STA 360: Assignment 7

Michael Lin

March 17, 2015

7.3 (a) The posterior samples of θ and Σ are obtained using a Gibbs sampler based on the full conditionals derived in Hoff. Here are the full conditional for θ:

$$\theta | y_1, y_2, \Sigma \sim \text{MVN}(\mu_n, \Lambda_n)$$

where

$$\mu_n = (\Lambda_0^{-1} + n\Sigma^{-1})^{-1}(\Lambda_0^{-1}\mu_0 + n\Sigma^{-1}\bar{y})$$
$$\Lambda_n = (\Lambda_0^{-1} + n\Sigma^{-1})^{-1}$$

and the full conditional for Σ:

$$\Sigma | y_1, y_2, \theta \sim \text{IWish}(\nu_n, S_n^{-1})$$

where

$$\nu_n = \nu_0 + n$$
$$S_n = S_0 + \sum_i (y_i - \theta)(y_i - \theta)^T$$

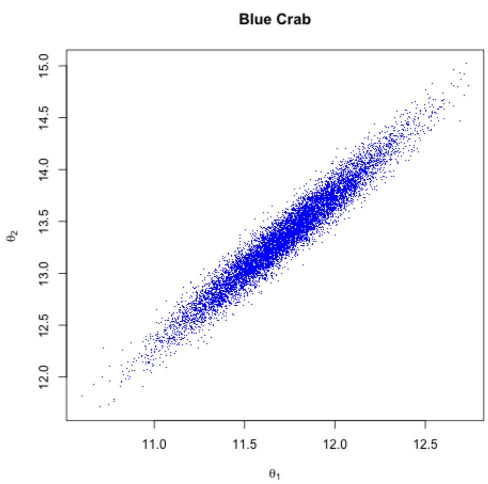
For the Gibbs sampler, we set:

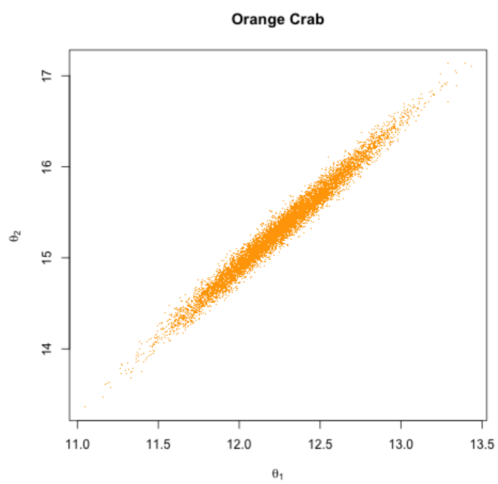
$$\mu_0 = \bar{y}$$
$$\Lambda_0 = S_0 = \Sigma_{y_1, y_2}$$

and we set the starting point:

$$\Sigma^{(0)} = S_0$$

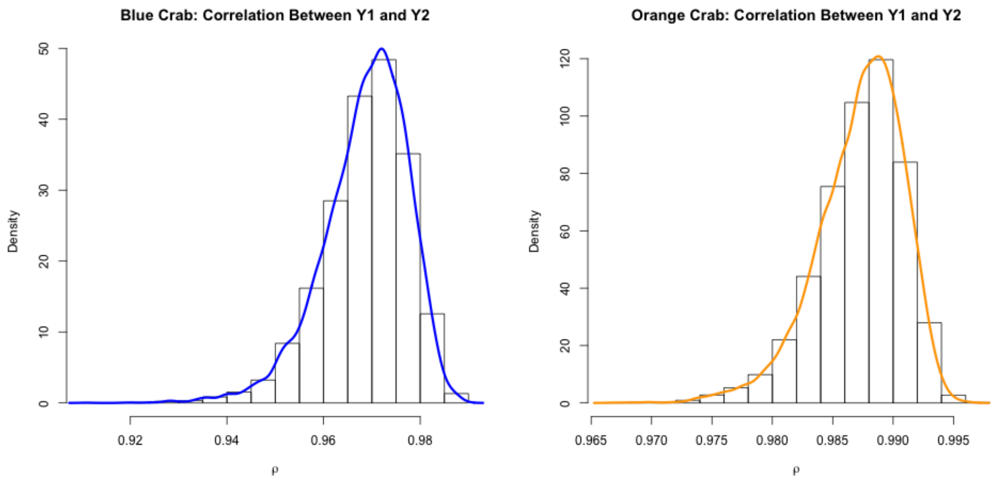
(b) Bivariate distribution plot of θ's for blue and orange crabs:





The plots suggest that while orange crabs on average have slightly larger body depth than blue crabs, orange crabs have significantly larger rear width than blue crabs.

(c) Posterior density plots of the correlations ρ_{blue} and ρ_{orange}:



A comparison of correlation coefficients from each update of the Gibbs sampler yielded approximately that:

$$\Pr(\rho_{\text{blue}} < \rho_{\text{orange}} | y_{\text{blue}}, y_{\text{orange}}) \approx 0.9908$$

The plots show that the distribution of correlations between body depth and rear width for orange crabs has a smaller range and higher mean. These results suggest that body depth and rear width are more correlated for orange crabs, and that orange crabs are more similar to one another than blue crabs.

7.5 (a) Below are empirical estimates based on the data:

$$\hat{\theta}_A = 24.200$$
$$\hat{\theta}_B = 24.805$$
$$\hat{\rho} = 0.616$$
$$\hat{\sigma}_A^2 = 4.093$$
$$\hat{\sigma}_B^2 = 4.692$$

(b) Using paired t-test on A and B responses, a t-value of -3.2807 was obtained with a p-value of 0.00177. As a result, we reject the null hypothesis and conclude that $\theta_A < \theta_B$. A 95% confidence interval for $\theta_A - \theta_B$ is [-0.9851, -0.2383].

(c) After implementing a Gibbs sampler with a unit information prior, a posterior mean for $\theta_A - \theta_B$ is -0.6730. A 95% confidence interval for $\theta_A - \theta_B$ is [-1.1210, -0.2250].

R code for 7.3:

```
##### Initialization ######
## Load Package ##
library(MASS)
library(MCMCpack)

## Load Data ##
bdata = read.table("bluecrab.dat")
odata = read.table("orangecrab.dat")

## Working Variables ##
n = 50
N = 10000


##### Blue Crab #####
## Blue Crab Posterior Sample ##
ybar = c(mean(bdata[,1]), mean(bdata[,2]))
mu0 = ybar
L0 = cov(bdata)

nun = 4+n
S0 = cov(bdata)

## Sampler ##
sigma.temp = S0
sigma.blue = list()
theta.blue = NULL
cor.blue = rep(NA,N)

for(i in 1:N){
  ## Update theta.blue ##
  Ln = solve(solve(L0) + n*solve(sigma.temp))
  mun = Ln%*%(solve(L0)%*%mu0 + n*solve(sigma.temp)%*%ybar)
  theta.temp = mvrnorm(1, mun, Ln)
  theta.blue = rbind(theta.blue, theta.temp)

  ## Update sigma.blue ##
  Sn = S0 + (t(bdata)-theta.temp)%*%t((t(bdata)-theta.temp))
  sigma.temp = solve(rwish(nun, solve(Sn)))
  sigma.blue[[i]] = sigma.temp

  ## Correlation Coefficient ##
  cor.blue[i] = sigma.temp[1,2]/sqrt(sigma.temp[1,1]*sigma.temp[2,2])
}


##### Orange Crab #####
## Orange Crab Posterior Sample ##
ybar = c(mean(odata[,1]), mean(odata[,2]))
mu0 = ybar
L0 = cov(odata)

nun = 4+n
```

```
S0 = cov(odata)

## Sampler ##
sigma.temp = S0
sigma.or = list()
theta.or = NULL
cor.or = NULL

for(i in 1:N){
  ## Update theta.or ##
  Ln = solve(solve(L0) + n*solve(sigma.temp))
  mun = Ln%*%(solve(L0)%*%mu0 + n*solve(sigma.temp)%*%ybar)
  theta.temp = mvrnorm(1, mun, Ln)
  theta.or = rbind(theta.or, theta.temp)

  ## Update sigma.or ##
  Sn = S0 + (t(odata)-theta.temp)%*%t((t(odata)-theta.temp))
  sigma.temp = solve(rwish(nun, solve(Sn)))
  sigma.or[[i]] = sigma.temp

  ## Correlation Coefficient ##
  cor.or[i] = sigma.temp[1,2]/sqrt(sigma.temp[1,1]*sigma.temp[2,2])
}

##### Plot #####
b = 1
png("bluetheta.png")
plot(theta.blue[b:10000,], pch = '.', col='blue', xlab = expression(theta[1]), ylab = expression(the
dev.off()

png("ortheta.png")
plot(theta.or[b:10000,], pch = '.', col='orange', xlab = expression(theta[1]), ylab = expression(the
dev.off()

##### Correlation Densities #####
## Blue ##
png("bluedist.png")
hist(cor.blue[b:10000], freq=F, xlab = expression(rho),
     main = 'Blue Crab: Correlation Between Y1 and Y2')
lines(density(cor.blue[b:10000]), col = 'blue', lwd = 3)
dev.off()

## Orange ##
png("ordist.png")
hist(cor.or[b:10000], freq=F,xlab = expression(rho),
     main = 'Orange Crab: Correlation Between Y1 and Y2')
lines(density(cor.or[b:10000]), col = 'orange', lwd = 3)
dev.off()

## Correlation Difference ##
print(mean(cor.blue < cor.or))
```

