hw6

Kun Zhou

March 15, 2016

1) Let Z be binomial random variable so that if $Z = 1$, deaths follows Poisson(μ_1) and if $Z = 2$, deaths follows Poisson(μ_2). Obviously, Z is the latent variable which we cannot observe. First we calculate the conditional distribution $(Z|X, \theta)$:

$$P(Z_i = 1|X = x_i, \theta_m) = \frac{\alpha_m \frac{\mu_{m,1} e^{-\mu_{m,1}}}{x_i!}}{\alpha_m \frac{\mu_{m,1} e^{-\mu_{m,1}}}{x_i!} + (1-\alpha_m) \frac{\mu_{m,2} e^{-\mu_{m,2}}}{x_i!}} = z_{x_i}(\theta_m)$$

$$P(Z_i = 2|X = x_i, \theta_m) = \frac{(1-\alpha_m) \frac{\mu_{m,2} e^{-\mu_{m,2}}}{x_i!}}{\alpha_m \frac{\mu_{m,1} e^{-\mu_{m,1}}}{x_i!} + (1-\alpha_m) \frac{\mu_{m,2} e^{-\mu_{m,2}}}{x_i!}} = 1 - z_{x_i}(\theta_m)$$

Now, we calculate the $Q(\theta|\theta_m)$.

$$Q(\theta|\theta_m) = \mathbb{E}_{Z|X,\theta_m}[\log L(\theta; X, Z)] = \sum_i^n \mathbb{E}_{Z|X,\theta_m}[\log L(\theta; x_i, z_i)]$$

$$= \sum_i^n [P(Z_i = 1|X = x_i, \theta_m) \log L(\theta; x_i, z_i) + P(Z_i = 2|X = x_i, \theta_m) \log L(\theta; x_i, z_i)]$$

$$= \sum_i^n \left[z_{x_i}(\theta_m) * \log(\alpha \frac{\mu_1^{x_i} e^{-\mu_1}}{x_i!}) + (1 - z_{x_i}(\theta_m)) * \log((1-\alpha) \frac{\mu_2^{x_i} e^{-\mu_2}}{x_i!}) \right]$$

$$= \sum_i^n [z_{x_i}(\theta_m) * (\log \alpha + x_i \log \mu_1 - \mu_1 - \log x_i!) + (1 - z_{x_i}(\theta_m)) * (\log(1-\alpha) + x_i \log \mu_2 - \mu_2 - \log x_i!)]$$

Calculate the maximum point θ_{m+1}.

$$\frac{\partial Q(\theta|\theta_m)}{\partial \mu_1} = \sum_i^n z_{x_i}(\theta_m) \left[\frac{x_i}{\mu_1} - 1 \right] = \sum_{i=0}^9 n_i z_i(\theta_m) \left[\frac{i}{\mu_1} - 1 \right] = 0$$

$$\Rightarrow \mu_{m+1,1} = \frac{\sum_{i=0}^9 z_i(\theta_m) i n_i}{\sum_{i=0}^9 z_i(\theta_m) n_i}$$

$$\frac{\partial Q(\theta|\theta_m)}{\partial \mu_2} = \sum_i^n (1 - z_{x_i}(\theta_m)) \left[\frac{x_i}{\mu_2} - 1 \right] = \sum_{i=0}^9 n_i (1 - z_i(\theta_m)) \left[\frac{i}{\mu_2} - 1 \right] = 0$$

$$\Rightarrow \mu_{m+1,2} = \frac{\sum_{i=0}^9 (1 - z_i(\theta_m)) i n_i}{\sum_{i=0}^9 (1 - z_i(\theta_m)) n_i}$$

$$\frac{\partial Q(\theta|\theta_m)}{\partial \alpha} = \sum_i^n \left[\frac{z_{x_i}(\theta_m)}{\alpha} - \frac{1 - z_{x_i}(\theta_m)}{1-\alpha} \right] = \sum_{i=0}^9 n_i \left[\frac{z_i(\theta_m)}{\alpha} - \frac{1 - z_i(\theta_m)}{1-\alpha} \right] = 0$$

$$\Rightarrow \alpha_{m+1} = \frac{\sum_{i=0}^9 n_i z_i(\theta_m)}{\sum_{i=0}^9 n_i}$$

Proof is finished.

