Efficient sampling for Gaussian linear regression with arbitrary priors

P. Richard Hahn and Jingyu He*
Booth School of Business, University of Chicago
and
Hedibert F. Lopes
INSPER Institute of Education and Research

July 9, 2017

Abstract

This paper develops a slice sampler for Bayesian linear regression models with arbitrary priors. The new sampler has two advantages over current approaches. One, it is faster than many custom implementations that rely on auxiliary latent variables, if the number of regressors is large. Two, it can be used with any prior with a density function that can be evaluated up to a normalizing constant, making it ideal for investigating the properties of new shrinkage priors without having to develop custom sampling algorithms. The new sampler takes advantage of the special structure of the linear regression likelihood, allowing it to produce better effective sample size per second than common alternative approaches.

Keywords: Bayesian computation, linear regression, shrinkage priors, slice sampling

*The authors gratefully acknowledge *please remember to list all relevant funding sources in the unblinded version*

1 Introduction

This paper develops a computationally efficient posterior sampling algorithm for Bayesian linear regression models with Gaussian errors. The new algorithm does not rely on case-specific auxiliary variable representations, which has two immediate advantages. First, it avoids expanding the parameter space by the number of predictor variables, which can have direct computational gains. Second, it allows for rapid prototyping of samplers for novel shrinkage priors that do not necessarily admit any obvious auxiliary representation, eliminating a barrier to using distributions outside the conditionally Gaussian framework.

Specifically, we propose a slice-within-Gibbs sampler based on the elliptical slice sampler of Murray et al. [2010]. These authors focus on sampling from posteriors that are proportional to a product of a multivariate Gaussian prior and an arbitrary likelihood. Intuitively, the elliptical slice sampler operates by drawing samples from the Gaussian factor of the posterior and then accepting or rejecting these samples by evaluating the non-Gaussian factor. The starting point of this paper is the observation that the elliptical slice sampler is also suited to the Bayesian Gaussian linear regression case, which has a multivariate Gaussian likelihood and an arbitrary prior. In fact, under independent priors over the regression coefficients, the Gaussian likelihood term contains all of the co-dependence information of the posterior, allowing us to pre-compute many key quantities, leading to a remarkably efficient, generic algorithm.

In this paper, the efficiency of the elliptical slice sampler will be illustrated on linear regression models using three widely-used shrinkage priors: ridge priors, Laplace priors [Park and Casella, 2008, Hans, 2009], and horseshoe priors [Carvalho et al., 2010]. We further demonstrate the flexibility of our approach by using it to perform posterior inference under two non-standard, "exotic", priors — an asymmetric Cauchy prior and a "non-local" two-component mixture prior.

The next section presents the elliptical slice sampler for regression with arbitrary priors. Section 3.1 reports the results of simulation studies. Section 3.2 presents an empirical demonstration and Section 4 concludes with a discussion.

2 Elliptical slice sampling for shrinkage regression with arbitrary priors

2.1 Review of elliptical slice sampling for Gaussian priors

To begin, we review the elliptical slice sampler of Murray et al. [2010]. In the following subsections we adapt the sampler specifically for use with Gaussian linear regression models. Unless otherwise noted, random variables (possibly vector-valued) are denoted by capital Roman letters, matrices are in bold, vectors are in Roman font, and scalars are italic. All vectors are column vectors.

The elliptical slice sampler considers cases where the goal is to sample from a distribution $p(\Delta) \propto \mathrm{N}(\Delta; 0, \mathbf{V})L(\Delta)$. The key idea is to take advantage of the elementary fact that the sum of two Gaussian random variables is a Gaussian random variable. Accordingly, for two independent (vector) random variables $v_0 \sim \mathrm{N}(0, \mathbf{V})$ and $v_1 \sim \mathrm{N}(0, \mathbf{V})$ and for any $\theta \in [0, 2\pi]$, $\Delta = v_0 \sin\theta + v_1 \cos\theta$ is also distributed according to $\mathrm{N}(0, \mathbf{V})$, since $\mathrm{cov}(\Delta) = \mathbf{V}\sin^2\theta + \mathbf{V}\cos^2\theta = \mathbf{V}$. Because this holds for each θ, the marginal distribution of Δ is $\mathrm{N}(0, \mathbf{V})$ for any distribution over θ.

Therefore, Murray et al. [2010] note that if one can sample from the parameter-expanded model $p(v_0, v_1, \theta) \propto \pi(\theta)N(v_0; 0, \mathbf{V})N(v_1; 0, \mathbf{V})L(v_0 \sin\theta + v_1 \cos\theta)$, then samples from $p(\Delta)$ can be obtained as samples of the transformation $v_0 \sin\theta + v_1 \cos\theta$. Sampling from this model is easiest to explain in terms of a singular Gaussian prior distribution over $(v_0^t, v_1^t, \Delta^t)^t$ with covariance

$$\boldsymbol{\Sigma}_\theta = \begin{pmatrix} \mathbf{V} & 0 & \mathbf{V}\sin\theta \\ 0 & \mathbf{V} & \mathbf{V}\cos\theta \\ \mathbf{V}\sin\theta & \mathbf{V}\cos\theta & \mathbf{V} \end{pmatrix}$$

and joint density $p(v_0, v_1, \Delta, \theta) \propto \mathrm{N}(0, \boldsymbol{\Sigma}_\theta)L(v_0 \sin\theta + v_1 \cos\theta)$. Using this model, we sample the parameters $(v_0, v_1, \Delta, \theta)$ via a two-block Gibbs sampler:

1. Sample from $p(v_0, v_1 \mid \Delta, \theta)$, which can be achieved by sampling $v \sim N(0, \mathbf{V})$ and setting $v_0 = \Delta \sin\theta + v\cos\theta$ and $v_1 = \Delta\cos\theta - v\sin\theta$.

