Bayesian Lasso

[STAT545 Final Project]

Xiangyu Xu 0029099478

1. Introduction

Lasso (least absolute shrinkage and selection operator) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model. What Lasso does is to force the sum of the absolute value of the regression coefficients to be less than a fixed value, which forces certain coefficients to be set to zero, effectively choosing a simpler model that does not include those coefficients.

Normally, we use cross validation to get a reasonable value of lambda, and then solve the Lasso function. But in Bayesian interpretation, Lasso can be interpreted as linear regression for which the coefficients have Laplace prior distributions, and the Lasso estimates can be seen as posterior mode estimates, which provide a new though for solving Lasso.

So, in this project, we will explore Bayesian Lasso by applying two methods:

1. Use MCEM to get the lambda, and then apply Gibbs Sampling to solve the coefficient estimates.
2. Place a hyperprior on Lambda, and sample Lambda from its conditional distribution.

After model coded, we design a simulation to prove our algorithm is right. Also, we apply the simulation data to normal Lasso which is solved by cross validation, to see the difference of two results.

2. Model Formulation

Park and Casella (2008) forms the following model as the full Bayesian Lasso linear regression model. Since the μ is usually given, the parameters we interested in are β, σ and τ.

$$\boldsymbol{y} \mid \mu, \boldsymbol{X}, \beta, \sigma^2 \sim N_n(\mu \mathbf{1}_n + \boldsymbol{X}\beta, \sigma^2 \boldsymbol{I}_n)$$

$$\beta \mid \tau_1^2, \dots, \tau_p^2, \sigma^2 \sim N_p\left(\mathbf{0}_p, \sigma^2 \boldsymbol{D}_\tau\right), \qquad \boldsymbol{D}_\tau = diag(\tau_1^2, \dots, \tau_p^2)$$

$$\tau_1^2, \dots, \tau_p^2 \sim \prod_{j=1}^{p} \frac{\lambda^2}{2} e^{-\frac{\lambda^2 \tau_j^2}{2}} d\tau_j^2, \quad \tau_1^2, \dots, \tau_p^2 > 0$$

$$\sigma^2 \sim \pi(\sigma^2) d\sigma^2$$

Note that σ and τ are independent. If we integrate the β over τ, we can find the conditional prior is:

$$\pi(\beta|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\frac{\lambda|\beta_j|}{\sqrt{\sigma^2}}}$$

3. Gibbs Sampling

Given the full model from the previous section, we have multiple parameters unknown. In the Gibbs sampling for Bayesian Lasso, the basic idea is that we only sample one of them each time conditioned on the rest of the parameters. Then update the current parameter value and sample the next.

To do this, we first assume we have some prior for μ, then with the model and prior from section 2, we can get a joint density of $\boldsymbol{y} \mid \mu, \beta, \sigma^2$. Consequently, we can get a conditional distribution of μ. Take the integration of such distribution will give us a joint density (function about β, σ^2 *and* τ) depends on *y*.

3.1 Sample β

According to Trevor and George (2008), if we let $\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{D}_\tau^{-1}$ and take the exponential terms involving β, it will be something like:

$$\beta^T\boldsymbol{A}\beta - 2\tilde{y}\boldsymbol{X}\beta + \tilde{y}^T\tilde{y} = (\beta - \boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{y})^T\boldsymbol{A}(\beta - \boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{y}) + \tilde{y}^T(\boldsymbol{I}_n - \boldsymbol{X}\boldsymbol{A}^{-1}\boldsymbol{X}^T)\tilde{y}$$

With the above equation, we can consider β as a multivariate Gaussian which has mean $\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{y}$ and variance $\sigma^2\boldsymbol{A}^{-1}$.

Multivariate Gaussian generation

First, assume that we have a way to generate independent and identical Gaussians. (e.g. Using Beasley Springer Moro method). Then, assume we have some constant c_{ij} and set:

$$x_1 = c_{11}z_1$$

$$x_2 = c_{21}z_1 + c_{22}z_2$$

$$\dots$$

$$x_d = c_{d1}z_1 + c_{d2}z_2 + \cdots + c_{dd}z_d$$

Because the sum of Gaussians are still Gaussians, these are some Gaussians centered at 0. To get them centered correctly, simply add the corresponding means to x_i. To adjust the variances, the Cholesky decomposition says, for a symmetric positive definite matrix **Σ** (here is the covariance matrix), there exists a unique decomposition of **Σ** into the product of a lower triangular matrix and its transpose. That lower triangular matrix is the coefficient matrix $\boldsymbol{C}$.

