STA Homework 6

Matt Johnson

September 20, 2013

1. The y_i data was simulated in R using the `rmvrnorm()` function in the `mvtnorm` package. In order to have marginal variances $\sigma^2 = 1$ and correlation $\rho = 0.8$, the covariance matrix must be $\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$.

```
> mu = c(0,0)
> Sigma = matrix(c(1,0.8,0.8,1),nrow=2)
> n = 100
> yi = rmvnorm(n, mu, Sigma)
```

2. For a sample $y_1, ..., y_n$, the maximum likelihood estimates of μ and Σ are

$$\hat{\mu}_{MLE} = \bar{y}$$

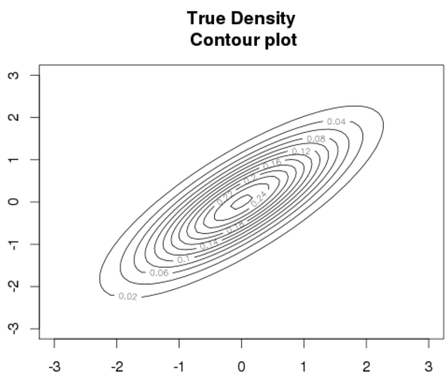
$$\hat{\Sigma}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T$$

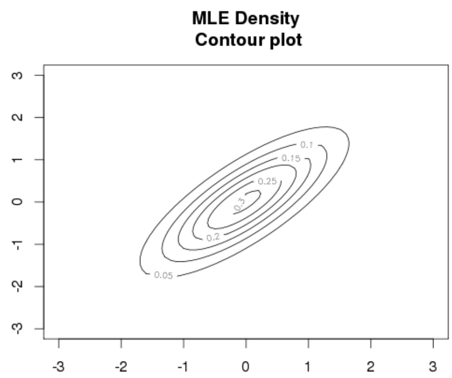
For this sample, those are approximately

$$\hat{\mu}_{MLE} = \begin{bmatrix} -0.005 \\ -0.026 \end{bmatrix}$$

$$\hat{\Sigma}_{MLE} = \begin{bmatrix} 0.86 & 0.63 \\ 0.63 & 0.76 \end{bmatrix} \Rightarrow \hat{\rho} = 0.78$$

```
> yBar = colMeans(yi)
> yBar
[1] -0.005178884 -0.025522028
> yBarMatrix = cbind(rep(yBar[1],n),rep(yBar[2],n))
> SigmaMLE = 1/n * t((yi-yBarMatrix)) %*% (yi-yBarMatrix)
> SigmaMLE
          [,1]      [,2]
[1,] 0.8643138 0.6344748
[2,] 0.6344748 0.7611970
```





3. We have no practical meaning for these variables, so prior parameters are somewhat arbitrary. In order to be "vaguely informative," I set most parameters close to their true values. The initial value μ_0 was set to be totally neutral. For Λ_0, I assumed marginal variances of 0.5, so that they would be smaller than the true variance of individual values, 1. Without any information about the data set, I set the implied correlation in Λ_0 to 0. In order to have vague beliefs about the variance matrix, I assumed $\nu_0 = p + 2 = 4$. For Σ_0, I assumed both marginal variances to be 1 and a positive correlation of $\rho = 0.5$. To sum up:

$$\mu_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \Lambda_0 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix},\ \nu_0 = 4,\ \Sigma_0 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

   A Gibbs sampler was run with 10,000 samples. Code was similar to that in the notes.

4. The Bayes estimate for the parameters were extremely close to the MLE estimates:

$$\hat{\mu}_{Bayes} = \begin{bmatrix} -0.003 \\ -0.024 \end{bmatrix} \qquad \hat{\mu}_{MLE} = \begin{bmatrix} -0.005 \\ -0.026 \end{bmatrix}$$

$$\hat{\Sigma}_{Bayes} = \begin{bmatrix} 0.87 & 0.64 \\ 0.64 & 0.77 \end{bmatrix} \Rightarrow \hat{\rho} = 0.78 \qquad \hat{\Sigma}_{MLE} = \begin{bmatrix} 0.86 & 0.63 \\ 0.63 & 0.76 \end{bmatrix} \Rightarrow \hat{\rho} = 0.78$$

```
> colMeans(MU)
[1] -0.003362901 -0.024415968
> colMeans(SIGMA)
[1] 0.8737492 0.6395737 0.6395737 0.7720473
```

   In both cases, the mean and correlation were predicted quite well, but the marginal variance estimates are not particularly close to the true values. These estimates depend heavily on the y_i data, so if the random sampling of y_i demonstrates smaller variance than the true distribution, the estimates are going to reflect the smaller variance.

```r
# Matt Johnson
# STA 601 Homework 6
# 9/20/2013

set.seed(0)
library(mvtnorm) #for dmvnorm

#############
# Problem 1 #
#############

mu = c(0,0)
SigmaTrue = matrix(c(1,0.8,0.8,1),nrow=2)
n = 100
yi = rmvnorm(n, mu, SigmaTrue)

#############
# Problem 2 #
#############

yBar = colMeans(yi)
yBar
yBarMatrix = cbind(rep(yBar[1],n),rep(yBar[2],n))
SigmaMLE = 1/n * t((yi-yBarMatrix)) %*% (yi-yBarMatrix)
SigmaMLE

x = seq(-3,3,0.1)
y = seq(-3,3,0.1)
mat = cbind(rep(x,each=length(y)),rep(y,length(x)))
z = dmvnorm(mat,mu,SigmaTrue)
matz = matrix(z,nrow=length(y))

zMLE = dmvnorm(mat,yBar,SigmaMLE)
matzMLE = matrix(zMLE,nrow=length(y))

par(mfrow=c(1,2))
contour(x,y,matz, main='True Density\n Contour plot')
contour(x,y,matzMLE, main='MLE Density\n Contour plot')

#############
# Problem 3 #
#############

rwish = function(n,nu0,S0){
  sS0 = chol(S0)
  S = array( dim=c( dim(S0),n ) )
  for(i in 1:n){
    Z = matrix(rnorm(nu0 * dim(S0)[1]), nu0, dim(S0)[1]) %*% sS0
    S[,,i] = t(Z) %*% Z
  }
  S[,,1:n]
}

mu0 = c(0,0)
L0 = matrix(c(0.5,0,0,0.5),nrow=2)
nu0 = 4
Sigma0 = matrix(c(1,0.5,0.5,1),nrow=2)

p=length(mu0)
S0 = (nu0-p-1)*Sigma0
Sigma = cov(yi)
MU=SIGMA=NULL
nSamples = 10000

for(s in 1:nSamples){
  #update mu
  Ln = solve(n*solve(Sigma)+solve(L0))
  mun = Ln %*%(n*solve(Sigma)%*%yBar+solve(L0)%*%mu0)
  mu = rmvnorm(1,mun,Ln)

  #update Sigma
  Sn = S0 + (t(yi)-c(mu)) %*% t(t(yi)-c(mu))
  Sigma = solve(rwish(1,nu0+n,solve(Sn)))

  #save results
  MU = rbind(MU,mu)
  SIGMA = rbind(SIGMA,c(Sigma))
}
colMeans(MU)
colMeans(SIGMA)
```