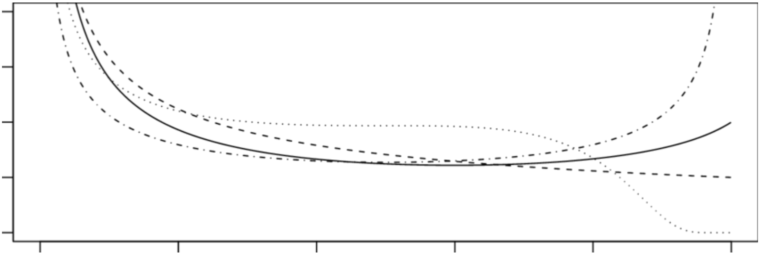
Scalable MCMC for Bayes Shrinkage Priors

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 $p \gg n$.

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

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

with $\beta_j = 0$ for many j.

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

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

with $\beta_j = 0$ for many j.

- How can we perform prediction and inference?

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

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

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

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

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

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

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

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

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

  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*:

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

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

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

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

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

* [Carvalho et. al, 2010]

Model

- The Horseshoe model*:

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

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

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

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

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

*[Carvalho et. al, 2010]

Model

- The Horseshoe model*:

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

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

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

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

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

*[Carvalho et. al, 2010]

Model

- The Horseshoe model*:

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

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

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

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

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

*[Carvalho et. al, 2010]

Model

- The Horseshoe model*:

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

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

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

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

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

*[Carvalho et. al, 2010]

Model

- Horseshoe has other good frequentist properties.

Model

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

Model

- 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],

$$\sup_{\beta:\|\beta\|_0 \le s_n} \mathbb{E}_\beta \left[\|\hat{\beta}_{HS} - \beta\|_2^2 \right] \le 4\sigma^2 s_n \log \frac{n}{s_n} \cdot (1 + o(1)),$$

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

$$\sup_{\beta:\|\beta\|_0 \le s_n} \mathbb{E}_\beta \left[\|\hat{\beta} - \beta\|_2^2 \right] \ge 2\sigma^2 s_n \log \frac{n}{s_n} \cdot (1 + o(1)).$$

Computation

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

Computation

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

Computation

- State-of-the-art: (i) $\tau \mid \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) \mid \tau, \lambda$, (iii) slice sampling for λ. *But...*
- We scale the model with two ideas.

Computation

- State-of-the-art: (i) $\tau \mid \beta, \sigma^2, \lambda$, (ii) $\left(\beta, \sigma^2\right) \mid \tau, \lambda$, (iii) slice sampling for λ. *But...*
- We scale the model with two ideas.
- First idea: **block** (β, σ^2, τ) 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.

Computation

- State-of-the-art: (i) $\tau \mid \beta, \sigma^2, \lambda$, (ii) $(\beta, \sigma^2) \mid \tau, \lambda$, (iii) slice sampling for λ. *But...*
- We scale the model with two ideas.
- First idea: **block** (β, σ^2, τ) 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**:

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

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) $\eta_j \mid u \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2}\eta_j} \mathbb{I}_{\left[\frac{1-u}{u} > \eta_j\right]}$.

Gibbs sampling

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

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

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) $\eta_j \mid u \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2}\eta_j} \mathbb{I}_{\left[\frac{1-u}{u} > \eta_j\right]}$.

Gibbs sampling

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

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

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) $\eta_j \mid u \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2}\eta_j} \mathbb{I}_{\left[\frac{1-u}{u} > \eta_j\right]}$.

Gibbs sampling

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

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

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) $\eta_j \mid u \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2}\eta_j} \mathbb{I}_{\left[\frac{1-u}{u} > \eta_j\right]}$.

Gibbs sampling

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

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

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) $\eta_j \mid u \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2}\eta_j} \mathbb{I}_{\left[\frac{1-u}{u} > \eta_j\right]}$.

Gibbs sampling

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

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

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) $\eta_j \mid u \sim e^{-\frac{\xi\beta_j^2}{2\sigma^2}\eta_j} \mathbb{I}_{\left[\frac{1-u}{u} > \eta_j\right]}$.