Algorithm to generate from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance **Σ**.

i. Generate some independent and identically distributed standard Normal $z_1, z_2, \dots, z_d \sim N(0,1)$.
ii. Use the Cholesky decomposition to decompose the variance-covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T$.
iii. Set $\boldsymbol{c} = \boldsymbol{\Lambda}$.
iv. Set $x_i = c_{i1}z_1 + c_{i2}z_2 + \cdots + c_{ii}z_i + \mu_i$ for all $i = 1,2,\dots,d$.
v. Return $(x_1, x_2, \dots, x_d)$

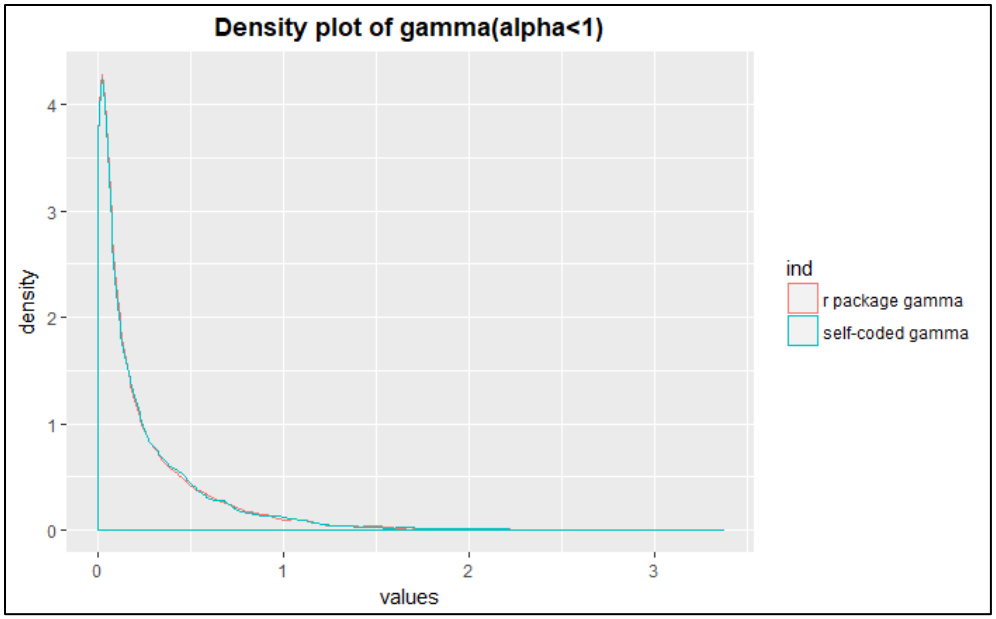

Figure 1

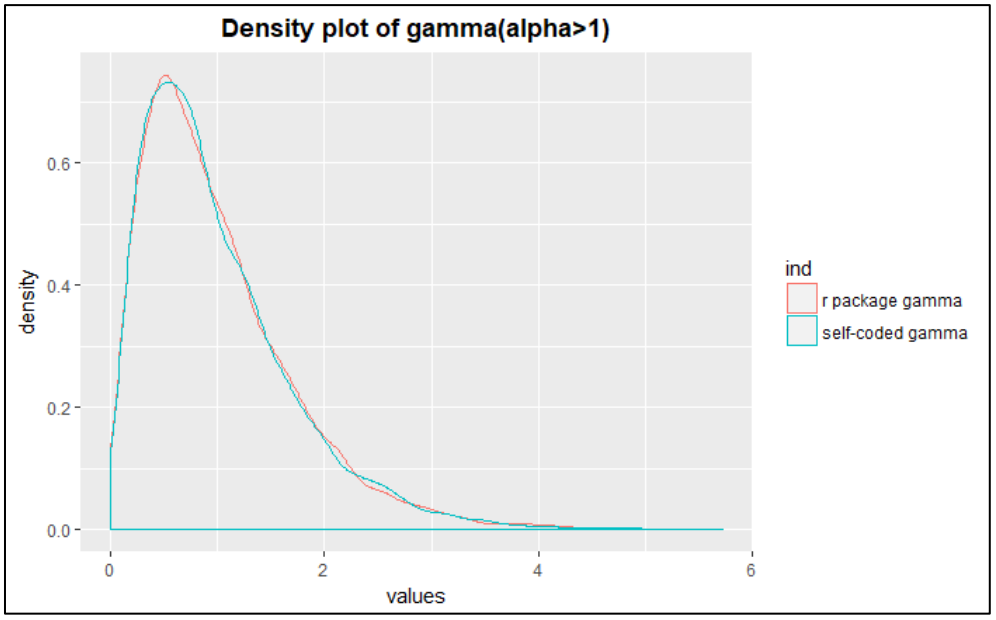

Figure 2

3.2 Sample σ^2

We can consider σ^2 as an inverse Gamma which has scale parameter $\frac{(\tilde{y}-X\beta)^T(\tilde{y}-X\beta)}{2} + \frac{\beta^T D_\tau^{-1}\beta}{2} + \gamma$ and shape parameter $\frac{n-1}{2} + \frac{p}{2} + a$.

Inverse Gamma generation

Algorithm to generate from an inverse Gamma distribution

