STA 601: Homework 11

Akbota Anuarbek

Part I

Gibbs steps:

(1) impute $X_{mis,i}$ from its full conditional posterior

For that first calculate full conditional of $\tilde{X}_i = (X_{ij}, j = 2, ..., p) \sim N(\mu, \Sigma)$

$p(\tilde{X}_i|\mu, \Sigma, y, \beta, \sigma^2) \propto N(y_i; X_i'\beta, \sigma^2)N(\tilde{X}_i; \mu, \Sigma) \propto \exp[-\frac{1}{2}\sigma^{-2}((X_i^T\beta)^2 - 2X_i^T\beta y_i)]\exp[-\frac{1}{2}(\tilde{X}_i^T\Sigma^{-1}\tilde{X}_i - 2\tilde{X}_i^T\Sigma^{-1}\mu] \propto \exp[-\frac{1}{2}\sigma^{-2}(\tilde{X}_i^T\tilde{\beta}\tilde{\beta}^T\tilde{X}_i - 2\tilde{X}_i^T\tilde{\beta}(y_i + \beta_1))]\exp[-\frac{1}{2}(\tilde{X}_i^T\Sigma^{-1}\tilde{X}_i - 2\tilde{X}_i^T\Sigma^{-1}\mu] \propto \exp[-\frac{1}{2}[\tilde{X}_i^T(\Sigma^{-1} + \tilde{\beta}\tilde{\beta}^T\sigma^{-2})\tilde{X}_i - 2\tilde{X}_i^T(\tilde{\beta}(y_i + \beta_1)\sigma^{-2} + \Sigma^{-1}\mu)]] \propto N(M, V)$, where

$$V = (\Sigma^{-1} + \tilde{\beta}\tilde{\beta}^T\sigma^{-2})^{-2}$$
$$M = V(\tilde{\beta}(y_i + \beta_1)\sigma^{-2} + \Sigma^{-1}\mu)$$

Here $\tilde{\beta} = (\beta_j, j = 2, ..., p)$

Then *full conditional for $X_{mis,i}$* is:

$p(X_{mis,i}|X_{obs,i}, \mu, \Sigma, y, \beta, \sigma^2) \propto N(\bar{\mu}_i, \bar{\Sigma}_i)$, where

$$\bar{\mu}_i = \mu_{mis,i} + \Sigma_{mis,obs,i}\Sigma_{obs,i}^{-1}(X_{obs,i} - \mu_{obs,i})$$
$$\bar{\Sigma}_i = \Sigma_{mis,i} - \Sigma_{mis,obs,i}\Sigma_{obs,i}^{-1}\Sigma_{obs,mis,i}$$

Here $\mu_{mis,i} = (\mu_j$ s.t. X_j is missing for subject $i)$, $\Sigma_{mis,i} = (\Sigma_{j1,j2}$ s.t. both X_{j1} and X_{j2} are missing), $\Sigma_{mis,obs,i} = (\Sigma_{j1,j2}$ s.t. X_{j1} is missing, but X_{j2} is not), etc.

(2) Sample parameters given completed data from step 1

N-IG jointly conjugate prior:

$p(\beta, \sigma^2) = p(\beta|\sigma^2)p(\sigma^2) = N(\beta; m_\beta, \sigma^2 V_\beta)IG(\sigma^2; a, b)$

N-IG posterior:

$p(\beta, \sigma^2|X, y) = N(\beta; m_\beta^*, \sigma^2 V_\beta^*)IG(\sigma^2; a^*, b^*)$, where

$$
\begin{aligned}
&V_\beta^* = (V_\beta^{-1} + X^T X)^{-1} \\
&m_\beta^* = V_\beta^*(V_\beta^{-1} m_\beta + X^T y) \\
&a^* = a + \tfrac{n}{2} \\
&b^* = b + \tfrac{1}{2}[m_\beta^T V_\beta^{-1} m_\beta + y^T y - m_\beta^{*T} V_\beta^{*-1} m_\beta^*]
\end{aligned}
$$

To sample from $N - IG$ posterior, first, sample σ^2 from $IG(a^*, b^*)$, then using the value of σ^2 sample β from $N(m_\beta^*, \sigma^2 V_\beta^*)$

N-IW jointly conjugate prior:

$p(\mu, \Sigma) = p(\mu|\Sigma)p(\Sigma) = N(\mu; \mu_0, \Sigma/k_0)IW(\Sigma; \Lambda_0, v_0)$

N-IW posterior:

$p(\mu, \Sigma|X) = N(\mu; \mu_n, \Sigma/k_n)IW(\Sigma; \Lambda_n, v_n)$, where

$$
\begin{aligned}
&k_n = k_0 + n \\
&\mu_n = \tfrac{k_0 \mu_0 + n\bar{y}}{k_n}
\end{aligned}
$$

$v_n = v_0 + n$

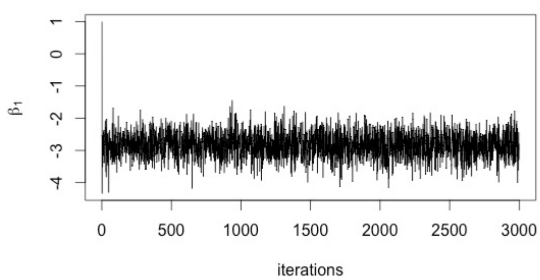
$\Lambda_n = \Lambda_0 + S + \frac{k_0 n}{k_0+n}(\bar{x} - \mu_0)(\bar{x} - \mu_0)^T$ s.t. $S = \sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$

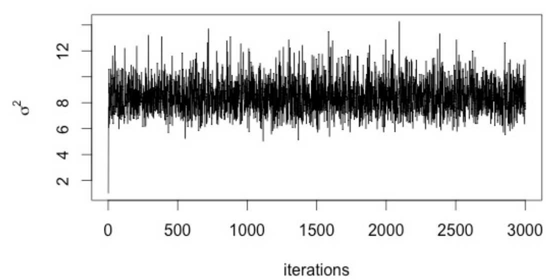
Sampling from $N - IW$ posterior is trivially done in the same way as sampling from $N - IG$.

Part II

For $p = 3$ and data from Sakai, implement Gibbs in R using above derived steps. R code is given at the end.

Check traceplots to see that sampler has converged, then burn-in first 1000 observations:





MLE estimates for β: -2.299, 0.760, 2.219

Bayesian estimates for β: -2.842, -0.282, 0.797

Bayesian 95% CIs for β: [-3.633 -2.101], [-0.743 0.200], [0.288 1.297]

Bayesian estimate for σ^2: 8.476

Bayesian 95% CI for σ^2: [6.231 11.310]

Part III

If we compare estimates of β obtained using MLE and Bayesian Approach, we ca see that estimates of the intercept are very close for two methods, however other β parameters are somewhat differ. The reason is that MLE method simply throws out missing observations, whereas Bayesian method uses two step Gibbs sampler with data imputation. Thus, results from Bayesian method should be more reliable, because it considers missingness and does not naivly throw out missing observations.

```
library(MCMCpack) #inv-wishart
library(pscl) #inv-gamma
library(mvtnorm)
library(optimbase)

setwd('/Users/akbota/Documents/STA601/HWs/hw11')
data = read.csv("data_set.csv", header = TRUE)

y=data$Y
n=length(y)
X=matrix(rep(1), nrow=n, ncol=3)
X[,2]=data$X.1
X[,3]=data$X.2

#MLE coeffs
beta.MLE = lm(y~X[,2:3])$coefficients



#Bayes Gibbs

T=3000

a=1
b=1
```

```
m.b=c(0,0,0)
m.b=transpose(m.b)
V.b=diag(3)

k0=1
v0=4
L0=diag(2)
mu0=c(0,0)

beta=matrix(ncol=T, nrow=3)
beta[,1]=c(1,1,1)
mu=matrix(ncol=T, nrow=2)
mu[,1]=c(1,1)
sig.sq=c(1)
Sigma=matrix(ncol=2*T, nrow=2)
Sigma[,1:2]=riwish(2,diag(2))

q<-rep(0,n)
for(i in 1:n){
  if(is.na(X[i,2])){
    q[i]=1
    if(is.na(X[i,3])){
      q[i]=3
    }
  }else if(is.na(X[i,3])){
    q[i]=2
  }
}
```

```
for(t in 2:T){

  #impute missing data

 V=solve(solve(Sigma[,(2*(t-1)-1):(2*(t-1))])+transpose(beta[2:3,t-1])%*%beta[2:3,t-1]/sig.sq[t-1])

  for(i in 1:n){

    if(q[i]!=0){
    M=V%*%(beta[2:3,t-1]*(y[i]+beta[1,t-1])/sig.sq[t-1]+solve(Sigma[,(2*(t-1)-1):(2*(t-1))])%*%mu[,t

      if(q[i]==1 ||q[i]==2){

     X[i,q[i]+1]=rnorm(1, mean=M[q[i]]+V[1,2]/V[q[i],q[i]]*(X[i,3-q[i]+1]-M[3-q[i]]), sd=sqrt(V[q[i]

      }else{
        X[i,2:3]=mvrnorm(n=1,M,V)
      }
    }
  }

  #update beta and sig.sq

  V.b.st=solve(solve(V.b)+transpose(X)%*%X)
  m.b.st=V.b.st%*%(solve(V.b)%*%m.b+transpose(X)%*%y)
  a.st=a+n/2
 b.st=b+0.5*(transpose(m.b)%*%solve(V.b)%*%m.b+y%*%transpose(y)-transpose(m.b.st)%*%solve(V.b.st)%*

  sig.sq[t]=rigamma(1,a.st,b.st)
```

```
  beta[,t]=mvrnorm(n=1,mu=m.b.st, Sigma=sig.sq[t]*V.b.st)

  #update mu and Sigma

  k.n=k0+n
  mu.n=(k0*mu0+sum(y))/k.n
  v.n=v0+n
  x.bar=c()
  x.bar[1]=mean(X[,2])
  x.bar[2]=mean(X[,3])
  S=sum((X[,2]-x.bar[1])^2)+sum((X[,3]-x.bar[2])^2)
  L.n=L0+S+k0*n/k.n*transpose(x.bar-mu0)%*%(x.bar-mu0)

  Sigma[,(2*t-1):(2*t)]=riwish(v.n, L.n)
  mu[,t]=mvrnorm(n=1, mu=mu.n, Sigma=Sigma[,(2*t-1):(2*t)]/k.n)

  if(t%%200==0){
    print(t)
  }

}

plot(beta[1,], type='l', xlab='iterations', ylab=expression(beta[1]))
plot(sig.sq, type='l', xlab='iterations', ylab=expression(sigma^2))

mean(beta[1,1001:T])
mean(beta[2,1001:T])
mean(beta[3,1001:T])
quantile(beta[1,1001:T], c(0.025, 0.975))
```

```
quantile(beta[2,1001:T], c(0.025, 0.975))
quantile(beta[3,1001:T], c(0.025, 0.975))
mean(sig.sq[1001:T])
quantile(sig.sq[1001:T], c(0.025, 0.975))
```