Homework 1

Ian Frankenburg

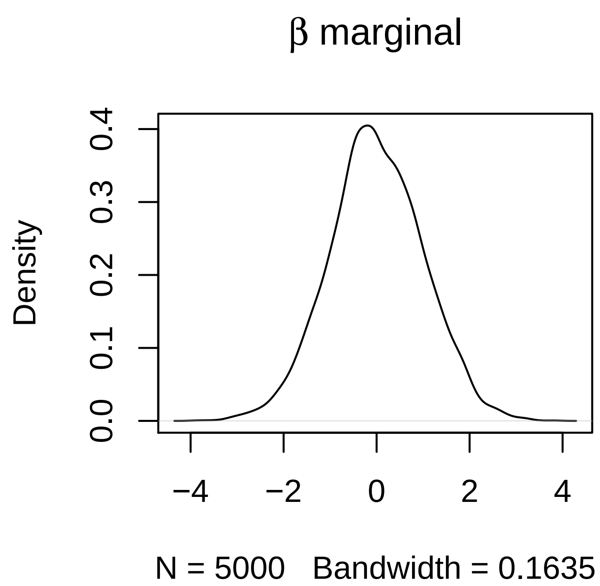
4/7/2020

Bayesian Adaptive Lasso

Part a.

Consider $p = 1$. Simulate 5,000 Monte Carlo samples from the conditional prior $\beta|\tau^2 = 1$ and obtain a plot of the density using the R function density.

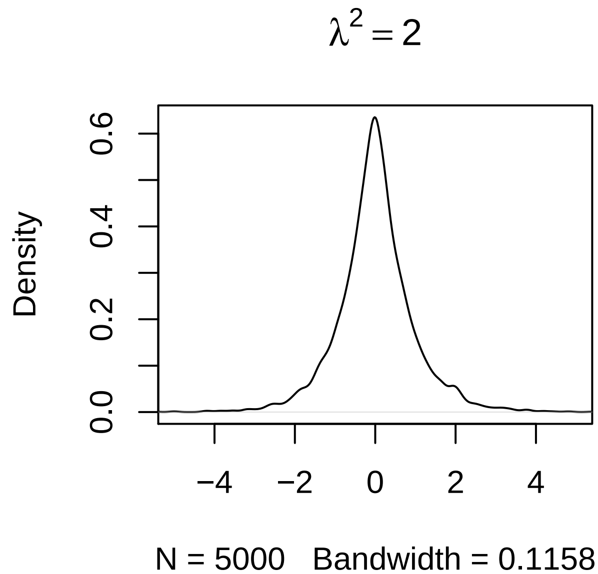
```
n <- 5000
plot(density(rnorm(n,0,1)), main=TeX(paste("$\\beta$", "marginal")))
```



Part b.

Consider $p = 1$. Simulate 5,000 Monte Carlo samples from the marginal prior $\boldsymbol{\beta}$, considering $\lambda^2 = 2$, so that $\mathbb{E}(\tau^2|\lambda) = 1$. Obtain a plot of the density as in **a.**

