Scalable MCMC for Bayes Shrinkage Priors

Paulo Orenstein
July 2, 2018

Stanford University

Joint work with James Johndrow and Anirban Bhattacharya
Introduction

Consider the high-dimensional setting: predict a vector \( y \in \mathbb{R}^n \) from a set of features \( X \in \mathbb{R}^{n \times p} \), with \( p \gg n \).
Consider the high-dimensional setting: predict a vector $y \in \mathbb{R}^n$ from a set of features $X \in \mathbb{R}^{n \times p}$, with $p \gg n$.

Assume a sparse Gaussian linear model

$$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n),$$

with $\beta_j = 0$ for many $j$. 

---

Paulo Orenstein  
Scalable MCMC for Bayes Shrinkage Priors  
Stanford University
Introduction

- Consider the high-dimensional setting: predict a vector $y \in \mathbb{R}^n$ from a set of features $X \in \mathbb{R}^{n \times p}$, with $p \gg n$.

- Assume a sparse Gaussian linear model

  $$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n),$$
  
  with $\beta_j = 0$ for many $j$.

- How can we perform prediction and inference?
Consider the high-dimensional setting: predict a vector $y \in \mathbb{R}^n$ from a set of features $X \in \mathbb{R}^{n \times p}$, with $p \gg n$.

Assume a sparse Gaussian linear model

$$y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n),$$

with $\beta_j = 0$ for many $j$.

How can we perform prediction and inference?
- Lasso
Introduction

- Consider the high-dimensional setting: predict a vector $y \in \mathbb{R}^n$ from a set of features $X \in \mathbb{R}^{n \times p}$, with $p \gg n$.

- Assume a sparse Gaussian linear model

$$y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n),$$

with $\beta_j = 0$ for many $j$.

- How can we perform prediction and inference?
  - Lasso
  - Point mass mixture prior
Introduction

Consider the high-dimensional setting: predict a vector \( y \in \mathbb{R}^n \) from a set of features \( X \in \mathbb{R}^{n \times p} \), with \( p \gg n \).

Assume a sparse Gaussian linear model

\[
y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n),
\]

with \( \beta_j = 0 \) for many \( j \).

How can we perform prediction and inference?

- Lasso, \textit{but}: convex relaxation; one parameter for sparsity and shrinkage
- Point mass mixture prior
Consider the high-dimensional setting: predict a vector $y \in \mathbb{R}^n$ from a set of features $X \in \mathbb{R}^{n \times p}$, with $p \gg n$.

Assume a sparse Gaussian linear model

$$y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n),$$

with $\beta_j = 0$ for many $j$.

How can we perform prediction and inference?

- Lasso, *but*: convex relaxation; one parameter for sparsity and shrinkage
- Point mass mixture prior, *but*: computation is prohibitive
Introduction

- Can we find a continuous prior that behaves like the point mass mixture prior?
Introduction

Can we find a continuous prior that behaves like the point mass mixture prior?

Desiderata:

- adaptive to sparsity
- easy to compute
- good predictive performance
- good frequentist properties
- decent compromise between statistical and computational goals
Introduction

- Can we find a continuous prior that behaves like the point mass mixture prior?

- Desiderata:
  - adaptive to sparsity
  - easy to compute
  - good predictive performance
  - good frequentist properties
  - decent compromise between statistical and computational goals

- Global-local priors can achieve this (with some qualifications).
Introduction

Can we find a continuous prior that behaves like the point mass mixture prior?

Desiderata:

- adaptive to sparsity
- easy to compute
- good predictive performance
- good frequentist properties
- decent compromise between statistical and computational goals

Global-local priors can achieve this (with some qualifications).

But... they are still slow.

