Bayesian Inference for Generalized Linear Model with Linear Inequality Constraints

Rahul Ghosal and Sujit Ghosh

07/17/2021

Introduction

This document presents the illustation of the BICGLM method proposed in Ghosal et al. (2021). We first consider the Heady study and illustrate the BICLS method.

```
#####Load the data############
set.seed(1)
library(agridat)
data(heady.fertilizer)
dat <- heady.fertilizer
d1 <- subset(dat, crop=="corn")     # considering corn yield as the response
mydata<-d1[-which(is.na(d1$yield)),]
head(mydata)
```

```
##   crop rep  P K N yield
## 1 corn   1  0 0 0  24.5
## 2 corn   2  0 0 0   6.2
## 3 corn   1 40 0 0  26.7
## 4 corn   2 40 0 0  29.6
## 5 corn   1 80 0 0  22.1
## 6 corn   2 80 0 0  30.6
```

BICLS Method: Heady Application

We setup the Gibbs sampler for the BICLS method below.

```
# @ y = response
# @ X = covariate matrix
# @ R = constraint matrix
# @ b = constrain bound b s.t R\beta>= b
# @ m1= Number of equalities, should be in top rows of R
# @ a0,b0 = paramter for inverse gamma distribution
# @ n.samples= Number of MCMC samples from the posterior
Bayes.con.slm<-function(y,X,R,b,m1,
                        a0=0.01,b0=0.01,
                        n.samples=5000){
  n<-length(y)
```

```
  p<-ncol(X)
  m<-length(b)
  Y<-y


  #intial values:
  ols     <- lm(Y~-1+X)
  sigma2  <- var(ols$residuals)

  #Initialize matrix to store the results:
  ef=p
  samples <- matrix(0,n.samples,p+1)
  mu1=coef(ols)
  Sigma1=vcov(ols)


  #needed things for beta
  Sigma1inv<-chol2inv(chol(Sigma1))
  xtx<-t(X)%*%(X)
  xty<-t(X)%*%(Y)

  ###initial value satisfying the constraint for Heady Application
  intsp<-rep(1,p)
  #Start the MCMC sampler:
  for(i in 1:n.samples){
    #update beta:     #assuming mu1=rep(0,p), Sigma1=100*diag(p)
    library(tmvmixnorm)
    term1<-xty*(1/sigma2)
    cov1<-chol2inv(chol((xtx/sigma2)+Sigma1inv))
    mun1<-cov1%*%(Sigma1inv%*%mu1+term1)
    cov1act<-cov1
    betapost<-rtmvn(n=1, Mean=mun1, cov1act, D=R, lower=b,
                    upper=rep(Inf,m),int=intsp, burn=10)
    #update sigma^2:
    SSE    <- sum((Y-X%*%betapost)^2)
    sigma2 <- 1/rgamma(1,n/2+a0,SSE/2+b0)


    #store results:
    samples[i,]  <- c(betapost,sigma2)
    }

  #return a list with the posterior samples:
  return(samples)}
```

We define the response and predictor variables and set up the constraint matrix.

```
y<-mydata$yield   #Response
x1<-(mydata$N)
x2<-(mydata$P)
x3<-sqrt(mydata$N)
x4<-sqrt(mydata$P)
x5<-sqrt((mydata$N)*(mydata$P))
```

```r
X<-cbind(rep(1,length(y)),x1,x2,x3,x4,x5)
R<-cbind(matrix(0,nrow=3,ncol = 3),diag(3)) #Constraint matrix
b<-rep(0,3)
R
```

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

We use the BICLS method to obtain posterior samples.

```r
n.samples=15000
burn=5000
samples<-Bayes.con.slm(y,X,R,b,m1=0,a0=0.01,b0=0.01,n.samples=15000)
m<-length(b)
p<-ncol(X)
betapost<-samples[burn:n.samples,(1):(p)] #Posterior Samples
betabayes<-colMeans(betapost)   #Bayes estimate under S.E.L
betasdbayes<-apply(betapost,2,sd) #standard deviation
betabayes
```

```
## [1] -5.7243585 -0.3160944 -0.4171974  6.3597210  8.5162357  0.3405918
```

```r
betasdbayes
```

```
## [1] 4.63049300 0.03350858 0.03251851 0.61645924 0.62006402 0.02737419
```

