Bayesian Ridge Regression

Consider the setup of a Bayesian ridge regression,

$$\begin{aligned} y \mid \beta, \sigma^2, \xi &\sim N(X\beta, \sigma^2 I_n) \\ \beta \mid \sigma^2, \xi &\sim N(0, \xi^{-1}\sigma^2 I_d) \\ \sigma^2 &\sim \mathrm{IG}(a_0/2, b_0/2) \\ \xi &\sim \pi_\xi. \end{aligned}$$

A default choice for the prior on ξ is induced by a half-Cauchy prior on the standard deviation $\tau = \xi^{-1/2}$. The half-Cauchy distribution has a density on $(0, \infty)$ proportional to $1/(1+x^2)$. A simple change-of-variable tells us that the induced prior on ξ is

$$\pi_\xi(\xi) = \frac{1}{\sqrt{\xi}\,(1+\xi)}.$$

We consider the blocked Gibbs sampler discussed in class last week to sample from the posterior distribution of $(\beta, \sigma^2, \xi) \mid y$. Our sampler cycles through the following steps: (i) update $\xi \mid y$ using a MH random walk on $\log \xi$, (ii) sample $\sigma^2 \mid \xi, y$, and (iii) sample $\beta \mid \sigma^2, \xi, y$. [To read more about this in a more general context, you may refer to https://arxiv.org/pdf/1705.00841.pdf.]

- Update ξ: Propose a new value ξ^* from the proposal $\log(\xi^*) \sim N(\log(\xi), s)$. We recommend the default $s = 0.8$. Accept ξ^* with probability $\min\{1.q\}$, where

  $$\log(q) = \log\left(\frac{L(\xi^*)\,\pi_\xi(\xi^*)\,\xi^*}{L(\xi)\,\pi_\xi(\xi)\,\xi}\right),$$

  with

  $$\log L(\xi) = -\frac{1}{2}\log|M_\xi| - \frac{n+a_0}{2}\log(b_0 + y' M_\xi^{-1} y), \quad M_\xi = I_n + \xi^{-1} X X'.$$

- Sample σ^2 from

  $$\mathrm{IG}\left(\frac{n+a_0}{2}, \frac{y' M_\xi^{-1} y + b_0}{2}\right),$$

  where M_ξ is formed with the current value of ξ.

- Sample β from

  $$N\big((X'X + \xi I_p)^{-1} X' y,\, \sigma^2 (X'X + \xi I_p)^{-1}\big).$$

Let's consider our first example. First, let's generate some data.

```
set.seed(1287)
n = 100                                # sample size
d = 7                                 # predictors
betastarm = runif(d,min=0.25,max=1.75)
sgn = 2*rbinom(d,1,0.5) - 1
betastar = betastarm*sgn               # true regression parameters
X = matrix(rnorm(n*d),nrow=n)          # design matrix drawn from iid N(0,1)
mustar = X%*%betastar                  # true mean
sigstar = 1.50                         # true standard deviation
y = mustar + sigstar*rnorm(n)          # generate response from linear model
betastar
```

```
## [1]  1.1296797  0.9847218 -0.5495943  0.7871724  1.5597676 -1.7439878
## [7] -0.4780140
```

Let us now fit the Bayesian ridge regression model. We run the MCMC for 20,000 iterations post burn-in and collect every fifth sample. The hyperparameters are set to $a_0 = b_0 = 0.5$. We also set the standard deviation of the MH random walk for ξ, $\sigma_{MH} = 0.9$.