- Lasso: $n \approx 1,000$, $p \approx 1,000,000$;
- Global-local: $n \approx 1,000$, $p \approx 1,000$. 
Model

The Horseshoe model*:

\[ y_i | \beta_j, \lambda_j, \tau, \sigma^2 \sim \text{N}(x_i \beta, \sigma^2) \]

\[ \beta_j \sim \text{N}(0, \tau^2 \lambda_j^2) \]

\[ \lambda_j \sim \text{Cauchy}_+(0, 1) \]

\[ \tau \sim \text{Cauchy}_+(0, 1) \]

\[ \sigma^2 \sim \text{InvGamma}(a_0/2, b_0/2) \]

* [Carvalho et. al, 2010]
The Horseshoe model:

\[ y_i \mid \beta_j, \lambda_j, \tau, \sigma^2 \sim \text{N}(x_i \beta, \sigma^2) \]

\[ \beta_j \sim \text{N}(0, \tau^2 \lambda_j^2) \]

\[ \lambda_j \sim \text{Cauchy}_+(0, 1) \]

\[ \tau \sim \text{Cauchy}_+(0, 1) \]

\[ \sigma^2 \sim \text{InvGamma}(a_0/2, b_0/2) \]

* [Carvalho et. al, 2010]
Model

- The Horseshoe model*: 
  \[ y_i \mid \beta_j, \lambda_j, \tau, \sigma^2 \overset{\text{ind}}{\sim} N(x_i \beta, \sigma^2) \]
  \[ \beta_j \overset{\text{ind}}{\sim} N(0, \tau^2 \lambda_j^2) \]
  \[ \lambda_j \overset{\text{ind}}{\sim} \text{Cauchy}_+(0, 1) \]
  \[ \tau \sim \text{Cauchy}_+(0, 1) \]
  \[ \sigma^2 \sim \text{InvGamma}(a_0/2, b_0/2) \]

* [Carvalho et. al, 2010]
The Horseshoe model*: 

\[ y_i \mid \beta_j, \lambda_j, \tau, \sigma^2 \overset{\text{ind}}{\sim} N(x_i \beta, \sigma^2) \]

\[ \beta_j \overset{\text{ind}}{\sim} N(0, \tau^2 \lambda_j^2) \]

\[ \lambda_j \overset{\text{ind}}{\sim} \text{Cauchy}_+(0, 1) \]

\[ \tau \overset{\text{ind}}{\sim} \text{Cauchy}_+(0, 1) \]

\[ \sigma^2 \overset{\text{ind}}{\sim} \text{InvGamma}(a_0/2, b_0/2) \]

* [Carvalho et. al, 2010]
The Horseshoe model*:

\[ y_i \mid \beta_j, \lambda_j, \tau, \sigma^2 \overset{\text{ind}}{\sim} N(x_i \beta, \sigma^2) \]

\[ \beta_j \overset{\text{ind}}{\sim} N(0, \tau^2 \lambda_j^2) \]

\[ \lambda_j \overset{\text{ind}}{\sim} \text{Cauchy}_+(0, 1) \]

\[ \tau \sim \text{Cauchy}_+(0, 1) \]

\[ \sigma^2 \sim \text{InvGamma}(a_0/2, b_0/2) \]

* [Carvalho et. al, 2010]
Model

- Horseshoe has other good frequentist properties.
Horseshoe has other good frequentist properties.

It achieves the minimax-adaptive risk for squared error loss up to a constant.
Horseshoe has other good frequentist properties.

It achieves the minimax-adaptive risk for squared error loss up to a constant.

Suppose $X = I$, $\|\beta\|_0 = s_n$, then [van der Pas et al., 2014],

$$\sup_{\beta: \|\beta\|_0 \leq s_n} \mathbb{E}_\beta \left[ \|\hat{\beta}_{HS} - \beta\|^2 \right] \leq 4\sigma^2 s_n \log \frac{n}{s_n} \cdot (1 + o(1)),$$

while, for any estimator $\hat{\beta}$, [Donoho et al., 1992] shows

$$\sup_{\beta: \|\beta\|_0 \leq s_n} \mathbb{E}_\beta \left[ \|\hat{\beta} - \beta\|^2 \right] \geq 2\sigma^2 s_n \log \frac{n}{s_n} \cdot (1 + o(1)).$$
State-of-the-art: (i) $\tau \mid \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) \mid \tau, \lambda$, (iii) slice sampling for $\lambda$. 

