HW09

Zach White

11/19/2016

Exercise 9.2

Part A

```
setwd("C:/Users/Zachary/Desktop/Fall 2016/STA 601/sta601/HW/HW09")
diab = read.table("azdiabetes.dat", header = TRUE)

# Fit a regression model using g-prior
g = n = nrow(diab)
nu.0 = 2
sigma2.0 = 1
nreps = 10000

y = diab$glu
X = as.matrix(diab[,-c(2,8)])
X = cbind(1,X)
p = ncol(X)

Hg = (g / (g+1)) * X %*% solve(t(X) %*% X) %*% t(X)
SSRg = t(y) %*% ( diag(1,n) - Hg ) %*% y

s2 = 1 / rgamma(nreps,(nu.0 + n)/2, (nu.0 *sigma2.0 + SSRg)/2 )

Vb =( g/(g+1)) * solve(t(X) %*% X)
Eb = Vb %*% t(X) %*% y

E = matrix(rnorm(nreps * p, 0 , sqrt(s2)),nreps,p)
beta = t( t(E %*% chol(Vb)) + c(Eb))
colnames(beta)[1] = "intercept"

apply(beta,2,quantile,c(.025,.975))
```

```
##       intercept      npreg          bp       skin      bmi       ped
## 2.5%   35.71369 -1.6170558 -0.02294221 -0.1123630 0.150554  3.129056
## 97.5%  69.61679  0.3109844  0.42697833  0.5013978 1.134208 17.769846
##             age
## 2.5%  0.4537319
## 97.5% 1.0792095
```

The above shows a 95% posterior credible intervals for β_i where $i = 1, ..., 7$. The first thing that I notice is that that the variables where the credible interval doesn't contain zero are intercept, bmi, ped, and age, which indicates that these might be the most influential variables.

Part B

```
g = n = nrow(diab)
nu.0 = 2
sigma2.0 = 1
nreps = 10000

y = diab$glu
X = as.matrix(diab[,-c(2,8)])
X = cbind(1,X)
p = ncol(X)

## Model selection and averaging procedure
lpy.X = function(y,X,g = length(y),nu.0 = 1, s20 = try(summary(lm(y~-1+X))$sigma^2, silent = TRUE) ){
  n = nrow(X)
  p = ncol(X)
  if(p == 0){
    Hg = 0
    s20 = mean(y^2)
  }
  if(p > 0){
    Hg = (g / (g+1)) * X %*% solve(t(X) %*% X) %*% t(X)
  }
  SSRg = t(y) %*% ( diag(1,n) - Hg ) %*% y

  -.5 * (n*log(pi) + p*log(1+g) + (nu.0 + n) * log(nu.0 * s20 + SSRg) - nu.0 * log(nu.0*s20)) + lgamma(
}

z = rep(1,p)
lpy.c = lpy.X(y,X[,z == 1, drop = FALSE])
nreps = 10000
Z = matrix(0,nreps,p)

beta = matrix(0,nreps,p)

## Gibbs step
for(i in 1:nreps){
  for(j in sample(1:p)){
    zp = z
    zp[j] = 1 - zp[j]
    lpy.p = lpy.X(y,X[,zp == 1, drop = FALSE])
    r = (lpy.p - lpy.c) * (-1)^(zp[j] == 0)
    z[j] = rbinom(1,1,1/(1+exp(-r)))
    if(z[j] == zp[j]){
      lpy.c = lpy.p
    }
  }
  Z[i,] = z
  Xz = X[,as.logical(Z[i,])]
  p = ncol(Xz)

  Hg = (g / (g+1)) * Xz %*% solve(t(Xz) %*% Xz) %*% t(Xz)
  SSRg = t(y) %*% ( diag(1,n) - Hg ) %*% y
```

```r
  s2 = 1 / rgamma(1,(nu.0 + n)/2, (nu.0 *sigma2.0 + SSRg)/2 )
  Vb = (g / (g+1)) * solve(t(Xz) %*% Xz)
  Eb = Vb %*% t(Xz) %*% y

  E = rnorm(p, 0, sqrt(s2))
  beta[i,Z[i,] == 1] = t( t(E %*%chol(Vb)) + c(Eb))
}


#Vb =( g/(g+1)) * solve(t(X) %*% X)
#Eb = Vb %*% t(X) %*% y

#E = matrix(rnorm(nreps * p, 0 , sqrt(s2)),nreps,p)
#beta = t( t(E %*% chol(Vb)) + c(Eb))
#colnames(beta)[1] = "intercept"
#sigma2.0

# Confidence intervals.  This should be shrunk thoug
apply(beta,2,quantile,c(.025,.975))
```

```
##           [,1]      [,2]      [,3] [,4]      [,5]     [,6]      [,7]
## 2.5%  45.85293 -1.067362 0.0000000    0 0.6072295 0.000000 0.5059104
## 97.5% 76.64059  0.000000 0.2907593    0 1.3863299 4.089266 1.0267223
```

```r
# Prob that var is included
apply(Z,2,mean)
```

```
## [1] 1.0000 0.1074 0.1384 0.0278 0.9944 0.0264 1.0000
```