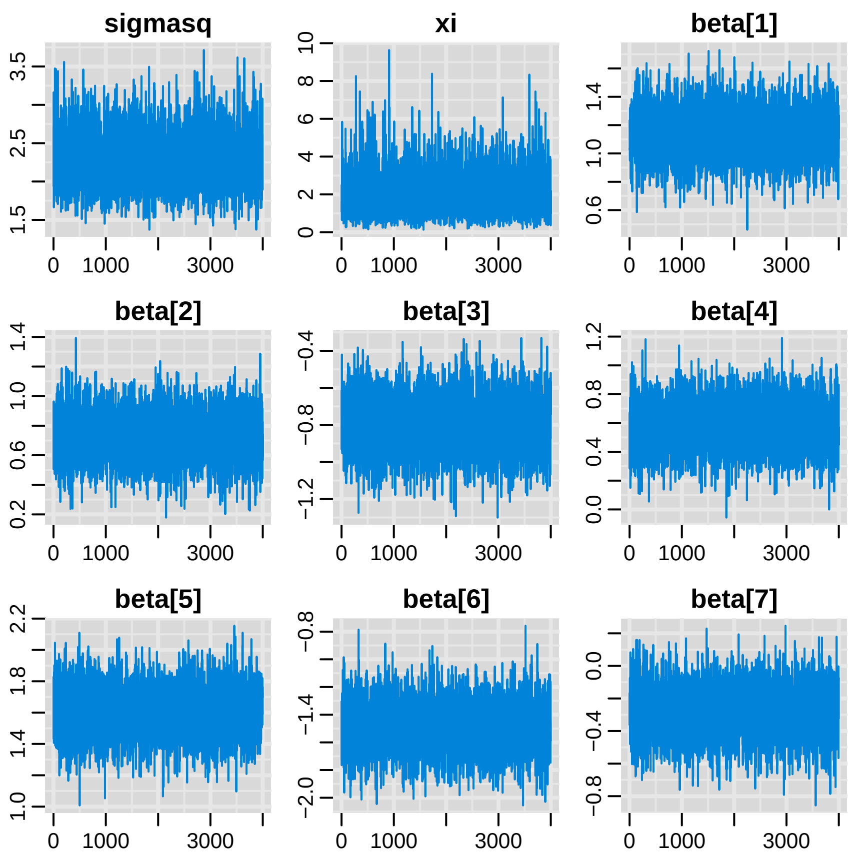
```
val = BayesRidge(y,X,sig_MH=0.9,a0=0.5,b0=0.5,BURNIN=5000,MCMC=20000,THIN=5)
val$postmean_beta                                  # posterior mean of beta
```

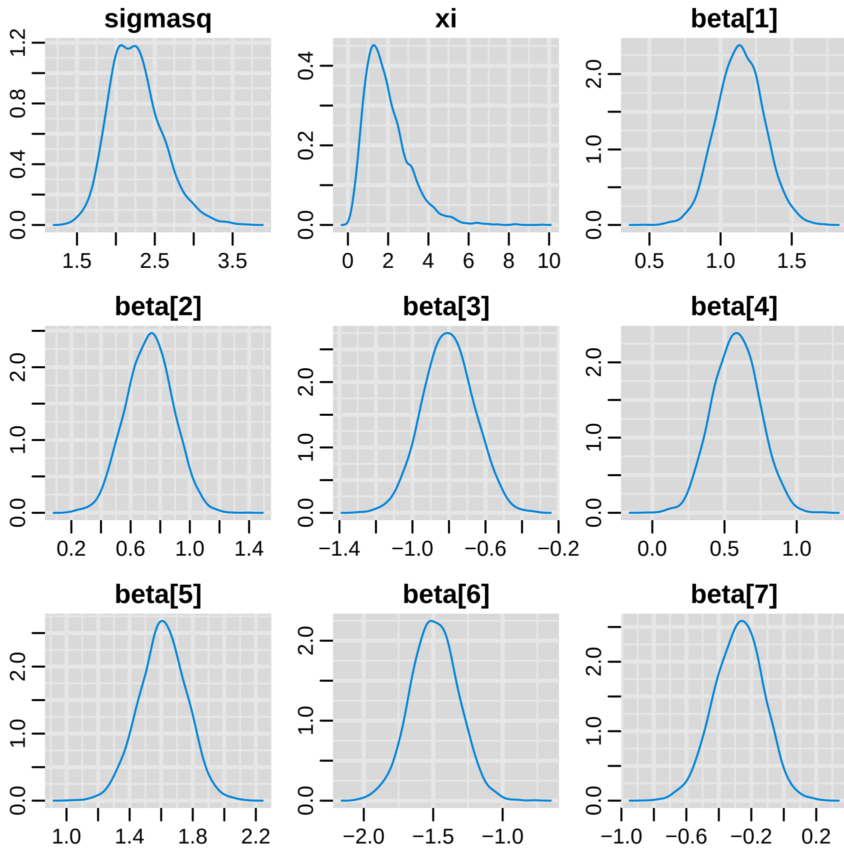
```
## [1]  1.1395297  0.7274015 -0.8030683  0.5795304  1.6096359 -1.4903483
## [7] -0.2765663
```

```
MLfit = glm.fit(X,y,family=gaussian())
mle_beta = MLfit$coefficients                      # OLS estimator of beta
mle_beta
```

```
## [1]  1.1737237  0.7425196 -0.8283323  0.5947388  1.6433100 -1.5253602
## [7] -0.2788524
```

Let us look at various plots.