2. Sample from $p(\Delta, \theta \mid v_0, v_1) \propto \mathrm{N}(0, \boldsymbol{\Sigma}_\theta)L(v_0 \sin\theta + v_1 \cos\theta)$ compositionally in two

steps:

(a) First draw from $p(\theta \mid v_0, v_1)$ by marginalizing over Δ, yielding $p(\theta \mid v_0, v_1) \propto L(v_0 \sin\theta + v_1 \cos\theta)$. We draw from this distribution via a traditional one-dimensional slice sampler [Neal, 2003]. Initialize $a = 0$ and $b = 2\pi$.

i. Draw ℓ uniformly on $[0, L(v_0 \sin\theta + v_1 \cos\theta)]$.

ii. Sample θ' uniformly on $\theta \in [a, b]$.

iii. If $L(v_0 \sin\theta' + v_1 \cos\theta') > \ell$, set $\theta \leftarrow \theta'$. Otherwise, shrink the support of θ' (if $\theta' < \theta$, set $a \leftarrow \theta'$; if $\theta' > \theta$, set $b \leftarrow \theta'$), and go to step ii.

(b) Then we draw from $p(\Delta \mid \theta, v_0, v_1)$, which is degenerate at $\Delta = v_0 \sin\theta + v_1 \cos\theta$.

Note that this version of the elliptical slice sampler is somewhat different than the versions presented in Murray et al. [2010], but as it reduces to a Gibbs sampler, its validity is more transparent and practically the algorithms are nearly equivalent.

2.2 Elliptical slice sampling for Gaussian linear regression

In this section we adapt the sampler described above to permit efficient sampling from Bayesian linear regression models. Specifically, we consider the standard Bayesian linear regression model:

$$Y = \mathbf{X}\beta + \epsilon, \tag{1}$$

where Y is an n-by-1 vector of responses, $\mathbf{X}$ is an n-by-p matrix of regressors, β is a p-by-1 vector of coefficients and $\epsilon \sim \mathrm{N}(0, \sigma^2\mathbf{I})$ is an n-by-1 vector of error terms. Denote the prior of β as $\pi(\beta)$. The objective is to sample from a posterior expressible as

$$\pi(\beta \mid \mathrm{y}, \mathbf{X}, \sigma^2) \propto f(\mathrm{y} \mid \mathbf{X}, \beta, \sigma^2\mathbf{I})\pi(\beta) \tag{2}$$

where $f(\mathrm{y} \mid \mathbf{X}, \beta)$ is a Gaussian likelihood $\mathrm{N}_{\mathrm{y}}(\mathbf{X}\beta, \sigma^2\mathbf{I})$.

Our approach is driven by the fact that, up to proportionality, $\mathrm{N}_{\mathrm{y}}(\mathbf{X}\beta, \sigma^2\mathbf{I})$ can be

Algorithm 1 : *Elliptical slice sampler for linear regression*

For initial value β, with $\Delta = \beta - \hat{\beta}$, and σ^2 fixed:

1. Draw $v \sim \mathrm{N}(0, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$. Set $v_0 = \Delta \sin\theta + v\cos\theta$ and $v_1 = \Delta\cos\theta - v\sin\theta$.
2. Draw ℓ uniformly on $[0, \pi(\hat{\beta} + v_0\sin\theta + v_1\cos\theta)]$. Initialize $a = 0$ and $b = 2\pi$.
   (a) Sample θ' uniformly on $[a, b]$.
   (b) If $\pi(\hat{\beta} + v_0\sin\theta' + v_1\cos\theta') > \ell$, set $\theta \leftarrow \theta'$. Otherwise, shrink the support of θ' (if $\theta' < \theta$, set $a \leftarrow \theta'$; if $\theta' > \theta$, set $b \leftarrow \theta'$), and go to step ii.
3. Return $\Delta = v_0\sin\theta + v_1\cos\theta$ and $\beta = \hat{\beta} + \Delta$.

Figure 1: The elliptical slice sampler for linear regression (with an arbitrary prior) samples all p elements of the regression coefficients simultaneously.

regarded as the posterior of β under a flat prior (indicated by the 0 subscript):

$$\begin{aligned}\pi_0(\beta \mid \sigma^2, \mathrm{y}, \mathbf{X}) &\propto \mathrm{N}_\mathrm{y}(\mathbf{X}\beta, \sigma^2\mathbf{I}) \\ &\propto \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(\mathrm{y} - \mathbf{X}\widehat{\beta})^T(\mathrm{y} - \mathbf{X}\widehat{\beta})\right] \times \\ &\quad \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(\beta - \widehat{\beta})^T\mathbf{X}^T\mathbf{X}(\beta - \widehat{\beta})\right] \\ &= \mathrm{N}_\beta(\hat{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})\end{aligned} \tag{3}$$

where $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathrm{y}$ is the ordinary least squares (OLS) estimator. Therefore, the slice sampler of [Murray et al., 2010] can be applied directly, using $\pi_0(\beta \mid \sigma^2, \mathrm{y}, \mathbf{X})$ as the Gaussian "prior" and $\pi(\beta)$ as the "likelihood". One minor modification is that, because $\pi_0(\beta \mid \sigma^2, \mathrm{y}, \mathbf{X})$ is centered at OLS estimator $\widehat{\beta}$, as opposed to 0, we sample the offset of β from $\widehat{\beta}$, which we denote $\Delta = \beta - \hat{\beta}$.