- We scale the model with two ideas.
  - First idea: block $(\beta, \sigma^2, \tau)$ to improve mixing; 1. sample $(\beta, \sigma^2, \tau) \mid \lambda$ by block sampling: $\tau \mid \lambda$, then $\sigma^2 \mid \tau, \lambda$, and finally $\beta \mid \sigma^2, \tau, \lambda$; 2. sample $\lambda \mid \beta, \sigma^2$ using slice sampling.
  - Second idea: truncate some of the matrices involved to improve the computational cost per step.

Paulo Orenstein
Scalable MCMC for Bayes Shrinkage Priors
Computation

- State-of-the-art: (i) $\tau | \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) | \tau, \lambda$, (iii) slice sampling for $\lambda$. But...
Computation

- State-of-the-art: (i) $\tau \mid \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) \mid \tau, \lambda$, (iii) slice sampling for $\lambda$. But...

- We scale the model with two ideas.
Computation

- State-of-the-art: (i) $\tau | \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) | \tau, \lambda$, (iii) slice sampling for $\lambda$. But...

- We scale the model with two ideas.

- First idea: **block** $(\beta, \sigma^2, \tau)$ to improve *mixing*;
  1. sample $(\beta, \sigma^2, \tau) | \lambda$ by block sampling: $\tau | \lambda$, then $\sigma^2 | \tau, \lambda$, and finally $\beta | \sigma^2, \tau, \lambda$;
  2. sample $\lambda | \beta, \sigma^2$ using slice sampling.
Computation

▶ State-of-the-art: (i) $\tau \mid \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) \mid \tau, \lambda$, (iii) slice sampling for $\lambda$. But...

▶ We scale the model with two ideas.

▶ First idea: **block** $(\beta, \sigma^2, \tau)$ to improve *mixing*;

   1. sample $(\beta, \sigma^2, \tau) \mid \lambda$ by block sampling: $\tau \mid \lambda$, then $\sigma^2 \mid \tau, \lambda$, and finally $\beta \mid \sigma^2, \tau, \lambda$;

   2. sample $\lambda \mid \beta, \sigma^2$ using slice sampling.

▶ Second idea: **truncate** some of the matrices involved to improve the *computational cost per step*. 
Gibbs sampling

Let \( M = X (\text{diag}(\xi \eta))^{-1} X^T + I \), \( \xi = \tau^{-2} \), \( \eta_j = \lambda_j^{-2} \), and **block update**:

\[
\begin{align*}
\rightarrow p(\tau \mid \lambda, y) & \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} (y^T M^{-1} y + b_0)^{-\frac{n+a_0}{2}} \\
\rightarrow p(\sigma^2 \mid \tau, \lambda, y) & \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} \left[ y^T M^{-1} y + b_0 \right] \right) \\
\rightarrow p(\beta \mid \sigma^2, \tau, \lambda, y) & \sim N \left( (X^T X + \text{diag}(\xi \eta))^{-1} X^T y, \sigma^2 (X^T X + \text{diag}(\xi \eta))^{-1} \right)
\end{align*}
\]

Then perform slice sampling:

\[
\begin{align*}
\rightarrow p(\lambda \mid \beta, \sigma^2, \tau, y): \ (i) \ U \mid \eta_j & \sim \text{Unif} \left[ 0, \frac{1}{1+\eta_j} \right]; \ (ii) \ \eta_j \mid u \sim e^{-\frac{\xi \beta_j^2}{2\sigma^2} \eta_j} \mathbb{I}_{\left[ \frac{1}{u} > \eta_j \right]}.
\end{align*}
\]
Gibbs sampling

Let $M = X(\text{diag}(\xi\eta))^{-1}X^T + I$, $\xi = \tau^{-2}$, $\eta_j = \lambda_j^{-2}$, and block update:

- $p(\tau \mid \lambda, y) \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} (y^TM^{-1}y + b_0)^{-\frac{n+a_0}{2}}$
- $p(\sigma^2 \mid \tau, \lambda, y) \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} \left[y^T M^{-1} y + b_0 \right] \right)$
- $p(\beta \mid \sigma^2, \tau, \lambda, y) \sim N \left( (X^TX + \text{diag}(\xi\eta))^{-1}X^Ty, \sigma^2 \left(X^TX + \text{diag}(\xi\eta) \right)^{-1} \right)$

Then perform slice sampling:

- $p(\lambda \mid \beta, \sigma^2, \tau, y)$: (i) $U \mid \eta_j \sim \text{Unif} \left( 0, \frac{1}{1+\eta_j} \right)$; (ii) $\eta_j \mid U \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2} \eta_j + \frac{\xi\beta_j^2}{2\sigma^2} \eta_j [\frac{1}{U} > \eta_j]}.$
Gibbs sampling

Let $M = X (\text{diag}(\xi \eta))^{-1} X^T + I$, $\xi = \tau^{-2}$, $\eta_j = \lambda_j^{-2}$, and block update:

- $p(\tau \mid \lambda, y) \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} \left( y^T M^{-1} y + b_0 \right)^{-\frac{n+a_0}{2}}$

- $p(\sigma^2 \mid \tau, \lambda, y) \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} \left[ y^T M^{-1} y + b_0 \right] \right)$

- $p(\beta \mid \sigma^2, \tau, \lambda, y) \sim N \left( (X^T X + \text{diag}(\xi \eta))^{-1} X^T y, \sigma^2 (X^T X + \text{diag}(\xi \eta))^{-1} \right)$

Then perform slice sampling:

- $p(\lambda \mid \beta, \sigma^2, \tau, y)$: (i) $U \mid \eta_j \sim \text{Unif} \left[ 0, \frac{1}{1+\eta_j} \right]$; (ii) $\eta_j \mid u \sim e^{-\frac{\xi \beta_j^2}{2\sigma^2} \eta_j} \mathbb{I}_{[\frac{1}{u} > \eta_j]}$. 

Gibbs sampling

Let $M = X(\text{diag}(\xi \eta))^{-1}X^T + I$, $\xi = \tau^{-2}$, $\eta_j = \lambda_j^{-2}$, and **block update**:

- $p(\tau | \lambda, y) \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} (y^T M^{-1} y + b_0)^{-\frac{n+a_0}{2}}$

- $p(\sigma^2 | \tau, \lambda, y) \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} [y^T M^{-1} y + b_0] \right)$

- $p(\beta | \sigma^2, \tau, \lambda, y) \sim N \left( (X^T X + \text{diag}(\xi \eta))^{-1} X^T y, \sigma^2 (X^T X + \text{diag}(\xi \eta))^{-1} \right)$

Then perform slice sampling:

- $p(\lambda | \beta, \sigma^2, \tau, y)$: (i) $U | \eta_j \sim \text{Unif} \left[ 0, \frac{1}{1+\eta_j} \right]$; (ii) $\eta_j | u \sim e^{-\frac{\xi \beta^2_j}{2\sigma^2} \eta_j^2} I_{[\frac{1}{1+u} > \eta_j]}$. 

Paulo Orenstein
Scalable MCMC for Bayes Shrinkage Priors
Stanford University
Gibbs sampling

Let \( M = X(\text{diag}(\xi \eta))^{-1}X^T + I, \ \xi = \tau^{-2}, \ \eta_j = \lambda_j^{-2}, \) and **block update:**

1. \[ p(\tau \mid \lambda, y) \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} (y^T M^{-1} y + b_0)^{-\frac{n+a_0}{2}} \]

2. \[ p(\sigma^2 \mid \tau, \lambda, y) \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} \left[ y^T M^{-1} y + b_0 \right] \right) \]

3. \[ p(\beta \mid \sigma^2, \tau, \lambda, y) \sim N \left( (X^T X + \text{diag}(\xi \eta))^{-1} X^T y, \sigma^2 (X^T X + \text{diag}(\xi \eta))^{-1} \right) \]

Then perform slice sampling:

