4.4e-01 4.6e-01 4.9e-01 5.0e-01
[,13] 5.5e-01 4.9e-01 5.0e-01 4.7e-01 4.8e-01 4.4e-01 4.5e-01 4.8e-01 4.9e-01
[,14] 5.4e-01 4.9e-01 4.9e-01 4.7e-01 4.8e-01 4.3e-01 4.5e-01 4.8e-01 4.9e-01
[,15] 5.2e-01 5.1e-01 4.9e-01 4.7e-01 4.7e-01 4.2e-01 4.5e-01 4.8e-01 4.9e-01

[,16] 5.1e-01 5.3e-01 4.9e-01 4.7e-01 4.7e-01 3.8e-01 4.4e-01 4.8e-01 4.8e-01
[,17] 5.1e-01 5.7e-01 4.8e-01 4.6e-01 4.8e-01 2.7e-01 4.2e-01 4.7e-01 4.8e-01
[,18] 5.0e-01 6.3e-01 4.8e-01 4.6e-01 4.8e-01 2.4e-01 3.7e-01 4.7e-01 4.7e-01
[,19] 4.8e-01 7.2e-01 4.7e-01 4.4e-01 4.8e-01 3.3e+00 2.4e-01 4.5e-01 4.5e-01
[,20] 4.6e-01 8.1e-01 4.5e-01 4.1e-01 4.9e-01 1.3e+00 4.3e-01 4.2e-01 4.1e-01
[,21] 4.1e-01 8.9e-01 4.0e-01 3.3e-01 5.1e-01 1.1e+00 2.5e+00 3.3e-01 3.0e-01
[,22] 2.8e-01 9.4e-01 2.6e-01 4.0e-02 5.5e-01 1.0e+00 1.3e+00 2.2e-02 1.3e-01