Module 9: The Multivariate Normal Distribution and Missing Data

Rebecca C. Steorts

Agenda

- Review of model
- Introduction to Pima Indian data set
- Approach to handling missing at random data
- Application to Pima Indian data set

Model set up

$$\boldsymbol{Y}_i \mid \boldsymbol{\theta}, \Sigma \sim MVN(\boldsymbol{\theta}_i, \Sigma).$$

$$\boldsymbol{\theta}_i \sim MVN(\boldsymbol{\mu_0}, \Lambda_0)$$

$$\Sigma \sim \text{inverseWishart}(\nu_o, S_o^{-1}).$$

Pima Indian heritage data

We consider a dataset involving health-related measurements on 200 women of Pima Indian heritage living near Phoenix, Arizona (Smith et al, 1988).

The four variables are `glu` (blood plasma glucose concentration), `bp` (diastolic blood pressure), `skin` (skin fold thickness) and `bmi` (body mass index).

Pima Indian data set

```
## Warning: package 'mvtnorm' was built under R version 3.4

## Warning: package 'MCMCpack' was built under R version 3.

## Loading required package: coda

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Qu

## ##
## ## Support provided by the U.S. National Science Foundat

## ## (Grants SES-0350646 and SES-0350613)
## ##
```

Pima Indian data set

```
## data with no missing values
data(Pima.tr)
Y0<-Pima.tr[,2:5]
Y<-Y0
n<-dim(Y)[1]
p<-dim(Y)[2]
head(Y)
```

```
##   glu bp skin  bmi
## 1  86 68   28 30.2
## 2 195 70   33 25.1
## 3  77 82   41 35.8
## 4 165 76   43 47.9
## 5 107 60   25 26.4
## 6  97 76   27 35.6
```

Pima Indian data set

```
## introduce missing values
set.seed(1)
O<-matrix(rbinom(n*p,1,.9),n,p)
## Make some of the Y's missing at random
Y[O==0]<-NA
```

Pima Indian data set

```
head(O)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    1    1    0
## [3,]    1    1    0    1
## [4,]    0    1    1    1
## [5,]    1    1    0    0
## [6,]    1    1    1    0
```

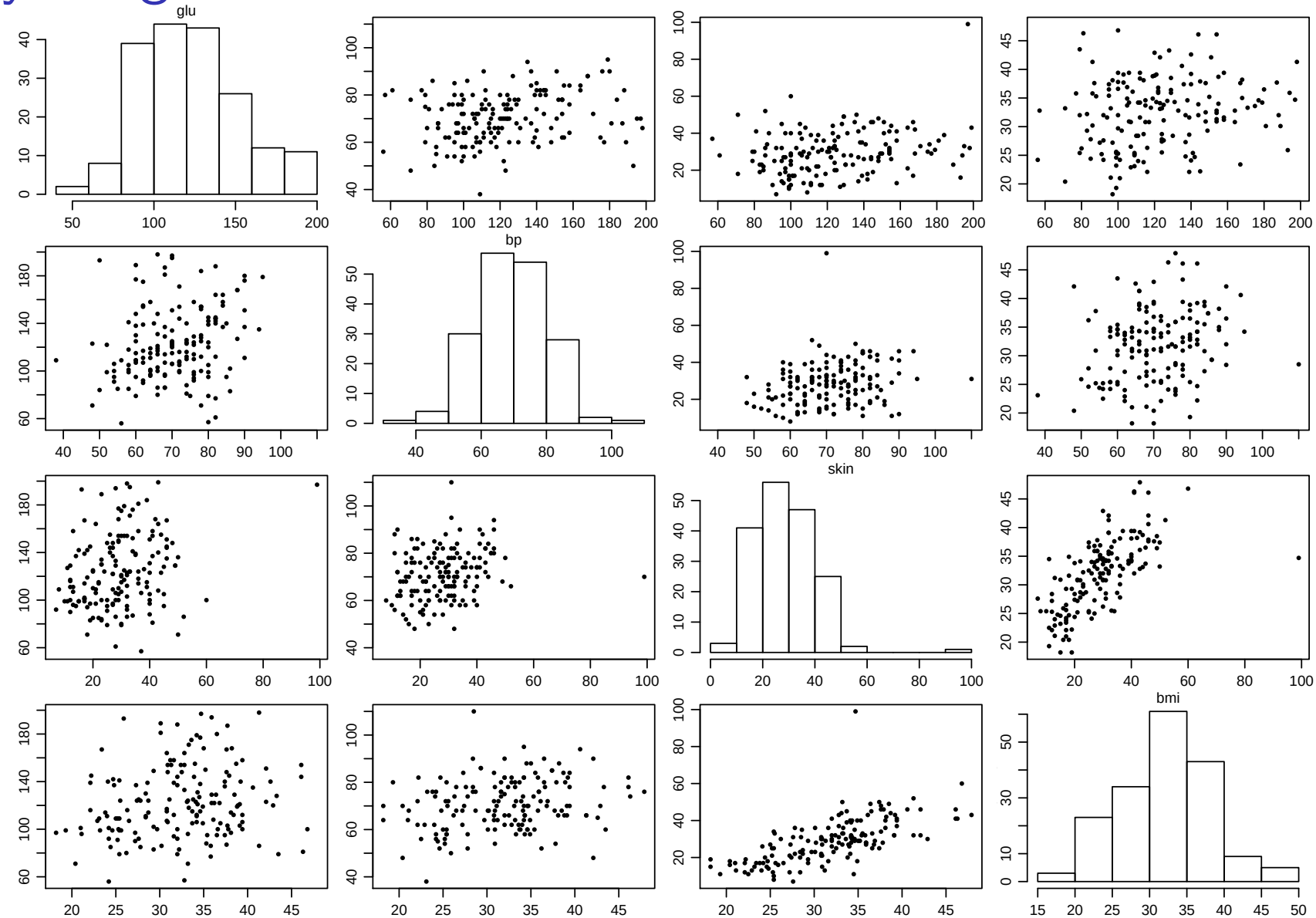
Missing data

- The NA's stand for "not available," and so some data for some individuals are "missing."
- Missing data are fairly common in survey data and other data sets.

```
head(Y)
```

```
##   glu bp skin  bmi
## 1  86 68   28 30.2
## 2 195 70   33   NA
## 3  77 82   NA 35.8
## 4  NA 76   43 47.9
## 5 107 60   NA   NA
## 6  97 76   27   NA
```

Physiological data on 200 women



Univariate histograms and bivariate scatterplots for four variables taken from the dataset.

Simple Approaches

1. Many software packages either throw away all subjects with incomplete data.

- We don't want to throw away data!

2. Others impute missing values with a population mean or some other fixed value, then proceed with the analysis.

- This approach is statistically incorrect, as it says we are certain about the values of the missing data when in fact we have not observed them.

Notation

Let $\boldsymbol{O}_i = (O_1, \ldots, O_p)^T$ be a binary vector of 0's and 1's.

Specifically, $O_{ij} = 1$ if Y_{ij} is observed and not missing.

$O_{ij} = 0$ if Y_{ij} is missing.

Our observed information about subject i is $\boldsymbol{O}_i = \boldsymbol{o}_i$ and $Y_{ij} = y_{ij}$ for variable j such that $o_{ij} = 1$.

Missing at Random

- We assume the data are missing at random, meaning that $\boldsymbol{O}_i$ and $\boldsymbol{Y}_i$ are independent and the distribution of $\boldsymbol{O}_i$ does not depend on θ or Σ.
- For modeling the data in a way that missing not a random, refer to Chapter 21 of Gelman et al (2004).

Missing at Random

The sampling probability for data for subject i is:

$$
\begin{aligned}
& p(\boldsymbol{o}_i, \{y_{ij} : o_{ij} = 1\} \mid \boldsymbol{\theta}, \Sigma) && (1)\\
& = p(\boldsymbol{o}_i) \times p(\{y_{ij} : o_{ij} = 1\} \mid \boldsymbol{\theta}, \Sigma) && (2)\\
& = p(\boldsymbol{o}_i) \times \int \left\{ p(\{y_{i,1}, \ldots, y_{ip} \mid \boldsymbol{\theta}, \Sigma) \prod_{y_{ij}:o_{ij}=0} dy_{ij} \right\} && (3)
\end{aligned}
$$

Missing at Random

Let's look at a special case so that the example is more concrete.

Let $\boldsymbol{y}_i = (y_{i1}, NA, y_{i3}, NA)^T$.

Then $o_i = (1, 0, 1, 0)^T$.

So, when data are missing at random we integrate over the missing data to obtain the marginal probability of the observed data:

$$p(\boldsymbol{o}_i, y_{i1}, y_{i3} \mid \boldsymbol{\theta}, \Sigma) = p(\boldsymbol{o}_i) p(y_{i1}, y_{i3} \mid \boldsymbol{\theta}, \Sigma) \tag{4}$$

$$= p(\boldsymbol{o}_i) \int p(\boldsymbol{y}_i \mid \boldsymbol{\theta}, \Sigma)\, dy_2 dy_4 \tag{5}$$

Notation

Let $\boldsymbol{Y}_{n\times p}$ be the matrix of all potential data in which $o_ij = 1$ if Y_ij$ is observed and $o_ij = 0$ if Y_ij$ is missing.

We can think of the matrix as consisting of two parts:

1. $$\boldsymbol{Y}_{obs} = \{y_{ij} : o_{ij} = 1\}$$
2. $$\boldsymbol{Y}_{miss} = \{y_{ij} : o_{ij} = 0\}$$

From the observed data, we want to obtain $p(\boldsymbol{\theta}, \Sigma, \boldsymbol{Y}_{miss} \mid \boldsymbol{Y}_{obs})$.

Gibbs sampler

Suppose the Gibbs sampler is at iteration s.

1. Sample $\boldsymbol{\theta}^{(s+1)}$ from it's full conditional:
   a) Compute μ_n and Σ_n from $\boldsymbol{Y}_{obs}$, $\boldsymbol{Y}_{miss}$ and $\Sigma^{(s)}$
   b) Sample $\boldsymbol{\theta}^{(s+1)} \sim MVN(\mu_n, \Sigma_n)$
2. Sample $\Sigma^{(s+1)}$ from its full conditional:
   a) Compute S_n from $\boldsymbol{Y}_{obs}$, $\boldsymbol{Y}_{miss}$, and $\theta^{(s+1)}$
   b) Sample $\Sigma^{(s+1)} \sim$ inverseWishart(ν_n, S_n^{-1})
3. Sample

$$\boldsymbol{Y}_{miss}^{s+1} \sim p(\boldsymbol{Y}_{miss} \mid \boldsymbol{Y}_{obs}, \theta^{(s+1)}, \Sigma^{(s+1)})$$

In steps 1–2, Note the fixed value of $\boldsymbol{Y}_obs$ combines with the current value of $\boldsymbol{Y}^(_{miss}s)$ to form a current version of a complete data matrix $\boldsymbol{Y}^(s)$ with no missing values.

Step 3 of Gibbs sampler

$$p(\boldsymbol{Y}_{miss} \mid \boldsymbol{Y}_{obs}, \boldsymbol{\theta}, \Sigma) \propto p(\boldsymbol{Y}_{miss}\boldsymbol{Y}_{obs} \mid \boldsymbol{\theta}, \Sigma) \tag{6}$$

$$= \prod_{i=1}^{n} p(\boldsymbol{y}_{i,miss}, \boldsymbol{y}_{i,obs} \mid \theta, \Sigma) \tag{7}$$

$$\propto \prod_{i=1}^{n} p(\boldsymbol{y}_{i,miss} \mid \boldsymbol{y}_{i,obs} \mid \theta, \Sigma) \tag{8}$$

We can compute the above quantity using a fact from multivariate methods.

Multivariate fact

Let $\boldsymbol{y}_{[b]}$ and $\boldsymbol{y}_{[a]}$ correspond to the elements of y corresponding to the indices of a and b. Let $\Sigma_{[a,b]}$ be a matrix with rows a and columns b.

Knowing infomation about partitioned matrices, one can show that

$$\boldsymbol{y}_{[b]} \mid \boldsymbol{y}_{[a]}, \boldsymbol{\theta}, \Sigma \sim MVN(\boldsymbol{\theta}_{b|a}, \Sigma_{b|a}),$$

where

$$\boldsymbol{\theta}_{b|a} = \boldsymbol{\theta}_{[b]} + \Sigma_{[b,a]}(\Sigma_{[a,a]})^{-1}(\boldsymbol{y}_{[a]} - \boldsymbol{\theta}_{[a]})$$

and

$$\Sigma_{[b,a]} = \Sigma_{[b,b]} - \Sigma_{[b,a]}(\Sigma_{[a,a]})^{-1}\Sigma_{[a,b]}.$$

Lab 9: Application to Pima Indian data set

0. Display univariate histograms and bivariate scatterplots for four variables taken from a dataset involving health-related measurements on 200 women of Pima Indian heritage living near Phoenix, Arizona.
1. Write down a generative model for the Pima Indian data set. Derive the full conditional distributions.
2. Code up the corresponding Gibbs sampler.
3. Specify hyper-parameters for your model and back up any assumptions that you give.
4. Provide diagnostic plots for all parameters in your model and provide commentary. (Note: do note use the word convegence in your commentary!)
5. Based on the number of S iterations of the Gibbs sampler you run, provide
   - 5.1 A posterior approximation of $E[\theta \mid y]$ and 95 posterior credible intervals.
   - 5.2 A posterior approximation of $E[\Sigma \mid y]$ and 95 posterior credible intervals.

Pima data set

We first talk about hyper-parameter selection and then implement the Gibbs sampler.

The full model can be written as the following:

$$\boldsymbol{Y}_i \mid \boldsymbol{\theta}, \Sigma \sim MVN(\boldsymbol{\theta}_i, \Sigma).$$

$$\boldsymbol{\theta}_i \sim MVN(\boldsymbol{\mu_0}, \Lambda_0)$$

$$\Sigma \sim \text{inverseWishart}(\nu_o, S_o^{-1}).$$

Hyper-parameter selection

The prior mean of $\boldsymbol{\mu_0} = (120, 64, 26, 26)^T$ is taken from national averages.

