Assignment 3 Report

Computational Methods for Bayesian Statistics

Trilokinath Modi

October 12, 2020

1 Question set 1

1.1 Q1 - a

The required likelihood function is,

$$\pi(Y_0, ..., Y_9, Z_0, ..., Z_9 \mid \lambda_1, \lambda_2, p) = \prod_{i=0}^{9} \left\{ \frac{p^{Z_i} \left(e^{-\lambda_1} \lambda_1^i\right)^{Z_i} (1-p)^{Y_i - Z_i} \left(e^{-\lambda_2} \lambda_2^i\right)^{Y_i - Z_i}}{i!^{Y_i}} \right\}$$
$$= C\, p^{\sum_i Z_i} (1-p)^{\sum_i Y_i - Z_i} \lambda_1^{\sum_i iZ_i} \lambda_2^{\sum_i iY_i - iZ_i} \exp(-\lambda_1 \sum_i Z_i - \lambda_2 \sum_i Y_i - Z_i)$$

Hence,

$$\log \pi(Y, Z | \lambda_1, \lambda_2, p) = \log C + \sum_i Z_i \log p + \sum_i (Y_i - Z_i) \log(1-p) +$$
$$\log \lambda_1 \sum_i (iZ_i) + \log \lambda_2 \sum_i (iY_i - iZ_i) - \lambda_1 \sum_i Z_i - \lambda_2 \sum_i Y_i - Z_i$$

Also, using the results of previous assignment, we can say,

$$\pi(Z_i \mid Y_i, p, \lambda_1, \lambda_2) \propto \text{Bin}\left(Y_i\,,\, \frac{p\lambda_1^i e^{-\lambda_1}}{p\lambda_1^i e^{-\lambda_1} + (1-p)\lambda_2^i e^{-\lambda_2}}\right)$$

This implies given the parameter values and day countsY_i, we can find the probability of number of days among Y_i that the deaths came from distribution with parameter λ_1. In other words, for a given set of parameter values, if $Y_2 = 271$ then, using the binomial distribution of $\pi(Z_i \mid \text{others})$ we can find the probability of number of days with 2 deaths coming from poisson distribution with parameter λ_1.

The expectation will hence be number of trials $\times$ probability. In terms of equation,

$$\implies \quad E(Z_i \mid Y_i, p, \lambda_1, \lambda_2) = \frac{Y_i\, p\lambda_1^i e^{-\lambda_1}}{p\lambda_1^i e^{-\lambda_1} + (1-p)\lambda_2^i e^{-\lambda_2}}$$
$$\implies \quad E(Z_i \mid Y_i, p^{'}, \lambda_1^{'}, \lambda_2^{'}) = \frac{Y_i\, p^{'} {\lambda_1^{'}}^i e^{-\lambda_1^{'}}}{p^{'} {\lambda_1^{'}}^i e^{-\lambda_1^{'}} + (1-p^{'}) {\lambda_2^{'}}^i e^{-\lambda_2^{'}}}$$
$$\implies \quad E(Z_i \mid Y_i, \theta^{'}) = Y_i\, g_i(\theta^{'})$$

1.2 Q1 - b

The E-step for log posterior or log likelihood can be written as,

$$E_{Z_i \mid Y_i, p', \lambda_1', \lambda_2'} \log \pi(Y, Z \mid \lambda_1, \lambda_2, p) = E(\log C + \sum_i Z_i \log p + \sum_i (Y_i - Z_i) \log(1-p) + \log\lambda_1 \sum_i (iZ_i) + \log\lambda_2 \sum_i (iY_i - iZ_i) - \lambda_1 \sum_i Z_i - \lambda_2 \sum_i Y_i - Z_i)$$

$$\implies \quad E_{Z_i \mid Y_i, \theta'} \log \pi(Y, Z \mid \theta) = \log C + \sum_i E(Z_i \mid Y_i, \theta') \log p + \sum_i (Y_i - E(Z_i \mid Y_i, \theta')) \log(1-p) + \log\lambda_1 \sum_i (iE(Z_i \mid Y_i, \theta')) + \log\lambda_2 \sum_i (iY_i - iE(Z_i \mid Y_i, \theta')) - \lambda_1 \sum_i E(Z_i \mid Y_i, \theta') - \lambda_2 \sum_i Y_i - E(Z_i \mid Y_i, \theta'))$$