This sampler is flexible because the only requirement is that the prior function $\pi(\beta)$ can be evaluated up to a normalizing constant. The sampler is efficient because in each Monte Carlo iteration, the sampler draws a single multivariate Gaussian random variable,

and then draws from a univariate uniform distribution within the while loop. The size of the sampling region for θ shrinks rapidly with each rejected value and is guaranteed to eventually accept. Sampling of σ^2 can be done after sampling β in each iteration, using a Metropolis-Hastings step.

Despite being quite fast, for larger regression problems, with p having more than a few dozen elements, the auto-correlation from this joint sampler can be prohibitively high, yielding very low effective sample sizes. Intuitively, this occurs because for any given auxiliary variables (v_0, v_1), the slice step over θ frequently has only a very narrow acceptable region, entailing that subsequent samples of θ (and hence β) will be very close to one another. Fortunately, the basic strategy of the elliptical slice sampler can be applied to smaller blocks, an approach we describe in the following section.

2.3 Elliptical slice-within-Gibbs for linear regression

As mentioned above, if the number of regression coefficients p is large, the slice which contains acceptable proposals is likely to be minuscule. Due to the shrinking bracket mechanism of the slice sampler, it rejects many proposals and shrinks the bracket strongly towards the initial value, thereby inducing high autocorrelation in the obtained samples. Here, we propose a slice-within-Gibbs sampler (Figure 2) to mitigate this problem.

Because β has a jointly Gaussian likelihood and independent priors, it is natural to implement a Gibbs sampler, updating a subset, which we denote β^k, given all other coefficients, which we denote β^{-k} (and other parameters), in each MCMC iteration. This is possible because the conditional distribution for the Gaussian portion of the distribution, which accounts for all the dependence, is well-known and easy to sample from. The basic idea is simply to apply the sampler from Figure 1 using the conditional distribution $\beta^k \mid \beta^{-k}$ as the "likelihood" instead of the full likelihood $L(\beta)$.

From equation 3, the joint likelihood of β is $N(\widehat{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$. Therefore, we group elements of β into several blocks $\beta = (\beta_1, \cdots, \beta_p) = \{\beta^1, \beta^2, \cdots, \beta^K\}$, constructing a Gibbs sampling scheme for all K blocks, using the elliptical slice sampler to update each block.

We can rearrange terms of the joint distribution as

$$\begin{bmatrix} \beta^k \\ \beta^{-k} \end{bmatrix} = N\left(\begin{bmatrix} \widehat{\beta}^k \\ \widehat{\beta}^{-k} \end{bmatrix}, \sigma^2 \begin{bmatrix} \mathbf{\Sigma}_{k,k} & \mathbf{\Sigma}_{k,-k} \\ \mathbf{\Sigma}_{-k,k} & \mathbf{\Sigma}_{-k,-k} \end{bmatrix} \right) \tag{4}$$

where $\begin{bmatrix} \widehat{\beta}^k \\ \widehat{\beta}^{-k} \end{bmatrix} = \widehat{\beta}$, the OLS estimator and $\begin{bmatrix} \mathbf{\Sigma}_{k,k} & \mathbf{\Sigma}_{k,-k} \\ \mathbf{\Sigma}_{-k,k} & \mathbf{\Sigma}_{-k,-k} \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}$.

The corresponding conditional distribution of β^k given β^{-k} is $N(\tilde{\beta}^k, \tilde{\mathbf{\Sigma}}^k)$ where

$$\tilde{\beta}^k = \widehat{\beta}^k + \mathbf{\Sigma}_{k,-k}\mathbf{\Sigma}_{-k,-k}^{-1}(\beta^{-k} - \widehat{\beta}^{-k}) \tag{5}$$

$$\tilde{\mathbf{\Sigma}}^k = \sigma^2 \left(\mathbf{\Sigma}_{k,k} - \mathbf{\Sigma}_{k,-k}\mathbf{\Sigma}_{-k,-k}^{-1}\mathbf{\Sigma}_{-k,k} \right). \tag{6}$$

Note that the grouping of coefficients is arbitrary; if all coefficients are grouped in a single block, we recover the original sampler from Figure 1. Empirically, we find that putting each coefficient in a different block so that $K = p$, and updating coefficients one by one, gives excellent performance. Our complete algorithm is given in Figure 2.

2.3.1 Computational cost

Although the slice-within-Gibbs sampler similarly updates the coefficients iteratively, it is more efficient than Gibbs samplers based on conditionally Gaussian representations because the structure of the slice sampler allows the necessary matrix factorizations and inversions to be pre-computed outside the main Gibbs loop. Specifically, to efficiently compute the K conditional mean vectors and covariance matrices given in expressions (5) and (6) we can precompute $\mathbf{\Sigma}_{k,-k}\mathbf{\Sigma}_{-k,-k}^{-1}$, $\mathbf{\Sigma}_{k,k} - \mathbf{\Sigma}_{k,-k}\mathbf{\Sigma}_{-k,-k}^{-1}\mathbf{\Sigma}_{-k,k}$, and Cholesky factors $\mathbf{L}_k$, with $\mathbf{L}_k\mathbf{L}_k^T = \mathbf{\Sigma}_{k,k} - \mathbf{\Sigma}_{k,-k}\mathbf{\Sigma}_{-k,-k}^{-1}\mathbf{\Sigma}_{-k,k}$, for each $k = 1, \ldots, K$. By contrast, Gibbs samplers based on conditionally Gaussian representations (e.g., Armagan et al. [2011]) have full conditional updates of the form

$$(\beta_j \mid -) \sim \mathrm{N}((\mathbf{X}^T\mathbf{X} + \mathbf{D})^{-1}\mathbf{X}^T\mathrm{y}, \sigma^2(\mathbf{X}^T\mathbf{X} + \mathbf{D})^{-1}\mathbf{X}^T\mathrm{y}), \tag{8}$$

which require costly Cholesky or eigenvalue decompositions of the matrix $(\mathbf{X}^T\mathbf{X} + \mathbf{D})^{-1}$ at each iteration as $\mathbf{D}$ is updated — eliminating this step at each iteration is the primary savings of the new algorithm.

