Week 8 Lab – MAST90125: Bayesian Statistical learning

Perform Gibbs sampling for linear models with proper priors for β.

In this week's lab, we discuss how to write Gibbs sampling code for linear models with proper priors. We consider the data in `USJudgeRatings.csv`, which is available on Canvas. We assume the variable `RTEN` is the response and the other variables as predictors.

Download `USJudgeRatings.csv` from Canvas.

Understand the code below that purports to perform Gibbs sampling for a variety of linear models. See if you can determine what the code is doing. You may find referring back to Lectures useful. Compare each other the posterior distributions obtained from the different priors.

Examples of Gibbs sampler for linear models

First, exercise the following two functions and see what they correspond to in Lecture?

```
Gibbs.lm1<-function(X,y,tau0,iter,burnin){
p <- dim(X)[2]
XTX <- crossprod(X)
XTXinv <-solve(XTX)
XTY <- crossprod(X,y)
betahat<-solve(XTX,XTY)
tau    <-tau0
library(mvtnorm)

par<-matrix(0,iter,p+1)
for( i in 1:iter){
  beta <- rmvnorm(1,mean=betahat,sigma=XTXinv/tau)
  beta <-as.numeric(beta)
  err  <- y-X%*%beta
  tau  <- rgamma(1,0.5*n,0.5*sum(err^2))
  par[i,] <-c(beta,tau)
}

par <-par[-c(1:burnin),]
return(par)
}
```

```
Gibbs.lm2<-function(X,y,tau0,iter,burnin){
p <- dim(X)[2]
svdX <-svd(X)
U    <-svdX$u
Lambda<-svdX$d
V    <-svdX$v
```

```
Vbhat <- crossprod(U,y)/Lambda
tau <-tau0

vbeta<-rnorm(p)
par<-matrix(0,iter,p+1)
for( i in 1:iter){
  sqrttau<-sqrt(tau)
  vbeta <- rnorm(p,mean=Vbhat,sd=1/(sqrttau*Lambda) )
  beta <-V%*%vbeta
  err  <- y-X%*%beta
  tau  <- rgamma(1,0.5*n,0.5*sum(err^2))
  par[i,] <-c(beta,tau)
}

par <-par[-c(1:burnin),]
return(par)
}
```

What is the different between the above functions?

Then, we go on practising those tasks discussed in Lecture.

- Linear mixed model/ ridge regression (flat prior for $\boldsymbol{\beta}_0$, $p(\tau) = \text{Ga}(\alpha_e, \gamma_e)$, where $\tau = (\sigma^2)^{-1}$), $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \sigma_\beta^2 \mathbf{I})$, $(\sigma_\beta^2)^{-1} = \tau_\beta \sim \text{Ga}(\alpha_\beta, \gamma_\beta)$.

```
normalmm.Gibbs<-function(iter,Z,X,y,burnin,taue_0,tauu_0,a.u,b.u,a.e,b.e){
  n   <-length(y) #no. observations
  p   <-dim(X)[2] #no of fixed effect predictors.
  q   <-dim(Z)[2] #no of random effect levels
  tauu<-tauu_0
  taue<-taue_0
  beta0<-rnorm(p)
  u0   <-rnorm(q,0,sd=1/sqrt(tauu))

  #Building combined predictor matrix.
  W<-cbind(X,Z)
  WTW <-crossprod(W)
  library(mvtnorm)

  #storing results.
  par <-matrix(0,iter,p+q+2)

  #Create modified identity matrix for joint posterior.
  I0  <-diag(p+q)
  diag(I0)[1:p]<-0

  for(i in 1:iter){
    #Conditional posteriors.
    tauu <-rgamma(1,a.u+0.5*q,b.u+0.5*sum(u0^2))
    #Updating component of normal posterior for beta,u
    Prec <-WTW + tauu*I0/taue
    P.mean <- solve(Prec)%*%crossprod(W,y)
    P.var  <-solve(Prec)/taue
```

```r
    betau <-rmvnorm(1,mean=P.mean,sigma=P.var)
    betau <-as.numeric(betau)
    err   <- y-W%*%betau
    taue  <-rgamma(1,a.e+0.5*n,b.e+0.5*sum(err^2))
    #storing iterations.
    par[i,]<-c(betau,1/sqrt(tauu),1/sqrt(taue))
    beta0  <-betau[1:p]
    u0     <-betau[p+1:q]
  }

par <-par[-c(1:burnin),]
colnames(par)<-c(paste('beta',1:p,sep=''),paste('u',1:q,sep=''),'sigma_b','sigma_e')
 return(par)
}
```

- LASSO. $\boldsymbol{\beta}_j$ drawn from Laplace prior with parameter λ assumed fixed. Implicit prior is $p(\beta_j|\sigma_j^2) = \mathcal{N}(0, \sigma_j^2), p(\sigma_j^2) = \text{Exp}(\gamma^2/2)$.

```
normallasso.Gibbs<-function(iter,Z,X,y,burnin,taue_0,lambda,a.e,b.e){
  library(LaplacesDemon)
  n   <-length(y) #no. observations
  p   <-dim(X)[2] #no of fixed effect predictors.
  q   <-dim(Z)[2] #no of random effect levels
  taue<-taue_0
  tauu <-rinvgaussian(q,lambda/abs(rnorm(q)),lambda^2)

  #Building combined predictor matrix.
  W<-cbind(X,Z)
  WTW <-crossprod(W)
  library(mvtnorm)

  #storing results.
  par <-matrix(0,iter,p+q+1)

  for(i in 1:iter){
    #Conditional posteriors.

    #Updating component of normal posterior for beta,u
    Kinv  <-diag(p+q)
    diag(Kinv)[1:p]<-0
    diag(Kinv)[p+1:q]<-tauu

    Prec <-taue*WTW + Kinv
    P.var  <-solve(Prec)
    P.mean <- taue*P.var%*%crossprod(W,y)
    betau <-rmvnorm(1,mean=P.mean,sigma=P.var)
    betau <-as.numeric(betau)
    err   <- y-W%*%betau
    taue  <-rgamma(1,a.e+0.5*n,b.e+0.5*sum(err^2))
    tauu <-rinvgaussian(q,lambda/abs(betau[-c(1:p)]),lambda^2)

    #storing iterations.
    par[i,]<-c(betau,1/sqrt(taue))
  }

par <-par[-c(1:burnin),]
colnames(par)<-c(paste('beta',1:p,sep=''),paste('u',1:q,sep=''),'sigma_e')
 return(par)
}
```

- Compare LASSO with linear mixed model regarding the posteriors for $\boldsymbol{\beta}$.

Below is code for plotting densities. However you need to write the code for creating the MCMC chains for the mixed model and LASSO examples.

```
chainmm<-rbind(chain10,chain11,chain12)
chainlasso<-rbind(chain13,chain14,chain15)
```

```
par(mfrow = c(4,3))
for(i in 1:12){
plot(density(chainmm[,i]),xlab=paste('b',i-1,sep=''),ylab='posterior',main='')
lines(density(chainlasso[,i]),col=2)
legend('topright',legend=c('mixed model','lasso'),col=1:2,pch=19)
abline(v=0)
}
```