

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:

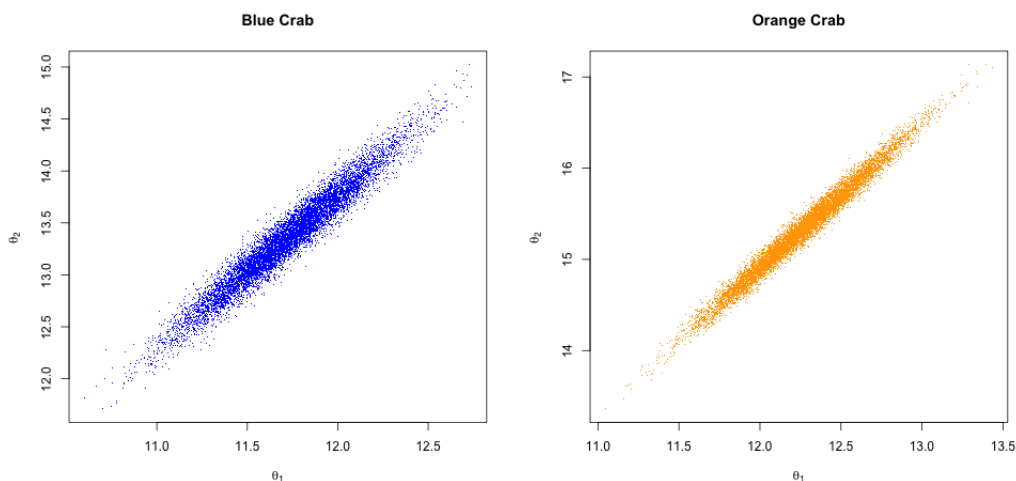
$$\mu_0 = \bar{y}$$

$$\Lambda_0 = S_0 = \Sigma_{y_1, y_2}$$

and we set the starting point:

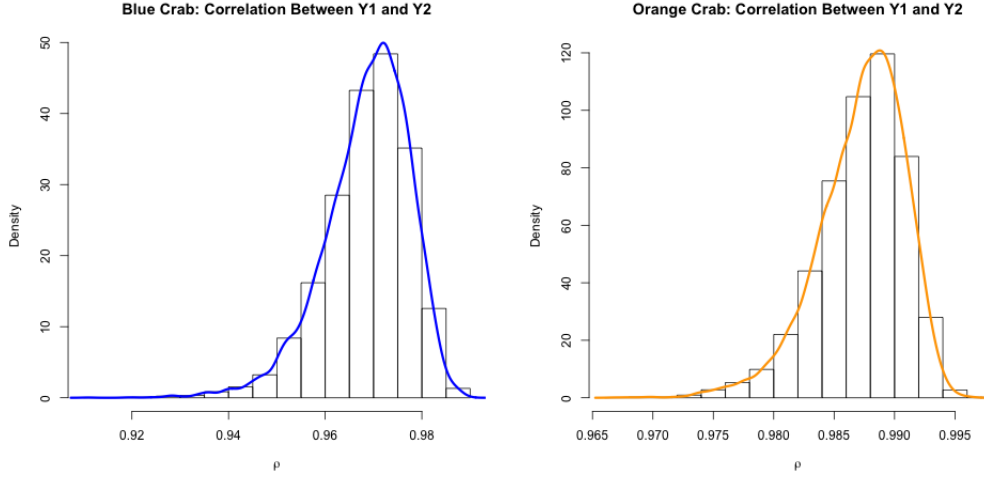
$$\Sigma^{(0)} = S_0$$

- (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:

```
1 ##### Initialization #####
2 ## Load Package ##
3 library(MASS)
4 library(MCMCpack)
5
6 ## Load Data ##
7 bdata = read.table("bluecrab.dat")
8 odata = read.table("orangecrab.dat")
9
10 ## Working Variables ##
11 n = 50
12 N = 10000
13
14
15 ##### Blue Crab #####
16 ## Blue Crab Posterior Sample ##
17 ybar = c(mean(bdata[,1]), mean(bdata[,2]))
18 mu0 = ybar
19 L0 = cov(bdata)
20
21 nun = 4+n
22 S0 = cov(bdata)
23
24 ## Sampler ##
25 sigma.temp = S0
26 sigma.blue = list()
27 theta.blue = NULL
28 cor.blue = rep(NA,N)
29
30 for(i in 1:N){
31   ## Update theta.blue ##
32   Ln = solve(solve(L0) + n*solve(sigma.temp))
33   mun = Ln%*(solve(L0)%*mu0 + n*solve(sigma.temp)%*ybar)
34   theta.temp = mvrnorm(1, mun, Ln)
35   theta.blue = rbind(theta.blue, theta.temp)
36
37   ## Update sigma.blue ##
38   Sn = S0 + (t(bdata)-theta.temp)%*t((t(bdata)-theta.temp))
39   sigma.temp = solve(rwish(nun, solve(Sn)))
40   sigma.blue[[i]] = sigma.temp
41
42   ## Correlation Coefficient ##
43   cor.blue[i] = sigma.temp[1,2]/sqrt(sigma.temp[1,1]*sigma.temp[2,2])
44 }
45
46
47 ##### Orange Crab #####
48 ## Orange Crab Posterior Sample ##
49 ybar = c(mean(odata[,1]), mean(odata[,2]))
50 mu0 = ybar
51 L0 = cov(odata)
52
53 nun = 4+n
```

```

54 S0 = cov(odata)
55
56 ## Sampler ##
57 sigma.temp = S0
58 sigma.or = list()
59 theta.or = NULL
60 cor.or = NULL
61
62 for(i in 1:N){
63   ## Update theta.or ##
64   Ln = solve(solve(L0) + n*solve(sigma.temp))
65   mun = Ln%*(solve(L0)%*mu0 + n*solve(sigma.temp)%*ybar)
66   theta.temp = mvrnorm(1, mun, Ln)
67   theta.or = rbind(theta.or, theta.temp)
68
69   ## Update sigma.or ##
70   Sn = S0 + (t(odata)-theta.temp)%*t((t(odata)-theta.temp))
71   sigma.temp = solve(rwish(nun, solve(Sn)))
72   sigma.or[[i]] = sigma.temp
73
74   ## Correlation Coefficient ##
75   cor.or[i] = sigma.temp[1,2]/sqrt(sigma.temp[1,1]*sigma.temp[2,2])
76 }
77
78 ##### Plot #####
79 b = 1
80 png("bluetheta.png")
81 plot(theta.blue[b:10000,], pch = '.', col='blue', xlab = expression(theta[1]), ylab = expression(theta[2]),
82 dev.off()
83
84 png("ortheta.png")
85 plot(theta.or[b:10000,], pch = '.', col='orange', xlab = expression(theta[1]), ylab = expression(theta[2]),
86 dev.off()
87
88 ##### Correlation Densities #####
89 ## Blue ##
90 png("bluedist.png")
91 hist(cor.blue[b:10000], freq=F, xlab = expression(rho),
92      main = 'Blue Crab: Correlation Between Y1 and Y2')
93 lines(density(cor.blue[b:10000]), col = 'blue', lwd = 3)
94 dev.off()
95
96 ## Orange ##
97 png("ordist.png")
98 hist(cor.or[b:10000], freq=F, xlab = expression(rho),
99      main = 'Orange Crab: Correlation Between Y1 and Y2')
100 lines(density(cor.or[b:10000]), col = 'orange', lwd = 3)
101 dev.off()
102
103 ## Correlation Difference ##
104 print(mean(cor.blue < cor.or))