The above shows the marginal inclusion probabilities, and it also shows the credible intervals for all of the coefficients, including the intercept.

```r
# HIghest probability models

# Now, I need to find the actual betas.
beta0.inc = beta[Z[,1] ==1,1]
beta1.inc = beta[Z[,2] ==1,2]
beta2.inc = beta[Z[,3] == 1, 3]
beta3.inc = beta[Z[,4] == 1, 4]
beta4.inc = beta[Z[,5] == 1, 5]
beta5.inc = beta[Z[,6] == 1, 6]
beta6.inc = beta[Z[,7] == 1, 7]

#Intercept
quantile(beta0.inc,c(.025,.975))
```

```
##     2.5%    97.5%
## 45.85293 76.64059
```

```r
# Npreg
quantile(beta1.inc,c(.025,.975))
```

```
##       2.5%      97.5%
## -1.7074517  0.3286216
```

```r
# Bp
quantile(beta2.inc,c(.025,.975))
```

```
##       2.5%      97.5%
## -0.0242549  0.4095864
```

```r
# Skin
quantile(beta3.inc,c(.025,.975))
```

```
##        2.5%       97.5%
## -0.07385796  0.68602534
```

```r
# bmi
quantile(beta4.inc,c(.025,.975))
```

```
##      2.5%     97.5%
## 0.6278962 1.3872336
```

```r
# ped
quantile(beta5.inc,c(.025,.975))
```

```
##     2.5%    97.5%
##  2.46268 17.32554
```

```r
# age
quantile(beta6.inc,c(.025,.975))
```

```
##      2.5%     97.5%
## 0.5059104 1.0267223
```

The above shows the values of $\beta_i \mid Z_i = 1$.

```r
# Highest Probability Models
Z.frame = as.data.frame(Z)
counts = Z.frame %>% count(V1,V2,V3,V4,V5,V6,V7)
counts$prob = counts$n / nreps
mod.prob = counts$n / nreps
counts[order(-counts$n),][1:5,]
```

```
## Source: local data frame [5 x 9]
## Groups: V1, V2, V3, V4, V5, V6 [5]
##
##      V1    V2    V3    V4    V5    V6    V7     n   prob
##   (dbl) (dbl) (dbl) (dbl) (dbl) (dbl) (dbl) (int)  (dbl)
## 1     1     0     0     0     1     0     1  7309 0.7309
## 2     1     0     1     0     1     0     1  1148 0.1148
## 3     1     1     0     0     1     0     1   883 0.0883
## 4     1     0     0     0     1     1     1   184 0.0184
## 5     1     0     0     1     1     0     1   165 0.0165
```

The above shows the highest probability models.

