Assignment 2

u1620789

December 6, 2019

1 a

Jeffrey's prior $\pi(\sigma)$ is defined as being proportional to the square root of the Fisher's information I_σ i.e. $\pi(\sigma) \propto \sqrt{I_\sigma}$. Therefore, using the fact that $\epsilon \sim N(0, \sigma^2)$:

$$
\begin{aligned}
\pi(\sigma) &\propto \left[-\mathbb{E}\left(\frac{\partial^2 lnL(\sigma, X)}{\partial \sigma^2} \right) \right]^{\frac{1}{2}} \\
&\propto \left[-\mathbb{E}\left(\frac{1}{\sigma^2} - \frac{3(y - X\beta)^2}{\sigma^4} \right) \right]^{\frac{1}{2}} \\
&\propto \left[-\frac{1}{\sigma^2} + \frac{3\mathbb{E}(y - X\beta)^2}{\sigma^4} \right]^{\frac{1}{2}} \\
&\propto \left[-\frac{1}{\sigma^2} + \frac{3\sigma^2}{\sigma^4} \right]^{\frac{1}{2}} \\
&\propto \left[\frac{2}{\sigma^2} \right]^{\frac{1}{2}} \\
&\propto \frac{1}{\sigma}
\end{aligned}
$$

where the last line follows by adding 2 to the arbitrary constant.

1 b

Posterior $p(\sigma^2, \beta | X, y)$ can be expressed as:

$$
\begin{aligned}
p(\sigma^2, \beta | X, y) &\propto f(y | X, \sigma^2, \beta) p(\sigma^2, \beta | X) \\
&\propto L(\sigma^2, \beta, | y, X) \pi(\beta | X, \sigma^2) \pi(\sigma^2 | X)
\end{aligned}
$$

Breaking the above into the smaller components and noting that $\hat{\beta} = (X^T X)^{-1} X^T y$, compute:

$$
\begin{aligned}
L(\sigma^2, \beta, | y, X) &= (2\pi\sigma^2)^{-n/2} exp\left(-\frac{(y - X\beta)^T (y - X\beta)}{2\sigma^2} \right) \\
&= (2\pi\sigma^2)^{-n/2} exp\left(-\frac{(y - X\hat{\beta})^T (y - X\hat{\beta}) + (\beta - \hat{\beta})^T X^T X (\beta - \hat{\beta})}{2\sigma^2} \right)
\end{aligned}
$$

and

$$\pi(\sigma^2|X)\pi(\beta|X,\sigma^2) \propto \sigma^{-2}(\sigma^2)^{-k/2}exp\Bigg(-\frac{(\beta-\beta_0)^T X^T X(\beta-\beta_0)}{2g\sigma^2}\Bigg)$$

Consequently,

$$p(\sigma^2,\beta|X,y) \propto (\sigma^2)^{-\frac{n+k}{2}-1}exp\Bigg(-\frac{(y-X\hat{\beta})^T(y-X\hat{\beta})+(\beta-\hat{\beta})^T X^T X(\beta-\hat{\beta})}{2\sigma^2}- \\ -\frac{(\beta-\beta_0)^T X^T X(\beta-\beta_0)}{2g\sigma^2}\Bigg)$$

1 c

$$\boldsymbol{\beta}|\sigma^2,\boldsymbol{X},\boldsymbol{y} = \frac{\boldsymbol{\beta},\sigma^2|\boldsymbol{X},\boldsymbol{y}}{\sigma^2|\boldsymbol{X},\boldsymbol{y}} \propto \boldsymbol{\beta},\sigma^2|\boldsymbol{X},\boldsymbol{y}$$

And eliminating all the given terms we are left with

$$\begin{aligned}
\boldsymbol{\beta}|\sigma^2,\boldsymbol{X},\boldsymbol{y} &\propto exp\Bigg(-\frac{(\beta-\hat{\beta})^T X^T X(\beta-\hat{\beta})}{2\sigma^2}-\frac{(\beta-\beta_0)^T X^T X(\beta-\beta_0)}{2g\sigma^2}\Bigg)\\
&\propto exp\Bigg(-\frac{(\beta^T-\hat{\beta}^T)X^T X(\beta-\hat{\beta})}{2\sigma^2}-\frac{(\beta^T-{\beta_0}^T)X^T X(\beta-\beta_0)}{2g\sigma^2}\Bigg)\\
&\propto exp\Bigg(-\frac{1}{2}\Big[(\beta^T-\hat{\beta}^T)\Sigma_1^{-1}(\beta-\hat{\beta})+(\beta^T-{\beta_0}^T)\Sigma_2^{-1}(\beta-\beta_0)\Big]\Bigg)
\end{aligned}$$

Now focusing on the terms only within the square brackets and expanding the terms we get,

$$(\beta^T-\hat{\beta}^T)\Sigma_1^{-1}(\beta-\hat{\beta})+(\beta^T-{\beta_0}^T)\Sigma_2^{-1}(\beta-\beta_0) \tag{1}$$

$$\propto \beta^T(\Sigma_1^{-1}+\Sigma_2^{-1})\beta-\beta^T(\Sigma_1^{-1}\hat{\beta}+\Sigma_2^{-1}\beta_0)-\hat{\beta}\Sigma_1^{-1}\beta-\beta_0\Sigma_2^{-1}\beta \tag{2}$$

The whole expression seems to resemble Multivariate Normal distribution of the form

$$(\beta-\mu)^T\Sigma_3^{-1}(\beta-\mu) \propto \beta^T\Sigma_3^{-1}\beta-\beta^T\Sigma_3^{-1}\mu-\mu^T\Sigma_3^{-1}\beta \tag{3}$$

The first term in equation 3 is equivalent to the first term in the equation 2 from which we can deduce that $\Sigma_3^{-1}=\Sigma_1^{-1}+\Sigma_2^{-1}$. The second terms in the equations 3 and 2 are also equivalent if we multiply term in the equation 2 by the identity:

$$\beta^T(\Sigma_1^{-1}\hat{\beta}+\Sigma_2^{-1}\beta_0)=\beta^T\Sigma_3^{-1}\Sigma_3(\Sigma_1^{-1}\hat{\beta}+\Sigma_2^{-1}\beta_0)$$

Thus $\mu=\Sigma_3(\Sigma_1^{-1}\hat{\beta}+\Sigma_2^{-1}\beta_0)$

Now plugging in the terms we get that

$$\begin{aligned}
\Sigma_3^{-1} &= \frac{X^TX}{\sigma^2}+\frac{X^TX}{\sigma^2 g}=\frac{(g+1)X^TX}{g\sigma^2}\\
\Sigma_3 &= \frac{g\sigma^2}{g+1}(X^TX)^{-1}\\
\mu &= \Sigma_3(\Sigma_1^{-1}\hat{\beta}+\Sigma_2^{-1}\beta_0)=\frac{g}{g+1}(\beta_0/g+\hat{\beta})
\end{aligned}$$

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{X}, \boldsymbol{y} \sim N_{p+1}\left(\frac{g}{g+1}\left(\frac{\beta_0}{g} + \hat{\beta}\right), \frac{g\sigma^2}{g+1}(X^T X)^{-1}\right)$$

The conditional distribution of σ^2 can be defined similarly as

$$\sigma^2|\boldsymbol{y}, \boldsymbol{X} \sim IG\left(\frac{n}{2}, \frac{s^2}{2} + \frac{1}{2(g+1)}(\beta_0 - \hat{\beta})^T X^T X(\beta_0 - \hat{\beta})\right)$$

where $s^2 = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T X^T X(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$

1 d

Algorithm 1: Gibbs sampling under Jeffrey's prior

Input: Conditional distributions, number of iterations (T) and the number of burnins (b)
Output: An approximated parameter distributions

---

for *iter in 1:T* **do**
  σ_t^2 according to $\sigma_{2|\boldsymbol{\beta}_{t-1}, \boldsymbol{y}, \boldsymbol{X}}$;
  $\boldsymbol{\beta}_t$ according to $\boldsymbol{\beta}|\sigma_t^2, \boldsymbol{y}, \boldsymbol{X}$;
end
return T-b iterations of $\boldsymbol{\beta}$ and σ^2

We need to identify $\sigma^2|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X}$.

$$\begin{aligned}\sigma^2|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X} &= \frac{\sigma^2, \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}}{\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}} \propto \sigma^2, \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X} \\ &\sim IG\left(\frac{n+p}{2}, \frac{1}{2}(y - X\beta)^T(y - X\beta) + \frac{1}{g}(\beta - \beta_0)^T X^T X(\beta - \beta_0))\right)\end{aligned}$$

Listing 1: Gibbs sampler under Jeffrey's and g-priors

```
gibbs <- function(T, b, X, y){
  # T - number of iterations
  # b - no. of initial iterations to be omitted from the result - burnin
  # X - covariates in a matrix form
  # y - response vector

  g <- 1 # for simplicity
  p <- ncol(X)

  beta0 <- rep(0, 10) # arbitrary choice of the hyperparameter
  beta <- rep(0, 10) # initial value of beta
  XX <- t(X) %*% X
```

```
beta_hat <- solve(XX) %*% t(X) %*% y # MLE solution

# result matrix
result <- matrix(ncol = 11, nrow = T)
for (i in 1: T) {

  # Defining sigma distribution
  # invgamma package defines rate as scale -- see the documentation
  rate_sigma <- 1/2 * t(y-X %*% beta) %*% (y-X %*% beta) + 1/(2 * g)
               *  t(beta -  beta0)  %*% XX %*% (beta-beta0)
  shape_sigma <-  (n+p)/2
  sigma <- rinvgamma(1, shape = shape_sigma, rate = rate_sigma)

  # Defining beta distribution
  beta_mean <- g/(g+1) * (beta0/g + beta_hat)
  beta_sd <- ((sigma * g)/(g+1) * solve(XX))
  beta <- mvrnorm(1, beta_mean, beta_sd)
  result[i, ] <- c(beta, sigma)
}
adjusted_result <- as.matrix(result[c(b:T),])
adjusted_result
}
```

1 d

Using the seed 1620789, the resulting representation of the iterations for a subset of the variables is presented in the Figure 1. It seems that both β_1 and σ^2 have achieved the required sampling, with no serial correlation visible. The resulting posterior distributions are presented in the Figure 2.

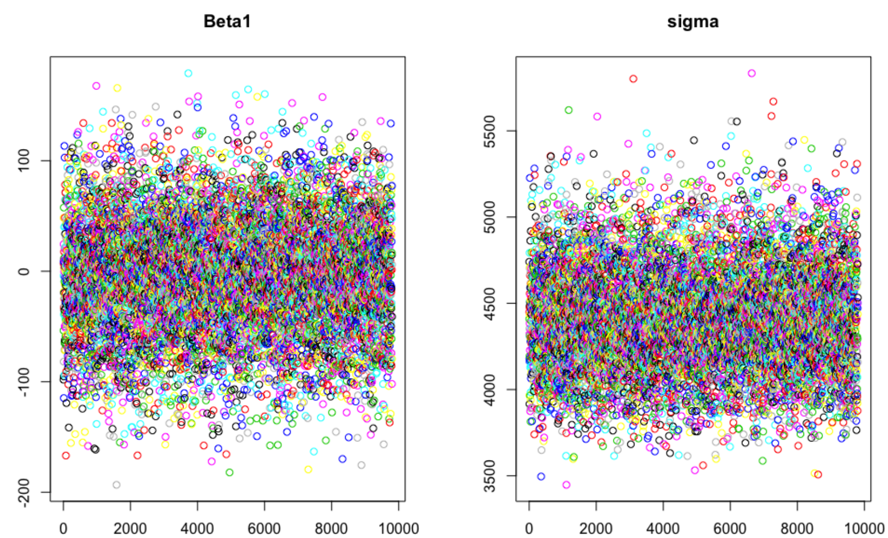
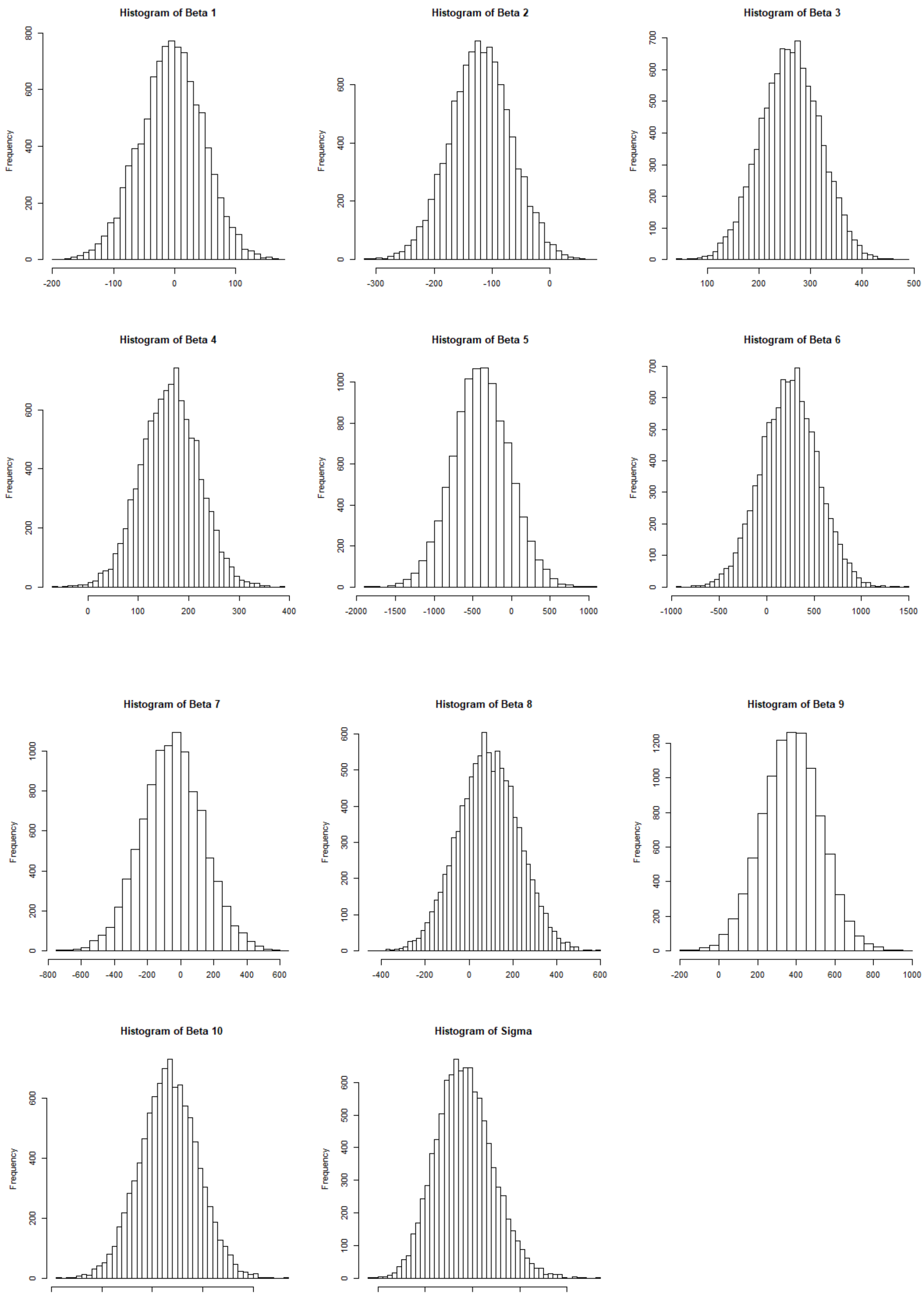


Figure 1: Diagnostic plots for β_1 and σ^2

Figure 2: Posterior distributions (10,000 iterations)



2 i

Logistic regression as well as logistic Lasso is estimated with MLE. To see what prior of β is needed for MAP to be equal to the $\hat{\beta}_{LASSO}$, let's begin by defining MLE estimation:

$$\hat{\beta}_{LASSO} = \beta^L_{MLE} = \underset{\beta}{\operatorname{argmax}}\, P^L(X|\beta) \implies \underset{\beta}{\operatorname{argmax}} \sum_{i=1}^{N} \ell(x_i, \beta) + \lambda \sum_{j=1}^{p} |\beta_j| \tag{4}$$

$$\tag{5}$$

Mazimum-a-posteriori, in turn, replaces all right hand side term with the expression of the posterior distribution

$$\beta_{MAP} \propto \underset{\beta}{\operatorname{argmax}}\, P(X|\beta)P(\beta) \tag{6}$$

$$\beta_{MAP} \propto \underset{\beta}{\operatorname{argmax}} \sum_{i=1}^{N} \ell(x_i, \beta) + logP(\beta) \tag{7}$$

From the last line above, we can see that $logP(\beta)$ needs to be equal to $\lambda \sum_{j=1}^{p} |\beta_j|$ for equations (1) and (3) to be identical.

$$logP(\beta) = \lambda \sum_{j=1}^{p} |\beta_j|$$

$$P(\beta) = \prod_{j=1}^{p} exp(\lambda|\beta_j|)$$

$$\beta \sim Laplace(0, 1/\lambda)$$

It follows that β follows Laplace distribution and thus MAP is equivalent to LASSO. This yields the same result as shown bellow:

$$\beta_{MAP} \propto \underset{\beta}{\operatorname{argmax}} \sum_{i=1}^{N} \ell(x_i, \beta) + log \prod_{j=1}^{p} \frac{\lambda}{2} exp\big(-\lambda|\beta_j|\big)$$

$$\propto \underset{\beta}{\operatorname{argmax}} \sum_{i=1}^{N} \ell(x_i, \beta) - \lambda \sum_{j=1}^{p} |\beta_j|$$

and since $\lambda \leq 0$ the expression above is equivalent to minimising the negative likelihood with LASSO penalty:

$$\underset{\beta}{\operatorname{argmin}} - \sum_{i=1}^{N} \ell(x_i, \beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

Thus Bayesian LASSO using MAP and logistic regression LASSO are identical, yet in practice we minimize the negative log-likelihood which is equivalent to maximising the log-likelihood.

2 ii

Listing 2: Gibbs sampler for Bayesian LASSO

```
library(invgamma)
library(MASS)
library(statmod)
library(SuppDists)

###### PREPARE DATA #######

summary(data) # V11 is the dependent variable
y <- data$V11
X <- data[,c(1:10)]
X <- as.matrix(sapply(X, as.numeric))

set.seed(1620789)

###### GIBBS SAMPLER #######
gibbs_blasso <- function(T, b=200, X, y){
  # T - number of iterations
  # b - number of initial iterations to be omitted from the final result - burnin

  r <- 1 # shape parameter for lambda - as in Park and Casella
  delta <- 1.78 # scale parameter for lambda as in Park and Casella
  n <- nrow(X) # no of observations
  p <- ncol(X) # no of covariates

  # Initial arbitrary D, beta and sigma
  D <- diag(1, p)
  beta <- rep(1, p)
  sigma <- 1
  XX <- t(X) %*% X

  # Result storage
  beta_result <- matrix(ncol = p, nrow = T)
  sigma_result <- matrix(ncol = 1, nrow = T)
  D_result <- matrix(ncol = p, nrow = T)
  lambda_result <- matrix(ncol=1, nrow = T)

  for (i in 1: T) {

    # Define lambda^2 as a hyperprior
    lambda2 <- rgamma(1, shape = p + r, rate = sum(diag(solve(D))/2) + delta)
    lambda  <- sqrt(lambda2)

    # Defining D
    for(t in 1:p){
      D[t,t] <- rinvGauss(1, nu = sqrt((lambda^2 * sigma)/beta[t]^2), lambda = lambda
    }
```

```
    A <- XX + D
    # Defining beta
    beta_mean <- solve(A) %*% t(X) %*% y
    beta_var <- sigma * solve(A)
    beta <- mvrnorm(1, beta_mean, beta_var)

    # Defining sigma
    resid_sigma <- t((y - X %*% beta)) %*% (y - X %*% beta)
    # invgamma package defines rate as scale -- see the documentation
    rate_sigma <- resid_sigma/2 + (t(beta) %*% D %*% beta) / 2
    sigma <- rinvgamma(1, shape = (n-1)/2 + p/2, rate = rate_sigma)

    # Storing results
    beta_result[i,] <- c(beta)
    sigma_result[i,] <- c(sigma)
    D_result[i,] <- c(diag(D))
    lambda_result[i,] <- c(lambda)
  }
  ad_beta_result <- as.matrix(beta_result[c(b:T),])
  ad_sigma_result <- as.matrix(sigma_result[c(b:T),])
  ad_D_result <- as.matrix(D_result[c(b:T),])
  ad_lambda_result <- as.matrix(lambda_result[c(b:T),])

  out <- list()
  out$beta <- ad_beta_result
  out$sigma <- ad_sigma_result
  out$D <- ad_D_result
  out$lambda <- ad_lambda_result

  return(out)
}
```

Figure 3: Sample of posterior BLASSO distributions (10,000 iterations)