Algorithm 2 : *Slice-within-Gibbs sampler for linear regression*

- For each k from 1 to K

  Update $\beta^k \mid \beta^{\{-k\}}, \sigma^2, \lambda$ according to Algorithm 1.

  1. Construct conditional mean $\tilde{\beta}^k$ and conditional covariance matrix $\tilde{\mathbf{\Sigma}}^k$ as expressions (5) and (6). Set $\Delta^k = \beta^k - \tilde{\beta}^k$. Draw $v \sim N(0, \tilde{\mathbf{\Sigma}}^k)$. Set $v_0 = \Delta^k \sin\theta^k + v\cos\theta^k$ and $v_1 = \Delta^k \cos\theta^k - v\sin\theta^k$.
  2. Draw ℓ uniformly on $[0, \pi(\Delta^k + \tilde{\beta}^k)]$. Initialize $a = 0$ and $b = 2\pi$.
     (a) Sample θ' uniformly on $[a, b]$.
     (b) If $\pi(\tilde{\beta}^k + v_0 \sin\theta' + v_1 \cos\theta') > \ell$, set $\theta^k \leftarrow \theta'$. Otherwise, shrink the support of θ' (if $\theta' < \theta^k$, set $a \leftarrow \theta'$; if $\theta' > \theta^k$, set $b \leftarrow \theta'$), and go to step (a).
  3. Return $\Delta^k = v_0 \sin\theta^k + v_1 \cos\theta^k$ and $\beta^k = \tilde{\beta}^k + \Delta^k$.

- Update $\sigma^2 \mid \boldsymbol{\beta}, \lambda$: let $s = (\mathrm{y} - \mathbf{X}\beta)^T(\mathrm{y} - \mathbf{X}\beta)$. Draw $\sigma^2 \sim \mathrm{IG}((n+\alpha)/2, (s+\gamma)/2)$, where IG denotes the inverse-Gamma distribution and (α, γ) are the associated prior parameters.

- Update $\lambda \mid \boldsymbol{\beta}, \sigma^2$, random walk Metropolis-Hastings update on log scale with a diffuse Gaussian prior:

  1. Draw $r \sim \mathrm{N}(0, 0.05^2)$, let $\lambda_{\text{proposal}} = \exp(\log(\lambda) + r)$.
  2. Compute the M-H ratio

$$\begin{aligned} \eta = \exp(&\log \pi(\beta, \lambda_{\text{proposal}}) + \log \phi(\lambda_{\text{proposal}}; 0, 100) \\ &- \log \pi(\beta, \lambda) - \log \phi(\lambda; 0, 100) + \log(\lambda_{\text{proposal}}) - \log(\lambda)) \end{aligned} \tag{7}$$

     where $\phi(\cdot; m, v)$ denotes the Gaussian density with mean m and variance v.
  3. Draw $u \sim \mathrm{Unif}(0, 1)$, if $u < \eta$, accept $\lambda_{\text{proposal}}$; otherwise keep the current λ.

Figure 2: The full slice-within-Gibbs sampler, including update steps for σ^2 and λ.

Finally, although the slice approach may incur additional computational cost if many proposals are rejected in each iteration prior to acceptance, we find that not to be the case. For instance, if $p = 100$, $n = 1000$ and signal-to-noise ratio is 1, the average number of rejections prior to acceptance (per iteration) is 1.2; if the signal-to-noise ratio is 1/4, the average number of rejections increases to 2.9.

2.3.2 The rank-deficient case

It is increasingly common to want to analyze regression models with more predictors than observations: $p > n$. Similarly, it is sometimes the case that $\mathbf{X}^T\mathbf{X}$ can be rank deficient due to perfect colinearity in the predictor matrix $\mathbf{X}$. Such cases would seem to pose a problem for our method for the following reason. Recall that the slice sampler draws from a target distribution of the form

$$\begin{aligned} p(\beta \mid \mathrm{y}, \mathbf{X}, \sigma) &\propto \mathrm{N}_Y(\mathbf{X}\beta, \sigma^2)\pi(\beta) \\ &\propto \mathrm{N}_\beta(\hat{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})\pi(\beta). \end{aligned} \tag{9}$$

If $\mathrm{rank}(\mathbf{X}^T\mathbf{X}) = r < p$, then the first term on the right-hand side is not absolutely continuous with respect to the second term, and the sampler will not function properly. Intuitively, the proposal distribution is supported on an $r < p$ dimensional hyperplane. The slice sampler will never propose values off of this subspace; hence it cannot have the correct target distribution. Operationally, $\hat{\beta}$ is not even unique. Fortunately, the algorithm can be salvaged with a very minor modification inspired by ridge regression analysis. We rewrite the above expression as

$$\begin{aligned} p(\beta \mid \mathrm{y}, \mathbf{X}, \sigma) &\propto \mathrm{N}_Y(\mathbf{X}\beta, \sigma^2)\mathrm{N}_\beta(0, c\sigma^2\mathbf{I})\frac{\pi(\beta)}{\mathrm{N}_\beta(0, c\sigma^2\mathbf{I})} \\ &\propto \mathrm{N}_\beta(\bar{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X} + c^{-1}\mathbf{I})^{-1})\frac{\pi(\beta)}{\mathrm{N}_\beta(0, c\sigma^2\mathbf{I})}. \end{aligned} \tag{10}$$

for $c > 0$, where $\bar{\beta} = (\mathbf{X}^T\mathbf{X} + c^{-1}\mathbf{I})^{-1}\mathbf{X}^T\mathrm{y}$. In the first line we merely "multiplied by 1"; subsequent lines reorganize the distributions in the familiar form required by the slice sampler. This reformulation solves the problem of absolute continuity; $\bar{\beta}$ is now well-defined and $(\mathbf{X}^T\mathbf{X} + c^{-1}\mathbf{I})$ is full rank. Otherwise, the algorithm operates exactly as before, only feeding in $\bar{\beta}$ rather than $\hat{\beta}$ and $(\mathbf{X}^T\mathbf{X} + c^{-1}\mathbf{I})$ rather than $\mathbf{X}^T\mathbf{X}$ and evaluating the ratio