1. \[ p(\lambda \mid \beta, \sigma^2, \tau, y): (i) \ U \mid \eta_j \sim \text{Unif} \left[ 0, \frac{1}{1+\eta_j} \right]; (ii) \ \eta_j \mid u \sim e^{-\frac{\xi \beta_j^2}{2\sigma^2 \eta_j} \mathbb{1}_{\left[ \frac{1-u}{u} > \eta_j \right]}}. \]
Gibbs sampling

Let $M = X(\text{diag}(\xi \eta))^{-1}X^T + I$, $\xi = \tau^{-2}$, $\eta_j = \lambda_j^{-2}$, and block update:

- $p(\tau \mid \lambda, y) \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} (y^T M^{-1} y + b_0)^{-\frac{n+a_0}{2}}$

- $p(\sigma^2 \mid \tau, \lambda, y) \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} [y^T M^{-1} y + b_0] \right)$

- $p(\beta \mid \sigma^2, \tau, \lambda, y) \sim N \left( (X^T X + \text{diag}(\xi \eta))^{-1} X^T y, \sigma^2 (X^T X + \text{diag}(\xi \eta))^{-1} \right)$

Then perform slice sampling:

- $p(\lambda \mid \beta, \sigma^2, \tau, y)$: (i) $U \mid \eta_j \sim \text{Unif} \left( 0, \frac{1}{1+\eta_j} \right)$; (ii) $\eta_j \mid U \sim e^{-\frac{\xi \beta^2}{2\sigma^2} \eta_j} I_{\frac{1}{1+u} > \eta_j}$. 
Gibbs sampling

Let $M = X (\text{diag}(\xi \eta))^{-1} X^T + I$, $\xi = \tau^{-2}$, $\eta_j = \lambda_j^{-2}$, and block update:

- $p(\tau | \lambda, y) \propto \frac{1}{\sqrt{\xi(1+\xi)}} |M|^{-1/2} (y^T M^{-1} y + b_0)^{-\frac{n+a_0}{2}}$

- $p(\sigma^2 | \tau, \lambda, y) \sim \text{InvGamma} \left( \frac{n+a_0}{2}, \frac{1}{2} [y^T M^{-1} y + b_0] \right)$

- $p(\beta | \sigma^2, \tau, \lambda, y) \sim \mathcal{N} \left( (X^T X + \text{diag}(\xi \eta))^{-1} X^T y, \sigma^2 (X^T X + \text{diag}(\xi \eta))^{-1} \right)$

Then perform slice sampling:

- $p(\lambda | \beta, \sigma^2, \tau, y)$: (i) $U | \eta_j \sim \text{Unif} \left[ 0, \frac{1}{1+\eta_j} \right]$; (ii) $\eta_j | u \sim e^{-\frac{\xi \beta^2}{2\sigma^2} \eta_j} \mathbb{I}_{\left\{ \frac{1}{u} > \eta_j \right\}}$. 

Paulo Orenstein
Scalable MCMC for Bayes Shrinkage Priors
Stanford University
Markov approximation

- We approximate $M = X \text{diag}(\xi \eta_j^{-1}) X^T + I$ with
  
  $$M_\delta = XD_\delta X^T + I, \quad D_\delta = \text{diag}(\xi \eta_j^{-1} \mathbb{I}[(\xi_{\text{max}} \eta_j)^{-1} > \delta])$$

  for $\delta \ll 1$, and $\xi_{\text{max}}$ the maximum of the current and proposed $\xi$. 

This makes computation much faster.
Markov approximation

We approximate $M = X \text{diag}((\xi \eta_j)^{-1}) X^T + I$ with

$$M_\delta = XD_\delta X^T + I, \quad D_\delta = \text{diag}((\xi \eta_j)^{-1}[\xi_{\max} \eta_j]^{-1} > \delta)$$

for $\delta \ll 1$, and $\xi_{\max}$ the maximum of the current and proposed $\xi$. 

This makes computation much faster.
Markov approximation

- We approximate $M = X \text{diag}((\xi \eta_j)^{-1}) X^T + I$ with

$$M_\delta = XD_\delta X^T + I, \quad D_\delta = \text{diag}((\xi \eta_j)^{-1} \mathbb{I}[(\xi_{\text{max}} \eta_j)^{-1} > \delta])$$

for $\delta \ll 1$, and $\xi_{\text{max}}$ the maximum of the current and proposed $\xi$.

- This makes computation much faster.
Markov approximation

- We approximate $M = X\text{diag}((\xi_j^{-1})X^T + I$ with

$$M_\delta = XD_\delta X^T + I,$$ $D_\delta = \text{diag}((\xi_j^{-1})I_{(\xi_j^{-1} > \delta)})$

for $\delta \ll 1$, and $\xi_{\text{max}}$ the maximum of the current and proposed $\xi$.

- This makes computation much faster.

Approximating Kernels

Let $P_\delta(x, \cdot)$ and $P(x, \cdot)$ denote the Markov operators for the approximate and exact algorithms, with $x = (\beta, \sigma^2, \tau, \lambda)$ the entire state vector. Then

$$\sup_x \|P_\delta(x, \cdot) - P(x, \cdot)\|_{TV} \leq \sqrt{\delta} \|X\| \sqrt{a + \frac{n + a_0}{b_0} + \frac{n}{2} \frac{\|y\|^2}{b_0}} + O(\delta),$$

for sufficiently small $\delta > 0$. 

Paulo Orenstein
Stanford University
Simulation

- We simulate data as follows:

\[ x_i \overset{iid}{\sim} N_p(0, \Sigma) \]
\[ y_i \sim N(x_i \beta, 4) \]
\[ \beta_j = \begin{cases} 2^{-j/4-9/4} & \text{if } j < 24, \\ 0 & \text{if } j \geq 24. \end{cases} \]
Simulation

We simulate data as follows:

\[
\begin{align*}
    x_i &\sim \text{iid } \mathcal{N}_p(0, \Sigma) \\
    y_i &\sim \mathcal{N}(x_i\beta, 4) \\
    \beta_j &= \begin{cases} 
        2^{-(j/4-9/4)} & \text{if } j < 24, \\
        0 & \text{if } j \geq 24.
    \end{cases}
\end{align*}
\]

There are nulls, clear non-nulls, and some subtle non-nulls.
Simulation

- We simulate data as follows:

  \[ x_i \overset{iid}{\sim} N_p(0, \Sigma) \]

  \[ y_i \sim N(x_i \beta, 4) \]

  \[ \beta_j = \begin{cases} 2^{-(j/4-9/4)} & \text{if } j < 24, \\ 0 & \text{if } j \geq 24. \end{cases} \]

- There are nulls, clear non-nulls, and some subtle non-nulls.

- We consider both \( \Sigma = I \) (independent design) and \( \Sigma_{ij} = 0.9^{|i-j|} \) (correlated design).
Autocorrelation

Autocorrelation for $\log(\xi) = -2 \log \tau$
Effective samples per second

- Approximate algorithm is $50 \times$ more efficient with $n = 2,000$ and $p = 20,000$. 

![Histogram of effective samples per second](chart.png)
Accuracy

- Existing algorithms failed to converge, due to numerical underflow.

Trace plots for $-2\log(\sigma)$ and $\log(\xi) = -2\log(\tau)$; truth in red
Accuracy

- In terms of MSE, the approximation costs us little.

![Box plot showing MSE comparison between old, new, and approximate algorithms.](image)
Dependence on $p$ and $n$

- Effective sample sizes seem independent of $n$ and $p$. 

![Box plots showing effective sample sizes for independent and correlated designs.](image)
Dependence on $p$ and $n$

- Effective sample sizes seem independent of $n$ and $p$. 

