Efficient sampling for Gaussian linear regression with arbitrary priors

P. Richard Hahn (Arizona State), Jingyu He (Chicago Booth), Hedibert Lopes (INSPER)

November 21, 2018
Motivation

Goal: run a Gaussian linear regression and a new prior

- Do math, try to write a new Gibbs sampler.
- Maybe hard to find easy conditional distributions.
- Probably require data augmentation, add lots of latent variables.

Why not take advantage of the **Gaussian** likelihood?

**Elliptical slice sampler** can be extended to arbitrary priors, as long as we can evaluate the prior density.
Outline

Brief review of (Bayesian) regularization
   - Ridge and Lasso regressions
   - Shrinkage priors

Algorithm 1: Elliptical slice sampler
Algorithm 2: Elliptical slice-within-Gibbs Sampler

Other issues
   - Computational considerations
   - Rank deficiency
   - Comparison metrics

Simulation exercise
   - $n > p$
   - $n < p$

Empirical illustrations
   - Illustration 1: Beauty and course evaluations
   - Illustration 2: Diabetes dataset
   - Illustration 3: Motorcycle data

References
Brief review of (Bayesian) regularization

Consider the Gaussian linear model

$$(y|X, \beta, \sigma^2) \sim N(X\beta, \sigma^2 I_n),$$

where $\beta$ is $p$-dimensional.

Ridge Regression: $\ell_2$ penalty on $\beta$:

$$\hat{\beta}_R = \arg \min_\beta \{||y - X\beta||^2 + \lambda||\beta||^2_2\}, \quad \lambda \geq 0,$$

leading to $\hat{\beta}_{ridge} = (X'X + \lambda I)^{-1}X'y$.

LASSO Regression: $\ell_1$ penalty on $\beta$:

$$\hat{\beta}_L = \arg \min_\beta \{||y - X\beta||^2 + \lambda||\beta||_1\}, \quad \lambda \geq 0,$$

which can be solved by using quadratic programming techniques such as a coordinate gradient descent algorithm.
Elastic net

The Elastic net combines the Ridge and the LASSO approaches:

\[ \hat{\beta}_{EN} = \arg \min_{\beta} \{ ||y - X\beta||^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||_2^2 \}, \lambda_1 \geq 0, \lambda_2 \geq 0, \]

The \( \ell_1 \) part of the penalty generates a sparse model.

The \( \ell_2 \) part of the penalty

- Removes the limitation on the number of selected variables;
- Encourages grouping effect;
- Stabilizes the \( \ell_1 \) regularization path.

R package \texttt{elasticnet}
Two dimension contour plots of the three penalty functions

Ridge (dot-dashed), LASSO (dashed) and Elastic net (solid)
Bayesian regularization

- Regularization and variable selection are done by assuming independent prior distributions from a scale mixture of normals (SMN) class:

\[
\beta | \psi \sim \mathcal{N}(0, \psi) \quad \text{and} \quad \psi \sim p(\psi),
\]

- The posterior mode or the maximum a posteriori (MAP) is

\[
\arg \max_{\beta} \{ \log p(y | \beta) + \log p(\beta | \psi) \}
\]

which is equivalent to penalizing the log-likelihood

\[
\log p(y | \beta)
\]

with penalty equal to the log prior

\[
\log p(\beta | \psi)
\]

when \( \psi \) is held fixed.
Bayesian regularization in linear regression problems

The marginal prior distribution of $\beta$

$$p(\beta) = \int_0^\infty p_N(\beta, 0, \psi) p(\psi) d\psi$$

can assume many forms depending on the mixing distribution $p(\psi)$:

<table>
<thead>
<tr>
<th></th>
<th>$p(\psi)$</th>
<th>$p(\beta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ridge</td>
<td>$\mathcal{IG}(\alpha, \delta)$</td>
<td>Student’s $t$</td>
</tr>
<tr>
<td>Lasso</td>
<td>$\mathcal{E}(\lambda^2/2)$</td>
<td>Laplace</td>
</tr>
<tr>
<td>NG prior</td>
<td>$\mathcal{G}(\lambda, 1/(2\gamma^2))$</td>
<td>Normal-Gamma</td>
</tr>
<tr>
<td>Horseshoe</td>
<td>$\sqrt{\psi} \sim \mathcal{C}^+(0, 1)$</td>
<td>No closed form</td>
</tr>
</tbody>
</table>

where,

$$p_{NG}(\beta|\lambda, \gamma) = \frac{1}{\sqrt{\pi}2^{\lambda-1/2}\gamma^\lambda+1/2\Gamma(\lambda)}|\beta|^{\lambda-1/2}K_{\lambda-1/2}(|\beta|/\gamma)$$

$$\log p_H(\beta) \approx \log \left(1 + \frac{4}{\beta^2}\right)$$
Comparing shrinkage priors
Comparing shrinkage priors