$\pi(\beta)/\mathrm{N}(0, c\sigma^2\mathbf{I})$ where one would have otherwise evaluated the prior $\pi(\beta)$. The optimal value of c will differ depending on the data and the prior being used; in practice small values near one seem to work fine, but tuning based on pilot runs could be performed if desired.

3 Demonstrations

3.1 Simulation studies

In this section we compare the performance of our new algorithm against several well-known alternatives. Specifically, we apply our approach to the horseshoe prior [Carvalho et al., 2010], the Laplace prior [Park and Casella, 2008, Hans, 2009] and the independent Gaussian or "ridge" prior. These three priors are frequently used in empirical studies in part because they have readily available implementations. Although other shrinkage priors have been proposed and studied, many have not been implemented in the regression setting and hence are not widely used outside of the normal-means context; consequently, we restrict our comparison to three popular regression priors. Recently developed priors we do not consider here include the Dirichlet-Laplace prior [Bhattacharya et al., 2015], the normal-gamma prior [Caron and Doucet, 2008, Griffin et al., 2010, Griffin and Brown, 2011], the Bayesian bridge [Polson et al., 2014] and many others [Armagan, 2009, Armagan et al., 2011, 2013, Neville et al., 2014, Polson and Scott, 2010].

The goal here is merely to demonstrate the efficacy of our computational approach, not to advocate for any particular prior choice. Indeed, our hope is that having a generic sampling scheme for any prior will make computational considerations secondary when choosing one's prior. Ideally, one would not select a prior merely on the grounds that it admits an efficient sampling algorithm. In other words, the selling point of the present approach is not that it is strictly better than the existing samplers for these models (it is not necessarily), rather it is that we are using the *same* underlying algorithm for all three of them, with no custom modification, and are still achieving competitive (or superior) computational performance.

In the following subsections, we detail the priors considered as well as our precise data

generating process.

3.1.1 Priors

The `R` package `monomvn` [Gramacy, 2017] implements the standard Gibbs samplers for the horseshoe prior (function `bhs`), the Laplace prior (function `blasso`), and the ridge (function `bridge`) prior. For Laplace prior, we additionally compare with the Gibbs sampler from Hans [2009]. All of the samplers are implemented in `C++`.

Horseshoe prior. The horseshoe prior can be expressed as a local scale-mixture of Gaussians

$$\beta \sim N(0, \lambda^2\Lambda^2), \lambda \sim C^+(0,1), \lambda_1, ..., \lambda_p \overset{\text{iid}}{\sim} C^+(0,1), \tag{11}$$

where $C^+(0,1)$ is half standard Cauchy distribution, $\Lambda = \text{diag}(\lambda_1, ...\lambda_p)$ represents the local shrinkage parameters and λ is the global shrinkage parameter. The standard approach to sampling from the posterior of regressions under horseshoe priors is a Gibbs sampler which samples $(\lambda_1, \ldots, \lambda_p)$ from their full conditionals.

The horseshoe density, integrating over the local scale factors λ_j, can be computed using special functions. However, the following bounds [Carvalho et al., 2010] provide an excellent approximation which is more straightforward to evaluate:

$$\frac{K}{2}\log\left(1+\frac{4}{(\beta_j/\lambda_0)^2}\right) < \pi(\beta_j/\lambda_0) < K\log\left(1+\frac{2}{(\beta_j/\lambda_0)^2}\right) \tag{12}$$

where $K = 1/(2\pi^3)^{1/2}$. In our implementation we use this lower bound as the density function.

Laplace prior. The Laplace (double-exponential) prior is given by

$$\pi(\beta_j \mid \lambda) = \frac{1}{2}\lambda^{-1}\exp(-|\beta_j|/\lambda). \tag{13}$$

Park and Casella [2008] gives the first treatment of Bayesian lasso regression and Hans [2009] proposes alternative Gibbs samplers.

Ridge prior. The ridge prior is given by

$$\beta \mid \lambda \sim N(0, \lambda^2\mathbf{I}_p). \tag{14}$$

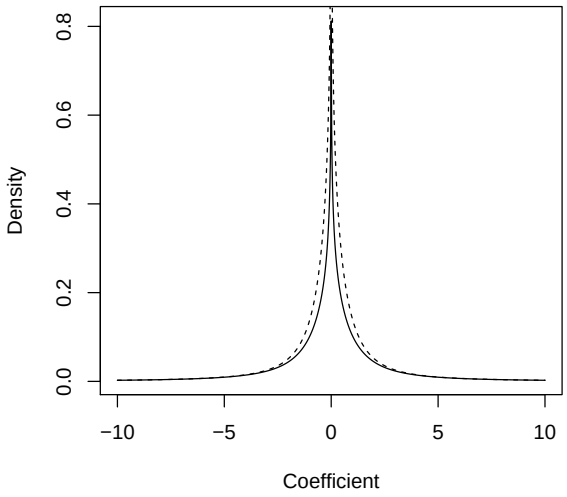
Section 2.3.2 of Gamerman and Lopes [2006] provides a nice exposition of Bayesian ridge regression.