Gibbs sampling

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

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

Then perform slice sampling:

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

Markov approximation

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

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

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

Markov approximation

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

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

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

Markov approximation

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

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

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

- This makes computation much faster.

Markov approximation

- We approximate $M = X\text{diag}((\xi\eta_j)^{-1})X^T + I$ with
$$M_\delta = XD_\delta X^T + I, \qquad D_\delta = \text{diag}((\xi\eta_j)^{-1}\mathbb{I}_{[(\xi_{\max}\eta_j)^{-1} > \delta]})$$
for $\delta \ll 1$, and $\xi_{\max}$ the maximum of the current and proposed ξ.
- This makes computation much faster.

Approximating Kernels

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

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

for sufficiently small $\delta > 0$.

Simulation

- We simulate data as follows:

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

Simulation

- We simulate data as follows:

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

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

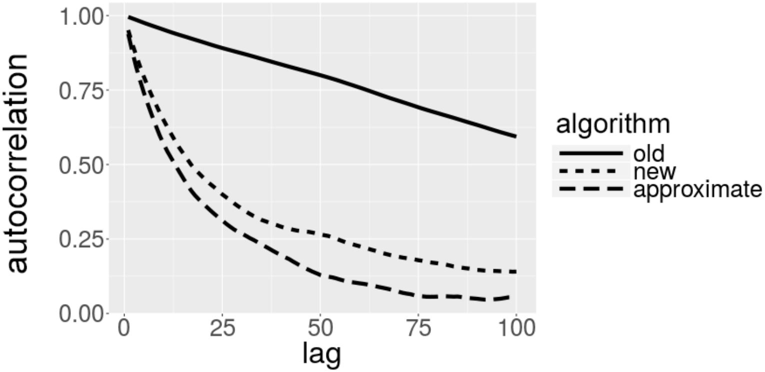
Simulation

- We simulate data as follows:

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

- There are nulls, clear non-nulls, and some subtle non-nulls.
- We consider both $\boldsymbol{\Sigma} = I$ (independent design) and $\boldsymbol{\Sigma}_{ij} = 0.9^{|i-j|}$ (correlated design).