![Shrinkage priors graph](image)

- Ridge
- Laplace
- Elastic-Net
- Horseshoe
- Normal-Gamma
Algorithm 1: Elliptical Slice Sampler

- The original elliptical slice sampler (Murray et. al. [2010]) was designed to sampling from a posterior arising from a normal prior and a general likelihood.

- It can also be used with a normal likelihood and general prior such as shrinkage priors.

- Advantages:
  - **Flexible**: It only requires evaluating the prior density or an approximation (no special samplers are required).
  - **Fast**: Sample all coefficients simultaneously. Not necessary to loop over variables.
Advantages of our sampling scheme

**Flexibility:** $\pi(\beta)$ evaluated up to a normalizing constant.

**MC efficiency:** In each MC iteration, single multivariate Gaussian draw and several univariate uniform draws.

**Acceptance rate:** The size of the sampling region for $\theta$ shrinks rapidly with each rejected value and is guaranteed to eventually accept.

**Single/block move:** The basic strategy of the elliptical slice sampler can be applied to smaller blocks.
Algorithm 1: Elliptical slice sampler

Goal: Sample $\Delta$ from

$$p(\Delta) \propto p_N(\Delta; 0, V)L(\Delta)$$

Key idea: For $v_0$ and $v_i$ iid $N(0, V)$ and any $\theta \in [0, 2\pi]$, it follows that

$$\Delta = v_0 \sin \theta + v_1 \cos \theta \sim N(0, V).$$

Parameter-expansion: Sampling from

$$p(v_0, v_1, \Delta, \theta) \propto p_N(0, \Sigma_{\theta})L(v_0 \sin \theta + v_1 \cos \theta)$$

can be done via two-block Gibbs sampler:

- Sample from $p(v_0, v_1 | \Delta, \theta)$
  - Sample $v$ from $N(0, V)$
  - Set $v_0 = \Delta \sin \theta + v \cos \theta$ and $v_1 = \Delta \cos \theta - v \sin \theta$
- Sample from $p(\Delta, \theta | v_1, v_2)$
  - Slice sampling from $p(\theta | v_0, v_1) \propto L(v_0 \sin \theta + v_1 \cos \theta)$
  - Set $\Delta = v_0 \sin \theta + v_1 \cos \theta$
Slice sampling from $p(\theta|v_0, v_1) \propto L(v_0 \sin \theta + v_1 \cos \theta)$
Second draw

\[ L(\theta) \]

\[ \theta' \sim U(a, b) \]

\[ L(\theta') < L(\theta^{(s-1)}) \]

\[ \theta' > \theta^{(s-1)} \]

\[ b = \theta' \]
Third draw

\[ L(\theta') \]

\[ \theta' \sim U(a, b) \]

\[ L(\theta') > L(\theta^{(s-1)}) \]

\[ \theta^{(s)} = \theta' \]
Elliptical slice sampler for Gaussian linear regression

Regression model:

\[ y | X, \beta, \sigma^2 \sim N(X\beta, \sigma^2 I_n), \]

Posterior

\[ p(\beta | X, y, \sigma^2) \propto f(y | X, \beta, \sigma^2) \pi(\beta) \]

\[ f(y | X, \beta, \sigma^2) \text{ can be rewritten as (based on OLS estimates)} \]

\[ \pi_0(\beta | X, y, \sigma^2) \propto \exp \left\{ -\frac{1}{2\sigma^2} (\beta - \hat{\beta})' X' X (\beta - \hat{\beta}) \right\}. \]

The slice sampler of Murray et al (2010) can be applied directly, using \( \pi_0(\beta) \) as the Gaussian “prior” and \( \pi(\beta) \) as the arbitrary “likelihood”

We actually sample \( \Delta = \beta - \hat{\beta} \), which is centered around zero.
Elliptical slice sampler for Gaussian linear regression

*For initial value $\beta$, $\Delta = \beta - \hat{\beta}$, $\sigma^2$ fixed, and $\theta \in [0, 2\pi]$:*

1. **Draw $v \sim N(0, \sigma^2 (X'X)^{-1})$.**
   Set $v_0 = \Delta \sin \theta + v \cos \theta$
   Set $v_1 = \Delta \cos \theta - v \sin \theta$.