Exercise 2

In this exercise, I use a different prior described in George and McCullouch

```r
n = nrow(diab)
nreps = 10000
```

```r
diab = diab[,-8]
X = scale(diab)
y = scale(diab$glu)
X = as.matrix(X[,-c(2)])
p = ncol(X)
Xtx =   t(X)%*%X

# Hyperparameters
nu.0 = 2
sigma2.0 = 1
pz = rep(.5,p)
tau = rep(.03,p)
c = rep(.5/.03,p)
R = diag(p)
R.inv = solve(R)

ols = summary(lm(y~-1+X))
s2 = ols$sigma^2
beta = beta.ols = ols$coefficients[,1]
z = rep(1,p)

# matrices
BETA = Z = matrix(0,nreps,p)
S2 = matrix(0,nreps,1)

for(i in 1:nreps){
  # Sample beta
  Dg = diag((z*c+(1-z))*tau )
  Dg.inv = solve(Dg)
  Ag = solve((1/s2)*Xtx + Dg.inv %*% R.inv %*% Dg.inv)
  beta = rmvnorm(1,(1/s2)* Ag %*% Xtx %*% beta.ols,Ag)
  BETA[i,] = beta
  # Sigma2
  SSR = sum((y - X %*% t(beta))^2)
  s2 = 1/rgamma(1,(nu.0 + n) /2, (SSR + nu.0*sigma2.0)/2)
  S2[i,] = s2
  # Sample z
  for(j in sample(1:p)){
    zp = z
    zp[j] = 1
    Dgp = diag((zp*c+(1-zp))*tau)
    a = dmvnorm(beta,rep(0,p),Dgp %*% R %*% Dgp) * pz[j]
    zp[j] = 0
    Dgp = diag((zp*c+(1-zp))*tau)
    b = dmvnorm(beta,rep(0,p), Dgp %*%R %*% Dgp) * (1-pz[j])
    r = a / (a+b)
    z[j] = rbinom(1,1,r)
  }
  Z[i,] = z
}

# Confidence intervals.  This should be shrunk thoug
apply(BETA,2,quantile,c(.025,.975))
```

```
##                [,1]        [,2]        [,3]       [,4]        [,5]      [,6]
## 2.5%  -0.12794314 -0.02206125 -0.03558521 0.03444449 0.002783489 0.1637928
## 97.5%  0.03391324  0.13744013  0.15558495 0.27787006 0.190100205 0.3557436
```

```
# Prob that var is included
apply(Z,2,mean)
```

```
## [1] 0.1877 0.2397 0.2112 0.9405 0.6040 1.0000
```

The above shows the marginal inclusion probabilities and scaled and centered coefficients for the beta.

```
# Now, I need to find the actual betas.
BETA1.inc = BETA[Z[,1] ==1,1]
BETA2.inc = BETA[Z[,2] ==1,2]
BETA3.inc = BETA[Z[,3] == 1, 3]
BETA4.inc = BETA[Z[,4] == 1, 4]
BETA5.inc = BETA[Z[,5] == 1, 5]
BETA6.inc = BETA[Z[,6] == 1, 6]

# Npreg
quantile(BETA1.inc,c(.025,.975))
```

```
##        2.5%       97.5%
## -0.16539070  0.03688723
```

```
# Bp
quantile(BETA2.inc,c(.025,.975))
```

```
##         2.5%        97.5%
## -0.009994903  0.168852921
```

```
# Skin
quantile(BETA3.inc,c(.025,.975))
```

```
##        2.5%       97.5%
## -0.03197487  0.19725511
```

```
# bmi
quantile(BETA4.inc,c(.025,.975))
```

```
##       2.5%      97.5%
## 0.08790957 0.27966434
```

```
# ped
quantile(BETA5.inc,c(.025,.975))
```

```
##       2.5%      97.5%
## 0.03554358 0.19842839
```

```
# age
quantile(BETA6.inc,c(.025,.975))
```

```
##      2.5%     97.5%
## 0.1637928 0.3557436
```

The above shows the posterior credible intervals of the coefficients given that he variable is included in the model.

```
# Highest Probability Models
Z.frame = as.data.frame(Z)
counts = Z.frame %>% count(V1,V2,V3,V4,V5,V6)
```

```r
counts$prob = counts$n / nreps
mod.prob = counts$n / nreps
counts[order(-counts$n),][1:5,]
```

```
## Source: local data frame [5 x 8]
## Groups: V1, V2, V3, V4, V5 [5]
##
##      V1    V2    V3    V4    V5    V6     n   prob
##   (dbl) (dbl) (dbl) (dbl) (dbl) (dbl) (int)  (dbl)
## 1     0     0     0     1     1     1  2891 0.2891
## 2     0     0     0     1     0     1  1971 0.1971
## 3     0     1     0     1     1     1   876 0.0876
## 4     1     0     0     1     1     1   649 0.0649
## 5     0     0     1     1     1     1   597 0.0597
```

Note that the highest probability model includes bmi, ped, and age.

Note that this is very sensitive to τ and c. However, it is definitely most sensitive to τ.