![Box plots showing effective sample sizes for independent and correlated designs](image-url)
Real application: GWAS

- \( n = 2267 \) observations, \( p = 98385 \) SNPs in the genome of maize.
Real application: GWAS

- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.
- $X$: maize seeds; $y$: growing degree days to silking (‘growth cycle’)

Bimodal posterior distribution for $\beta | y$; Lasso (red) shrinks more than Horseshoe (blue)
Real application: GWAS

- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.
- $X$: maize seeds; $y$: growing degree days to silking (‘growth cycle’)

Bimodal posterior distribution for $\beta \mid y$; Lasso (red) shrinks more than Horseshoe (blue)
Real application: GWAS

- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.
- $X$: maize seeds; $y$: growing degree days to silking (‘growth cycle’)

Bimodal posterior distribution for $\beta | y$; Lasso (red) shrinks more than Horseshoe (blue)
Real application: GWAS

- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.
- $X$: maize seeds; $y$: growing degree days to silking (‘growth cycle’)

Bimodal posterior distribution for $\beta | y$; Lasso (red) shrinks more than Horseshoe (blue)
Variable selection with Horseshoe

Number of variables for which $\hat{\beta}_{HS,j} = \mathbb{E}[\beta_j \mid y] > t$ or $\hat{\beta}_{Lasso,j} > t$ vs threshold $t$; both methods largely agree on the identities of the signals.
There is no point in having a great model, like the Horseshoe, if it can't be computed.
Conclusion

▶ There is no point in having a great model, like the Horseshoe, if it can’t be computed.

▶ There is a need to scale more Bayesian models to the level of Frequentists.
Conclusion

- There is no point in having a great model, like the Horseshoe, if it can’t be computed.

- There is a need to scale more Bayesian models to the level of Frequentists.

- We manage to do that for the Horseshoe prior with two ideas: blocking and truncation.
Conclusion

- There is no point in having a great model, like the Horseshoe, if it can’t be computed.

- There is a need to scale more Bayesian models to the level of Frequentists.

- We manage to do that for the Horseshoe prior with two ideas: blocking and truncation.

- We observed interesting and novel statistical phenomena, e.g., bimodality of $\beta$. 
Conclusion

- There is no point in having a great model, like the Horseshoe, if it can’t be computed.

- There is a need to scale more Bayesian models to the level of Frequentists.

- We manage to do that for the Horseshoe prior with two ideas: blocking and truncation.

- We observed interesting and novel statistical phenomena, e.g., bimodality of $\beta$.

- There is likely more room for improvement.
References

Extra slides

- More simulation results
- Why “Horseshoe”?
More simulations

- We let $n = 1000$ and $p = 20,000$. 

![Graph showing the relationship between beta and a numerical value.](graph.png)
More simulations
More simulations
More simulations
More simulations

- The new algorithm lead to significant improvement in the autocorrelation:

![Markov chain run graph]

- Sampler:
  - new
  - old
More simulations

![Graph showing efficiency ratio and number of samples vs. number of features.]

- Efficiency ratio
- Number of samples, n
- Number of features, p

Equation: \( \frac{n_{e,new}}{n_{e,old}} \)
Why "Horseshoe"?

- In the orthogonal case with $n \geq p$ and $\sigma^2 = \tau = 1$, and defining a shrinkage profile $\kappa_j = 1/(1 + n\lambda_j^2)$, we can write $E[\beta_j|y] = (1 - E[\kappa_j|y])\hat{\beta}_j$. 

Prior density for $\kappa_j$: 

Paulo Orenstein  
Scalable MCMC for Bayes Shrinkage Priors  
Stanford University
Why "Horseshoe"?

- In the orthogonal case with $n \geq p$ and $\sigma^2 = \tau = 1$, and defining a shrinkage profile $\kappa_j = 1/(1 + n\lambda_j^2)$, we can write $E[\beta_j | y] = (1 - E[\kappa_j | y])\hat{\beta}_j$. 
Why "Horseshoe"?

- In the orthogonal case with $n \geq p$ and $\sigma^2 = \tau = 1$, and defining a shrinkage profile $\kappa_j = 1/(1 + n\lambda_j^2)$, we can write $\mathbb{E}[\beta_j|y] = (1 - \mathbb{E}[\kappa_j|y])\hat{\beta}_j$.

- Prior density for $\kappa_j$: