EM Algorithm - Survival Data

Matthew Scott | mscott24@bu.edu

January 2023

Question Let $t_1, t_2, ..., t_n$ i.d.d. with pdf $\rho exp(-\rho t)$, i.e., exponential with parameter ρ. Supposed that we observe (y_i, δ_i), where $y_i = min(t_i, c_i)$ and $\delta_i = i$ if $t_i < c_i$ and 0 otherwise for $i = 1, 2, ..., n$. Implement an EM-algorithm for the Stanford Heart Transplant data (for the variables survival time y and survival status δ) and compute the standard error of the estimate $\hat{\rho}_{MLE}$.

Derivation Assume that $Y_1, ..., Y_m$ are non-censored data and the rest of the observation are censored with censoring times given by $c_{m+1}, ..., c_n$. Let $Z_{m+1}, ..., Z_n$ be the survival times for the censored data, with the complete data likelihood given by

$$p(\rho|y, z) \propto \rho^n exp[-\rho(\sum_{i=1}^{m} y_i + \sum_{i=m+1}^{n} z_i)]$$

$$logp(\rho|y, z) \propto nlog\rho - \rho(\sum_{i=1}^{m} y_i + \sum_{i=m+1}^{n} z_i)$$

The conditional predictive distribution of z given $Z > c$ is a truncated exponential distribution. Due to the memoryless property of exponential distributions, we have

$$E(Z_i|Z_i > c_i, \rho^{(k)}) = c_i + \frac{1}{\rho^{(k)}}$$

Thus

$$Q(\rho, \rho^{(k)}) = nlog\rho - \rho(\sum_{i=1}^{m} y_i + \sum_{i=m+1}^{n} (c_i + \frac{1}{\rho^{(k)}}))$$

In the M-step, we maximize $Q(\rho, \rho^{(k)})$, leading to the EM update

$$\rho^{(k+1)} = \frac{n}{\sum_{i=1}^{m} y_i + \sum_{i=m+1}^{n} (c_i + \frac{1}{\rho^{(k)}})}$$

The SE can be found via Louis's method

$$\frac{d^2}{d\rho^2} Q(\rho, \rho_{EM}) = \frac{n}{\rho_{EM}^2}$$

$$\rightarrow Var(\frac{d}{d\rho} logp(\rho|y, z)) = \sum_{i=m+1}^{n} Var(z_i|z_i > c_i, \rho_{EM}) = \frac{n-m}{\rho_{EM}^2}$$

Thus the observed Fisher information evaulated at ρ_{EM} is

$$\frac{n}{\rho_{EM}^2} - \frac{n-m}{\rho_{EM}^2} = \frac{m}{\rho_{EM}^2}$$

$$\rightarrow \boxed{SE = \frac{\hat{\rho}_{EM}}{\sqrt(m)}}$$

EM Algorithm

```
library(survival)

#initialize
r <- rep(NA,1000)
r[1] <- 0.0001
n <- length(stanford2$time)

#perform iterations
for(k in 1:999) {
  r[k+1]<-n/{sum(stanford2$time[stanford2$status==0]) + sum(stanford2$time[stanford2$status==1]+(1/r[k])
  mle_em <- r[k+1]
  #stop is convergence is reached
  if(abs(r[k+1]-r[k]) < 10^(-10)) {break}
}
#calculate SE
se_em <- mle_em/sqrt(length(stanford2$time[stanford2$status==0]))
```

Estimated $\hat{\rho}_{MLE}$

```
mle_em
```

```
## [1] 0.00055366
```

SE of $\hat{\rho}_{MLE}$

```
se_em
```

```
## [1] 6.570736e-05
```