2. **Draw $\ell$ from $U[0, \pi(\hat{\beta} + v_0 \sin \theta + v_1 \cos \theta)]$.**
   Initialize $a = 0$ and $b = 2\pi$.
   2.1 **Sample $\theta'$ from $U[a, b]$.**
   2.2 If $\pi(\hat{\beta} + v_0 \sin \theta' + v_1 \cos \theta') > \ell$
      Then: Set $\theta \leftarrow \theta'$ Go to step 3.
      Else: If $\theta' < \theta$, set $a \leftarrow \theta'$, else set $b \leftarrow \theta'$ Go to step 2.1.

3. **Return $\Delta = v_0 \sin \theta + v_1 \cos \theta$ and $\beta = \hat{\beta} + \Delta$.**
One problem of naive slice sampler: If the number of regression coefficients $p$ is large, the green slice region is so tiny thus too many rejections before one acceptable sample.

Solution:

- $\beta$ has a jointly Gaussian likelihood and independent priors, it’s natural to write a Gibbs sampler, update a subset $\beta^k$ given other coefficients $\beta^{-k}$ in each MCMC iteration.

- Apply elliptical slice sampler to conditional likelihood $\beta^k | \beta^{-k}$ instead of full likelihood.

- Thus it is a “slice-within-Gibbs sampler”
Update a subset $\beta^k \mid \beta^{-k}$

Let us assume that $\beta$ can be split into $\beta^k$ and $\beta^{-k}$.

\[
\begin{bmatrix}
\beta^k \\
\beta^{-k}
\end{bmatrix}
\sim \mathcal{N}
\left(
\begin{bmatrix}
\hat{\beta}^k \\
\hat{\beta}^{-k}
\end{bmatrix},
\sigma^2
\begin{bmatrix}
\sum_{k,k} & \sum_{k,-k} \\
\sum_{-k,k} & \sum_{-k,-k}
\end{bmatrix}
\right)
\]

where

\[
\begin{bmatrix}
\hat{\beta}^k \\
\hat{\beta}^{-k}
\end{bmatrix} = \hat{\beta}
\text{ and }
\begin{bmatrix}
\sum_{k,k} & \sum_{k,-k} \\
\sum_{-k,k} & \sum_{-k,-k}
\end{bmatrix} = (X'X)^{-1}.
\]

Therefore, the conditional distribution of $\beta^k$ given $\beta^{-k}$ is $\mathcal{N}(\tilde{\beta}^k, \tilde{\Sigma}^k)$:

\[
\tilde{\beta}^k = \hat{\beta}^k + \sum_{k,-k} \sum_{-k,-k}^{-1} (\beta^{-k} - \hat{\beta}^{-k})
\]

\[
\tilde{\Sigma}^k = \sigma^2 \left( \sum_{k,k} - \sum_{k,-k} \sum_{-k,-k}^{-1} \sum_{-k,k} \right).
\]

Simulation shows that let $k = 1$, sampling one element of $\beta$ at a time has highest effective sample size per second!
Algorithm 2: Slice-within-Gibbs for linear regression

For each $k$ from 1 to $K$.

1. Construct $\tilde{\beta}^k$ and $\tilde{\Sigma}^k$ as in expressions (2) and (3).
   Set $\Delta^k = \beta^k - \tilde{\beta}^k$.
   Draw $v \sim N(0, \tilde{\Sigma}^k)$.
   Set $v_0 = \Delta^k \sin \theta^k + v \cos \theta^k$
   Set $v_1 = \Delta^k \cos \theta^k - v \sin \theta^k$.

2. Draw $\ell$ uniformly on $[0, \pi(\Delta^k + \tilde{\beta}^k)]$.
   Initialize $a = 0$ and $b = 2\pi$.
   2.1 Sample $\theta'$ uniformly on $[a, b]$.
   2.2 If $\pi(\tilde{\beta}^k + v_0 \sin \theta' + v_1 \cos \theta') > \ell$, set $\theta^k \leftarrow \theta'$.
      Go to step 3.
      Otherwise, shrink the support of $\theta'$ (if $\theta' < \theta^k$, set $a \leftarrow \theta'$; if $\theta' > \theta^k$, set $b \leftarrow \theta'$), and go to step 2.1.