Our second example. We now introduce correlation between certain predictors.

```
set.seed(1287)
n = 100                                     # sample size
d = 7                                      # predictors
betastarm = runif(d,min=0.25,max=1.75)
sgn = 2*rbinom(d,1,0.5) - 1
betastar = betastarm*sgn                    # true regression parameters
X = matrix(rnorm(n*d),nrow=n)              # design matrix drawn from iid N(0,1)
X[,3] = sqrt(0.98)*X[,1] + sqrt(0.02)*X[,3]
X[,4] = sqrt(0.95)*X[,1] + sqrt(0.05)*X[,4]
#X[,6] = sqrt(0.9)*X[,2] + sqrt(0.1)*X[,6]
mustar = X%*%betastar                       # true mean
sigstar = 1.50                              # true standard deviation
y = mustar + sigstar*rnorm(n)               # generate response from linear model
betastar
```

```
## [1]  1.1296797  0.9847218 -0.5495943  0.7871724  1.5597676 -1.7439878
## [7] -0.4780140
```

Let us now fit the Bayesian ridge regression model with the exact same specifications.

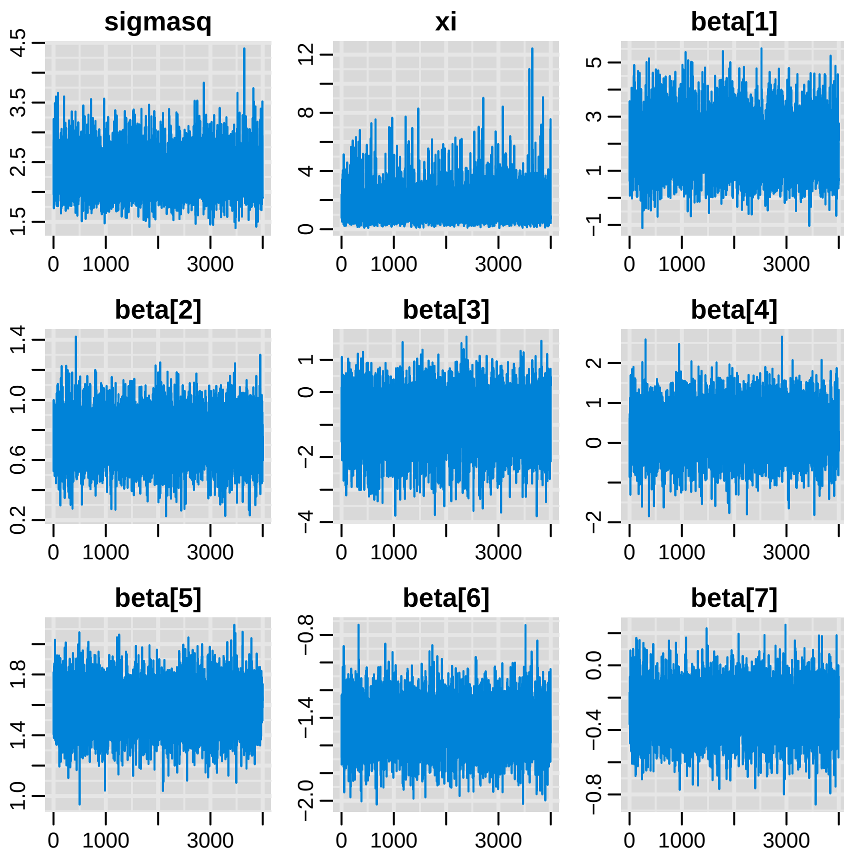
```
val = BayesRidge(y,X,sig_MH=0.9,a0=0.5,b0=0.5,BURNIN=5000,MCMC=20000,THIN=5)
val$postmean_beta                                      # posterior mean of beta
```