i. Set d $= 1.0334 - 0.0766e^{2.2942\alpha}$, $a = 2^{\alpha}\left(1 - e^{-\frac{d}{2}}\right)^{\alpha}$, $b = \alpha d^{\alpha-1}e^{-d}$, c = a + b.

ii. Generate a random variate from the Uniform distribution q ~ U(0,1).

iii. If q $\leq \frac{a}{a+b}$, then set X $= -2\ln[1 - \frac{(cq)^{\frac{1}{\alpha}}}{2}]$, else set X $= -\ln[\frac{c(1-q)}{\alpha d^{\alpha-1}}]$.

iv. Generate a random variate from the Uniform distribution p ~ U(0,1) identically independent to q.

v. When X $\leq$ d, if p $\leq \frac{X^{\alpha-1}e^{-X/2}}{2^{\alpha-1}(1-e^{-X/2})^{\alpha-1}}$, return 1/X; else go to step ii.

vi. When X $>$ d, if p $\leq (\frac{d}{X})^{1-\alpha}$, return 1/X; else go to step ii.

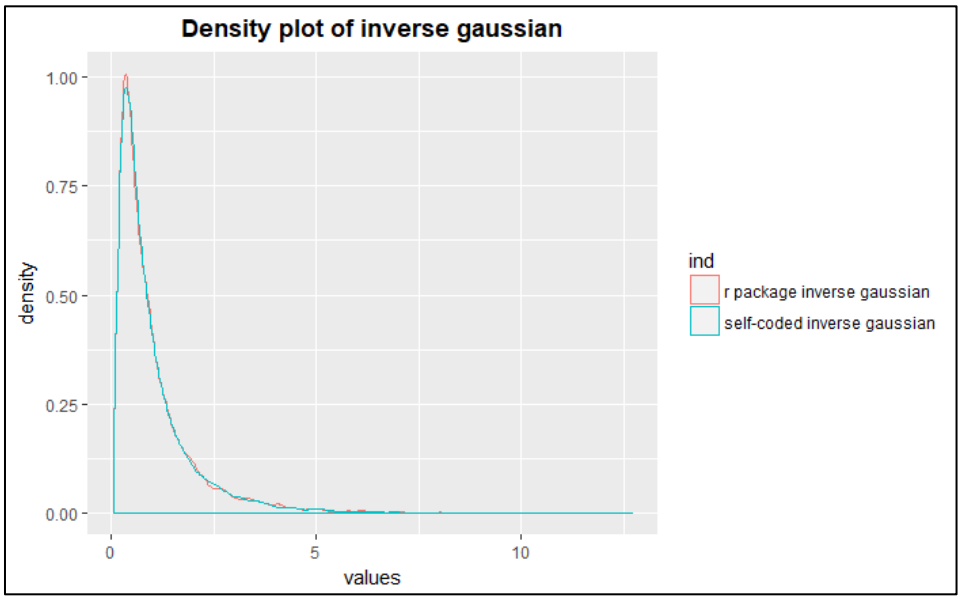

Figure 3

3.3 Sample τ_j^2

Instead of sampling τ_j^2 directly, we can consider $1/\tau_j^2$ as an inverse Gaussian which has mean $\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}}$ and variance λ^2.

