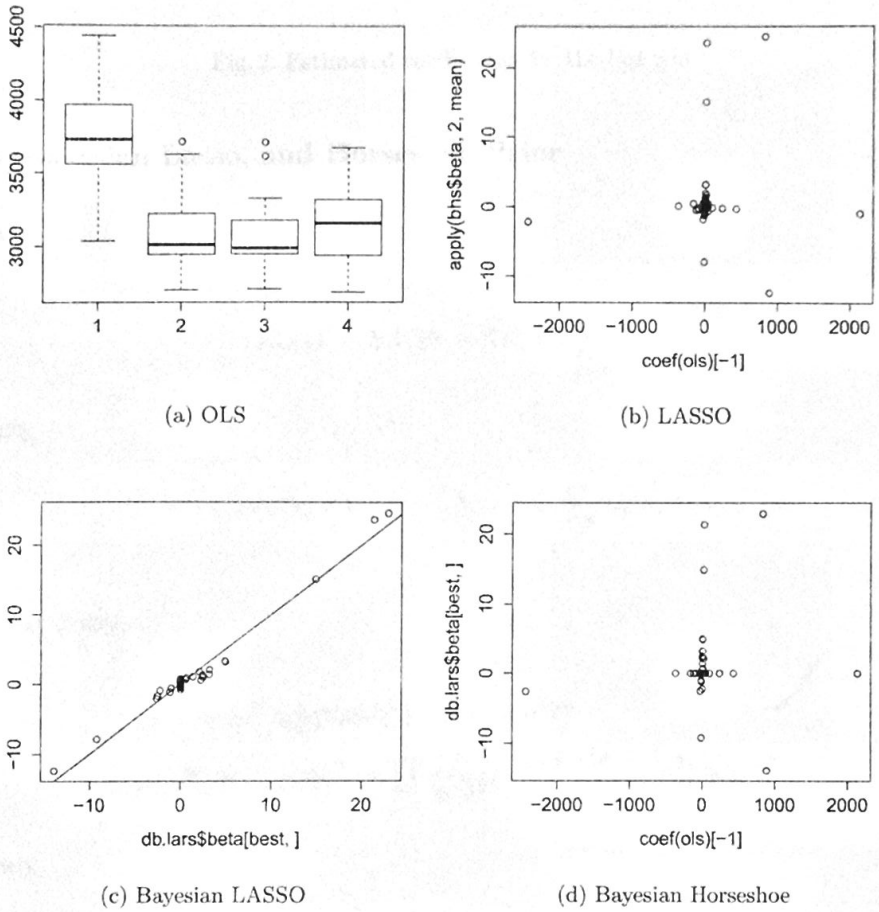
SDS 386D HW 4

Maurice Diesendruck, Sukyung Park, Bowei Yan, Michael Zhang

Department of Statistics and Data Sciences

1 Plots



(a) OLS

(b) LASSO

(c) Bayesian LASSO

(d) Bayesian Horseshoe

Fig. 1: MSE of OLS, LASSO, Bayesian LASSO, Horseshoe

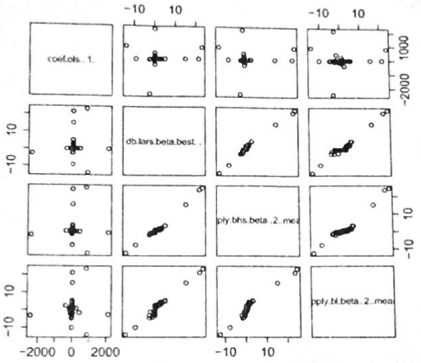


Fig. 2: Estimated coefficients for the last run

2 Lasso, Bayesian Lasso, and Horseshoe Prior

2.1 Lasso

$$\min\ (Y - X\beta)'(Y - X\beta) + k\sum_{j=1}^{p}|\beta_j|$$

or equivalently,

$$\min\ (Y - X\beta)'(Y - X\beta)\ \text{ s.t. }\ \sum_{j=1}^{p}|\beta_j| \le t$$