R code for 7.5:

```
##### Initialization ######
## Load Package ##
library(MASS)
library(MCMCpack)

## Load Data ##
data = read.table('interexp.dat',header=T)
Y = data

##### Imputation #####
## Empirical estimates ##
theta.A = mean(na.omit(data$yA))
theta.B = mean(na.omit(data$yB))
var.A = var(na.omit(data$yA))
var.B = var(na.omit(data$yB))
cor.AB = cor(na.omit(data))
cov.AB = cov(na.omit(data))

## Imputation Calculation ##
n = dim(data)[1]

for(i in 1:n){
  # Impute A response
  if(is.na(data[i,1])){
    data[i,1] = theta.A + (data[i,2] - theta.B)*cor.AB[1,2]*sqrt(var.A/var.B)
  }

  # Impute B response
  if(is.na(data[i,2])){
    data[i,2] = theta.B + (data[i,1] - theta.A)*cor.AB[1,2]*sqrt(var.B/var.A)
  }
}

## T-Test ##
data.test = t.test(data[,1], data[,2], paired=T)

##### Gibbs Sampler #####
## Prior Parameters ##
N = 1000
n = dim(Y)[1]
p = dim(Y)[2]
mu0 = c(theta.A, theta.B)
sd0 = c(sqrt(var.A), sqrt(var.B))
L0 = cov(na.omit(Y))
S0 = L0
nu0 = p + 2

## Starting values ##
sigma.temp = S0
Y.full = Y
O = 1*(!is.na(Y))
for(i in 1:n){
  if(is.na(Y.full[i,1])){
```

```
    Y.full[i,1] = theta.A
  }

  if(is.na(Y.full[i,2])){
    Y.full[i,2] = theta.B
  }
}

theta = NULL
sigma = list()

## Sampler ##
for(i in 1:N){
  ## update theta ##
  ybar = apply(Y.full, 2, mean)
  Ln = solve(solve(L0) + n*solve(sigma.temp))
  mun = Ln %*% (solve(L0)%*%mu0 + n*solve(sigma.temp)%*%ybar)
  theta.temp = mvrnorm(1,mun,Ln)
  theta = rbind(theta, theta.temp)

  ## update sigma ##
  Sn = S0 + (t(Y.full)-c(theta.temp))%*%t(t(Y.full)-c(theta.temp))
  sigma.temp = solve(rwish(nu0 + n, solve(Sn)))
  sigma[[i]] = sigma.temp

  ## update missing data ##
  for(j in 1:n){
    Y.full.temp = O[j,]
    j.prior = abs(cov.AB[1,2])^(-(p+2)/2)
    if(Y.full.temp[1]==0 & Y.full.temp[2]==1){
      mean.temp = theta.A + cov.AB[1,2]*(1/var.B)*(Y.full[j,2]-theta.B)
      sd.temp = sqrt(var.A - cov.AB[1,2]*(1/var.B)*cov.AB[1,2] + j.prior)
      Y.full[j,1]=rnorm(1, mean = mean.temp, sd = sd.temp)
    }

    if(Y.full.temp[1]==1 & Y.full.temp[2]==0){
      mean.temp = theta.B + cov.AB[1,2]*(1/var.A)*(Y.full[j,1]-theta.A)
      sd.temp = sqrt(var.B - cov.AB[1,2]*(1/var.A)*cov.AB[1,2] + j.prior)
      Y.full[j,2]=rnorm(1, mean = mean.temp, sd = sd.temp)
    }
  }
}

## Posterior Mean and Conf. Int. ##
Y.full.test = t.test(Y.full[,1], Y.full[,2], paired=T)

plot(data[,1],data[,2], col = "red", pch = 20)
points(Y.full[,1], Y.full[,2], col = "blue", pch=20)
points(data[,1][data==Y.full],data[,2][data==Y.full], col = "purple", pch = 20)
```