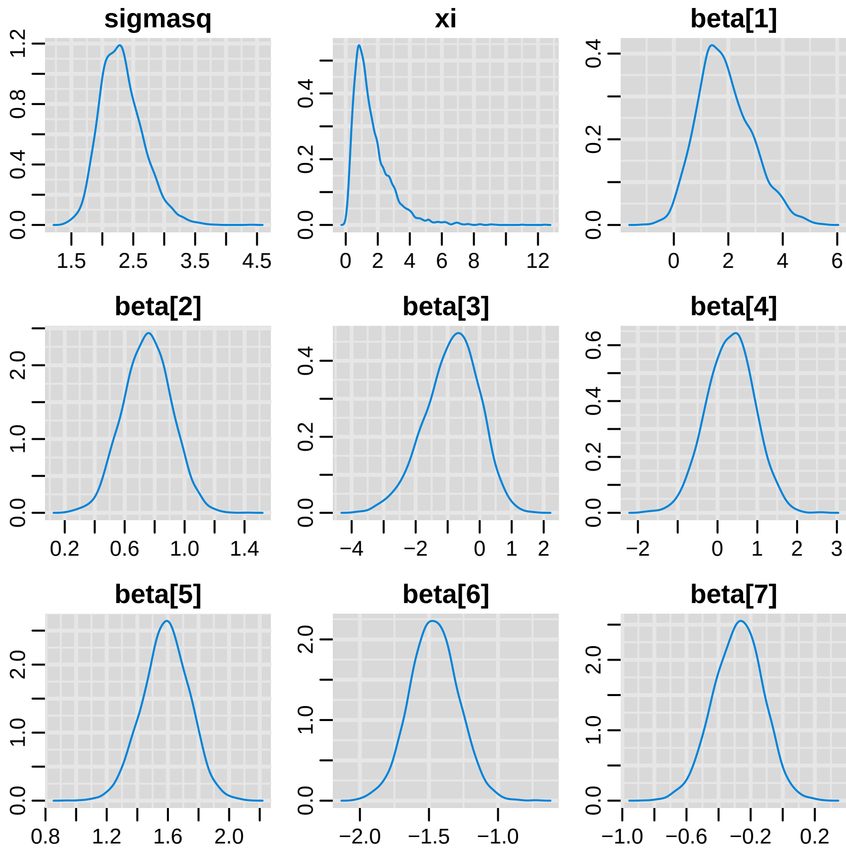
```
## [1]  1.8745397  0.7512194 -0.8654315  0.3243732  1.5866312 -1.4624520
## [7] -0.2773316
```

```
MLfit = glm.fit(X,y,family=gaussian())
mle_beta = MLfit$coefficients                              # OLS estimator of beta
mle_beta
```

```
## [1]  3.96368873  0.74251963 -2.52057002 -0.07341679  1.64331001 -1.52536017
## [7] -0.27885240
```

Let us look at various plots.





Our third example. We now introduce yet more correlation between certain predictors.

```
set.seed(1287)
n = 100                                        # sample size
d = 7                                         # predictors
betastarm = runif(d,min=0.25,max=1.75)
sgn = 2*rbinom(d,1,0.5) - 1
betastar = betastarm*sgn                       # true regression parameters
X = matrix(rnorm(n*d),nrow=n)                  # design matrix drawn from iid N(0,1)
X[,3] = sqrt(0.98)*X[,1] + sqrt(0.02)*X[,3]
X[,4] = sqrt(0.95)*X[,1] + sqrt(0.05)*X[,4]
X[,6] = sqrt(0.9)*X[,2] + sqrt(0.1)*X[,6]
mustar = X%*%betastar                          # true mean
sigstar = 1.50                                 # true standard deviation
y = mustar + sigstar*rnorm(n)                  # generate response from linear model
betastar
```

```
## [1]  1.1296797  0.9847218 -0.5495943  0.7871724  1.5597676 -1.7439878
## [7] -0.4780140
```

Let us now fit the Bayesian ridge regression model with the exact same specifications.