2.2 Bayesian Lasso

$$\text{Likelihood}:\ p(y|X,\beta,\sigma^2) = N(X\beta,\sigma^2 I)$$

$$\text{Priors}:\ p(\beta|\sigma^2) = \prod_{j=1}^{p}\frac{\lambda}{2\sqrt{\sigma^2}}e^{-\lambda|\beta_j|/\sqrt{\sigma^2}},\ \ p(\sigma^2)\propto 1/\sigma^2$$

or equivalently,

$$p(y|X,\beta,\sigma^2) = N(X\beta,\sigma^2 I)$$
$$p(\beta_j|\sigma^2,\tau_j^2) = N(0,\sigma^2\tau_j^2),\ \ p(\sigma^2)\propto 1/\sigma^2$$
$$p(\tau_j^2) = Exp(\lambda^2/2)$$

2.3 Horseshoe Prior

$$
\begin{aligned}
p(y|\theta) &= N(\theta, \sigma^2 I) \\
p(\theta_i|\lambda_i) &= N(0, \lambda_i^2) \\
p(\lambda_i|\tau) &= C^+(0, \tau) \\
p(\tau) &= C^+(0, \sigma)
\end{aligned}
$$

where C^+ is a half-Cauchy distribution on the positive reals.

1 Lasso, Bayesian Lasso, and Horseshoe Prior

1.1 Strengths and Weaknesses

The strength for lasso is its simplicity and efficiency compared to Bayesian approaches. And by tuning λ it's easy to see the entire solution path. The strength for Bayesian lasso is we could get the posterior distribution instead of a single point estimation. But for double-exponential prior, small values of τ can lead to strong shrinkage near the origin, and could severely compromise performance in the tails. When θ is sparse, estimation of τ under the double-exponential model must balance between risk due to under-shrinking noise and risk due to over-shrinking large signals. The horseshoe prior requires no compromise of this sort and have better tail robustness for large signals.

SDS 387 HW 4

Maurice Diesendruck, Sukyung Park, Bowei Yan, Michael Zhang

March 25, 2015

3 Comparison

This is an analysis of the LASSO, BLASSO, and Horseshoe priors on diabetes data.

1. LASSO
   The LASSO is an L1-penalized least squares estimate of β, with $\tilde{y}$ as the mean-centered outcome variable, where the following is optimized:

$$\min_{\beta}(\tilde{y} - XB)'(\tilde{y} - XB) + \lambda \sum_{j=1}^{p} |\beta_j|$$

   (a) Strengths
      i. Easy to implement.
   (b) Weaknesses
      i. Proposed standard error estimators are not considered satisfactory.

2. BLASSO
   The BLASSO interprets the L1-penalty as a prior on β and σ^2. In this case, y is normal, the prior $\pi(\beta|\sigma^2)$ is Laplace (i.e. double exponential), and the prior $\pi(\sigma^2)$ is non-informative and defined as such to ensure a unimodal posterior:

$$\begin{aligned} y|\mu, X, \beta, \sigma^2 &\sim N(\mu 1_n + X\beta, \sigma^2 I_n) \\ \pi(\beta|\sigma^2) &= \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}} \quad \text{(Laplace distribution)} \\ \pi(\sigma^2) &= \frac{1}{\sigma^2} \end{aligned}$$

   (a) Strengths
      i. Easy to implement
      ii. Automatically provides interval estimates for all parameters, including the error variance.
   (b) Weaknesses
      i. More computationally intensive than LASSO.

3. Horseshoe
   The Horseshoe (HS) prior assumes $y|\theta \sim N(\theta, \sigma^2 I)$, and aims to (1) estimate θ, and (2) predict future realizations of y. It is especially useful in cases where most covariates are nearly zero (i.e. sparse θ). The model is as follows:

$$\begin{aligned} \theta_i|\lambda_i &\sim N(0, \lambda_i^2) \\ \lambda_i|\tau &\sim C^+(0, \tau) \\ \tau &\sim C^+(0, \sigma) \\ E(\theta_i|y) &= \int_0^1 (1 - \kappa_i) y_i p(\kappa_i|y) \mathrm{d}\kappa_i = (1 - E(\kappa_i|y)) y_i \end{aligned}$$

   Where $\kappa_i = 1/(1 + \lambda_i^2)$, assuming fixed values $\sigma^2 = \tau^2 = 1$. Given $\lambda_i \geq 0$, this implies that λ_i's are indirectly related to amount of shrinkage (e.g. high λ_i means low κ_i, and less shrinkage).

(a) Strengths

i. Easy to implement.

ii. Has fewer hyperparameters.

iii. Robust and flexible to high or low sparsity situations.

iv. Converges efficiently.