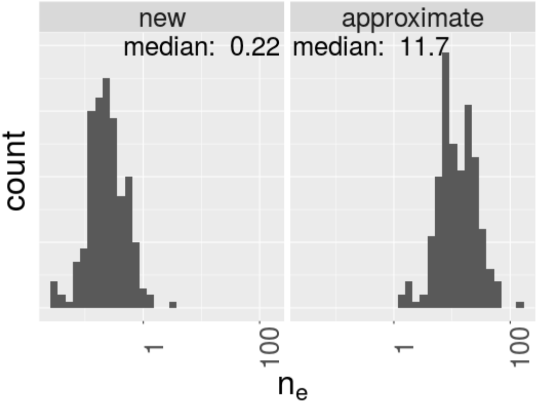
Autocorrelation



Autocorrelation for $\log(\xi) = -2 \log \tau$

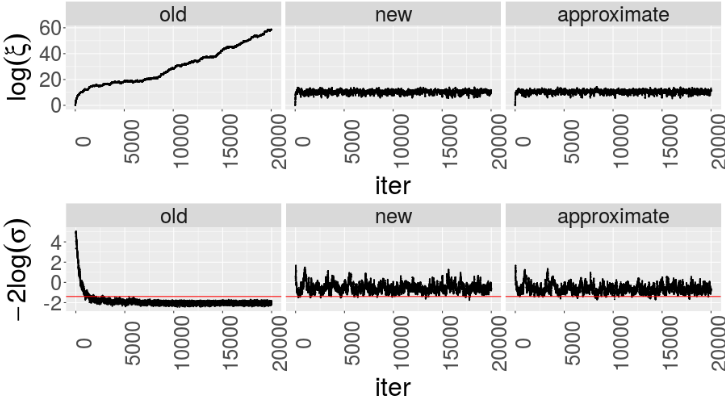
Effective samples per second

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



Accuracy

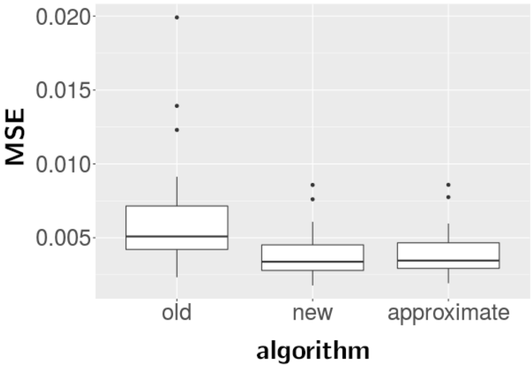
- Existing algorithms failed to converge, due to numerical underflow.



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

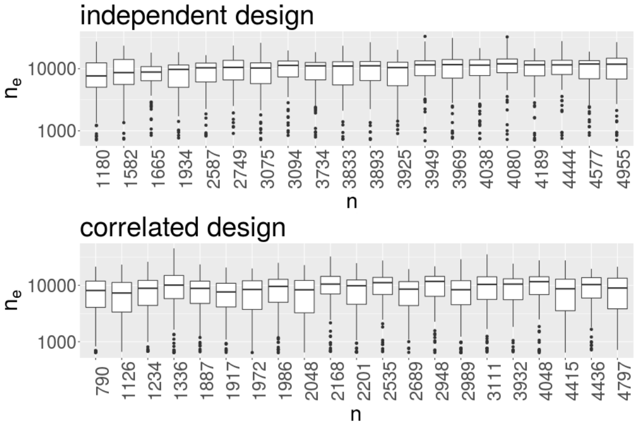
Accuracy

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



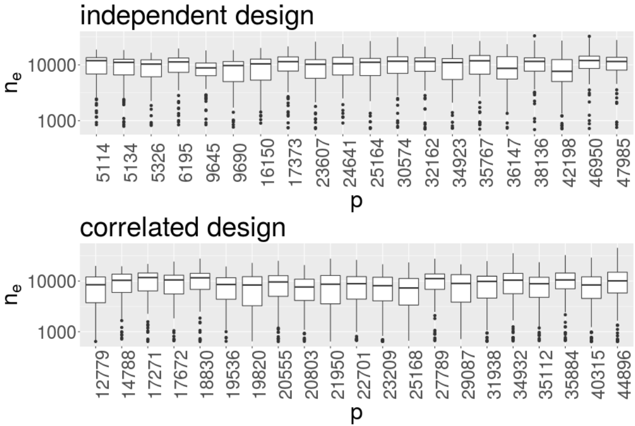
Dependence on p and n

- Effective sample sizes seem independent of n and p.



Dependence on p and n

- Effective sample sizes seem independent of n and p.



Real application: GWAS

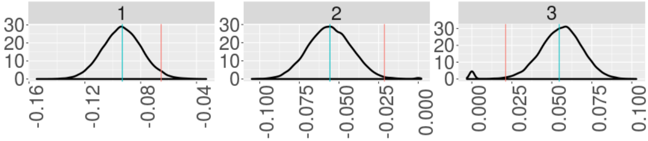
- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.

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')

Real application: GWAS

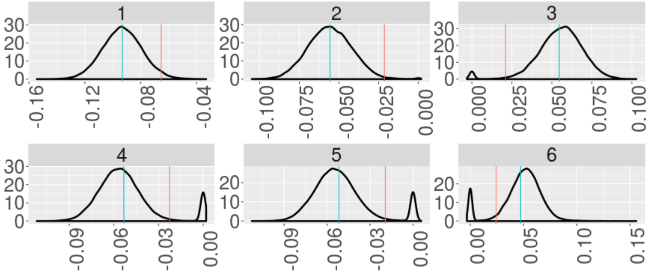
- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.
- X: maize seeds; y: growing degree days to silking ('growth cycle')