3. Return $\Delta^k = v_0 \sin \theta^k + v_1 \cos \theta^k$ and $\beta^k = \tilde{\beta}^k + \Delta^k$. 21
Computational considerations

**Question:** Is slice-within-Gibbs sampler (updating one element of $\beta$ at a time) better than regular standard Gibbs sampler?

**Answer:** Yes!, All the conditional covariance are fixed in MCMC iterations. We can precompute all the conditional likelihoods.

We can precompute $\Sigma_{k,-k}\Sigma_{-k,-k}^{-1}$, $\Sigma_{k,k} - \Sigma_{k,-k}\Sigma_{-k,-k}^{-1}\Sigma_{-k,k}$, and Cholesky factors $L_k$, with $L_kL_k^T = \Sigma_{k,k} - \Sigma_{k,-k}\Sigma_{-k,-k}^{-1}\Sigma_{-k,k}$, for each $k = 1, \ldots, K$

By contrast, regular Gibbs samplers have full conditional updates of the form

\[
(\beta \mid X, y, \sigma^2) \sim \mathcal{N}((X'X + D)^{-1}X'y, \sigma^2(X'X + D)^{-1}),
\]

which require costly Cholesky or eigenvalue decompositions of the matrix $(X'X + D)^{-1}$ at each iteration as $D$ is updated.
Rank deficient case

**Question:** What if $X'X$ is not invertible?

**Answer:** Recall that the slice sampler draws from

$$p(\beta \mid y, X, \sigma) \propto N_Y(X\beta, \sigma^2)\pi(\beta) \propto N_\beta(\hat{\beta}, \sigma^2(X'X)^{-1})\pi(\beta), \quad (5)$$

It can be written as

$$p(\beta \mid y, X, \sigma) \propto N_Y(X\beta, \sigma^2)N_\beta(0, c\sigma^2 I)\frac{\pi(\beta)}{N_\beta(0, c\sigma^2 I)}$$

$$\propto N_\beta(\bar{\beta}, \sigma^2(X'X + c^{-1}I)^{-1})\frac{\pi(\beta)}{N_\beta(0, c\sigma^2 I)}. \quad (6)$$

$c > 0$ and makes $(X'X + c^{-1}I)$ invertible, then we evaluate $\frac{\pi(\beta)}{N_\beta(0, c\sigma^2 I)}$ rather than $\pi(\beta)$. Pick small $c$ around 1 works fine in practice.
Comparison metrics

Effective sample size (ESS) per second: Letting $N$ denote the Monte Carlo sample size, then ESS for parameter $\beta_j$ is

$$
ESS(\beta_j) = \frac{N}{1 + 2 \sum_{k=1}^{\infty} \rho_k},
$$

where $\rho_k = \text{corr}(\beta_j^{(0)}, \beta_j^{(k)})$ is the autocovariance of lag $k$. We divide ESS by running time in seconds to compute ESS per second as a measure of efficiency of each sampler.

Root mean square error: Suppose $\{\bar{\beta}_j\}$ are posterior means of each variable and $\{\beta_j\}$ are true values. The estimation error is measured by

$$
\text{error} = \sqrt{\frac{\sum_{j=1}^{p} (\bar{\beta}_j - \beta_j)^2}{\sum_{j=1}^{p} \beta_j^2}}.
$$
Simulation exercise

1. Draw elements of $\beta$ from a “sparse Gaussian” where $\lceil p \rceil$ entries of $\beta$ are non-zero, drawn from a standard Gaussian distribution, and all other entries are zero.

2. Generate the regressors matrix $X$ in one of two ways.
   - **Independent regressor:** $X_{ij}$ are iid from standard Gaussian.
   - **Factor structure:** Suppose there are $k = p/5$ factors. Factors are iid $F \sim N(0, 1)$. The factor loading matrix, $B$, has exactly five ones in each column and a single 1 in each row, all others 0, so that $BB'$ is block diagonal, with blocks of all ones and all other elements being zero. The regressors are then set as $X = F'B' + \varepsilon$ where $\varepsilon_{ij}$ are iid $N(0, 0.01)$.