(b) Weaknesses

i.

ii.

4 Perform Gibbs Sampling for Bayesian Lasso

Reference: Park and Casella (2008) (Section 2).

1. Model and Prior for BLASSO
   See 3.2, above. The full model is as follows:
   Again, the Laplace can be re-written as the integrated product of a Normal and Exponential distribution:

$$\begin{aligned}\pi(\beta|\sigma^2) &= \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}} && \text{(Laplace distribution)}\\ &= \prod_{j=1}^{p} \int_0^{\infty} \text{Normal * Exponential}\end{aligned}$$

   This way, full model becomes:

$$\begin{aligned} y|\mu, X, \beta, \sigma^2 &\sim N(\mu 1_n + X\beta, \sigma^2 I_n),\\ \beta|\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim N_p(0_p, \sigma^2 D_\tau), && \text{(Normal part of Laplace)}\\ \text{where } D_\tau &= diag(\tau_1^2, \ldots, \tau_p^2),\\ \sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim \pi(\sigma^2) d\sigma^2 \prod_{j=1}^{p} \frac{\lambda^2}{2} e^{\lambda^2 \tau_j^2/2} d\tau_j^2, && \text{(Exponential part of Laplace)}\\ \sigma^2, \tau_1^2, \ldots, \tau_p^2 &> 0. \end{aligned}$$

2. Complete Conditional for each (set of) Parameters

$$\begin{aligned} \beta|\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim N_p((X'X + D_\tau^{-1})X'\tilde{y},\ \sigma^2(X'X + D_\tau^{-1}))\\ \sigma^2|\beta, \tau_1^2, \ldots, \tau_p^2 &\sim InvGamma(\frac{n-1}{2} + \frac{p}{2},\ \frac{(\tilde{y} - X\beta)'(\tilde{y} - X\beta)}{2} + \frac{\beta' D_\tau^{-1} \beta}{2}\end{aligned}$$

   $\tau_1^2, \ldots, \tau_p^2$ are conditionally independent, with $\frac{1}{\tau_j^2}$ sampled from the Inverse-Gaussian:

$$\frac{1}{\tau_j^2} \sim InvGauss(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\ \lambda^2) \quad \text{(Useful: rinvgauss, in package "statmod")}$$