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

Real application: GWAS

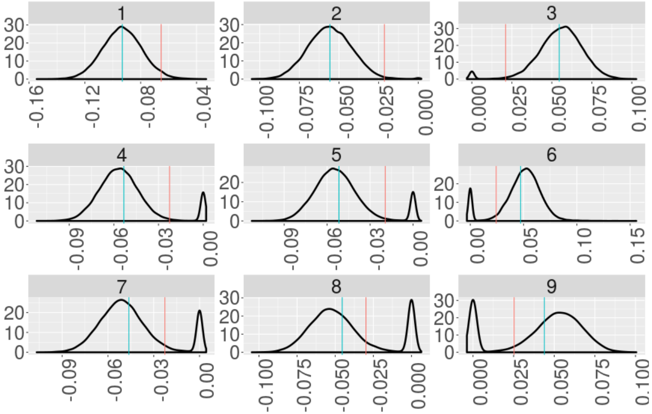
- $n = 2267$ observations, $p = 98385$ SNPs in the genome of maize.
- X: maize seeds; y: growing degree days to silking ('growth cycle')



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

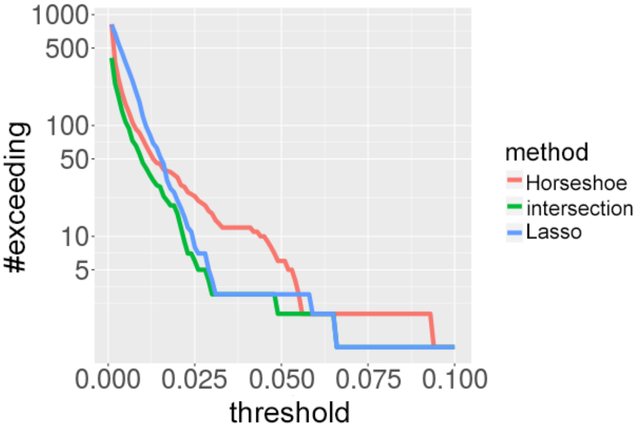
Real application: GWAS

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



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

Variable selection with Horseshoe



Number of variables for which $\hat{\beta}_{\mathrm{HS},j} = \mathbb{E}[\beta_j \mid y] > t$ or $\hat{\beta}_{\mathrm{Lasso},j} > t$ vs threshold t; both methods largely agree on the identities of the signals

Conclusion

- 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 β.

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 β.
- There is likely more room for improvement.

References

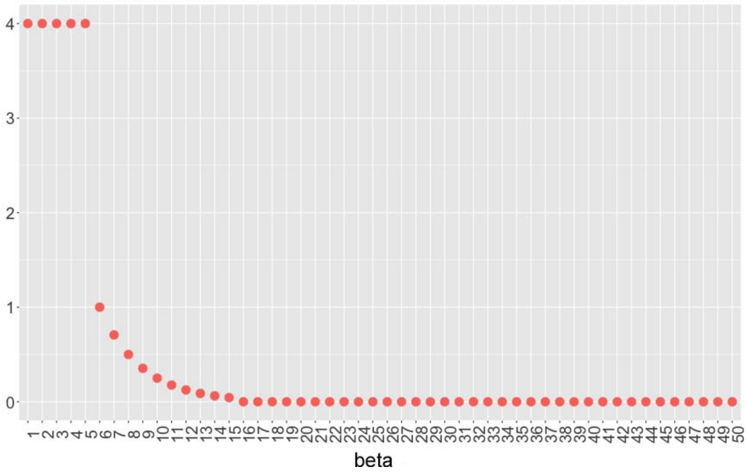
- Bhattacharya, Anirban, et al. "Bayesian shrinkage". *arXiv preprint*, *arXiv: 1212.6088* (2012).
- Carvalho, Carlos M., Nicholas G. Polson, and James G. Scott. "The horseshoe estimator for sparse signals". *Biometrika* 97.2 (2010): 465-480.
- Johndrow, James E., and Jonathan C. Mattingly. "Error bounds for Approximations of Markov chains." *arXiv preprint*, *arXiv:1711.05382* (2017).
- Johndrow, James E., P. O., Bhattacharya, Anirban "**Scalable MCMC for Bayes Shrinkage Priors**". *arXiv preprint*, *arXiv: 1705.00841* (2018).
- Rudolf, Daniel, and Nikolaus Schweizer. "Perturbation theory for Markov chains via Wasserstein distance." *Bernoulli* 24.4A (2018): 2610-2639.
- Van Der Pas, S. L., B. J. K. Kleijn, and A. W. Van Der Vaart. "The horseshoe estimator: Posterior concentration around nearly black vectors." *Electronic Journal of Statistics* 8.2 (2014): 2585-2618.