Note that all three priors have a "global" shrinkage parameter λ, which is given a hyperprior. A key feature of Bayesian shrinkage regression is the inference of this parameter; as opposed to setting it at a fixed value or selecting it by cross-validation, point estimates of β are obtained as the marginal posterior mean, integrating over (the posterior of) λ. Figure 3 plots these three densities for comparison.

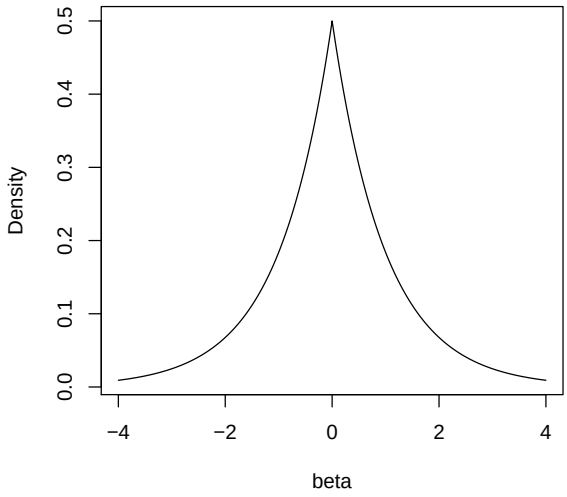
3.1.2 Data generating process

The predictor matrix $\mathbf{X}$ is generated in two different ways: independently from a standard Gaussian distribution, or according to a Gaussian factor model so that variables have strong linear codependencies. Details of the data generating process are shown below.

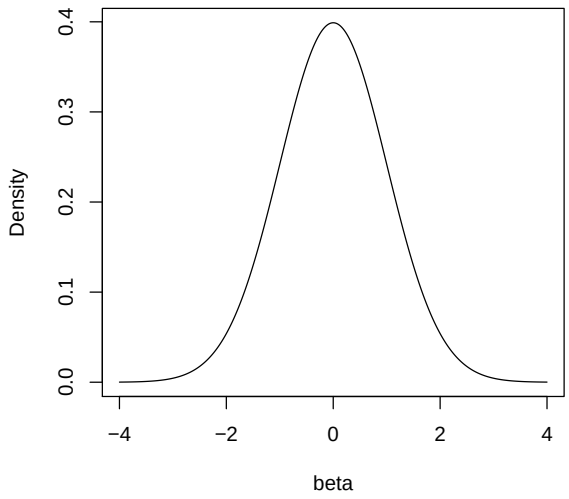
1. Generate the regression coefficients β
   - Horseshoe prior : fix $\lambda = 1$ draw $\lambda_1 \ldots, \lambda_p \sim C^+(0, 1)$, $\beta_j \sim \mathrm{N}(0, \lambda^2\lambda_j^2)$.
   - Laplace prior : fix $\lambda = 1$, draw $\beta_j \sim \mathrm{Laplace}(0, \lambda)$.
   - Ridge prior : fix $\lambda = 1$, draw $\beta_j \sim \mathrm{N}(0, 1)$.
2. Generate regressors $\mathbf{X}$
   - Independent regressors. Draw data matrix $\mathbf{X}_{n\times p}$ from standard Gaussian distribution.
   - Regressors with factor structure. In our simulation study we assume there are $k = p/5$ factors, therefore every 5 regressors have same loading on one factor. We draw factors $\mathbf{F}_{k\times n} \sim \mathrm{N}(0, 1)$. $\mathbf{B}_{p\times k}$ is the loading matrix where every 5 regressors have loading 1 on one factor. $\mathbf{X} = (\mathbf{BF})^T + \Omega$ where all entries of Ω are i.i.d. $N(0, 0.01)$.
3. Set $\sigma = \kappa\sqrt{\sum_{j=1}^{p} \beta_j^2}$, where κ controls noise level.
4. Draw $y_i = \mathrm{x}_i\beta + \epsilon_i, \epsilon_i \sim \mathrm{N}(0, \sigma^2)$ for $i = 1, \ldots, n$.



(a) Approximations to density of horseshoe prior, $\lambda = 1$. The solid line is the lower bound and the dashed line is the upper bound.



(b) Density of Laplace prior, $\lambda = 1$.



(c) Density of ridge prior, $\lambda = 1$.

Figure 3: Plots of several prior densities.

Additionally, we vary the noise level, letting $\kappa = 1$ or $\kappa = 2$, corresponding to signal-to-noise ratios of 1 and 1/2, respectively. We consider size 1 blocks in the independent

regressors case, and size 1 and size 5 blocks in the factor structured case. Since in the factor structured case every 5 regressors are highly correlated, the size 5 block slice-within-Gibbs sampler puts all highly correlated regressors into blocks together and samples these coefficients jointly.

3.1.3 Simulation results

To gauge the performance of our new algorithm, we must judge not only the speed, but also the quality of the posterior samples. To address this concern, we compare our approach with alternative samplers using effective sample size per second (see e.g. Gamerman and Lopes [2006] pages 126 - 127). Letting N denote the Monte Carlo sample size, the effective sample size $N_{\text{eff}}(\beta_j)$ is

$$N_{\text{eff}}(\beta_j) = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_k}, \tag{15}$$

where $\rho_k = \text{corr}\left(\beta_j^{(0)}, \beta_j^{(k)}\right)$ is the autocovariance of lag k. To verify that the samplers are giving comparable results (they ought to be fitting the same model) we also report the estimation error of the associated posterior point estimates[1] . Suppose $\{\bar{\beta}_j\}$ are posterior means of each variable and $\{\beta_j\}$ are true values, the estimation error is measured by

$$\text{error} = \sqrt{\frac{\sum_{j=1}^{p}(\bar{\beta}_j - \beta_j)^2}{\sum_{j=1}^{p}\beta_j^2}}. \tag{16}$$

Although we do not report it here, we also examined posterior standard deviations and found all algorithms to be comparable up to Monte Carlo error. For each simulation, 50,000 posterior samples are drawn, 20,000 of which are burn-in samples (with no thinning). We divide N_{eff} by running time in seconds to compute ESS per second as a measure of efficiency of each sampler.