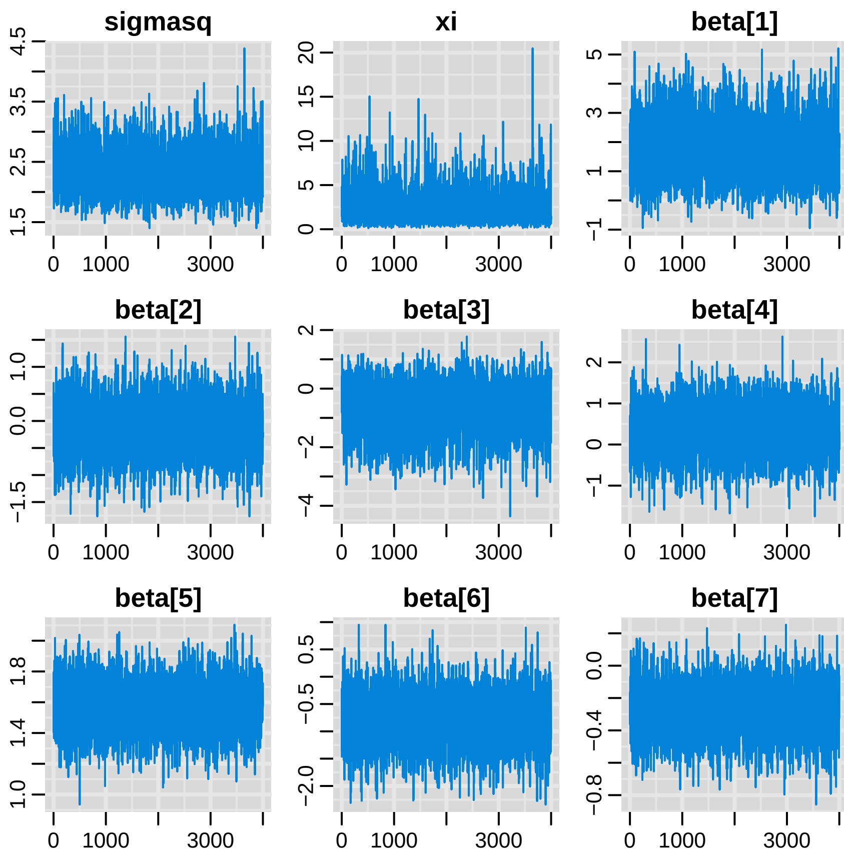
```
val = BayesRidge(y,X,sig_MH=0.9,a0=0.5,b0=0.5,BURNIN=5000,MCMC=20000,THIN=5)
val$postmean_beta                                        # posterior mean of beta
```

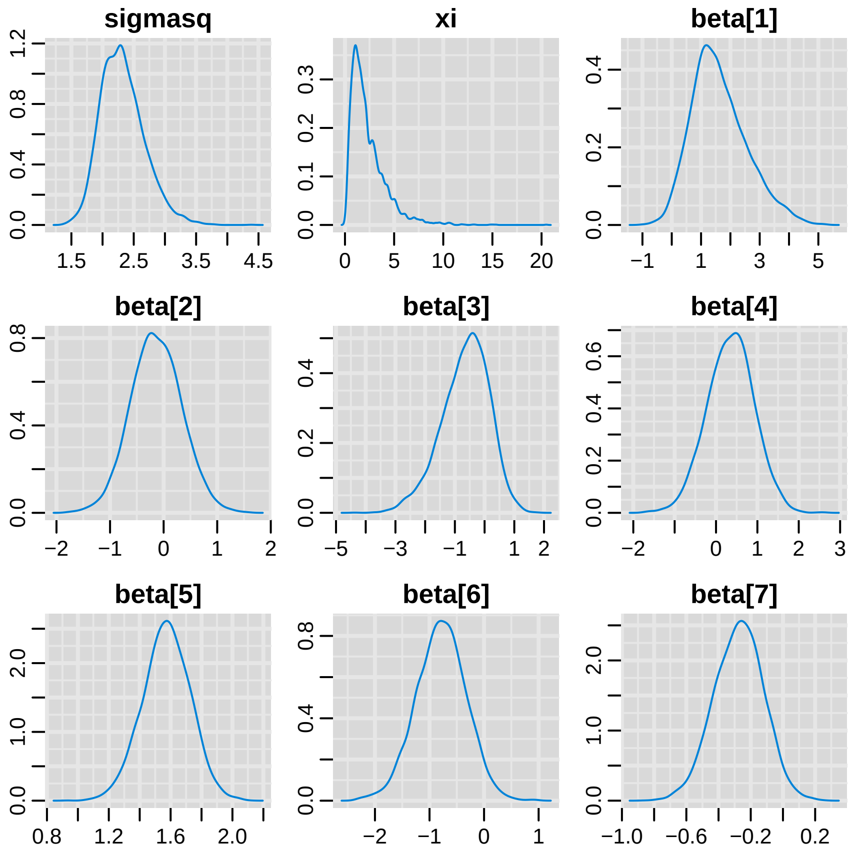
```
## [1]  1.6247032 -0.1359704 -0.6527097  0.3528157  1.5723459 -0.7878050
## [7] -0.2745075
```

```
MLfit = glm.fit(X,y,family=gaussian())
mle_beta = MLfit$coefficients                      # OLS estimator of beta
mle_beta
```

```
## [1]  3.96368873  0.08663676 -2.52057002 -0.07341679  1.64331001 -1.05262654
## [7] -0.27885240
```

Let us look at various plots.




sigmasq
xi
beta[1]
beta[2]
beta[3]
beta[4]
beta[5]
beta[6]
beta[7]