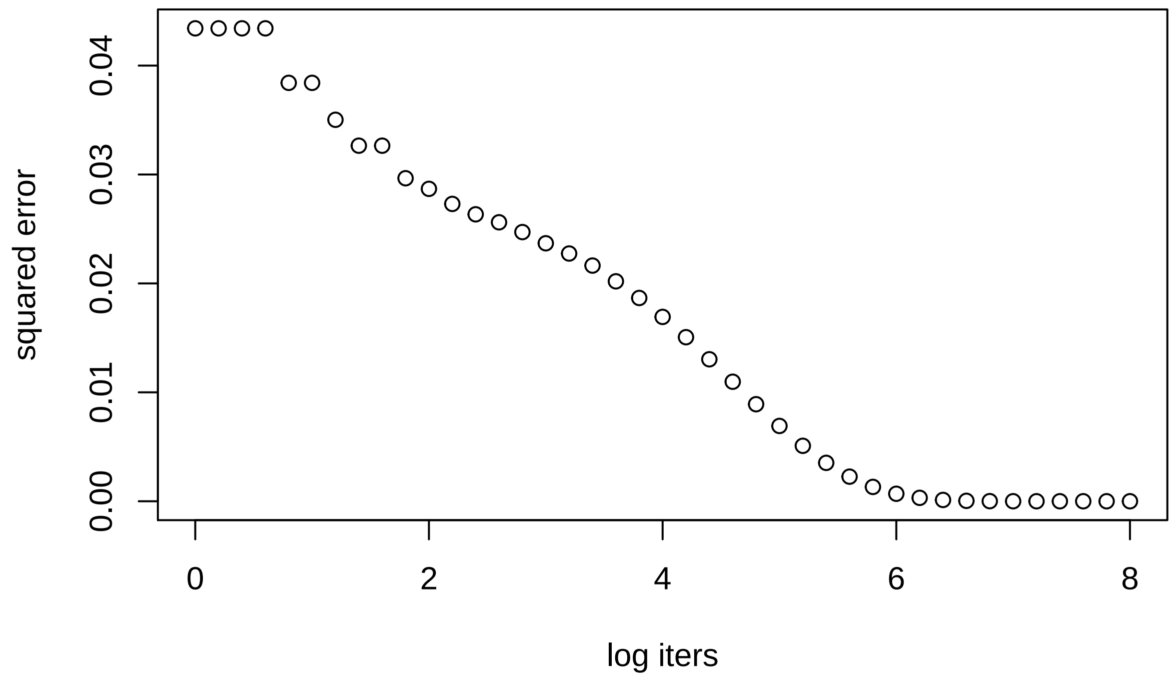
The following is calculation part.

```r
EM <- function(n, alpha, mu1, mu2, iter)
{
  i = 1:length(n) - 1
  alpha.m = alpha
  mu1.m = mu1
  mu2.m = mu2
  z.m = alpha.m * exp(-mu1.m) * mu1.m^i/(alpha.m * exp(-mu1.m) * mu1.m^i +
                                          (1-alpha.m) * exp(-mu2.m) * mu2.m^i)
  for(j in 1:iter)
  {
    alpha.m = sum(z.m*n) / sum(n)
    mu1.m = sum(n*i*z.m) / sum(n*z.m)
    mu2.m = sum(n*i*(1-z.m)) / sum(n*(1-z.m))
    z.m = alpha.m * exp(-mu1.m) * mu1.m^i/
      (alpha.m * exp(-mu1.m) * mu1.m^i + (1-alpha.m) * exp(-mu2.m) * mu2.m^i)
  }
  return(list(alpha=alpha.m, mu1=mu1.m, mu2=mu2.m))
}
n = c(162, 267, 271, 185, 111, 61, 27, 8, 3, 1)
EM(n, 0.3, 1, 2.5, 2000)
```

```
## $alpha
## [1] 0.359873
##
## $mu1
## [1] 1.256074
##
## $mu2
## [1] 2.663389
```

The above result is calculated by 2000 iterations. The next graph will show the iterations against squared error.

```r
a = seq(from = 0, to = 8, by = 0.2)
iters = floor(exp(a))
errors = c()
for(i in iters)
{
  result = EM(n, 0.3, 1, 2.5, i)
  err = (result[[1]] - 0.3599)^2 + (result[[2]] - 1.2561)^2
  + (result[[3]] - 2.6634)^2
  errors = c(errors, err)
}
plot(a, errors, xlab = "log iters", ylab = "squared error")
```