3. Set $\sigma = \kappa \sqrt{\sum_{j=1}^{p} \beta_j^2}$, where $\kappa$ controls noise level.

4. Draw $y_i = x_i'\beta + \epsilon_i$, $\epsilon_i \sim N(0, \sigma^2)$ for $i = 1, \ldots, n$.

Additionally, we vary the noise level, letting $\kappa = 1$ or $\kappa = 2$, corresponding to signal-to-noise ratios of 1 and $1/2$, respectively.
\( n > p \), regressors are independent

\( \kappa = 1 \), signal-to-noise ratio is 1:1. The regressors are independent. We show effective sample size per second and RMSE of estimation.

| Prior     | \( p \) | \begin{tabular}{c|c|c|c|c|c|c|c} \hline & & \multicolumn{3}{c|}{RMSE} & \multicolumn{3}{c}{ESS per second} \\ \hline & & OLS & slice & mono & Gibbs & slice & mono & Gibbs \\ \hline Horseshoe & 100 & 3.38\% & 1.52\% & 1.51\% & 1.51\% & \textbf{1399} & 613 & 567 \\ & 1000 & 1.05\% & 0.27\% & 0.27\% & 0.27\% & 91 & 5 & 5 \\ \hline Laplace & 100 & 3.38\% & 2.39\% & 2.38\% & – & \textbf{2362} & 809 & – \\ & 1000 & 1.04\% & 0.63\% & 0.63\% & – & 168 & 8 & – \\ \hline Ridge & 100 & 3.38\% & 3.20\% & 3.20\% & – & \textbf{3350} & 959 & – \\ & 1000 & 1.06\% & 0.99\% & 0.99\% & – & 178 & 5 & – \\ \hline

We compare elliptical slice sampler, Gibbs sampler in R package \texttt{monomvn} (column mono) and our own implementation of Gibbs sampler for horseshoe regression (column Gibbs).

The elliptical slice sampler has similar error to the Gibbs sampler, but much higher effective sample size per second.
\( n > p \), regressors have factor structure

\[ \kappa = 1 \), signal-to-noise ratio is 1:1. The regressors have underlying factor structure where every 5 regressors are highly correlated.

<table>
<thead>
<tr>
<th>Prior</th>
<th>( p )</th>
<th>( \text{OLS} )</th>
<th>( 1\text{block} )</th>
<th>( \text{mono} )</th>
<th>( \text{Gibbs} )</th>
<th>( \text{ESS per second} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1block mono Gibbs</td>
<td></td>
<td></td>
<td>1block mono Gibbs</td>
<td></td>
</tr>
<tr>
<td>Horseshoe</td>
<td>100</td>
<td>16.47%</td>
<td>6.06%</td>
<td>6.04%</td>
<td>6.03%</td>
<td>387</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>6.85%</td>
<td>1.64%</td>
<td>1.64%</td>
<td>1.64%</td>
<td>36</td>
</tr>
<tr>
<td>Laplace</td>
<td>100</td>
<td>17.06%</td>
<td>7.21%</td>
<td>7.15%</td>
<td>–</td>
<td>573</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>6.77%</td>
<td>1.95%</td>
<td>1.94%</td>
<td>–</td>
<td>38</td>
</tr>
<tr>
<td>Ridge</td>
<td>100</td>
<td>16.90%</td>
<td>8.50%</td>
<td>8.75%</td>
<td>–</td>
<td>669</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>6.85%</td>
<td>2.93%</td>
<td>3.09%</td>
<td>–</td>
<td>38</td>
</tr>
</tbody>
</table>

We compare elliptical slice sampler, Gibbs sampler in R package monomvn (column mono) and our own implementation of Gibbs sampler for horseshoe regression (column Gibbs).