Tables 1 and 2 report a representative subset of our simulation results; comprehensive tables can be found in the Appendix. Here we summarize the broad trends that emerge.

[1]Our results reveal that the error of the new sampler is slightly larger than that of the Gibbs sampler when the number of observations n is small; this is likely due to our approximate version of the horseshoe prior which matches the data generating process less well than the exact representation used in the standard Gibbs sampler implementation. For larger values of n, this small discrepancy in the prior becomes increasingly negligible. Observe also that there is no such discrepancy in Laplace and ridge regression because the prior density and the data generating process are exactly the same in those cases.

Prior	p	n	Error			ESS per second		
			OLS	slice	`monomvn`	slice	`monomvn`	ratio
Horseshoe	100	10p	3.40%	2.01%	1.90%	1677	756	2.22
	1000	10p	1.04%	0.35%	0.31%	29	1	29.00
Laplace	100	10p	3.38%	3.11%	3.11%	1243	478	2.60
	1000	10p	1.07%	0.97%	0.97%	70	5	14.00
Ridge	100	10p	3.33%	3.11%	3.10%	865	263	3.29
	1000	10p	1.05%	1.00%	1.00%	127	7	18.14

Table 1: Simulation results of all three priors, $\kappa = 1$, independent regressors.

Prior	p	n	Error				ESS per second		
			OLS	1-block	5-block	`monomvn`	1-block	5-block	`monomvn`
Horseshoe	100	10p	29.50%	6.93%	6.79%	6.11%	106	107	618
	1000	10p	9.41%	1.06%	1.06%	0.95%	7	7	3
Laplace	100	10p	29.08%	8.52%	8.52%	8.52%	118	552	360
	1000	10p	9.39%	2.68%	2.67%	2.67%	7	47	4
Ridge	100	10p	29.70%	8.67%	8.69%	8.69%	305	1588	1416
	1000	10p	9.42%	2.71%	2.71%	2.71%	8	42	6

Table 2: Simulation results of all three priors, $\kappa = 1$. There is underlying factor structure in the predictor matrix, with every 5 regressors being highly correlated with one another. The designation "1-block" refers to the slice-within-Gibbs sampler grouping every coefficient individually to its own block. The designation "5-block" refers to the slice-within-Gibbs sampler grouping every 5 highly correlated coefficients together in a single block.

First, the slice sampler enjoys a substantial advantage in terms of effective sample size (ESS) per second compared to the standard samplers. For example, in the independent regressor case, when $p = 1,000$ and $n = 10 \times p$, our approach is about 29 times faster than the `monomvn` Gibbs sampler.

Second, the ESS per second decreases for all samplers when the data is generated with strong colinearities in the regressor matrix. Which approach achieves higher ESS depends on the specific data generating process. Roughly speaking, the more difficult the problem — higher dimension, higher noise variance — the more the new sampler outperforms the `monomvn` sampler. Additionally, specifying the "correct" blocks appears to improve performance. The size-5-block slice-within-Gibbs sampler has much higher ESS per second for Laplace and ridge regression. This observation squares with the intuition that by grouping colinear predictor variables, one can better explore the contours of the posterior distribution. Surprisingly, under the horseshoe prior, the benefit of this grouping is negligible.

In addition to effective sample size per second, we also consider raw computing time. The code is tested on a machine with an Intel i7-6920HQ CPU and 16GB RAM. For the horseshoe regression with independent regressors, with $p = 500$ and $n = 5,000$, the slice-within-Gibbs sampler takes 101 seconds running time to draw 50,000 posterior samples, of which 19 seconds are fixed computing time and 82 seconds are spent within the loop. By comparison, the standard Gibbs sampler takes 1 second of fixed computing time and 5,310 seconds within the loop to draw the same number of posterior samples.

3.2 Empirical demonstration: beauty and course evaluations

In this section we consider an interesting data set first presented in Hamermesh and Parker [2005]. The data are course evaluations from the University of Texas at Austin between 2000 and 2002. The data are on a 1 to 5 scale, with larger numbers being better. In addition to the course evaluations, information concern the class and the instructor were also collected. To quote Hamermesh and Parker [2005]:

> We chose professors at all levels of the academic hierarchy, obtaining professorial staffs from a number of departments that had posted all faculty members'

pictures on their departmental websites. An additional ten faculty members' pictures were obtained from miscellaneous departments around the University. The average evaluation score for each undergraduate course that the faculty member taught during the academic years 2000-2002 is included. This sample selection criterion resulted in 463 courses, with the number of courses taught by the sample members ranging from 1 to 13. The classes ranged in size from 8 to 581 students, while the number of students completing the instructional ratings ranged from 5 to 380. Underlying the 463 sample observations are 16,957 completed evaluations from 25,547 registered students.

For additional details on how the beauty scores were constructed and on how to interpret the regression results, please see **?**. Here we do not aim to provide any sort of definitive reanalysis of the results in Hamermesh and Parker [2005]. Instead, our goal is to fit a plausible, but over-parametrized, model to their data and to employ a variety of priors, including some non-standard priors in addition to the usual shrinkage priors (horseshoe, Laplace and ridge). We are interested if substantive conclusions may change under "exotic" priors that are not likely to be used by the typical social scientist.