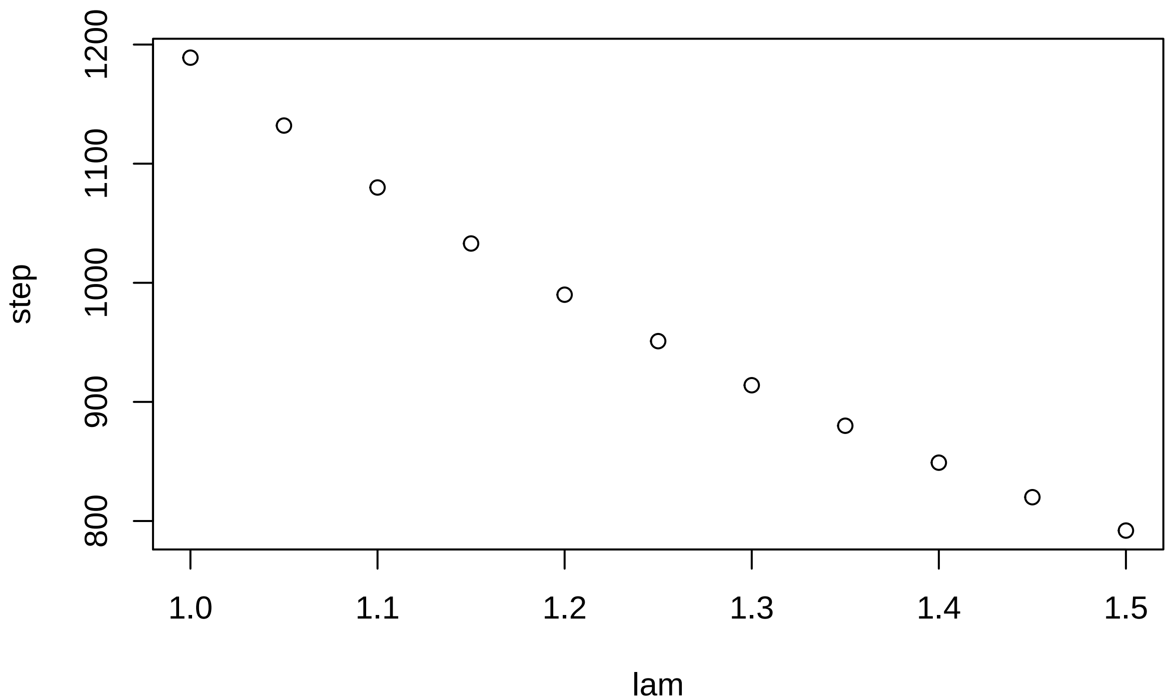
So after about 5 iterations, the EM converges in linear speed with iterations growing exponetially.

2)

```
EM.acc <- function(n, alpha, mu1, mu2, lambda)
{
  i = 1:length(n) - 1
  alpha.m = alpha
  mu1.m = mu1
  mu2.m = mu2
  z.m = alpha.m * exp(-mu1.m) * mu1.m^i/
    (alpha.m * exp(-mu1.m) * mu1.m^i + (1-alpha.m) * exp(-mu2.m) * mu2.m^i)
  step = 0
  err = 1
  while(err > 1e-6)
  {
    alpha.n = sum(z.m*n) / sum(n)
    mu1.n = sum(n*i*z.m) / sum(n*z.m)
    mu2.n = sum(n*i*(1-z.m)) / sum(n*(1-z.m))
    alpha.m = (alpha.n - alpha.m) * lambda + alpha.m
    mu1.m = (mu1.n - mu1.m) * lambda + mu1.m
    mu2.m = (mu2.n - mu2.m) * lambda + mu2.m
    z.m = alpha.m * exp(-mu1.m) * mu1.m^i/
      (alpha.m * exp(-mu1.m) * mu1.m^i + (1-alpha.m) * exp(-mu2.m) * mu2.m^i)
    err = (alpha.m - 0.3599)^2 + (mu1.m - 1.2561)^2 + (mu2.m - 2.6634)^2
    step = step + 1
```

```
  }
  return(list(alpha=alpha.m, mu1=mu1.m, mu2=mu2.m, step = step))
}

lam = seq(from = 1.5, to = 1, by = -0.05)
step = c()
for(i in lam)
  step = c(step,  EM.acc(n, 0.3, 1, 2.5, i)[[4]])
plot(lam, step)
```



The accuracy is set so that the squared error is below 1e-6. From the graph, when λ decreases to 1, the number of steps increases linearly. So accelerated EM improves convergence.