HOMEWORK 7

Gustavo Esparza

4/18/2019

Neccesarry Libraries

```
library(readr) #Import Data
library(MCMCpack) #Vectorize Sigma
```

A

The data set consists of data from a trivariate normal distribution with parameters μ and Σ. The data set also contains missing values from this distribution. Then, the EM algorithm can be used to find the MLE estimates in this missing data case.

Let us consider the complete dataset $x^T = (y_o^T, y_m^T)$ where y_o represents the observed data and y_m represents the missing data. Then, based on this data, we can partition our parameters μ and Σ in the following way.

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

Where o represents an observed value and m represents a missing value. Now, we can define the complete log-likelihood as

$l_c(\mu, \Sigma) = \frac{-n}{2}(plog(2\pi) + log(|\Sigma|)) - \frac{1}{2}\sum_{i=1}^{n}\left(x_i - \mu\right)^T \Sigma^{-1}(x_i - \mu)$

Then, we have the following EM algorithm:

E:

We wish to maximize $Q(\theta', \theta)$, which can be written as

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

where

$$\bar{x}^* = \frac{1}{n}\sum_{i=1}^{n} E^*(x_i)$$

$$E^*(x_i) = \begin{pmatrix} y_o \\ y_m^* \end{pmatrix}, y_m^* = \mu_m + \Sigma_{mo}\Sigma_{oo}^{-1}(y_o - \mu_o)$$

$$S^* = \frac{1}{n}\sum_{i=1}^{n} E^*(x_i x_i^T)$$

$$E^*(x_i x_i^T) = \begin{pmatrix} y_o y_o^T & y_o (y_m^*)^T \\ y_m^* y_o^T & E^*(y_m y_m^T) \end{pmatrix}, E^*(y_m y_m^T) = \Sigma_{mm} - \Sigma_{mo}\Sigma_{oo}^{-1}\Sigma_{om} + y_m^*(y_m^*)^T$$

M:

Now, we may use our E-step to solve for the paramter estimates of μ and Σ. Then, we have

$$\hat{\mu} = \bar{x}^*$$

and

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}\left(E^*(x_i x_i^T) - E^*(x_i)\mu^{'T} - \mu^{'} E^*(x_i) + \mu^{'}\mu^{'T}\right)$$

Then, we have the following steps:

1.) Start with initial paramters μ and Σ.
2.) E-step: Compute $\bar{x}^*$ and S^*
3.) M-step: Compute $\hat{\mu}$ and $\hat{\Sigma}$ using the values from our E-step.
4.) Replace our parameters with the values from step 3 and repeat until MLEs are estimated.

B

We will begin by imprting our data and deleting any rows of data that are completely missing:

```
x= read.table("/Users/gustavo/Desktop/Math 534/10/trivariatenormal.txt",header=T)

x=x[rowSums(is.na(x)) != ncol(x),]
```

Now, here is the E-M Algorithm for missing data:

```
EM= function(data,Mu,Sig,maxit)
{
  data=as.matrix(data)
  n=nrow(data)
  p=ncol(data)

  for(i in 1:maxit){

    sum.E.star = matrix(0,p,1)  #used for mu hat
    sum.E.star2 = matrix(0,p,p) #used for sigma hat

    for(j in 1:n){

      S=matrix(0,p,p) # used for S*

      mis = which(is.na(data[j,]))
      obs = which(!is.na(data[j,]))

      yobs = as.numeric(data[j,obs])
```

```
    #All Observed Data Case

    if(length(obs)==p){

      E.star=yobs

      S = yobs %*%t(yobs)

      sum.E.star= sum.E.star + E.star # gathering the sum

      S.star = S -Mu%*%t(E.star) -E.star%*%t(Mu) +Mu%*%t(Mu) #Expanded form of E(x,x)

      sum.E.star2 = sum.E.star2 + S.star # gathering the sum
    }

    #Missing Data Case

    if(length(obs)!=p){

      Mmu = Mu[mis]
      Omu = Mu[obs]

      OOsig = Sig[obs,obs]
      MMsig = Sig[mis,mis]
      MOsig = Sig[mis,obs]
      OMsig = Sig[obs,mis]

      Eymis = Mmu + (MOsig%*%solve(OOsig)%*%(yobs-Omu)) #missing y

      E.star = matrix(0,p,1)

      E.star[mis] = Eymis
      E.star[obs] = yobs

      sum.E.star = sum.E.star + E.star #gathering the sum

      yoyo = yobs%*%t(yobs)
      yoEym = yobs%*%t(Eymis)
      Eymyo = t(yoEym)
      Eymym = MMsig - (MOsig%*%solve(OOsig)%*%OMsig) + (Eymis%*%t(Eymis))

      S[obs,obs]= yoyo
      S[obs,mis]= yoEym
      S[mis,obs]= Eymyo
      S[mis,mis]= Eymym

      S.star = S - Mu%*%t(E.star) - E.star%*%t(Mu) + Mu%*%t(Mu) #Expanded form of E*(x,x)

      sum.E.star2 = sum.E.star2 + S.star #gathering the sum
      }
    }

  #Dividing sum by n to get estimates and defining them for the next iteration
```

```
    Mu = (sum.E.star)/n
    Sig = (sum.E.star2)/n
    Sigvec=vech(Sig)

    cat("Iteration = ",i,"\n")
    cat("Mean =  ",Mu,"\n")
    cat("Sigma = ",Sigvec,"\n")
    cat("\n")}
}
```