Inverse Gaussian generation

Algorithm to generate from an inverse Gaussian distribution

i. Generate a random variate from the Standard Normal distribution p ~ $N(0,1)$.

ii. Set y $= p^2$

iii. Set x $= \mu + \frac{\mu^2 y}{2\lambda} - \frac{\mu}{2\lambda}\sqrt{4\mu\lambda y + \mu^2 y^2}$

iv. Generate a random variate from the Uniform distribution q ~ U(0,1).

v. If q $\leq \frac{\mu}{\mu+x}$, return x; else, return $\frac{\mu^2}{x}$.

4. Parameter Choosing

4.1 Monte Carlo EM

The parameter of the ordinary Lasso can be chosen by cross validation. But the Bayesian Lasso also offers some uniquely method, MCEM.

First, we initialize the parameter λ, then we generate from the posterior distribution of $\beta, \sigma^2, {\tau_1}^2, \dots, {\tau_p}^2$ by using the Gibbs Sampling like we did before. At the E-step approximate the expected log likelihood for λ by replacing average based on the Value of the parameters we get from the previous step. The log likelihood is:

$$-\left(\frac{\mathrm{n}+\mathrm{p}-1}{2}+\mathrm{a}+1\right)\ln(\sigma^2)-\frac{1}{\sigma^2}\left(\frac{(Y-X\beta)^T(Y-X\beta)}{2}+\gamma\right)-\frac{1}{2}\sum_{j=1}^{p}\ln({\tau_j}^2)$$

$$-\frac{1}{2}\sum_{j=1}^{p}\frac{{\beta_j}^2}{\sigma^2\tau_j^2}+p\ln(\lambda^2)-\frac{\lambda^2}{2}\sum_{j=1}^{p}\tau_j^2$$

At the M-step, we solve for λ^{k+1} to be the value of λ that maximize the log likelihood. Repeat these steps until convergence.

$$\lambda^{k+1}=\sqrt{2p/\sum_{j=1}^{p}E_{\lambda^{(k)}}\left[\tau_j^2|\tilde{y}\right]}$$

4.2 Set Hyperprior on Lambda

The Bayesian LASSO parameter Lambda can be chosen by using marginal maximum likelihood or an appropriate hyperprior. As it is suggested by Park and Casella (2008), the class of gamma priors on λ^2, which is

$$\pi(\lambda^2)=\frac{\delta^r}{\Gamma(r)}(\lambda^2)^{r-1}e^{-\delta\lambda^2},\ \ \lambda^2>0\ \ (r>0,\delta>0)$$

The improper scale-invariant prior $\frac{1}{\lambda^2}$ for λ^2 (formally obtained by setting r = 0 and δ = 0) is a tempting choice. The parameter δ must be sufficiently larger than zero to avoid computational and conceptual problems.

The full conditional distribution of λ^2 is gamma with shape parameter p + r and rate parameter $\sum_{i=1}^{p}\frac{{\tau_i}^2}{2}+\delta$. With the specification above, λ can simply join the hierarchical model we mentioned before with the conditional distribution of other parameters unchanged. And we simply set r = 0, $\delta = 1$.

5. Simulation

To test the model constructed above, we to run a simulation to see if this Bayesian lasso model can shrink the less contribution coefficients. The model is as such: we randomly generate values of X, which is a 50 by 200 matrix. Y is a 50 by 1 matrix, which is the response of x_1, x_2, x_3 and fairly large noise, which means in fact only x_1, x_2, x_3 contribute to the reponse.

```
n=200
m=50
x=matrix(rep(0,m*n),nrow=n)
y=rep(0,n)
for (i in 1:n){
  z=rnorm(1,0,1)
  for(j in 1:m){
    x[i,j]=(z+rnorm(1,0,1))/2
  }
  y[i]=1*x[i,1]-1*x[i,2]+0.5*x[i,3]+0.6*rnorm(1,0,1)
}
```

