534 Homework 5.3

Michael Pena

2024-04-02

```
# import the data
data <- as.matrix(read.table("trivariatenormal.txt",header = T))
```

part (a).

In class we found the Q-function in class to be

$$Q(\mu', \Sigma'|\vec{x}) = \frac{-3n}{2} log|2\pi| - \frac{n}{2} log||\Sigma'| - Tr[\Sigma'^{-1}(s^* - \bar{x}^*\mu'^T - \mu'\bar{x}^{*T} + \mu'\mu'^T)]$$

where $s^* = \frac{1}{n}\sum_{i=1}^{n} E^*[x_i x_i^T]$ and $\bar{x}^* = \frac{1}{n}\sum_{i=1}^{n} E^*[x_i]$

This implies that

$$dQ(d\mu') = -n \cdot Tr[\Sigma^{-1}(-\bar{x}^* + \mu')]$$

which gives

$$\frac{dQ}{d\mu_i} = -n[\Sigma^{-1}(-\bar{x}^* + \mu')]_i$$

This also implies that

$$dQ(d\Sigma') = \frac{-n}{2} \cdot Tr[\Sigma'^{-1} d\Sigma (I - \Sigma^{-1}(s^* - \bar{x}^*\mu'^T - \mu'\bar{x}^{*T} + \mu'\mu'^T))]$$

which gives

$$\frac{dQ}{d\Sigma'_{ij}} = \frac{-n}{2} \cdot [\Sigma'^{-1}(I - \Sigma^{-1}(s^* - \bar{x}^*\mu'^T - \mu'\bar{x}^{*T} + \mu'\mu'^T))]_{ij}$$

when $i = j$

$$\frac{dQ}{d\Sigma'_{ij}} = -n \cdot [\Sigma'^{-1}(I - \Sigma^{-1}(s^* - \bar{x}^*\mu'^T - \mu'\bar{x}^{*T} + \mu'\mu'^T))]_{ij}$$

when $i \neq j$

part (b).

```
# to.theta function
to.theta <- function(mu,Sig){
  theta = c(0)
  p = length(Sig[,1])
```

```
  q = p*(p+1)/2
  v = matrix(c(1,1,2,1,2,2,3,1,3,2,3,3),ncol = 2, nrow = 6, byrow = T)
  for(i in 1:p){theta[i] = mu[i]}
  for(i in 1:q){theta[p+i] = Sig[v[i,1],v[i,2]]}
  return(theta)
}
```

```
 # EM algorithm

 EMfunc <- function(y,mu,Sig,tolgrad){
  # initials
  p = length(Sig[,1])
  n = length(y[,1])
  th = to.theta(mu,Sig)
  ip = norm(th, type='2')
  it = 1
  # save iterations in here
  ITER = matrix(c(it,mu[1],mu[3],Sig[1,1],Sig[3,3],ip), nrow =1, ncol = 6, byrow = T)
    while(ip >= tolgrad){
      xbar.star = c(0,0,0)
      S.star = matrix(0,p,p)

      for(i in 1:n){

        # noting missing and observed data
        obs = which(!is.na(y[i,]))
        mis = which(is.na(y[i,]))

        # use that info to get mus, sigmas, and data
        mu_o = mu[obs]
        mu_m = mu[mis]
        Sig_oo = Sig[obs,obs]
        Sig_om = Sig[obs,mis]
        Sig_mo = Sig[mis,obs]
        Sig_mm = Sig[mis,mis]
        y_o = y[i,obs]
        y_m = y[i,mis]

        # initializing expectations for the xbar and S
        E.xi = c(0)
        E.S = matrix(0,p,p)

        # get mu tilde
        Estar.y_m = mu_m + (Sig_mo %*% solve(Sig_oo)) %*% (y_o - mu_o)
        E.xi[obs] = y_o
        E.xi[mis] = Estar.y_m
        xbar.star = xbar.star + E.xi/n
        mu.tilde = xbar.star

        # get sigma tilde
        E.S[obs,obs] = y_o %*% t(y_o)
        E.S[mis,obs] = Estar.y_m %*% t(y_o)
        E.S[obs,mis] = Estar.y_m %*% t(y_o)
        E.S[mis,mis] = Sig_mm - (Sig_mo %*% solve(Sig_oo) %*% Sig_om) + (Estar.y_m %*% t(Estar.y_m))
```

```
        S.star = S.star + E.S/n
        Sig.tilde = S.star - (xbar.star %*% t(xbar.star))
    }

    # finding the gradients
    del.mu = (-1)*n*solve(Sig.tilde) %*% (mu - mu.tilde)
    J = S.star - xbar.star%*%t(mu.tilde) - mu.tilde%*%t(xbar.star) + mu.tilde%*%t(mu.tilde)
    I = diag(3)
    del.Sig = (-n/2) * solve(Sig.tilde)%*%(I - solve(Sig.tilde)%*%J)
    del.theta = to.theta(del.mu,del.Sig)

    # get the innerproduct of del.theta
    ip <- norm(del.theta, type = '2')

    # plug back in for iteration
    mu.tilde -> mu
    Sig.tilde -> Sig
    it = it + 1

    # save into dataframe
    ITER <- rbind(ITER,c(it,mu[1],mu[3],Sig[1,1],Sig[3,3],ip))
    }
# print first three and last three rows of dataframe
u <- length(ITER[,1])
colnames(ITER) = c("Iteration","mu1","mu2","Sigma_11","Sigma_33","gradnorm")
print("first 3 iterations")
print(ITER[1:3,])
print("last 3 iterations")
print(ITER[(u-2):u,])
# print final mu and Sigma
return(list("mu estimator" = mu, "Sigma estimator" = Sig))
 }
```

```
# run the program
mu0 <- c(0,0,0)
Sigma0 <- diag(3)
EMfunc(data,mu0,Sigma0,1e-06)
```

```
## [1] "first 3 iterations"
##      Iteration       mu1      mu2 Sigma_11  Sigma_33  gradnorm
## [1,]         1 0.0000000 0.000000 1.000000  1.000000  1.732051
## [2,]         2 0.5964583 7.686042 1.414481 12.502374 59.246824
## [3,]         3 0.8006115 8.834786 1.353736  4.105603 21.800091
## [1] "last 3 iterations"
##      Iteration      mu1      mu2 Sigma_11 Sigma_33     gradnorm
## [1,]        30 0.878601 9.025745 1.413132 2.522443 2.291258e-06
## [2,]        31 0.878601 9.025745 1.413132 2.522443 1.108805e-06
## [3,]        32 0.878601 9.025745 1.413132 2.522443 5.360867e-07
```

```
## $`mu estimator`
## [1] 0.878601 2.850157 9.025745
##
## $`Sigma estimator`
##          [,1]      [,2]      [,3]
## [1,] 1.413132 1.0015800 1.3184716
```

```
## [2,] 1.001580 0.7779349 0.7043226
## [3,] 1.318472 0.7043226 2.5224430
```