STA 601: Homework 7

Akbota Anuarbek

(1)

$y_i = B_0 + B_1 x_{i1} + B_2 x_{i2} + e_i$, $e_i \sim N(0, \sigma^2)$

Setting $B_0 = -1, B_1 = 1, B_2 = 1, \sigma^2 = 1$ and $\begin{pmatrix} x_{i1} \\ x_{i2} \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$, simulate data in R (for different values of ρ):

```
r=0
n<-100
err<-rnorm(n, mean=0, sd=1)
mu=c(0,0)
Sigma=matrix(c(1,r,r,1),2,2)
x2_x3<-mvrnorm(n=n,mu=mu,Sigma=Sigma)
x1<-c()
for(i in 1:n){
  x1[i]=1
}
x<-matrix(c(x1,x2_x3),n,3)
b<-c(-1,1,1)
y<-x%*%b+err
```

(2) <u>MLE</u>

Estimate $\hat{B} = (X'X)^{-1}X'Y$ using R. Results of estimations are summarizes in the table below. It is clear that as x_{i1} and x_{i2} become more and more correlated with each other, MLE gives very inaccurate estimates for betas, and have higher MSE.

ρ	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	MSE
0	-1.029216	1.090837	1.114210	0.842
0.99	-0.9323507	-39.9323529	41.8521128	1.36

```
b_mle<-solve(t(x)%*%x)%*%t(x)%*%y
mse_mle<-t(y-x%*%b_mle)%*%(y-x%*%b_mle)/100
```

(3)
(i) Fully Bayes

likelihood:
$p(y|X, \beta, \sigma^2) \propto (\sigma^2)^{-n/2} \exp(-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta))$

priors:
$\beta \sim N(0, \tau^{-1}\sigma^2 I_3)$
$\sigma^{-2} \sim Gamma(\frac{v_0}{2}, \frac{v_0\sigma_0^2}{2})$
$\tau \sim Gamma(0.5, 0.5)$ (Jeffrey's prior)

Now derive full conditional distributions and use Gibbs sampling.

$\beta \sim N(E, V)$, where
$V = (\Sigma_0^{-1} + X^T X/\sigma^2)^{-1}$
$E = (\Sigma_0^{-1} + X^T X/\sigma^2)^{-1}(X^T y/\sigma^2)$, $\Sigma_0 = \tau^{-1}\sigma^2 I_3$

$\sigma^{-2} \sim Gamma(\frac{1}{2}(v_0 + n), (v_0\sigma_0^2 + SSR(\beta))/2)$, $SSR(\beta) = (y - X\beta)^T(y - X\beta)$

$\tau \sim Gamma((1 + n)/2, \beta^T\beta\frac{1}{2\sigma^2} + 0.5)$

Results of estimations are summurized in the table below:

ρ	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	MSE
0	-0.5013178	0.4741322	0.6481512	1.75
0.99	-0.4708323	0.6019733	0.6048684	1.86

```
N<-1000
tau<-c(1)
sig_sq<-c(1)
B<-matrix(c(10,10,10), N, 3)
c=0.5
d=0.5
v0=1
sig0sq=1
```

```
for(i in 2:N){

  tau[i]=rgamma(1, shape=c+n/2, rate=1/(2*sig_sq[i-1])*(B[i-1,]%*%t(B[i-1,]))+d)

  SSR=t(y)%*%y-2*t(B[i-1,])%*%t(x)%*%y+t(B[i-1,])%*%t(x)%*%x%*%B[i-1,]
  f=rgamma(1, shape=(v0+n)/2, rate=(v0*sig0sq+SSR)/2)
  sig_sq[i]=1/f

  Sigma0<-(tau[i]^(-1))*sig_sq[i]*diag(3)
  V=solve(solve(Sigma0)+1/(sig_sq[i])*t(x)%*%x)
  E=solve(solve(Sigma0)+t(x)%*%x*1/(sig_sq[i]))%*%(t(x)%*%y*1/(sig_sq[i]))
  B[i,]=mvrnorm(n=1, mu=E, Sigma=V)
}

b_bayes<-c(mean(B[100:N,1]), mean(B[100:N,2]), mean(B[100:N,3]))
mse_bayes<-t(y-x%*%b_bayes)%*%(y-x%*%b_bayes)/100
```

(ii)Cross Validation

Split the data into two subsets: use 90 observations for training, and 10 for testing.
For different values of τ using Gibbs sampling calculate posterior mean of β and then MSE's. Choose tuning parameter τ that minimizes MSE. Results of the Cross Validation for different values of ρ (correlation between x_1 and x_2) are sumurized in the table below.

ρ	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	MSE
0	-0.7871652	0.8629834	1.0814191	1.56
0.99	-0.8962344	0.8910776	0.9148277	1.01

Overall, if we compare three methods of estimating parameters β, the MLE gives very good estimates with low MSE if there is no correlation between x_1 and x_2. But as ρ increases, estimates became highly innacurate and also with very high variance. Both other methods don't have this problem with high correlation. But results of Cross Validation are more closer to the true parameter values and have lower MSE's. Fully Bayes method gives fairly good estimates and doesn't have problems with high variance as MLE method has.

```
x_trng<-x[1:90,]
x_test<-x[91:100,]
y_trng<-y[1:90]
y_test<-y[91:100]
n=90
mse<-c()
post_mean<-matrix(c(0,0,0), 10, 3)
```

```
N<-1000
tau2<-c(1, 10, 25, 34, 40, 50, 63, 75, 83, 100)
sig_sq<-c(1)
B<-matrix(c(10,10,10), N, 3)
c=0.5
d=0.5
v0=1
sig0sq=1

for(j in 1:10){
  for(i in 2:N){
   SSR=t(y_trng)%*%y_trng-2*t(B[i-1,])%*%t(x_trng)%*%y_trng+t(B[i-1,])%*%t(x_trng)%*%x_trng%*%B[i-1,
    f=rgamma(1, shape=(v0+n)/2, rate=(v0*sig0sq+SSR)/2)
    sig_sq[i]=1/f

    Sigma0<-(tau2[j]^(-1))*sig_sq[i]*diag(3)
    V=solve(solve(Sigma0)+1/(sig_sq[i])*t(x_trng)%*%x_trng)
   E=solve(solve(Sigma0)+t(x_trng)%*%x_trng*1/(sig_sq[i]))%*%(t(x_trng)%*%y_trng*1/(sig_sq[i]))
    B[i,]=mvrnorm(n=1, mu=E, Sigma=V)
  }

  post_mean[j,]<-c(mean(B[100:N,1]), mean(B[100:N,2]), mean(B[100:N,3]))
  mse[j]=(t(y_test-x_test%*%post_mean[j,])%*%(y_test-x_test%*%post_mean[j,]))/10
}

index<-which.min(mse)
b_cv<-post_mean[index,]
mse_cv=mse[index]
```