5.1 Simulate with Bayesian Lasso

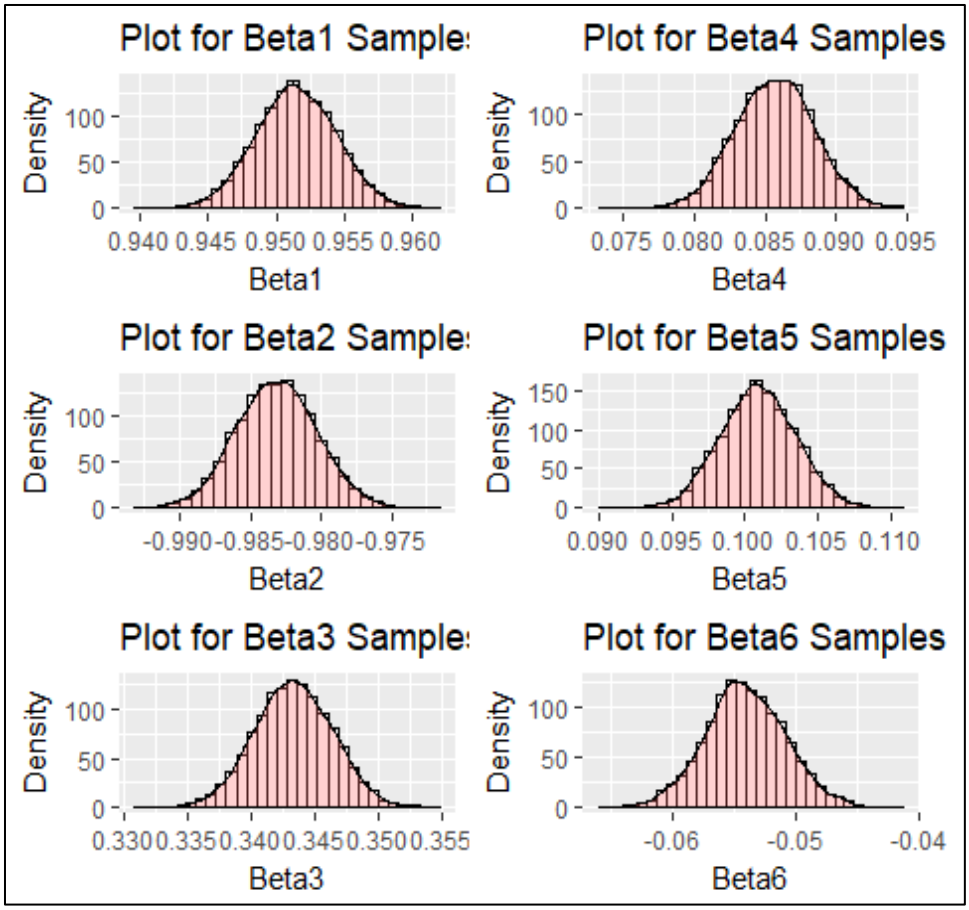

Figure 4

We plot the Beta samples based on 10000 Gibbs samples with 3000 burn-in period. Due to the limitation of the length of the report, we only display the first six Betas and they follow the Laplace distribution. See Figure 4.

From figure 5, it is obvious that only the first three coefficients are significantly larger than zero while others lingering around 0, which sufficiently support the model efficiency.

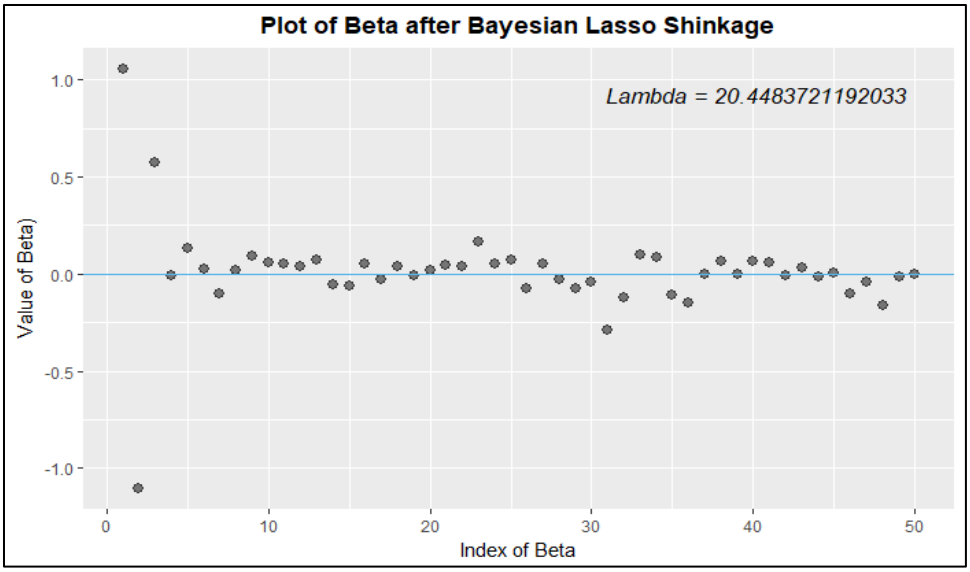

Figure 5

5.2 Simulate with R package "glmnet"

We try to compare the Bayesian Lasso with ordinary Lasso which implements n fold cross validation. We use R package 'glmnet' to get the lambda, which is around 0.04, while the lambda we got from Bayesian is about 20.