$$\implies \quad E_{Z_i \mid Y_i, \theta'} \log \pi(Y, Z \mid \theta) = \log C + \sum_i Y_i\, g_i(\theta') \log p + \sum_i (Y_i - Y_i\, g_i(\theta')) \log(1-p) + \log\lambda_1 \sum_i (iY_i\, g_i(\theta')) + \log\lambda_2 \sum_i (iY_i - iY_i\, g_i(\theta')) - \lambda_1 \sum_i Y_i\, g_i(\theta') - \lambda_2 \sum_i Y_i - Y_i\, g_i(\theta'))$$

$$\text{Where,} \quad g_i(\theta') = \frac{p' {\lambda_1'}^i e^{-\lambda_1'}}{p' {\lambda_1'}^i e^{-\lambda_1'} + (1-p') {\lambda_2'}^i e^{-\lambda_2'}}$$

1.3 Q1 - c

The M step involves maximizing the expectation with prior for each parameter.

$$\text{Maximize,} \quad Q(\theta) = E_{Z_i \mid Y_i, \theta'} \log \pi(Y, Z \mid \theta) + \log \pi(\theta)$$

$$\implies \frac{\partial Q}{\partial \theta} = 0$$

$$\implies \frac{\partial Q}{\partial p} = 0 \,, \frac{\partial Q}{\partial \lambda_1} = 0 \,, \frac{\partial Q}{\partial \lambda_2} = 0$$

$$\frac{\partial Q}{\partial p} = \frac{1}{p} \sum_i Y_i g_i(\theta') - \frac{1}{1-p} \sum_i Y_i (1 - g_i(\theta')) = 0$$

$$\implies p = \frac{\sum_i Y_i g_i(\theta')}{\sum_i Y_i}$$

$$\frac{\partial Q}{\partial \lambda_1} = \frac{1}{\lambda_1} \sum_i (iY_i g_i(\theta')) - \sum_i Y_i g_i(\theta') - 1 = 0$$

$$\implies \lambda_1 = \frac{\sum_i (iY_i g_i(\theta'))}{1 + \sum_i Y_i g_i(\theta')}$$

$$\frac{\partial Q}{\partial \lambda_2} = \frac{1}{\lambda_2} \sum_i (iY_i (1 - g_i(\theta'))) - \sum_i Y_i (1 - g_i(\theta')) - 1 = 0$$

$$\implies \lambda_2 = \frac{\sum_i (iY_i (1 - g_i(\theta')))}{1 + \sum_i Y_i (1 - g_i(\theta'))}$$

$$\text{Where,} \quad g_i(\theta') = \frac{p' {\lambda_1'}^i e^{-\lambda_1'}}{p' {\lambda_1'}^i e^{-\lambda_1'} + (1-p') {\lambda_2'}^i e^{-\lambda_2'}}$$

Eigen values are easily computable and they are all negative implying it is a maxima and not minima or saddle point.

1.4 Q1 - d

A comparison plot for observed counts against predicted counts obtained when uni or bi Poisson distribution was considered is shown in figure 1. Clearly, the combination of Poisson distribution i.e. with parameter p, λ_1, λ_2 has much better fit compared to a model made by single Poisson distribution with parameter λ. The convergence of parameter values and the difference in observed and predicted count is shown in figure 2.

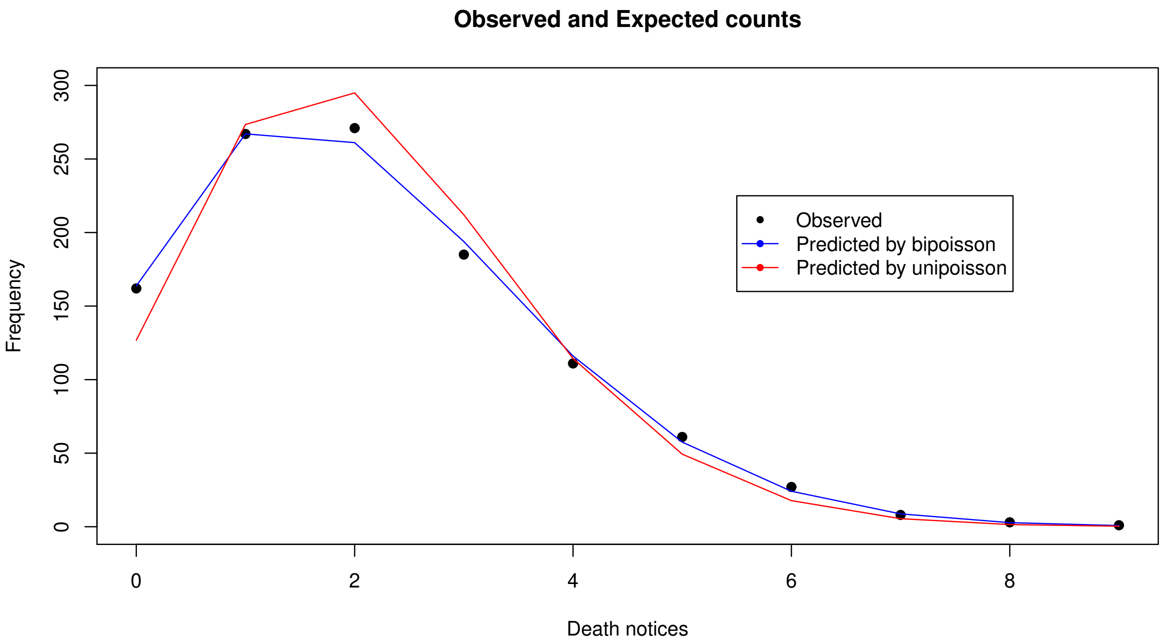

Figure 1: *Observed against expected counts*

To compare the results, we can look at the histogram of posterior of each of the individual parameter made in assignment 2. The highest frequency range in the histograms correspond well with the estimated parameter values using EM algorithm.

The R code used to obtain figures and predicted counts is,

```
setwd("D:/Masters Program Chalmers/Projects and Labs/CMBS")

# Q1 - d ------------------------------------------------------------------

dayCount <- c(162,267,271,185,111,61,27,8,3,1)
deathNotices <- c(0:9)

functionG <- function(p, lambda1, lambda2){
  num <- p*(lambda1^deathNotices)*exp(-lambda1)
  denom <- num + (1-p)*(lambda2^deathNotices)*exp(-lambda2)
```

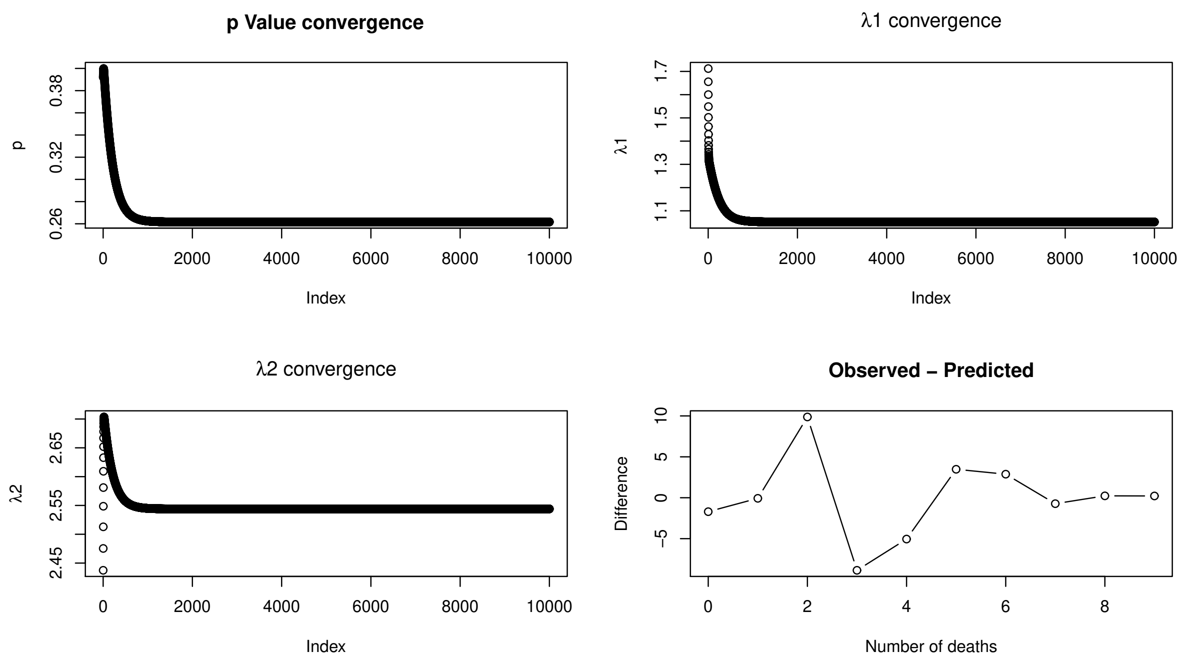

Figure 2: *Convergence(plot 1,2,3) and difference in counts(plot 4)*

```
  functionG <- num/denom
}

N <- 10000
pOld <- 0.5
lambda1Old <- rbeta(1,1,1)
lambda2Old <- rbeta(1,1,1)

PVector <- rep(0, times = N)
lambda1Vector <- rep(0, times = N)
lambda2Vector <- rep(0, times = N)

for (iter in 1:N){
  numerator <- sum(dayCount*functionG(pOld,lambda1Old, lambda2Old))
  p <- numerator/sum(dayCount)
  lambda1 <- sum(deathNotices*dayCount*functionG(pOld,lambda1Old, lambda2Old))
  lambda1 <- lambda1/(1 + numerator)
  lambda2 <- sum(deathNotices * dayCount * (1 - functionG(pOld,lambda1Old,
     ↪ lambda2Old)))
  lambda2 <- lambda2/(1 + sum(dayCount*(1 - functionG(pOld,lambda1Old, lambda2Old
     ↪ ))))

  pOld <- p
  PVector[iter] <- p
  lambda1Old <- lambda1
  lambda1Vector[iter] <- lambda1
```

```
  lambda2Old <- lambda2
  lambda2Vector[iter] <- lambda2
}

#Prediction using above p, lambda_1 and lambda_2
predictedCounts <- sum(dayCount)*(pOld*dpois(deathNotices, lambda1Old) +
  (1 - pOld)*dpois(deathNotices, lambda2Old))
legend(x = 5.5, y= 225, legend = c("Observed", "Predicted␣by␣bipoisson"),col = c(
    ↪ "black","blue"), pch = c(20,20))

#Plotting predicted counts, observed counts and new predicted counts if there
    ↪ was one distribution
# i.e. Assignment 2 Q2-b

# If there was only one distribution
lambda <- sum(deathNotices*dayCount)/sum(dayCount)
newPredictedCounts <- sum(dayCount)*dpois(deathNotices,lambda)
plot(deathNotices, y = dayCount, col = "black", pch = 20, ylim = c(0,300),
     cex = 1.5,xlab = "Death␣notices", ylab = "Frequency", main = "Observed␣and␣
       ↪ Expected␣counts")
lines(deathNotices, y = predictedCounts, col = "blue", pch = 20, cex = 1.5)
lines(deathNotices, y = newPredictedCounts, col = "red", pch = 20, cex = 1.5)
legend(x = 5.5, y= 225, legend = c("Observed","Predicted␣by␣bipoisson","Predicted
    ↪ ␣by␣unipoisson"),col = c("black","blue","red"), pch = c(20), lty = c(0,1,1)
    ↪ )


differenceCount <- dayCount - predictedCounts
differenceCount

par(mfrow = c(2,2))
plot(PVector, ylab = "p", main = "p␣Value␣convergence")
plot(lambda1Vector, ylab = expression(paste(lambda, "1")), main = expression(
    ↪ paste(lambda, "1␣convergence")))
plot(lambda2Vector, ylab = expression(paste(lambda, "2")), main = expression(
    ↪ paste(lambda, "2␣convergence")))
plot(deathNotices,differenceCount,type = "b", ylab = "Difference", xlab = "Number
    ↪ ␣of␣deaths",main = "Observed␣-␣Predicted")
dev.off()
```