Extra slides

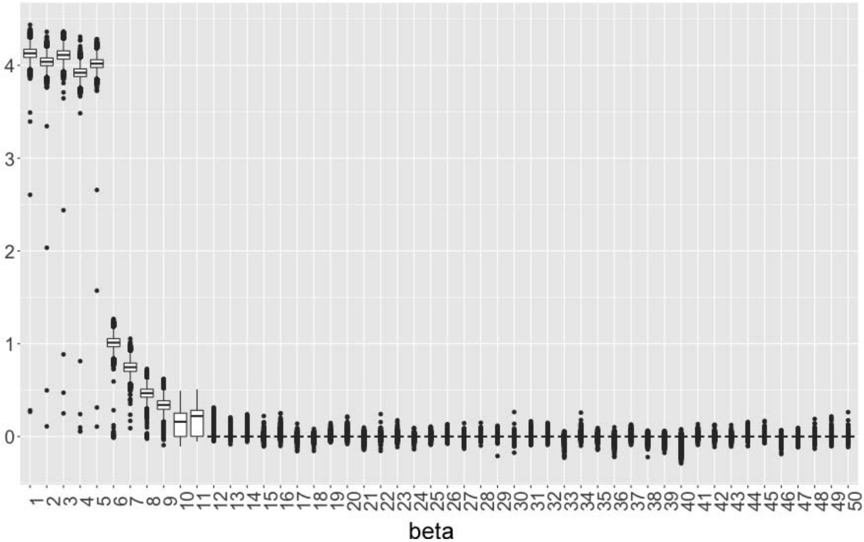
- More simulation results
- Why “Horseshoe”?

More simulations

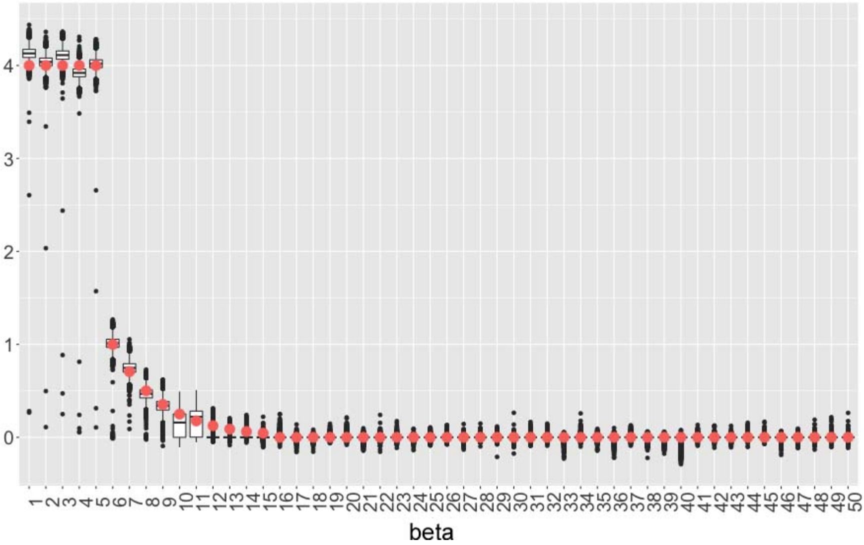
- We let $n = 1000$ and $p = 20,000$.



