assigment

abc

11/29/2019

## Question 1

### a)

Recall that the *Fisher information* is defined as:

In our regression model, which means we want to find the *Jeffreys prior* for the normal variance with known mean.

For known mean equal to 0 and unknown variance , we have log-likelihood

Then we can calculate the fisher information for is

Because and

so Then

So the *Jeffreys prior* is

### b)

Suppose that is full rank:

And the

The liklihood of ordinary normal linear model is

Also the MLE of is the solution of the least square problem and defined as

Unbiased estimator of is

Since the design matrix is known and fixed, the is constant, follow the process in question (a) we can directly conclude that the *g-prior* is

Because the is used in both prior and likelihood, then joint posterior distribution can be simplified into

Since

We know that

### c)

From the joint posterior , we obtain a Gaussian posterior on .

Since prior is inverse and likelihood is normal distribution, then posterior is inverse Gamma distribution. So we have an Inverse gamma posterior on

Since , we also replace by it’s posterior mean.

Which means

Hence , and

### d)

Since there is no precise prior information about and . Try and

For the problem of setting , we can find that if the influence of prior will be vanish and we recover the frequentist estimate of . Let takes the posterior to the prior distribution. Some other options for choosing include using BIC, empirical Bayes, and full Bayes.

Initialise , i.e. find starting values for . For

1. Draw from multinormal distribution based on and draw from Inverse Gamma distribution based on
2. Draw from multinormal distribution based on and draw from Inverse Gamma distribution based on
3. …
4. Draw from multinormal distribution based on and draw from Inverse Gamma distribution based on
5. Put and , set

Here we used as the posterior for the . And as the posterior for the

library("invgamma")  
library("mvtnorm")  
  
## input   
## response variable y  
## predictors data X  
  
gibbs = function(y, X){  
 beta = matrix(0, nrow=10, ncol = 1100)  
 sigma = rep(0, 1100)  
   
 sig = rep(0, 1100)  
 T = 100 # burn-in  
  
 n = dim(X)[1]  
 k = dim(X)[2]  
 g = 100  
 beta\_hat = solve(t(X) %\*% X) %\*% t(X) %\*% y  
 s = t(y - X %\*% beta\_hat) %\*% (y - X %\*% beta\_hat)  
 ## initialisation  
 sigma[1] = 1  
   
 for(i in 2:1000){  
 beta[,i] = rmvnorm(n=1, mean = g/(g+1) \* beta\_hat,   
 sigma = g\*sigma[i-1]^2/(g+1)\*solve(t(X)%\*%X))  
 rat = s^2/2 + (t(y-X%\*%beta\_hat)%\*%(y-X%\*%beta\_hat) -  
 t(beta[,i]-beta\_hat)%\*%t(X)%\*%X%\*%  
 (beta[,i]-beta\_hat))/2  
 sigma[i] = rinvgamma(n=1, shape = n/2, rate = rat)  
   
 }  
 par(mfrow=c(3,4))  
   
 # remove burn-in  
 beta = beta[,-(1:T)]  
 sigma = sigma[-(1:T)]  
   
 for(j in 1:10){  
 hist(beta[j,], xlab = paste0("simulated beta ", j),   
 main = paste("Histogram of beta" , j), nclass = 50)  
 }  
 hist(sigma, xlab = "simulated sigma", main = "Histogram of sigma",   
 nclass = 50)  
  
}

### e)

For this dataset, we have predictors with sample size . The regression model can be written like:

To find the posterior distribution of , we used the function defined in (d):

diabete = read.table("https://web.stanford.edu/~hastie/Papers/LARS/diabetes.sdata.txt", skip = 12)  
  
X = as.matrix(diabete[, 1:10])  
#ONE = rep(1,nrow(X))  
#X = cbind(ONE, X)  
  
y = as.matrix(diabete[, 11])  
  
gibbs(y, X)

## Question 2

### (i)

For the case in logistic regression, we have the model form:

### (ii)

library("statmod")

## Warning: package 'statmod' was built under R version 3.6.1

library("stats")  
## input:  
## target variable y  
## data matrix X  
  
bayeslasso = function(y, X, lambda){  
 y\_centered = scale(y)  
 ybar = mean(y)  
 n = dim(X)[1]  
 p = dim(X)[2]  
   
 tau2 = rep(0, p)  
 D = matrix(0, p, p)  
 lambda = rep(0, 1000)  
 sigma2 = rep(0, 1000)  
 beta = matrix(0, nrow = p, ncol = 1000)  
   
 ## initial  
 r = 1  
 sigma2[1] = 1.78  
 beta[,1] = rep(1, p)  
   
 for(j in 2:1000){  
   
 for(i in 1:p){  
 tau2[i] = rinvgauss(1, sqrt(lambda^2 \* sigma2[j-1] / beta[i, j-1]),   
 lambda^2)^(-1)  
 }  
   
 tau2[is.na(tau2)] = 10^(-10)  
 diag(D) = tau2  
 A = t(X)%\*%X + solve(D)  
 beta[,j] = rmvnorm(n=1, mean = solve(A)%\*%t(X)%\*%y\_centered, sigma = sigma2[j-1] \* solve(A))  
 sigma2[j] = rinvgamma(n=1, shape=(n-1)/2+p/2, rate=t(y\_centered-X%\*%beta[,j])%\*%(y\_centered-X%\*%beta[,j])/2+t(beta[,j])%\*%solve(D)%\*%beta[,j]/2)  
 # Using Hyperpriors for the Lasso Parameter lambda  
 #lambda[j] = sqrt(rgamma(n = 1, shape = p+r, rate = sum(tau2 / 2) + sqrt(sigma2[j])))  
 }  
 return(list(beta, sigma2, lambda))  
}