# Homework 1

Ian Frankenburg

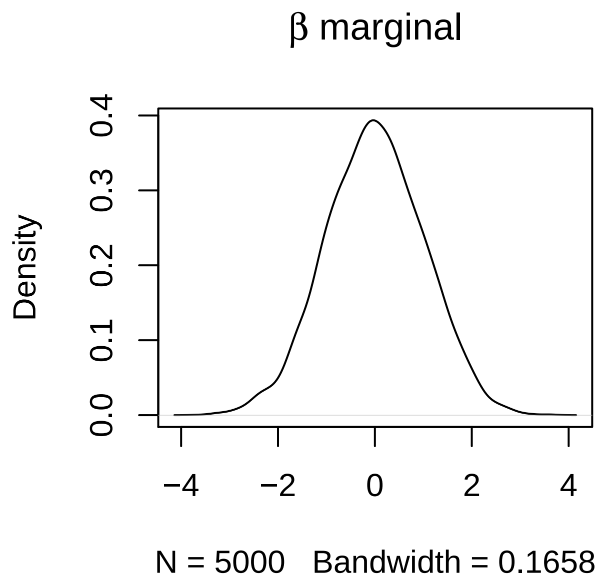
4/7/2020

## 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.

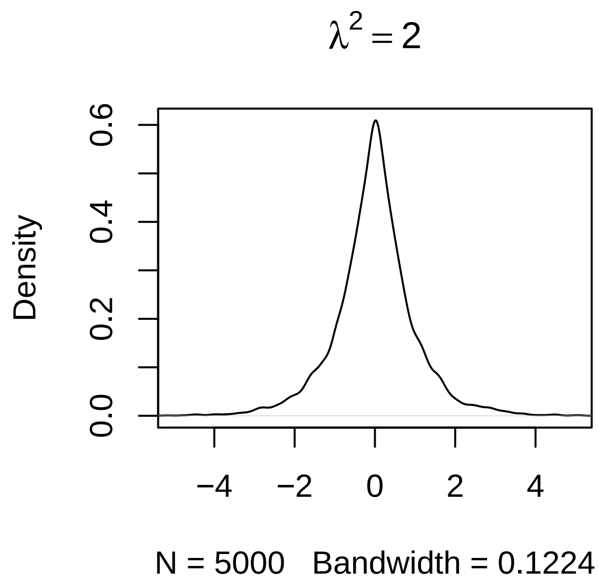
```
n <- 5000
plot(density(rnorm(n,0,1)), main=TeX(paste("$\\beta$", "marginal")))
```



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.**

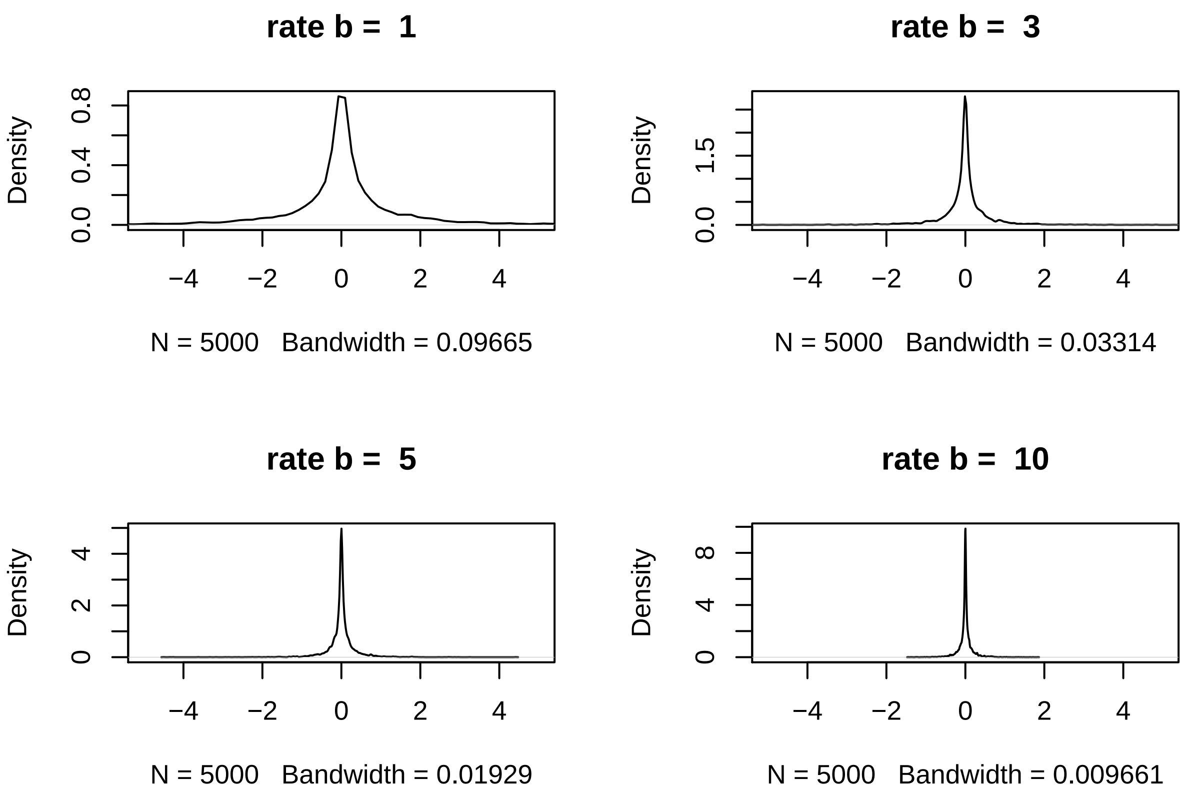
```
lambda <- sqrt(2)
tau.sq <- rgamma(n,shape=1,rate = lambda^2/2)
beta.marginal <- rnorm(n,0,sqrt(tau.sq))
plot(density(beta.marginal), main=TeX(paste("$\\lambda^2 = 2$")), xlim=c(-5,5))
```



**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 $\sigma^2$.

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

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

To start, I need the full conditional

$$\{\sigma^2|\boldsymbol{Y},\beta_1,\ldots,\beta_p,\tau_1^2,\ldots,\tau_p^2,\lambda\},$$

so I'll start with the posterior

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

As a function of just $\sigma^2$, this is proportional to

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

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 non-standard, but it's proportional to

$$\begin{aligned}
& p(\boldsymbol{Y}|\beta_1,\ldots,\beta_p,\tau_1^2,\ldots,\tau_p^2,\sigma^2,\lambda)p(\beta_1,\ldots,\beta_p|\tau_1^2,\ldots,\tau_p^2)\\
=\ & N(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I})\cdot\prod_{i=1}^{p}N(0,\tau_i^2).
\end{aligned}$$

So my sampling routine will combine Gibbs to sample from $\sigma^2$ and Metropolis to sample from $\boldsymbol{\beta}$.

In the actual sampling routine, I can simply generate the $\tau_j$'s conditional on a $\lambda$ value and then place these values in a matrix $\Sigma$, which I'll have appropriately hard-coded. Now that I have the full conditionals, at a high-level, the Gibbs Sampling routine will progress as. Suppose I have a $\boldsymbol{\beta}^{(0)}$. Then I sample a $\sigma^{2(0)}$

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

**for** *s in # Samples* **do**

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

$\lambda \sim IG(0.1, 0.1)$

$\tau_1^2, \ldots, \tau_p^2 \overset{iid}{\sim} IG(1, \lambda^2/2)$

$\boldsymbol{\beta}^* \sim N_p(\boldsymbol{\beta}^{(s)}, \delta \boldsymbol{I})$

$r = \frac{p(\boldsymbol{y}|\boldsymbol{\beta}^*, \tau_1^2, \ldots, \tau_p^2, \sigma^{2(s)})p(\boldsymbol{\beta}^*|\tau_1^2, \ldots, \tau_p^2)}{p(\boldsymbol{y}|\boldsymbol{\beta}^{(s)}, \tau_1^2, \ldots, \tau_p^2, \sigma^{2(s)})p(\boldsymbol{\beta}^{(s)}|\tau_1^2, \ldots, \tau_p^2)}$

$u \sim Unif(0, 1)$

**if** $u < r$ **then**

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

**else**

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

**end**

**end**

**Algorithm 1:** Combining Gibbs and Metropolis

**Part e.**

temp

```
set.seed(1)
data("diabetes")
X <- cbind(rep(1,n),diabetes$x); y <- diabetes$y; n <- nrow(X); p <- ncol(X)
samples <- 500; a <- b <- 0.1; delta <- 0.1
beta.s <- solve(t(X)%*%X)%*%t(X)%*%y
sigma2 <- 0; beta <- rep(0,p)
for(s in 1:samples){
  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)))
  beta.star <- t(rmvnorm(1,mean = beta.s, sigma = delta * diag(p)))
  logr <-
    dmvnorm(y,X%*%beta.star, sigma = sigma2.s*diag(n),log=T)+
    dmvnorm(t(beta.star),mean=matrix(0,p,1),sigma=diag(tau2), log=T)-
    dmvnorm(y,X%*%beta.s, sigma = sigma2.s*diag(n),log=T)-
    dmvnorm(t(beta.s),mean=matrix(0,p,1),sigma=diag(tau2), log=T)
  if(log(runif(1))<logr){
    beta.s <- beta.star
  }else{
    beta.s <- beta.s
  }
  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,
```

```r
colMeans(beta[250:samples,]), coef(fit,0))
colnames(df) <- c("LS", "Bayes", "glmnet")
df
```

```
## 11 x 3 sparse Matrix of class "dgCMatrix"
##             LS       Bayes      glmnet
##      152.13348  164.724854  152.133484
## age  -10.01220   -9.649369   -9.220679
## sex -239.81909 -227.238635 -239.063410
## bmi  519.83979  525.936521  520.458638
## map  324.39043  324.740108  323.616100
## tc  -792.18416 -797.949162 -716.483385
## ldl  476.74584  469.074282  418.448411
## hdl  101.04457   92.662924   65.386847
## tch  177.06418  183.645684  164.562823
## ltg  751.27932  738.215125  723.735577
## glu   67.62539   67.863606   67.540420
```

**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 $\lambda$ and carry out a sensitivity analysis assessing the behavior of the posterior distribution of $\boldsymbol{\beta}$ and $\sigma^2$, 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