Here, we will use standard initial parameters of $\mu = (0,0,0)$ and $\Sigma = \text{I}$. Here is the result that our EM algorithm provides:

```
Mu=matrix(c(0,0,0),3,1)
Sig=diag(3)
maxit=20
EM(x,Mu,Sig,maxit)
```

```
Iteration =  1
Mean =   0.5964583 2.483542 7.686042
Sigma =  1.770244 2.1923 5.099367 7.91039 20.72931 71.57761

Iteration =  2
Mean =   0.7896385 2.843576 8.940618
Sigma =  1.387537 0.8971596 1.210184 1.263053 1.94324 8.296319

Iteration =  3
Mean =   0.8283592 2.837764 8.979068
Sigma =  1.316176 0.8821732 1.057707 0.8391231 0.871409 3.218823


.
.
.
.

Iteration =  19
Mean =   0.8786018 2.850156 9.025742
Sigma =  1.413128 1.001578 1.31847 0.7779349 0.7043278 2.522432

Iteration =  20
Mean =   0.8786014 2.850156 9.025744
Sigma =  1.41313 1.001579 1.318471 0.7779349 0.7043251 2.522438
```

Thus, our parameter estimates are:

$$\hat{\mu} = \begin{pmatrix} .879 \\ 2.85 \\ 9.03 \end{pmatrix}, \hat{\Sigma} = \begin{pmatrix} 1.41313 & 1.001579 & 1.318471 \\ 1.001579 & 0.7779349 & 0.7043251 \\ 1.318471 & 0.7043251 & 2.522438 \end{pmatrix}$$

Now, lets try starting parameters that are different than the MLE estimates found in the previous solution.

```
Mu=matrix(c(10,10,10),3,1)
Sig=7*diag(3)
maxit=20
EM(x,Mu,Sig,maxit)
```

```
Iteration =  1
Mean =   3.304792 3.733542 9.144375
Sigma =  64.38274 43.4173 7.168117 46.48956 5.858477 4.148444

Iteration =  2
Mean =   1.368786 2.807513 8.98696
Sigma =  13.60277 4.413932 1.958723 4.788881 1.050686 2.781469

Iteration =  3
Mean =   1.031928 2.816089 8.988587
Sigma =  4.25755 1.146527 1.438581 1.21386 0.7949335 2.633873

.
.
.

Iteration =  19
Mean =   0.8786038 2.850156 9.025735
Sigma =  1.413122 1.001575 1.318468 0.777935 0.7043383 2.52241

Iteration =  20
Mean =   0.8786024 2.850156 9.02574
Sigma =  1.413127 1.001578 1.31847 0.777935 0.7043301 2.522427
```

We can see that our EM algorithm still converges to equivalent MLE estimates despite the distant initial values. We can also note that the amount of iterations required is relatively small. This algorithm can also be extended to any p-variate normal set of data that has missing values, which extends the power of this algorithm.

Ultimately, this algorithm has shown it's strength by being able to estimate the MLE values for this set of data in spite of the missing values encountered. It's simplicity is also a factor in considering how useful this algorithm is.