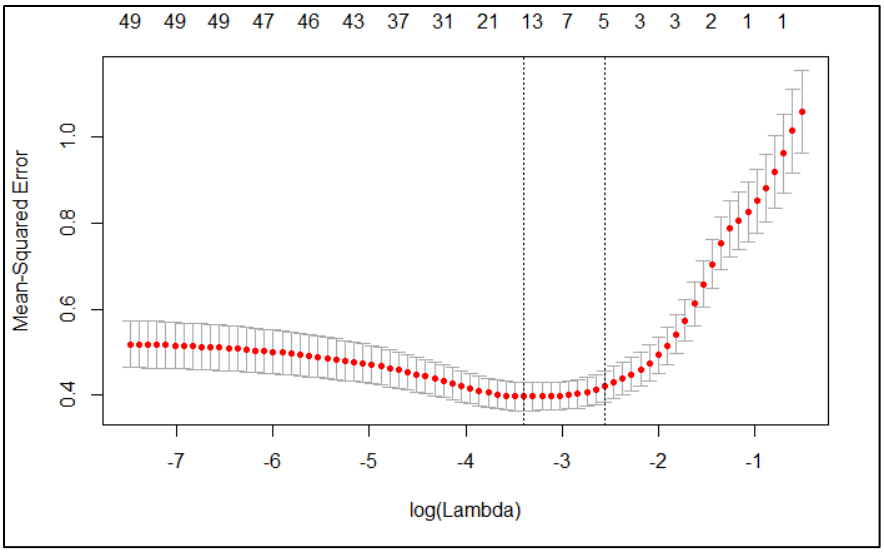

Figure 6

This is because in Bayesian Lasso:

$$\text{argmax}\ \mathrm{p}(\beta|\alpha) \propto \mathrm{p}(\mathrm{x}|\beta)\ p(\beta)$$

$$\text{argmax}\quad \log\big(\mathrm{p}(\beta|x)\big) \propto \log\big(p(x|\beta)\big) + \log\big(p(\beta)\big)$$

$$\propto\ -\|Y - X\beta\|^2 - \lambda|\beta|$$

But in Lasso (glmnet):

$$\text{argmin } \frac{1}{2n} \|Y - X\beta\|^2 + \lambda|\beta|$$

Which means if we divide the beta we got from Bayesian Lasso, it will be 0.05112, which is remarkably close to the lambda.

6. Personal Summary

What I did in this project is mainly about set hyperprior on lambda and do Gibbs Sampling to get beta estimate; then run simulation to check the answer, visualize data, write report.

What I learned in this project:

1. In MCEM, it is really hard to get convergence. So we can calculate the likelihood ratio then find the best lambda with the largest loglikelihood ratio.
2. When I was doing the simulation, I adjusted the value of n and p. It turns out that Bayesian Lasso does not perform well on small n large p problem or small p large n problem.
3. The method we used in homework for sampling from Gamma distribution has relatively high rejection rate.

Appendix

References

[1] Trevor Park & George Casella (2008) The Bayesian Lasso, Journal of the American Statistical Association, 103:482, 681-686, DOI: 10.1198/016214508000000337

[2] CHRIS HANS Bayesian lasso regression Biometrika, Vol. 96, No. 4 (DECEMBER 2009), pp. 835-845 http://www.jstor.org/stable/27798870?seq=1&cid=pdf-reference#references_tab_contents

[3] Debasis Kundu & Rameshwar D. Gupta A Convenient Way of Generating Gamma Random Variables Using Generalized Exponential Distribution https://pdfs.semanticscholar.org/5294/12dffa396771edece9f73dc83b0e80f12858.pdf

[4] Wikipedia https://en.wikipedia.org/wiki/Inverse_Gaussian_distribution#Generating_random_variates_from_an_inverse-Gaussian_distribution