```

R code for 7.5:

```
1 ##### Initialization #####
2 ## Load Package ##
3 library(MASS)
4 library(MCMCpack)
5
6 ## Load Data ##
7 data = read.table('interexp.dat',header=T)
8 Y = data
9
10 ##### Imputation #####
11 ## Empirical estimates ##
12 theta.A = mean(na.omit(data$yA))
13 theta.B = mean(na.omit(data$yB))
14 var.A = var(na.omit(data$yA))
15 var.B = var(na.omit(data$yB))
16 cor.AB = cor(na.omit(data))
17 cov.AB = cov(na.omit(data))
18
19 ## Imputation Calculation ##
20 n = dim(data)[1]
21
22 for(i in 1:n){
23   # Impute A response
24   if(is.na(data[i,1])){
25     data[i,1] = theta.A + (data[i,2] - theta.B)*cor.AB[1,2]*sqrt(var.A/var.B)
26   }
27
28   # Impute B response
29   if(is.na(data[i,2])){
30     data[i,2] = theta.B + (data[i,1] - theta.A)*cor.AB[1,2]*sqrt(var.B/var.A)
31   }
32 }
33
34 ## T-Test ##
35 data.test = t.test(data[,1], data[,2], paired=T)
36
37 ##### Gibbs Sampler #####
38 ## Prior Parameters ##
39 N = 1000
40 n = dim(Y)[1]
41 p = dim(Y)[2]
42 mu0 = c(theta.A, theta.B)
43 sd0 = c(sqrt(var.A), sqrt(var.B))
44 L0 = cov(na.omit(Y))
45 S0 = L0
46 nu0 = p + 2
47
48 ## Starting values ##
49 sigma.temp = S0
50 Y.full = Y
51 O = 1*(!is.na(Y))
52 for(i in 1:n){
53   if(is.na(Y.full[i,1])){
```

```

54     Y.full[i,1] = theta.A
55 }
56
57 if(is.na(Y.full[i,2])){
58     Y.full[i,2] = theta.B
59 }
60 }
61
62 theta = NULL
63 sigma = list()
64
65 ## Sampler ##
66 for(i in 1:N){
67     ## update theta ##
68     ybar = apply(Y.full, 2, mean)
69     Ln = solve(solve(L0) + n*solve(sigma.temp))
70     mun = Ln %%(solve(L0)%%mu0 + n*solve(sigma.temp)%%ybar)
71     theta.temp = mvrnorm(1,mun,Ln)
72     theta = rbind(theta, theta.temp)
73
74     ## update sigma ##
75     Sn = S0 + (t(Y.full)-c(theta.temp))%*%t(t(Y.full)-c(theta.temp))
76     sigma.temp = solve(rwish(nu0 + n, solve(Sn)))
77     sigma[[i]] = sigma.temp
78
79     ## update missing data ##
80     for(j in 1:n){
81         Y.full.temp = 0[j,]
82         if(Y.full.temp[1]==0 & Y.full.temp[2]==1){
83             mean.temp = theta.A + cov.AB[1,2]*(1/var.B)*(Y.full[j,2]-theta.B)
84             sd.temp = sqrt(var.A - cov.AB[1,2]*(1/var.B)*cov.AB[1,2])
85             Y.full[j,1]=rnorm(1, mean = mean.temp, sd = sd.temp)
86         }
87
88         if(Y.full.temp[1]==1 & Y.full.temp[2]==0){
89             mean.temp = theta.B + cov.AB[1,2]*(1/var.A)*(Y.full[j,1]-theta.A)
90             sd.temp = sqrt(var.B - cov.AB[1,2]*(1/var.A)*cov.AB[1,2])
91             Y.full[j,2]=rnorm(1, mean = mean.temp, sd = sd.temp)
92         }
93     }
94 }
95
96 ## Posterior Mean and Conf. Int. ##
97 Y.full.test = t.test(Y.full[,1], Y.full[,2], paired=T)
98
99 plot(data[,1],data[,2], col = "red", pch = 20)
100 points(Y.full[,1], Y.full[,2], col = "blue", pch=20)
101 points(data[,1][data==Y.full],data[,2][data==Y.full], col = "purple", pch = 20)

```