SDS 387 HW 4

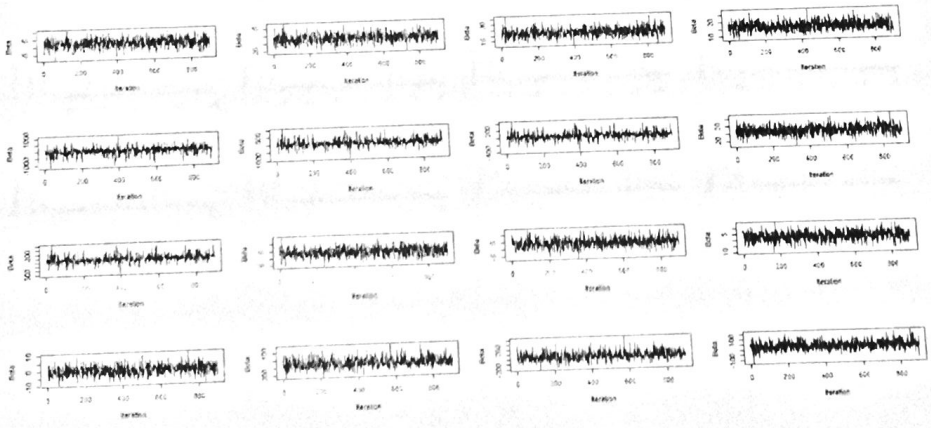
Michael Zhang

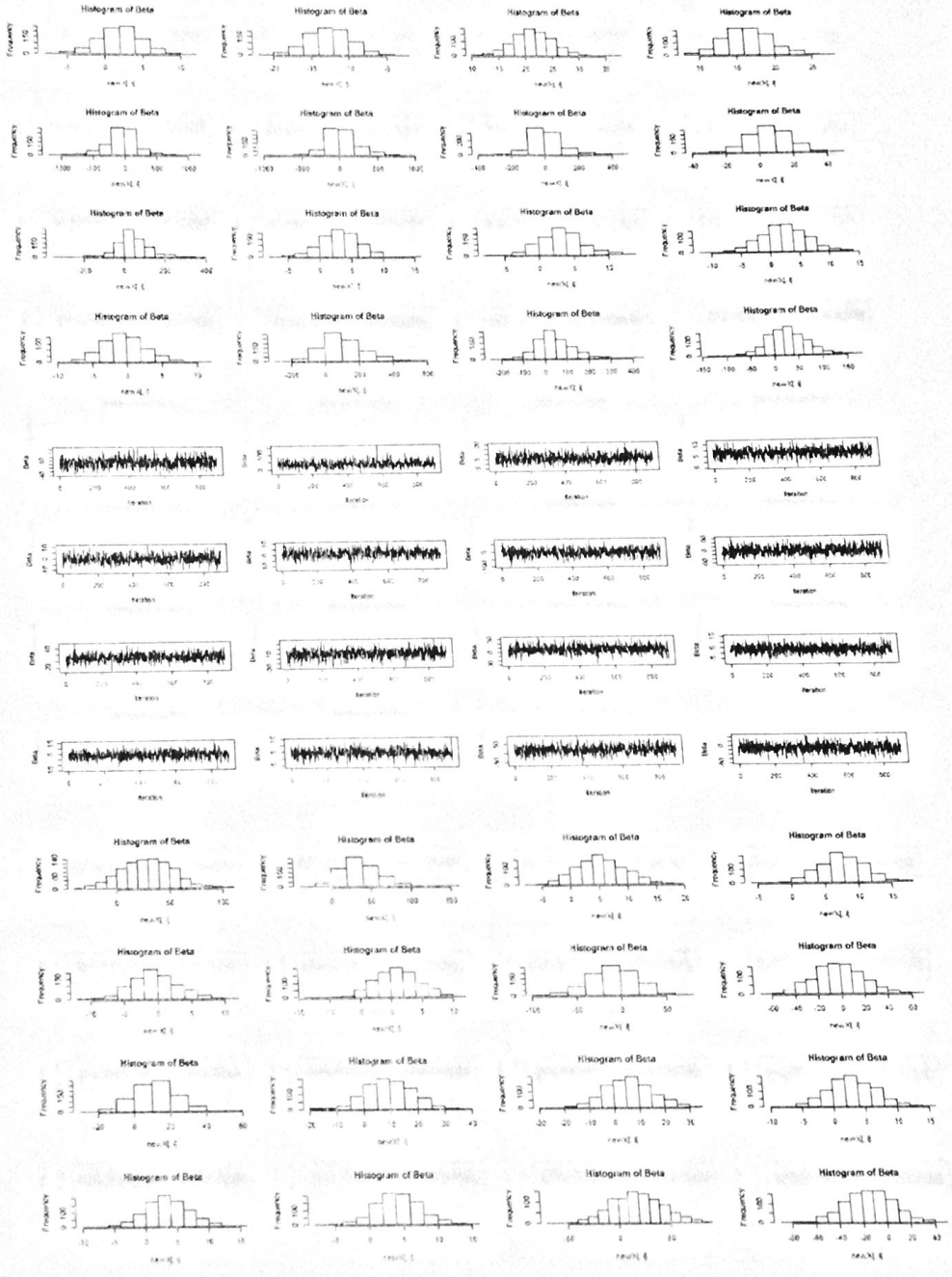
March 29, 2015

4.3. The following is our Gibbs sampler:

```
> for(i in 2:iterations){
+   A = xTx + solve(D_tau)
+   beta_mean = solve(A) %*% xTy
+   beta_var = sigma2 * solve(A)
+   beta = mvrnorm(1, beta_mean, beta_var)
+   beta_chain[,i] = beta
+   gamma_a = (n+p) / 2
+   gamma_b = as.numeric((1/2) *
+                        (t(yf - Xf %*% beta) %*%
+                           (yf - Xf %*% beta) +
+                           t(beta) %*% solve(D_tau) %*%
+                           beta))
+   sigma2 = 1 / rgamma(1, gamma_a, gamma_b)
+   sigma_chain[i] = sigma2
+   invgaus_a = sqrt((lambda_prior^2 * sigma2)/beta^2)
+   tau2 = 1/ sapply(invgaus_a, rinvgauss, n=1, shape=lambda_prior^2)
+   tau_chain[,i] = tau2
+   D_tau = diag(tau2)
+ }
```

The resulting output shows the following traceplots:




Histogram of Beta
Frequency
Beta
Iteration