The corresponding prior variances are based primarily on keeping most of the prior mass on values that are above zero.

These prior distributions are likely much more diffuse than more informed prior distributions that could be provided by an expert in this field of study.

The data set

```
head(Y)
```

```
##   glu bp skin  bmi
## 1  86 68   28 30.2
## 2 195 70   33   NA
## 3  77 82   NA 35.8
## 4  NA 76   43 47.9
## 5 107 60   NA   NA
## 6  97 76   27   NA
```

```
n <- dim(Y)[1]
p <- dim(Y)[2]
```

Prior parameter specification

```
mu0 <- c(120,64,26,26)
(sd0 <- mu0/2)
```

```
## [1] 60 32 13 13
```

```
(L0 <- matrix(0.1, p,p))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  0.1  0.1  0.1  0.1
## [2,]  0.1  0.1  0.1  0.1
## [3,]  0.1  0.1  0.1  0.1
## [4,]  0.1  0.1  0.1  0.1
```

```
diag(L0) <- 1
L0 <- L0*outer(sd0,sd0)
nu0 <- p + 2
S0 <- L0
```

Starting values

```
Sigma <- S0
Y.full <- Y
# 1 for observed values
# 0 for NA's
O <- 1*(!is.na(Y))

# replace the NA values with #average of all the observed
# values in column j

for(j in 1:p){
Y.full[is.na(Y.full[,j]),j]<- mean(Y.full[,j],na.rm=TRUE)
}
```

Gibbs sampler

```
THETA<-SIGMA<-Y.MISS<-NULL
set.seed(1)
n.iter <- 1000
for(s in 1:n.iter) {
  ## update theta
  ybar <- apply(Y.full,2,mean)
  Ln <- solve(solve(L0) + n*solve(Sigma))
  mun <- Ln %*% (solve(L0) %*% mu0 + n*solve(Sigma) %*% ybar)
  theta <- rmvnorm(1, mun, Ln)

  ## update Sigma
  Sn <- S0 + (t(Y.full) - c(theta)) %*% t(t(Y.full)-c(theta))
  Sigma <- solve(rwish(nu0 + n, solve(Sn)))

  ###update missing data
  for(i in 1:n) {
    b <- (O[i,]==0)
    if (sum(b) > 0){
      a <- ( O[i,]==1 )
      iSa<- solve(Sigma[a,a])
      beta.j <- Sigma[b,a]%*%iSa
      s2.j   <- Sigma[b,b] - Sigma[b,a]%*%iSa%*%Sigma[a,b]
```

Posterior appx of $E[\theta \mid y]$

```
(theta_mean <- apply(THETA,2,mean))
```

```
## [1] 123.56644  71.08184  29.35342  32.17966
```

95 percent credible interval

```
library(knitr)
interval.theta <-  apply(THETA, 2, quantile, c(0.025, 0.975
kable(data.frame(interval.theta))
```

	X1	X2	X3	X4
2.5%	119.4802	69.40729	27.74316	31.29883
97.5%	127.9806	72.76456	30.96127	33.07434

Posterior appx of $E[\Sigma \mid y]$

```
COR <- array( dim=c(p,p,1000) )
for(s in 1:1000)
{
  Sig<-matrix( SIGMA[s,] ,nrow=p,ncol=p)
  COR[,,s] <- Sig/sqrt( outer( diag(Sig),diag(Sig) ) )
}



apply(COR,c(1,2),mean)
```

```
##             [,1]      [,2]      [,3]      [,4]
## [1,] 1.0000000 0.2230165 0.2505873 0.1914125
## [2,] 0.2230165 1.0000000 0.2501551 0.2388068
## [3,] 0.2505873 0.2501551 1.0000000 0.6491677
## [4,] 0.1914125 0.2388068 0.6491677 1.0000000
```

95 % confidence intervals for $E[\Sigma \mid y]$

```
interval.sigma <- apply(COR, c(1,2), quantile,prob=c(.025,
kable(data.frame(interval.sigma))
```

	X1.1	X2.1	X3.1	X4.1	X1.2	X2.2
2.5%	1	0.0617685	0.1038804	0.0444056	0.0617685	1
97.5%	1	0.3703619	0.3901463	0.3426900	0.3703619	1

95 percent posterior confidence intervals

```
library(sbgcop)
```

```
##
## Attaching package: 'sbgcop'

## The following object is masked from 'package:MCMCpack':
##
##     rwish
```

```
colnames(COR)<-rownames(COR)<-colnames(Y)
par(mfcol=c(4,2),mar=c(1,2.75,1,1),mgp=c(1.75,.75,0),oma=c(
plotci.sA(COR)
REG<-sR.sC(COR)
plotci.sA(REG)
```