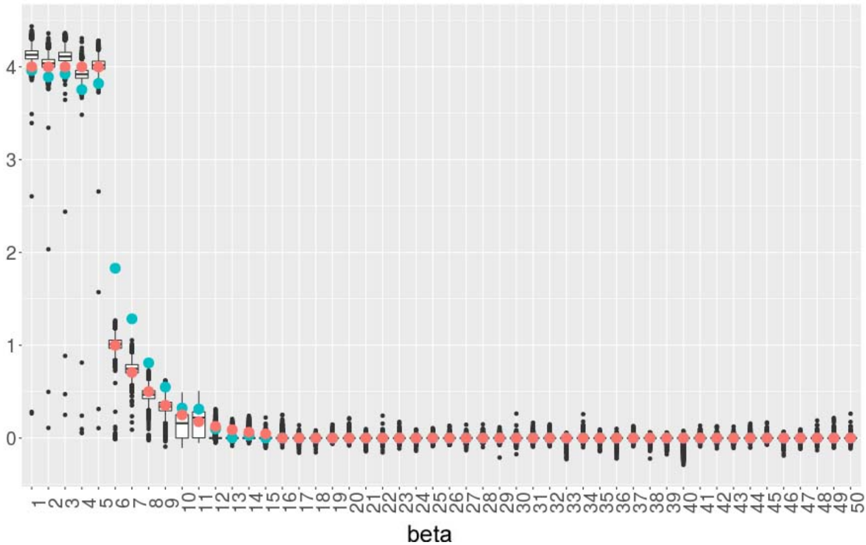
More simulations



More simulations

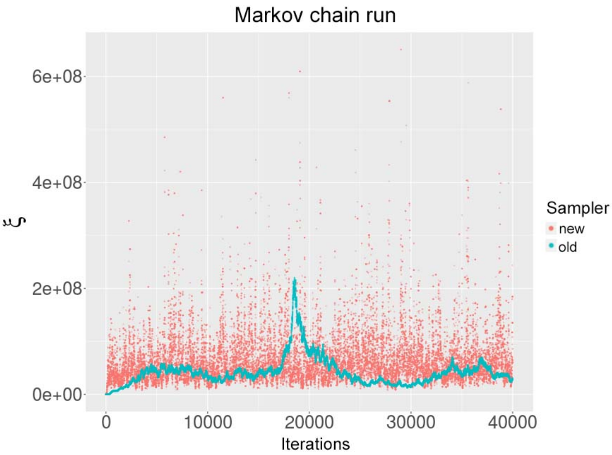


More simulations

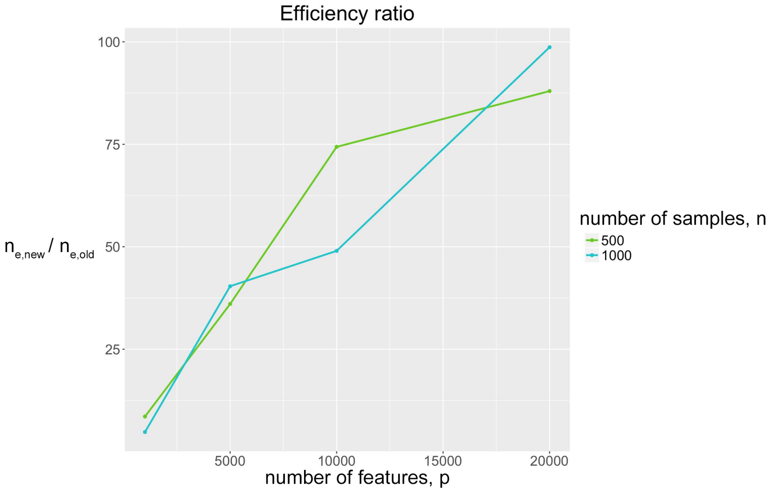


More simulations

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



More simulations



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 $\mathbb{E}[\beta_j|y] = (1 - \mathbb{E}[\kappa_j|y])\hat{\beta}_j$.

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 $\mathbb{E}[\beta_j|y] = (1 - \mathbb{E}[\kappa_j|y])\hat{\beta}_j$.

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 κ_j: