HW11

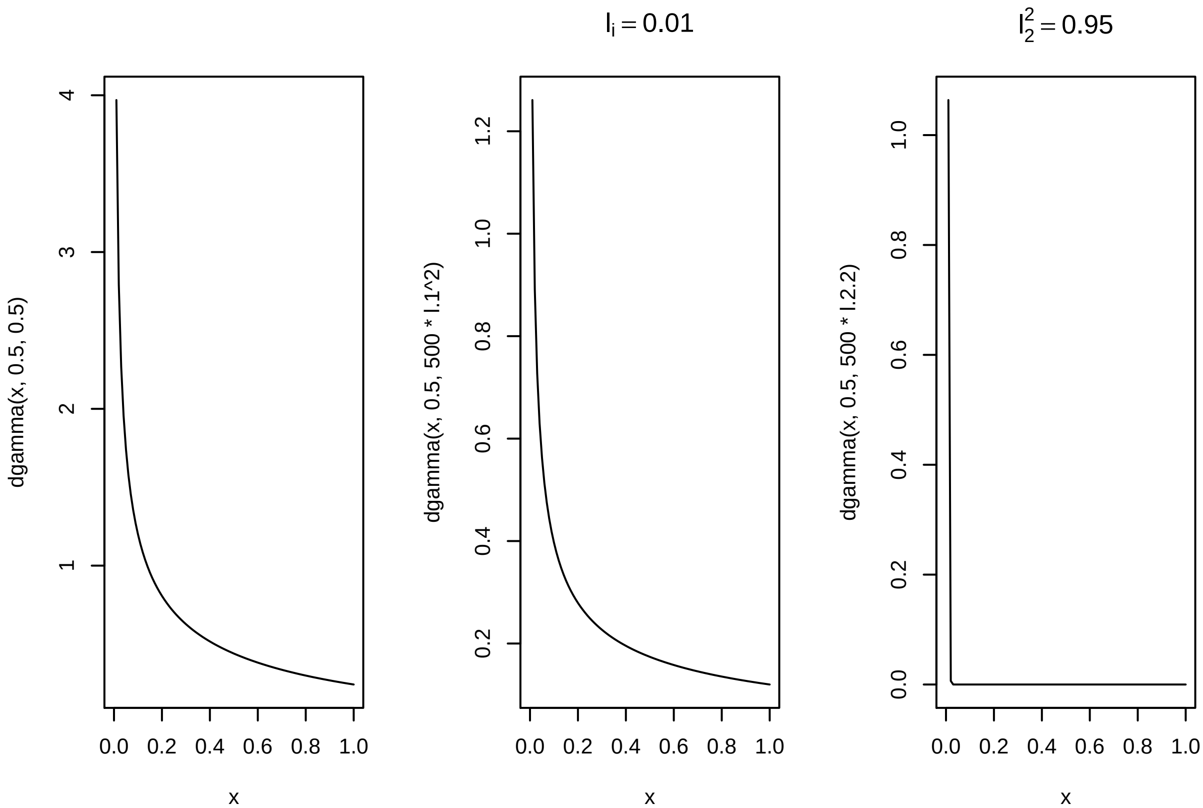
Zach White

10/25/2016

Problem 1

As I was thinking about a potential prior, I used the two bullet points to come with a reasonable prior. The prior I propose is $\kappa_i \sim \text{gamma}(\frac{1}{2}, 500 l_i^2)$. If we examine this prior, for large values of of l_i^2, we want all the prior weight on κ_i to be around zero, and this prior would certainly do this. Note the folowing illustrations

```
l.1 = .01
l.2.2 = .95

par(mfrow = c(1,3))
curve(dgamma(x,.5,.5))
curve(dgamma(x,.5,500*l.1^2), main = expression(l[i] == .01) )
curve(dgamma(x,.5,500*l.2.2), main = expression(l[2]^2== .95) )
```



Now it's clear that these plots meet our criteria.

Problem 2

A prior that would correspond to the g-prior. This prior makes sense, but it is degenerate is $\kappa_i \sim \frac{1}{g}$. This would give us what we need in the other priors.

Problem 3

From our previous prior $\kappa_j \sim G(\frac{1}{2}, \frac{1}{2})$, the only thing we change is the second hyperparameter, and so as we look the the full conditionals, the only one that we have to change is the $\kappa_j|\gamma, \alpha, \phi, Y^*$. Note that it will change in the following way.

$$\begin{aligned}
\kappa_j|\gamma, \alpha, \phi, Y^* &\propto \exp(-\kappa_j(500l_i^2))\exp(-\frac{\kappa_j\phi}{2}\gamma_2^2) \\
&\propto \exp(-\kappa_j(500l_i^2 + \frac{\phi\gamma_j^2}{2})) \\
&\sim \exp(500l_i^2 + \frac{\phi\gamma_j^2}{2})
\end{aligned}$$

The other full conditionals remain unchanged.

Problem 4

```
data(longley)

y = longley$Employed

X = as.data.frame(longley[,-7])
Xs = scale(longley[,-7])

svdX = svd(Xs)
n = nrow(Xs)
p = ncol(Xs)

# Computations
svd_X <- svd(Xs)

n <- nrow(Xs)

p <- ncol(Xs)

##### Compute a lot of stuff outside the loop
Up <- svd_X$u
V <- svd_X$v
l <- svd_X$d
l2 <- l^2
L <- diag(l)
ybar <- mean(y)
LL <- diag(l2)
ys <- t(Up) %*% y
```

```r
ly <- c(l * ys)
ghat <- ys / l
ahat <- ybar
gLLg <- t(ghat) %*% LL %*% ghat
SSE <- t(y) %*% (diag(n) - rep(1,n) %*% t(rep(1,n))/n - Up %*% t(Up) ) %*% y
LUY <- L %*% (t(Up) %*% y)

n.iter = 5000

gamma = matrix(1, ncol = p, nrow = n.iter)
kappa = matrix(1,ncol = p, nrow = n.iter)
alpha = numeric(n.iter)
phi = numeric(n.iter)
alpha[1] = 10
phi[1] = 1


for( i in 2:n.iter ){

gamma[i,] <- rnorm(p, ly / (l2 + kappa[i-1,]), sqrt(1/( phi[i-1] * (l2 + kappa[i-1,]) )) )

alpha[i] <- rnorm(1,ybar, sqrt( 1 / (phi[i-1] * n) ) )

phi[i] <- rgamma(1, (n+p)/2, 1/2 * (SSE + n * ahat^2 + gLLg + n * alpha[i]^2 - 2 * alpha[i] * n * ybar +

t(gamma[i,]) %*% (LL %*% gamma[i,] ) - t(gamma[i,]) %*% LUY + sum(kappa[i-1,] * gamma[i,]^2 ) ) )

kappa[i,] <- rexp(p, (phi[i] * gamma[i,]^2 + 1)/2)

}

burn.in = 1000
post.phi = phi[-c(1:1000)]
post.alpha = alpha[-c(1:1000)]
post.gamma = gamma[-c(1:1000),]
post.kappa = kappa[-c(1:1000),]

apply(post.kappa,2,mean)
```

```
## [1] 1.700169 1.880462 1.642262 1.248430 1.095436 1.012380
```

```r
apply(post.kappa,2,quantile, c(.025,.975))
```

```
##             [,1]       [,2]       [,3]       [,4]       [,5]        [,6]
## 2.5%  0.04222875 0.05397333 0.04154388 0.01653665 0.01051564 0.005290321
## 97.5% 6.34833775 6.94800285 6.28100852 5.33305649 5.17002969 5.005019075
```

```r
apply(post.gamma,2,mean)
```

```
## [1]  1.52589680 -0.34925966  1.33951771  0.01070878  0.79984032  0.29453204
```

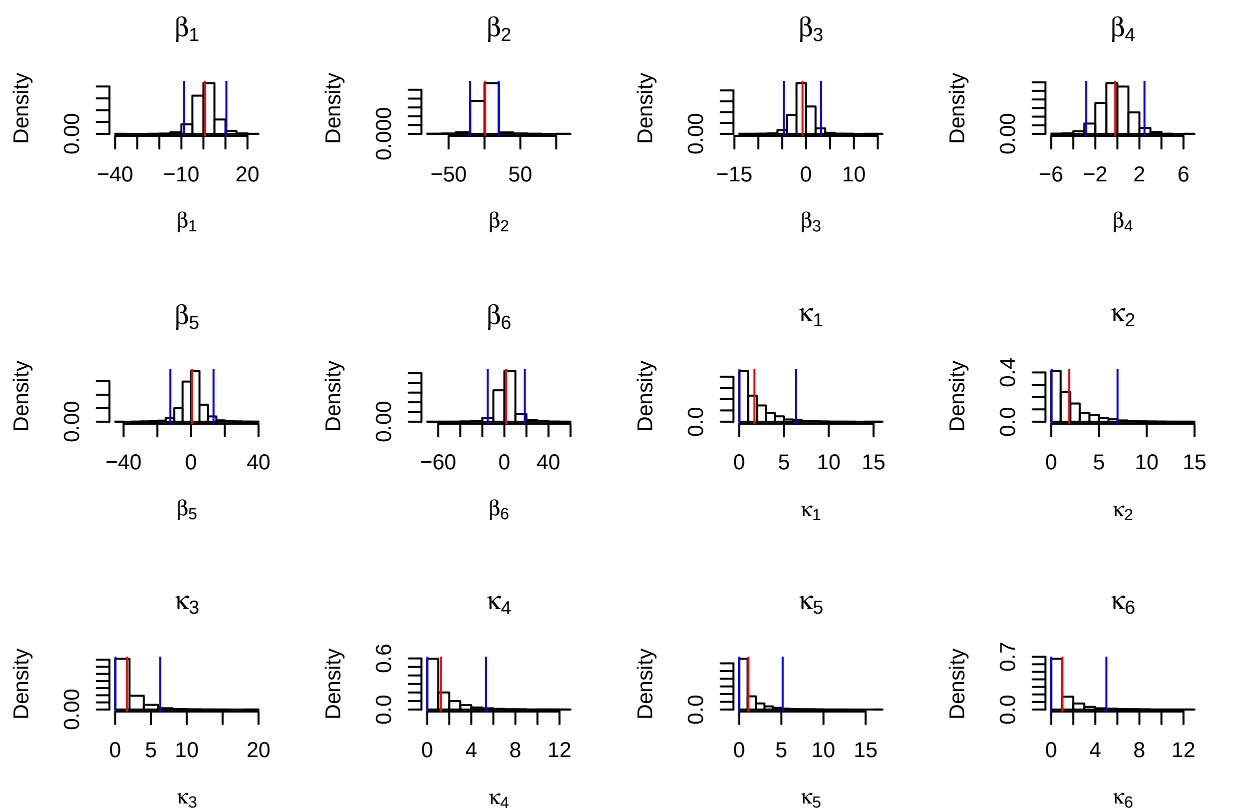
```r
apply(post.gamma,2,quantile,c(.025,.975))
```

```
##            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## 2.5%  0.6419659 -2.026981 -2.212564 -9.102352 -14.98077 -22.10632
## 97.5% 2.4172203  1.325183  5.250537  9.172429  18.08359  26.21282
```

```r
post.beta = t(V %*% t(post.gamma))

apply(post.beta,2,mean)
```

```
## [1]  0.7026746  0.9720167 -0.7322217 -0.1915421  0.5986858  1.6128948
```

```r
post.beta.quant = apply(post.beta,2,quantile,c(.025,.975))



par(mfrow = c(3,4))
for(i in 1:p){
  hist(post.beta[,i], main =bquote(beta[.(i)]), freq = FALSE, xlab = bquote(beta[.(i)]) )
  abline(v = mean(post.beta[,i]), col = "red")
  abline(v = quantile(post.beta[,i],.025), col = "blue")
  abline(v = quantile(post.beta[,i],.975), col = "blue")
}
for(i in 1:p){
  hist(post.kappa[,i], main = bquote(kappa[.(i)]),freq = FALSE, xlab = bquote(kappa[.(i)]))
  abline(v = mean(post.kappa[,i]), col = "red")
  abline(v = quantile(post.kappa[,i],.025), col = "blue")
  abline(v = quantile(post.kappa[,i],.975), col = "blue")
}
```



```r
n.iter = 10000

gamma = matrix(1, ncol = p, nrow = n.iter)
```

```r
kappa = matrix(1,ncol = p, nrow = n.iter)
alpha = numeric(n.iter)
phi = numeric(n.iter)
alpha[1] = 10
phi[1] = 1


for( i in 2:n.iter ){

gamma[i,] <- rnorm(p, ly / (l2 + kappa[i-1,]), sqrt(1/( phi[i-1] * (l2 + kappa[i-1,]) )) )

alpha[i] <- rnorm(1,ybar, sqrt( 1 / (phi[i-1] * n) ) )

phi[i] <- rgamma(1, (n+p)/2, 1/2 * (SSE + n * ahat^2 + gLLg + n * alpha[i]^2 - 2 * alpha[i] * n * ybar +

kappa[i,] <- rexp(p, (phi[i] * gamma[i,]^2)/2 + 500 *l2)

}

burn.in = 5000
post.phi = phi[-c(1:burn.in)]
post.alpha = alpha[-c(1:burn.in)]
post.gamma = gamma[-c(1:burn.in),]
post.kappa = kappa[-c(1:burn.in),]

apply(post.kappa,2,mean)
```

```
## [1] 2.915881e-05 1.108904e-04 6.658739e-04 8.713855e-03 4.029798e-02
## [6] 2.056005e-01
```

```r
apply(post.kappa,2,quantile,c(.025,.975))
```

```
##               [,1]         [,2]         [,3]         [,4]         [,5]
## 2.5%  7.854255e-07 2.749275e-06 1.516484e-05 0.0002574258 0.0008138869
## 97.5% 1.076799e-04 4.267045e-04 2.501733e-03 0.0310882348 0.1558264821
##              [,6]
## 2.5%  0.002034222
## 97.5% 0.989301666
```

```r
apply(post.gamma,2,mean)
```

```
## [1]  1.5752020 -0.3932169  1.8374222  0.2165028  3.5728394  0.9188785
```

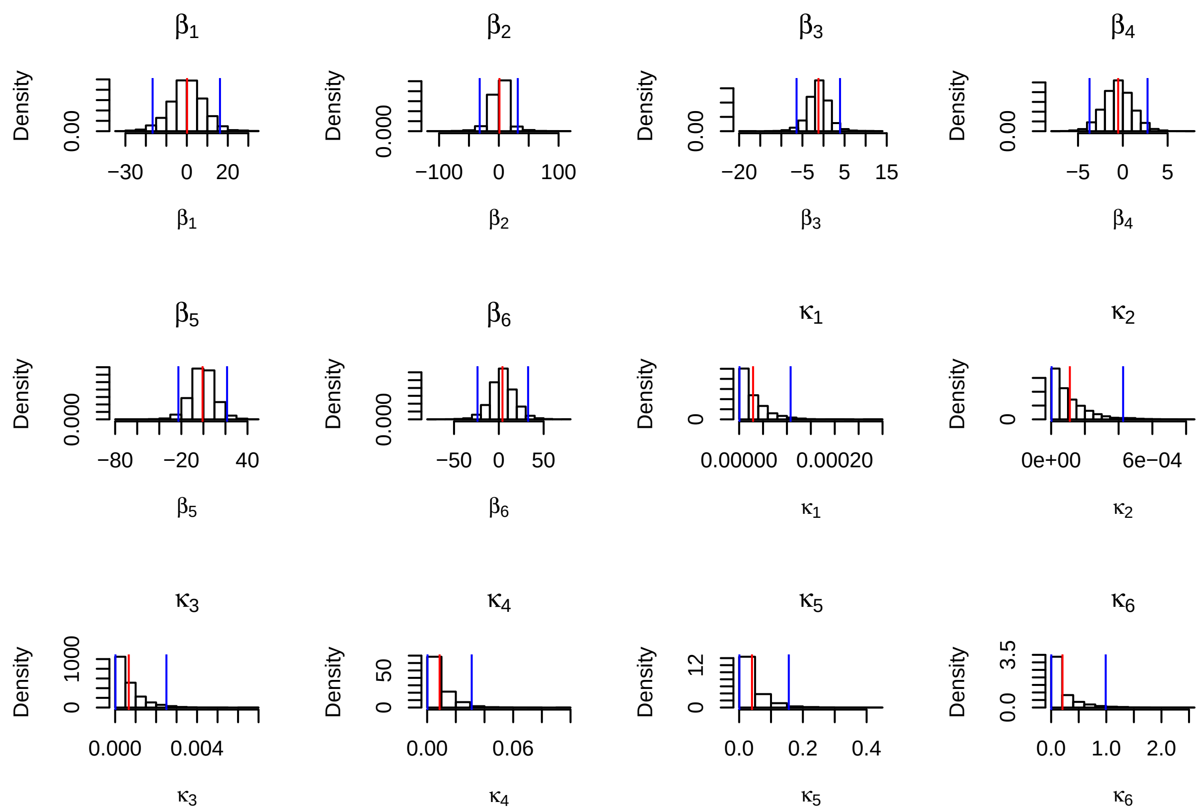
```r
apply(post.gamma,2,quantile,c(.025,.975))
```

```
##            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## 2.5%  0.6743523 -2.221359 -2.553834 -15.45063 -25.39678 -36.39324
## 97.5% 2.4763521  1.356193  6.008819  15.35180  34.20884  41.26342
```

```r
post.beta = t(V %*% t(post.gamma))
apply(post.beta,2,quantile,c(.025,.975))
```

```
##            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## 2.5%  -16.70199 -32.01734 -6.383236 -3.713767 -22.67086 -23.79991
## 97.5%  16.18138  31.72684  3.919467  2.757162  21.47210  32.66728
```

```r
par(mfrow = c(3,4))
for(i in 1:p){
  hist(post.beta[,i], main =bquote(beta[.(i)]), freq = FALSE, xlab = bquote(beta[.(i)]) )
  abline(v = mean(post.beta[,i]), col = "red")
  abline(v = quantile(post.beta[,i],.025), col = "blue")
  abline(v = quantile(post.beta[,i],.975), col = "blue")
}
for(i in 1:p){
  hist(post.kappa[,i], main = bquote(kappa[.(i)]),freq = FALSE, xlab = bquote(kappa[.(i)]))
  abline(v = mean(post.kappa[,i]), col = "red")
  abline(v = quantile(post.kappa[,i],.025), col = "blue")
  abline(v = quantile(post.kappa[,i],.975), col = "blue")
}
```




```r
apply(post.beta,2,mean)
```

```
## [1]  0.1127065  1.0140915 -1.1962973 -0.5205180 -0.7309864  4.0485679
```

```r
apply(post.beta,2,quantile,c(.025,.975))
```

```
##             [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## 2.5%  -16.70199 -32.01734 -6.383236 -3.713767 -22.67086 -23.79991
## 97.5%  16.18138  31.72684  3.919467  2.757162  21.47210  32.66728
```

```r
lm.ridge(y~Xs)
```

```
##                XsGNP.deflator          XsGNP   XsUnemployed XsArmed.Forces
##      65.3170000      0.1625410     -3.5602451     -1.8878325     -0.7190428
```

```
##   XsPopulation         XsYear
##     -0.3554853      8.7085028
```

```
# Code from class with GCV
mod = lm(y~Xs)
CV <- cv.glmnet(Xs,y, lambda=seq(0,1,length=10000),nfolds=5 )

coef(mod)
```

```
##    (Intercept) XsGNP.deflator          XsGNP   XsUnemployed XsArmed.Forces
##     65.3170000      0.1625410     -3.5602451     -1.8878325     -0.7190428
##   XsPopulation         XsYear
##     -0.3554853      8.7085028
```

If we notice, the prior I created of $\kappa_j \sim \text{gamma}(\frac{1}{2}, 500l_2)$ isn't that much different than the previous prior. There is a little bit more shrinkage with my prior in some cases, but in others, there is not. The last results are the generalize cross validation. However, it is important to note that the our shrinkage parameter estimations are larger than the ones from the the GCV, which means that they are more shrunken from the classical. In general, our prior gives us an estimate that is fairly close to the OLS. Also, note that the intercept in this case is 65.317

Problem 5

While holding all other variables constant, for every increase of 1 GNP implicit price deflator, there is a .95 probability that the number of people employed changes by -8.7522193, 10.4262385.

While holding all other variables constant, for every increase of 1 gross domestic product, there is a .95 probability that the number of people employed changes by -20.0850538, 19.8458161.

While holding all other variables constant, for every increase of 1 in number of people unemployed, there is a .95 probability that the number of people employed changes by -4.6385926, 3.1279105.

While holding all other variables constant, for every increase of 1 person in armed forces, there is a .95 probability that the number of people employed changes by -2.8257138, 2.4644127.

While holding all other variables constant, for every increase of 1 noninstituitionalized population greater than 14, there is a .95 probability that the number of people employed changes by -12.3075582, 13.3406535.

While holding all other variables constant, for every increase of 1 year, there is a .95 probability that the number of people employed changes by -15.0379097, 18.5422313.

Problem 6

The computational advantage of using the canonical form is that we don't have to do a matrix inversion for every iteration of the Gibbs sampler. Under the traditional ridge paradigm, we would have to invert $(X^T X + \kappa I)$ every iteration, which would greatly slow down this process. However, under this representation, there isn't any matrix inversion.