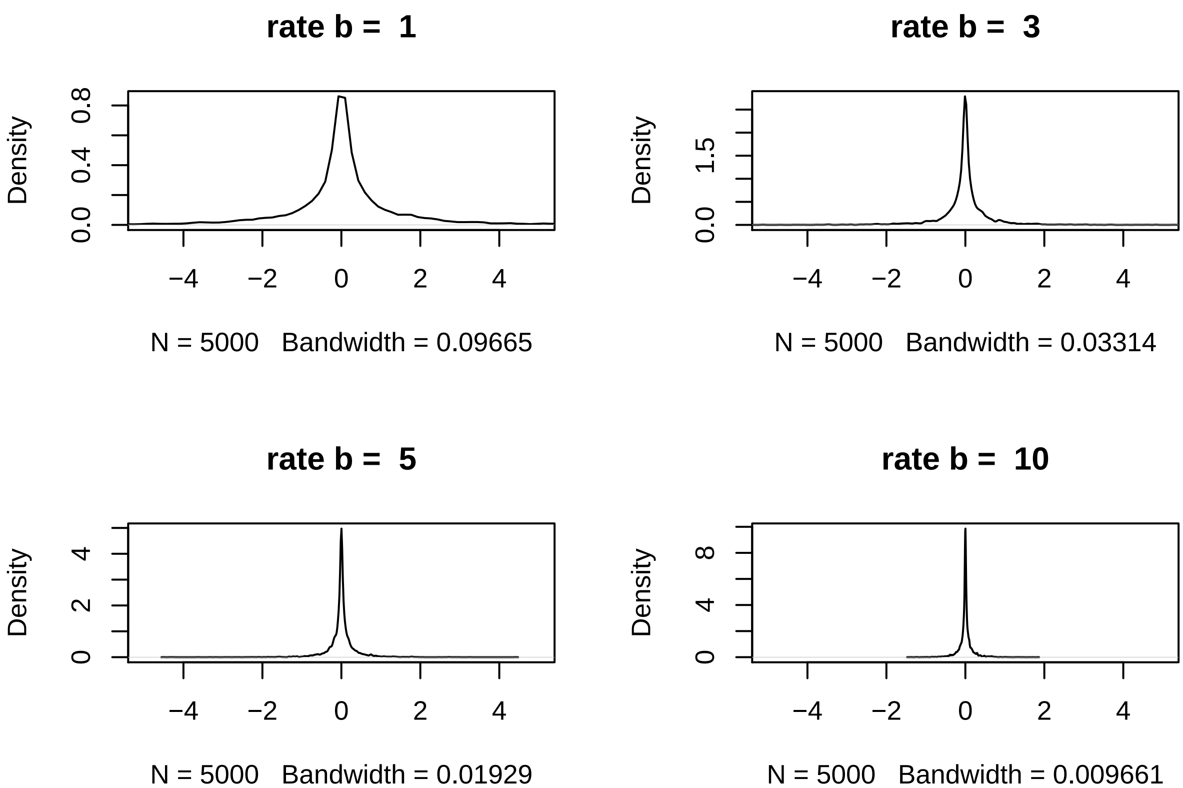
```r
lambda <- sqrt(2)
tau.sq <- rgamma(n,shape=1,rate = lambda^2/2)
beta.marginal <- rnorm(n,0,sqrt(tau.sq))
plot(density(beta.marginal), main=TeX(paste("$\\lambda^2 = 2$")), xlim=c(-5,5))
```



Part c.

Consider $p = 1$. Add a hyper prior on $\gamma = 1/\gamma \sim Gamma(a, rate = b)$. Assess how the marginal prior of $\boldsymbol{\beta}$ changes for $a = 1$ and values of $b \geq 1$.

```r
set.seed(1)
par(mfrow=c(2,2))
rates <- c(1,3,5,10)
for(b in rates){
  lambda <- 1/rgamma(n,1,b)
  tau.sq <- rgamma(n,shape=1,rate = lambda^2/2)
  beta.marginal <- rnorm(n,0,sqrt(tau.sq))
  plot(density(beta.marginal), main=paste("rate b = ",b),xlim=c(-5,5))
}
```



Part d.

Considering the hyper prior in **c.**, describe a Markov Chain Monte Carlo algorithm to sample from the posterior distribution of $\boldsymbol{\beta}$ and σ^2.

I will implement a joint Gibbs and Metropolis sampler. The model is

$$\begin{aligned}
\boldsymbol{Y}|\boldsymbol{\beta},\sigma^2 &\sim N(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I})\\
\beta_j|\tau_j^2 &\sim N(0,\tau_j^2)\\
\tau_j^2 &\sim \text{Gamma}(1,\frac{\lambda^2}{2})\\
\lambda^2 &\sim \text{Inverse-Gamma}(a,1/b)\\
\sigma^2 &\sim \text{Inverse-Gamma}(0.1,0.1).
\end{aligned}$$

I need the full conditionals

$$\begin{aligned}
&\{\beta_1,\dots,\beta_p|\boldsymbol{Y},\sigma^2,\tau_1^2,\dots,\tau_p^2,\lambda\},\\
&\{\sigma^2|\boldsymbol{Y},\beta_1,\dots,\beta_p,\tau_1^2,\dots,\tau_p^2,\lambda\},\\
&\{\tau_1^2,\dots,\tau_p^2|\boldsymbol{Y},\beta_1,\dots,\beta_p,\sigma^2,\lambda\},\\
&\{\lambda|\boldsymbol{Y},\beta_1,\dots,\beta_p,\sigma^2,\tau_1^2,\dots,\tau_p^2\}
\end{aligned}$$

