

MTH 511a - 2020: Lecture 19

Instructor: Dootika Vats

The instructor of this course owns the copyright of all the course materials. This lecture material was distributed only to the students attending the course MTH511a: “Statistical Simulation and Data Analysis” of IIT Kanpur, and should not be distributed in print or through electronic media without the consent of the instructor. Students can make their own copies of the course materials for their use.

1 The EM algorithm

Recall, the objective function to maximize is

$$l(\theta|X) = \log \int f(\mathbf{x}, \mathbf{z}|\theta) d\nu_{\mathbf{z}},$$

The EM algorithm iterates through the following: Consider a starting value θ_0 . Then for any $k + 1$ iteration

1. **E-Step:** Compute

$$q(\theta; \theta_{(k)}) = \mathbb{E}_{Z|x} \left[\log f(\mathbf{x}, \mathbf{z}|\theta) \mid X = x, \theta_{(k)} \right]$$

where the expectation is computed with respect to the conditional distribution of Z given $X = x$ for the current iterate $\theta_{(k)}$.

2. **M-Step:** Compute

$$\theta_{k+1} = \arg \max_{\theta \in \Theta} q(\theta; \theta_k).$$

We will look at another EM algorithm example.

1.1 Censored data example

A light bulb company is testing the failure times of their bulbs and know that failure times follow $\text{Exp}(\lambda)$ for some $\lambda > 0$. They test n light bulbs, so the failure time of each light bulb is

$$Z_1, \dots, Z_n \stackrel{iid}{\sim} \text{Exp}(\lambda).$$

However, officials recording these failure times walked into the room only at time T and observed that $m < n$ of the bulbs had already failed. Their failure time cannot be recorded. So observed data is

$$E_1 = 1, \dots, E_m = 1, Z_{m+1}, Z_{m+2}, \dots, Z_n,$$

where $E_i = I(Z_i < T)$. Our goal is to find the MLE for λ . Note that

- If we ignore the first m light bulbs, then not only do we have a smaller sample size, but we also have a biased sample which do not contain the bottom tail of the distribution of failure times.

So we must account for the “missing data”. First, we find the observed data likelihood.

Note that $E_i \sim \text{Bern}(p)$ where $p = \Pr(E_i = 1) = \Pr(Z_i \leq T) = 1 - e^{-\lambda T}$ (from the CDF of an exponential distribution). So,

$$\begin{aligned} L(\lambda | E_1, \dots, E_m, Z_{m+1}, \dots, Z_n) &= \prod_{i=1}^m \Pr(E_i = e_i) \cdot \prod_{j=m+1}^n f(z_j | \lambda) \\ &= \prod_{i=1}^m (e^{-\lambda T})^{1-e_i} (1 - e^{-\lambda T})^{e_i} \prod_{j=m+1}^n \lambda \exp\{-\lambda Z_j\} \\ &= (1 - e^{-\lambda T})^m \lambda^{n-m} \exp\left\{-\lambda \sum_{j=m+1}^n Z_j\right\}. \end{aligned}$$

Closed-form MLEs are difficult here, and some sort of numerical optimization is useful. We will resort to the EM algorithm, that has the added advantage that the estimates of Z_1, \dots, Z_m may be obtained as well.

Also, note that if we choose to “throw away” the censored data, then the likelihood is

$$L_{bad}(\lambda | Z_{m+1} \dots Z_n) = \lambda^{n-m} \exp\left\{-\lambda \sum_{j=m+1}^n Z_j\right\}$$

and the MLE is

$$\lambda_{\text{MLE, bad}} = \frac{n - m}{\sum_{j=m+1}^n Z_j}$$

The above MLE is a bad estimator since the data thrown away is not censored at random, and in fact, all those bulbs fused early. So the bulb company cannot just throw that data away, as that would be dishonest!

Now we implement the EM algorithm for this example. First, note that the complete unobserved data is Z_1, \dots, Z_n and the complete log likelihood is

$$\log f(z|\lambda) = \log \left\{ \prod_{i=1}^n \lambda e^{-\lambda z_i} \right\} = n \log \lambda - \lambda \sum_{i=1}^n Z_i.$$

In order to implement the EM algorithm, we need the distribution of

$$\begin{aligned} \Pr(\text{unobserved} \mid \text{observed}) &= f(Z_1, \dots, Z_m \mid E_1, \dots, E_m, Z_{m+1}, \dots, Z_n) \\ &= f(Z_1, \dots, Z_m \mid E_1, \dots, E_m) \\ &= \prod_{i=1}^m f(Z_i \mid E_i) \\ &= \prod_{i=1}^m f(Z_i \mid Z_i \leq T) \\ &= \prod_{i=1}^m \frac{\lambda e^{-\lambda z_i}}{1 - e^{-\lambda T}} \mathbb{I}(z_i \leq T). \end{aligned}$$

Further,

$$\begin{aligned} \mathbb{E}[Z_i \mid E_i = 1] &= \mathbb{E}[Z_i \mid Z_i \leq T] \\ &= \int_0^T z_i \frac{\lambda e^{-\lambda z_i}}{1 - e^{-\lambda T}} = \dots = \frac{1}{\lambda} - \frac{T e^{-\lambda T}}{1 - e^{-\lambda T}}. \end{aligned}$$

Implementing the EM steps now

1. **E-Step:** In the E -step, we find the expectation of the complete log likelihood under $Z_{1:m} \mid (E_{1:m}, Z_{(m+1):n})$. That is

$$\begin{aligned} q(\lambda \mid \lambda_{(k)}) &= \mathbb{E} \left[\log f(Z_1, \dots, Z_n \mid \lambda) \mid E_1, \dots, E_m, Z_{m+1}, \dots, Z_n \right] \\ &= n \log \lambda - \lambda \mathbb{E}_{\lambda_{(k)}} \left[\sum_{i=1}^n Z_i \mid E_1 = 1, \dots, E_m = 1, Z_{m+1}, \dots, Z_n \right] \\ &= n \log \lambda - \lambda \sum_{i=m+1}^n Z_i - \lambda \sum_{i=1}^m [\mathbb{E}[Z_i \mid Z_i \leq T]] \end{aligned}$$

2. **M-Step:** To implement the M-step:

$$\lambda_{(k+1)} = \arg \max_{\lambda} \left[n \log \lambda - \lambda \sum_{i=m+1}^n Z_i - \lambda \sum_{i=1}^m [\mathbb{E}[Z_i \mid Z_i \leq T]] \right]$$

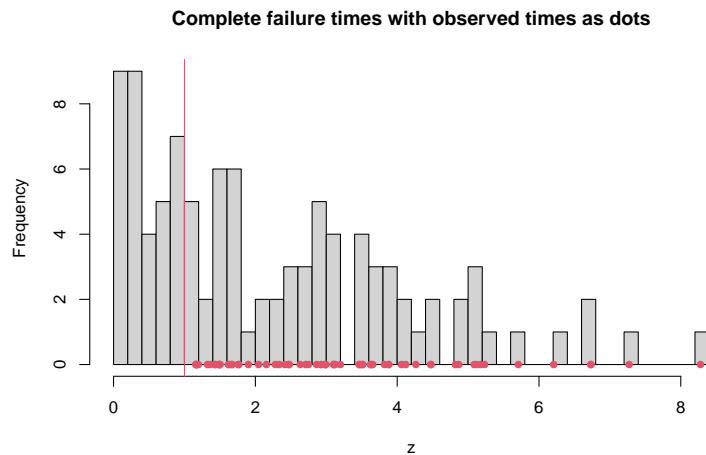
It is then easy to show that the M step makes the following update (show by yourself):

$$\lambda_{(k+1)} = \frac{n}{\sum_{i=m+1}^n Z_i + \sum_{i=1}^m [E[Z_i | Z_i \leq T]]} = \frac{n}{\sum_{i=m+1}^n Z_i + m \left[\frac{1}{\lambda_{(k)}} - \frac{T e^{-\lambda_{(k)} T}}{1 - e^{-\lambda_{(k)} T}} \right]}$$

```
#####
## EM Algorithm for the Censored exponential data
#####

# First we will simulate data
set.seed(10)
m <- 60
n <- 100
lam <- .45
z <- rexp(n, rate = lam)
T <- 1

# Observed
obs.z <- z[(z > T)]
hist(z, breaks = 30, main = "Complete failure times with observed times as
  dots")
points(obs.z,rep(0, length(obs.z)), col = 2, pch = 16)
abline(v = 1, col = 2)
```



```
diff <- 100
tol <- 1e-5
iter <- 0
current <- 1/mean(obs.z) # good starting values
```

```

lam.k <- current
store.z <- T
while(diff > tol)
{
  iter <- iter + 1
  # E-step
  Estep <- 1/current - T*exp(-current * T)/(1 - exp(-current * T))

  # Storing Zs
  store.z <- c(store.z, Estep)

  #Update
  update <- n/(sum(obs.z) + m*Estep)

  diff <- abs(current - update)
  current <- update
  lam.k <- c(lam.k, update)
}

current
# [1] 0.420317
(false.mle <- (n-m)/sum(obs.z)) # MLE if you ignore the Es (bad MLE)
#[1] 0.1904662

# Estimate of Z_i not observed (also called imputation)
(Z_unobs <- 1/current - T*exp(-current * T)/(1 - exp(-current * T)))
# [1] 0.4650763

```

1.2 Monte Carlo EM

Sometimes, the E-step is not analytically tractable, in which case the following step is not obtainable.

$$q(\theta|\theta_{(k)}) = E_{\theta_{(k)}} [\log f(x, z|\theta) | X]$$

Instead, we *estimate this expectation* using draws from $Z|X$ via Monte Carlo. That is, the new E-step is

$$\hat{q}(\theta|\theta_{(k)}) = \frac{1}{m} \sum_{t=1}^m \log(f(x, z^{(t)}|\theta)),$$

where $z^{(t)}$ are drawn from $Z|X$.

Example 1 (Censored Exponential). Although EM is possible in this example, we can choose to do MCEM. The original steps are:

1. **E-Step:**

$$q(\lambda \mid \lambda_{(k)}) = n \log \lambda - \lambda \sum_{i=m+1}^n Z_i - \lambda \sum_{i=1}^m \mathbb{E}_{\lambda_{(k)}}[Z_i \mid Z_i \leq T]$$

2. **M-Step:** It is then easy to show that the M step makes the following update:

$$\lambda_{(k+1)} = \frac{n}{\sum_{i=m+1}^n Z_i + m \sum_{i=1}^m \mathbb{E}_{\lambda_{(k)}}[Z_i \mid Z_i \leq T]}$$

Instead, we can estimate $\mathbb{E}_{\lambda_{(k)}}[Z_i \mid Z_i \leq T]$ using Monte Carlo:

1. **MCE-Step:** Draw $z^{(1)} = (z_1^{(1)}, \dots, z_m^{(1)}), \dots, z^{(K)}$ from $Z \mid Z \leq T$ (from a truncated exponential).

$$q(\lambda \mid \lambda_{(k)}) = n \log \lambda - \lambda \sum_{i=m+1}^n Z_i - \lambda \sum_{i=1}^m \left(\frac{1}{K} \sum_{t=1}^K z_i^{(t)} \right).$$

2. **M-Step:** It is then easy to show that the M step makes the following update:

$$\lambda_{(k+1)} = \frac{n}{\sum_{i=m+1}^n Z_i + \left(\frac{1}{K} \sum_{t=1}^K z_i^{(t)} \right)}.$$

```
#####
## Monte Carlo EM Algorithm for the
## Censored exponential data
#####
set.seed(1)
#function draws from truncated exponential
trexp <- function(nsim, T, lambda)
{
  count <- 1
  samp <- numeric(length = nsim)
  while(count < nsim)
  {
    draw <- rexp(1, rate = lambda)
    if(draw <= T)
    {
      samp[count] <- draw
      count <- count + 1
    } else{
      next
    }
  }
}
return(samp)
```

```

}

diff <- 100
tol <- 1e-5
iter <- 0
current <- 2
lam.k <- current
store.z <- T
mc.size <- 100 # Monte Carlo size
while(diff > tol)
{
  iter <- iter + 1

  # E-step with Monte Carlo

  trunExp.samples <- trexp(nsim = mc.size, T = T, lambda = current)
  Estep <- mean(trunExp.samples)
  #Estep <- 1/current - T*exp(-current * T)/(1 - exp(-current * T))

  # Storing Zs
  store.z <- c(store.z, Estep)

  #Update
  update <- n/(sum(obs.z) + m*Estep)

  diff <- abs(current - update)
  current <- update
  lam.k <- c(lam.k, update)
}

current
# [1] 0.4240852

```

2 Questions to think about

- In Monte Carlo EM, how can you improve the expectation step?
- What is an added benefit of doing EM rather than NR or Gradient Ascent on the observed likelihood?