The elliptical slice sampler has similar error to the Gibbs sampler, but much higher effective sample size per second when \( p = 1,000 \).
Compare with Johndrow and Orenstein [2017] (denoted J&O) for the $p > n$ case. 12,000 posterior draws with the first 2,000 as burn-in.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$n$</th>
<th>$\kappa$</th>
<th>J&amp;O</th>
<th>Slice</th>
<th>J&amp;O</th>
<th>Slice</th>
<th>J&amp;O</th>
<th>Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>300</td>
<td>0.25</td>
<td>119.11</td>
<td>91.50</td>
<td>0.0041</td>
<td>0.0038</td>
<td>46.71</td>
<td>43.19</td>
</tr>
<tr>
<td>1000</td>
<td>600</td>
<td>0.25</td>
<td>394.02</td>
<td>88.61</td>
<td>0.0028</td>
<td>0.0026</td>
<td>14.68</td>
<td>47.26</td>
</tr>
<tr>
<td>1000</td>
<td>900</td>
<td>0.25</td>
<td>905.36</td>
<td>88.91</td>
<td>0.0021</td>
<td>0.0020</td>
<td>6.60</td>
<td>48.85</td>
</tr>
<tr>
<td>1000</td>
<td>300</td>
<td>1</td>
<td>127.33</td>
<td>90.19</td>
<td>0.0189</td>
<td>0.0189</td>
<td>43.92</td>
<td>39.25</td>
</tr>
<tr>
<td>1000</td>
<td>600</td>
<td>1</td>
<td>399.50</td>
<td>91.17</td>
<td>0.0129</td>
<td>0.0129</td>
<td>14.39</td>
<td>44.12</td>
</tr>
<tr>
<td>1000</td>
<td>900</td>
<td>1</td>
<td>927.96</td>
<td>91.58</td>
<td>0.0098</td>
<td>0.0099</td>
<td>6.35</td>
<td>46.09</td>
</tr>
<tr>
<td>1500</td>
<td>450</td>
<td>0.25</td>
<td>346.37</td>
<td>187.91</td>
<td>0.0029</td>
<td>0.0027</td>
<td>16.37</td>
<td>21.26</td>
</tr>
<tr>
<td>1500</td>
<td>900</td>
<td>0.25</td>
<td>1073.28</td>
<td>185.57</td>
<td>0.0022</td>
<td>0.0021</td>
<td>5.50</td>
<td>23.08</td>
</tr>
<tr>
<td>1500</td>
<td>1350</td>
<td>0.25</td>
<td>2629.52</td>
<td>183.68</td>
<td>0.0018</td>
<td>0.0017</td>
<td>2.27</td>
<td>24.04</td>
</tr>
<tr>
<td>1500</td>
<td>450</td>
<td>1</td>
<td>326.63</td>
<td>183.66</td>
<td>0.0164</td>
<td>0.0164</td>
<td>17.39</td>
<td>20.28</td>
</tr>
<tr>
<td>1500</td>
<td>900</td>
<td>1</td>
<td>1021.47</td>
<td>174.52</td>
<td>0.0100</td>
<td>0.0101</td>
<td>5.73</td>
<td>23.72</td>
</tr>
<tr>
<td>1500</td>
<td>1350</td>
<td>1</td>
<td>2515.37</td>
<td>176.51</td>
<td>0.0071</td>
<td>0.0071</td>
<td>2.36</td>
<td>24.78</td>
</tr>
<tr>
<td>3000</td>
<td>100</td>
<td>0.25</td>
<td>85.95</td>
<td>985.68</td>
<td>0.0067</td>
<td>0.0075</td>
<td>69.72</td>
<td>3.89</td>
</tr>
<tr>
<td>3000</td>
<td>500</td>
<td>0.25</td>
<td>575.92</td>
<td>983.64</td>
<td>0.0024</td>
<td>0.0022</td>
<td>9.85</td>
<td>4.17</td>
</tr>
</tbody>
</table>

Similar RMSE. The elliptical slice sampler has higher ESS per second in most cases considered here, especially when $p \approx n$. The Johndrow et al. sampler is much more efficient only when $p \gg n$, such as $p = 3000$ and $n = 100$. 28
Empirical illustration 1: beauty and course evaluations

Use data from Hamermesh and Parker [2005].

Course evaluations from the University of Texas at Austin between 2000 and 2002 and additional information of class and instructor.

We want to study how the beauty score of instructor affect his or her course evaluation scores (on 1 to 5 scale, 5 is the best).

We fit regression model with horseshoe, ridge, Laplace (lasso) and two exotic priors.

The purpose is not to argue advantage of exotic priors, but just to show we are able to fit them easily.
Exotic priors

Consider “sharkfin” prior

\[
\pi(\beta) = \begin{cases} 
2qf(\beta) & \beta \leq 0 \\
2f(\beta/s)(1 - q)/s & \beta > 0 
\end{cases},
\]

where \( f(x) = \frac{1}{\pi(1+x^2)} \) is the density.