which are all proportional to

$$p(\boldsymbol{Y}|\beta_1,\dots,\beta_p,\tau_1^2,\dots,\tau_p^2,\sigma^2,\lambda)\times p(\beta_1,\dots,\beta_p|\tau_1^2,\dots,\tau_p^2)\times p(\tau_1^2,\dots,\tau_p^2|\lambda)p(\lambda)p(\sigma^2)$$

so I'll start with the posterior

$$\begin{aligned}
p(\beta_1,\dots,\beta_p,\tau_1^2,\dots,\tau_p^2,\sigma^2,\lambda|\boldsymbol{Y}) \propto\ & p(\boldsymbol{Y}|\beta_1,\dots,\beta_p,\tau_1^2,\dots,\tau_p^2,\sigma^2,\lambda)\\
&\times p(\beta_1,\dots,\beta_p|\tau_1^2,\dots,\tau_p^2)\\
&\times p(\tau_1^2,\dots,\tau_p^2|\lambda)p(\lambda)p(\sigma^2).
\end{aligned}$$

As a function of just σ^2, this is proportional to

$$\begin{aligned}
&p(\boldsymbol{Y}|\beta_1,\dots,\beta_p,\tau_1^2,\dots,\tau_p^2,\sigma^2,\lambda)p(\sigma^2)\\
=\ & N(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I})IG(a,b).
\end{aligned}$$

Time to show this is inverse-gamma distributed.

$$\begin{aligned}
&N(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I})IG(a,b)\\
\propto\ &(\sigma^2)^{-n/2}\exp\big\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\big\}(\sigma^2)^{a-1}\exp\big\{-\frac{b}{\sigma^2}\big\}\\
=\ &(\sigma^2)^{-(n/2+a)-1}\exp\big\{-\frac{1}{\sigma^2}(2b+\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\big\}\\
=\ &IG(n/2+a,2b+(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})/2)
\end{aligned}$$

As a function of $\boldsymbol{\beta}$, the conditional is proportional to

$$\begin{aligned}
&p(\boldsymbol{Y}|\beta_1,\dots,\beta_p,\tau_1^2,\dots,\tau_p^2,\sigma^2,\lambda)p(\beta_1,\dots,\beta_p|\tau_1^2,\dots,\tau_p^2)\\
=\ & N(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I})\cdot\prod_{i=1}^p N(0,\tau_i^2)\\
=\ & N(\boldsymbol{X}\beta,\Sigma)\cdot N(0,\Omega),\ \text{where }\Omega=\text{diag}(\tau_1^2,\dots,\tau_p^2)\\
=\ & N(m,\boldsymbol{M})
\end{aligned}$$

with $m = \boldsymbol{M}\boldsymbol{X}^\top\Sigma^{-1}y$ and $\boldsymbol{M} = (\boldsymbol{X}^\top\Sigma^{-1}\boldsymbol{X} + \Omega^{-1})^{-1}$. As a function of $\tau_1^2, \ldots, \tau_p^2$, the conditional is non-standard, but it's proportional to

$$\begin{aligned} & p(\beta_1, \ldots, \beta_p | \tau_1^2, \ldots, \tau_p^2) p(\tau_1^2, \ldots, \tau_p^2 | \lambda) \\ = & \prod_{i=1}^{p} N(0, \tau_i^2) \cdot \prod_{i=1}^{p} IG(1, \frac{\lambda^2}{2}) \end{aligned}$$

Finally, as a function of λ, it's proportional to

$$\begin{aligned} & p(\tau_1^2, \ldots, \tau_p^2 | \lambda) p(\lambda) \\ = & \prod_{i=1}^{p} IG(1, \frac{\lambda^2}{2}) \cdot IG(0.1, 0.1) \end{aligned}$$

Now I can build an algorithm to iteratively update through these conditional distributions. I take the starting value of $\boldsymbol{\beta}^{(0)}$ to be the least-squares solution $\hat{\boldsymbol{\beta}}$.