BICGLM Method: SCRAM Rate Modelling

We now illustrate the BICGLM method for SCRAM rate modelling. We consider the year-specific model (17) in the paper.

```r
#####Load the data and fit a GLM for modelling nonzero scrams###########
set.seed(1)
mydata=read.table("Scram-NEC-data.txt",header=T)
n.scram=as.numeric(mydata$n.scram)
year=as.factor(mydata$year)
plant=as.factor(mydata$plant)
c.time=as.numeric(mydata$c.time/7000)
scram.data=data.frame(n.scram,c.time,year,plant)
head(scram.data)
```

```
##   n.scram    c.time year plant
## 1       0 0.9986000    1    33
## 2       0 0.1158000    1    35
## 3       0 1.2548571    1    39
## 4       0 0.9171571    1    43
## 5       0 1.0777429    1    44
## 6       0 1.1205714    1    46
```

```
nzero.indx=which(n.scram!=0)
plant<-factor(mydata$plant)
###GLM####
myfit1=glm(n.scram~ year+plant,offset=log(c.time),subset=nzero.indx,
family="poisson")
#summary(myfit1)
#######Extract Model matrix X and response y#################
X<-model.matrix(myfit1)
y<-n.scram[nzero.indx]
```

We setup the Slice sampler for using the BICGLM method below.

```
# @ y = response
# @ X = covariate matrix
# @ R = constraint matrix
# @ b = constrain bound b s.t R\beta>= b
# @ delta= offset term in GLM
# @ n.samples= Number of MCMC samples from the posterior
# this is a sampler for Poisson distribution
Bayes.icon.glm<-function(y,X,R,b,delta,n.samples=5000){
  n<-length(y)
  p<-ncol(X)
  m<-length(b)
  #Get Glm Betahat
  glmod<-glm(y~-1+X,offset = delta,family = poisson(link = "log"))
  betaglm<-as.numeric(glmod$coefficients)
  #function for calculating information matrix
  Ibetafunc<-function(beta)
  {
    gamma2<-c()
    for(i in 1:n)
    {temp=crossprod(X[i,],beta)+delta[i]
    gamma2[i]=exp(temp)  #depends on the \psi function, exp for Poisson
    }
    GamaMat2<-diag(gamma2)
    Ibetahat<-t(X)%*%GamaMat2%*%X
    return(Ibetahat)
  }
  Ibetahat<-Ibetafunc(betaglm)
  mu1= betaglm
  Sigma1= chol2inv(chol(Ibetahat))
  Sigmapost<-Sigma1
  mupost<-mu1+Sigma1%*%t(X)%*%y
  beta<- betaglm
  samples <- matrix(0,n.samples,p)
  Rstar<-rbind(R,-X)
  #Start the MCMC sampler:
  for(i in 1:n.samples){
    #update U_i
    u<-c()
    for(l in 1:n)
    {up<- as.numeric(exp(-exp(t(X[l,])%*%beta+delta[l])))
    u[l]<-runif(1,0,up)
```

```
    }

    #update beta:
    library(tmvmixnorm)
    bstar2<--log(-log(u))+delta
    bstar<-c(b,bstar2)
    mstar<-length(bstar)
    beta<-rtmvn(n=1, Mean=mupost, Sigmapost, D=Rstar, lower=bstar,
                upper=rep(Inf,mstar),int=beta, burn=10)

    #store results:
    samples[i,]  <- c(beta)
    #print(i)
  }

  #return a list with the posterior samples:
  return(samples)
}
```

The constraint matrix for the application is set up below.

```
R<-matrix(0,nrow=8,ncol=9)
for(i in 1:8)
{for(j in 1:9)
{
  if(j==i){R[i,j]=1}
  if(j==(i+1)){R[i,j]=-1}
}
}
R<-cbind(rep(0,8),R)
mat<-matrix(0,nrow=8,ncol=65)
R<-cbind(R,mat)
b<-rep(0,8)
```

We run the slice sampler now the the variables and the constraint matrix have been defined.

```
outglm<- Bayes.icon.glm(y=y,X=X,R=R,b=b,delta=log(c.time)[nzero.indx],
n.samples=40000)
##takes time to run, we provide saved output below
burn=20000
n.samples=40000
betapost<-outglm[burn:n.samples,] #Posterior samples
```

The estimated parameters for year-specific effects are shown below.

```
###loading saved posterior samples
load("betapost_scram_slice_1s.RData")
colMeans(betapost)[2:10] #Bayes estimate under S.E.L
```

```
## [1] -0.2887305 -0.4953470 -0.8401664 -1.0165751 -1.1258800 -1.1837125 -1.2435252
## [8] -1.2973178 -1.3639937
```

```r
apply(betapost,2,sd)[2:10] #Bayesian estimate of sd
```

```
## [1] 0.05259218 0.05703580 0.06459602 0.06461134 0.05620075 0.05332575 0.05410466
## [8] 0.05591090 0.06556231
```