Figure: Density of sharkfin prior.
Exotic priors

“non-local” prior is a mixture of Cauchy priors, which is anti-sparse

$$\pi(\beta) = 0.5t(\beta; -1.5, 1) + 0.5t(\beta; 1.5, 1)$$

(10)

Figure: Density of non-local prior.
Results of beauty and course evaluations

Table: Posterior points estimates of regression coefficients under each prior; those whose posterior 95% credible intervals exclude zero are shown in bold.

<table>
<thead>
<tr>
<th>variable name</th>
<th>horseshoe</th>
<th>lasso</th>
<th>ridge</th>
<th>sharkfin</th>
<th>non-local</th>
</tr>
</thead>
<tbody>
<tr>
<td>class size 61 to 150</td>
<td>−0.13</td>
<td>−0.19</td>
<td>−0.20</td>
<td>−0.14</td>
<td>−0.22</td>
</tr>
<tr>
<td>class size 151 to 600</td>
<td>−0.36</td>
<td>−0.41</td>
<td>−0.43</td>
<td>−0.36</td>
<td>−0.46</td>
</tr>
<tr>
<td>tenure track</td>
<td>0.22</td>
<td>0.29</td>
<td>0.31</td>
<td>0.27</td>
<td>0.40</td>
</tr>
<tr>
<td>non-minority</td>
<td>0.65</td>
<td>0.65</td>
<td>0.53</td>
<td>0.63</td>
<td>0.71</td>
</tr>
<tr>
<td>highly beautiful</td>
<td>0.14</td>
<td>0.36</td>
<td>0.54</td>
<td>0.25</td>
<td>0.38</td>
</tr>
</tbody>
</table>
Empirical illustration 2: Diabetes

The data consist of $p = 10$ baseline measurements on $n = 442$ diabetic patients; the response variable is a numerical measurement of disease progression (Efron et al., 2004). Below are the OLS estimates.
Ridge and Lasso penalties

Ridge regression

Lasso
Cross validation
Out-of-sample Root MSE

The diagram shows a scatter plot with coefficients of various regressors. The legend indicates the following models and their RMSE:

- OLS (RMSE=0.694)
- Ridge (RMSE=0.678)
- Lasso (RMSE=0.672)
- Horseshoe (RMSE=0.69)

The x-axis represents the different regressors, while the y-axis shows the coefficient values. Each model is represented by a different color dot.
Out-of-sample Root MSE: replications

Train=50%

MSE

0.45 0.50 0.55 0.60

OLS RIDGE LASSO HORSESHOE
OLS: Including squares and interactions ($p = 64$)
Ridge and Lasso penalties
Cross validation

![Diagram of Cross Validation](image)

- **Ridge**
  - Mean-Squared Error against log(Lambda)

- **Lasso**
  - Mean-Squared Error against log(Lambda)
Out-of-sample Root MSE

OLS (RMSE=0.768)
Ridge (RMSE=0.664)
Lasso (RMSE=0.674)
Horseshoe (RMSE=0.688)
Out-of-sample Root MSE: replications
Description
This table gives the results of 133 simulations showing the effects of motorcycle crashes on victims heads: time after a simulated impact with motorcycles and head acceleration of a PTMO (post mortem human test object) were recorded.

Usage
data(motorcycledata)

Format
A 133 by 2 data frame.

References
Data

Head acceleration vs. Time after impact with motorcycles.
Spline regression: OLS and horseshoe

- beta 1
- beta 5
- beta 10
- beta 15
- beta 19
- beta 24
- beta 29
- beta 34
- beta 38
- beta 43
- beta 48
- beta 53
Shrinkage
Head acceleration vs. Time after impact with motorcycles

OLS
Comparison

Time after impact with motorcycles

Head acceleration

Horseshoe

OLS

Horseshoe

−3 −2 −1 0 1 2 3

10 20 30 40 50

Horseshoe

OLS

Time after impact with motorcycles

Head acceleration

−3 −2 −1 0 1 2 3

10 20 30 40 50

Horseshoe

OLS

Horseshoe
install.packages("adlift")
install.packages("bayeslm")
require(splines)
library("adlift")
library("bayeslm")

data(motorcycledata)
y = motorcycledata[,2]
y = (y-mean(y))/sqrt(var(y))
x = motorcycledata[,1]
n = length(x)

cuts = quantile(x,seq(0.02,0.98,by=0.02))
X = bs(x,knots=cuts)
p = ncol(X)

fit.ols <- lm(y~X)
fit.hs = bayeslm(y~X)
References