Result: Samples from joint posterior $p(\boldsymbol{\beta}, \sigma^2 | \boldsymbol{y})$

for *s in # samples* **do**

$\sigma^{2(s+1)} \sim IG(n/2 + a, 2b + (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^{(s)})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^{(s)})/2)$

$\boldsymbol{\beta}^s \sim N_p(m, \boldsymbol{M})$

$m^\top = \qquad \boldsymbol{M} = \tau_1^2, \ldots, \tau_p^2 \sim N(\boldsymbol{\tau}^{(s)}, \delta\boldsymbol{I})^2$

$\lambda^* \sim N(\lambda.s, \delta)^2$

$$\begin{aligned} \log(r) = & \log p(\boldsymbol{y} | \boldsymbol{\beta}^*, \tau_1^2, \ldots, \tau_p^2, \sigma^{2(s)}) + \log p(\boldsymbol{\beta}^* | \tau_1^2, \ldots, \tau_p^2) \\ & - \log p(\boldsymbol{y} | \boldsymbol{\beta}^{(s)}, \tau_1^2, \ldots, \tau_p^2, \sigma^{2(s)}) - \log p(\boldsymbol{\beta}^{(s)} | \tau_1^2, \ldots, \tau_p^2) \end{aligned}$$

$u \sim Unif(0, 1)$

if $\log(u) < \log(r)$ **then**

$\boldsymbol{\tau}^{(s+1)} = \boldsymbol{\beta}^*$

else

$\boldsymbol{\tau}^{(s+1)} = \boldsymbol{\beta}^{(s)}$

end

end

Algorithm 1: Gibbs and Metropolis

Part e.

temp

```
set.seed(1)
data("diabetes")
X <- cbind(diabetes$x); y <- diabetes$y; n <- nrow(X);
X <- cbind(rep(1,n),X);p <- ncol(X)
samples <- 5000; a <- b <- 0.1; delta <- 0.1
beta.s <- solve(t(X)%*%X)%*%t(X)%*%y
lambda.s <- 0.5
tau2.s <- rep(1,p)
s <- 1
beta<-sigma2<-rep(0,samples)
for(s in 1:samples){
  lambda.star <- rnorm(1,lambda2.s, 0.01)^2
  logr <-
    sum(dgamma(p,1,lambda.star/2, log = T))+
    sum(dgamma(1,0.1,10, log=T))-sum(dgamma(p,1,lambda.s/2, log=T))-
    sum(dgamma(1,0.1,10,log=T))
  if(log(runif(1))<logr){
    lambda.s <- lambda.star
  }else{
    lambda.s <- lambda.s
  }

  tau2.star <- rnorm(p,tau2.s, 0.01)^2
  logr <-
    dmvnorm(rep(0,p),mean=matrix(1,p,1),sigma=diag(p),log = T)+
    sum(dgamma(p,1,lambda2.s/2, log=T))-
    dmvnorm(rep(0,p),mean=rep(0,p),sigma=diag(tau2.s),log = T)-
    sum(dgamma(p,1,lambda2.s/2, log=T))
  if(log(runif(1))<logr){
    tau2.s <- tau2.star
  }else{
    tau2.s <- tau2.s
  }
  lambda <- 1/rgamma(1,0.1,10)
  tau2 <- 1/rgamma(p,1,lambda^2/2)
  sigma2.s <- rgamma(1,n/2+a, 1/(2*b+t(y-X%*%beta.s)%*%(y-X%*%beta.s)))
  M <- matrix(solve(1/sigma2.s*t(X)%*%X+diag(1/tau2)),p,p)
  m <- M%*%t(X)%*%y/sigma2.s
  beta.s <- t(rmvnorm(n=1,mean=m,sigma=diag(p)))
  sigma2 <- c(sigma2, sigma2.s)
  beta <- rbind(beta,t(matrix(beta.s)))
}


beta.ls <- solve(t(X)%*%X)%*%t(X)%*%y
fit <- glmnet(X[,-1], y)
df <- cbind(beta.ls,
colMeans(beta[floor(samples/4):samples,]), coef(fit,s=0))
colnames(df) <- c("Least Square", "Bayes", "glmnet")
df
```

Part f.

Implement such algorithm in R and compare your results with estimates obtained using **glmnet()**. In particular, you should test your results on the diabetes data available from lars, (use the matrix of predictors x).

Part g.

Free λ and carry out a sensitivity analysis assessing the behavior of the posterior distribution of $\boldsymbol{\beta}$ and σ^2, as hyper parameters a and b are changed. Explain clearly the rationale you use to assess sensitivity and provide recommendations for the analysis of the diabetes data.

Part h.

Implementation and benchmarking in Julia