The model we fit allows for fixed effects for each of 95 instructors[2]. We include additive effects for the following factors: class size (number of students), language in which the professor earned his or her degree, whether or not the instructor was a minority, gender, beauty rating, and age. Each of these variables was included in the model via dummy variables according to the following breakdown:

- class size: below 31, 31 to 60, 61 to 150, or 151 to 600 (four levels by quartile),
- language: English or non-English (two levels),
- minority: ethnic minority or non-minority (two levels),
- gender: male or female (two levels),
- beauty: four levels by quartile,

[2]We recover the instructors by matching on teacher characteristics, including a variable denoting if the professor's photo is in black and white or in color. Using this method we find 95 uniques, although the original paper says there are 94 instructors. The data we used can be found at `http://faculty.chicagobooth.edu/richard.hahn/teaching/hamermesh.txt`

- age: below 43, 43 to 47, 48 to 56, 57 to 73 (four levels by quartile).

Finally, we include up to three-way interactions between age, beauty, and gender. We include an intercept in our model so that individual effects can be interpreted as deviation from the average. Clearly, this model is over-parameterized. Indeed, while $\mathbf{X}$ is 463-by-242, the rank of $\mathbf{X}^T\mathbf{X}$ is only $155 < 242$.

In addition to the horseshoe, Laplace and ridge regression priors, we also analyze these data using two exotic priors: an asymmetric Cauchy prior and a "non-local" two-component mixture of Cauchys.

The asymmetric Cauchy prior is

$$\pi(\beta) = \begin{cases} 2qf(\beta) & \beta \leq 0 \\ 2f(\beta/s)(1-q)/s & \beta > 0 \end{cases}, \tag{17}$$

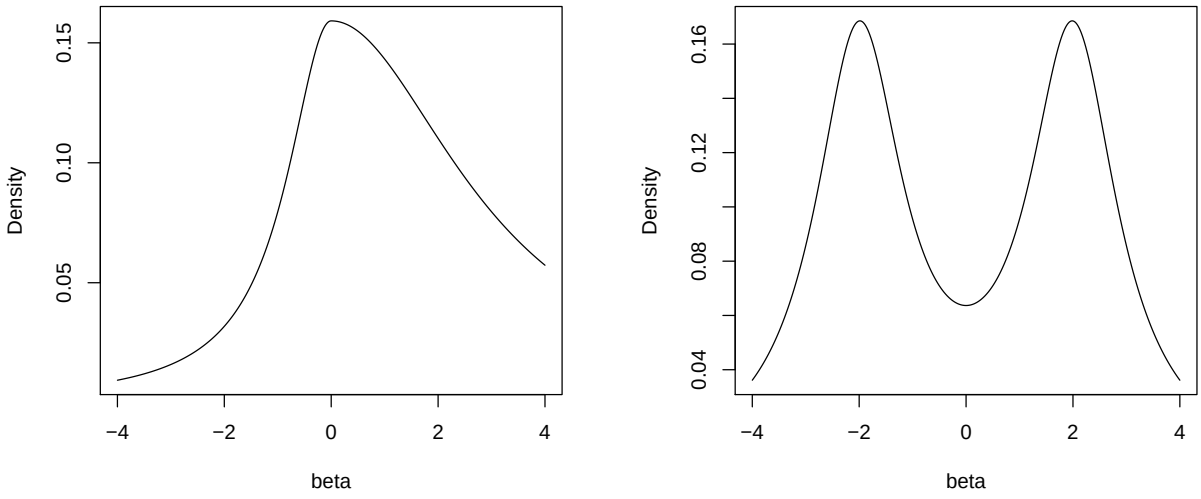
where $f(x) = \frac{1}{\pi(1+x^2)}$ is density of standard Cauchy distribution and $s = (1-q)/q$. Here, q is the prior probability that the coefficient is negative. We refer to this prior as having a shark fin density, as suggested by the shape shown in Figure 4. The bivariate mixture of Cauchys prior is

$$\pi(\beta) = 0.5t(\beta; -1.5, 1) + 0.5t(\beta; 1.5, 1) \tag{18}$$

where $t(x; m, v)$ is density of Student-t distribution with location m and degrees of freedom v. The non-local mixture of Cauchys is a sort of "anti-sparsity" prior: it asserts that the coefficients are all likely to be similar in magnitude and non-zero, but with unknown sign. A global scale parameters can be accommodated within the above forms by using density $\pi(\beta/\lambda)/\lambda$. Figure 4 shows density functions of these two priors.

When applying the exotic priors to the course evaluations model, we differentiated between regressors in terms of hyperparameter selection. Specifically, in the asymmetric Cauchy model we defaulted to $q = 0.5$, except for the following: the largest class size we set $q = 0.75$, for tenure track status we gave $q = 0.25$, for non-english we set $q = 0.75$, and for each of the three higher beauty levels we set $q = 0.25$. Similarly, for the non-local Cauchy mixture we defaulted to a standard Cauchy, using the nonlocal prior only for the class size, tenure track, language, and minority variables.

Our results are summarized in Table 3, which reports any variable whose posterior 95% credible interval excluded zero for at least one of the five priors. A number of features stand



(a) Density of sharkfin density.

(b) Density of mixture of Cauchy, location parameter -2 and 2.

Figure 4: The left panel is density of "shark fin" prior with $q = 0.25$. The right panel is density of mixture of two components Cauchy distribution where the weight for each component is 0.5, scale parameter 1 and location parameters are -2, 2 respectively.

out. First, there is a relatively small set of factors that are isolated between the various models as statistically significant (in the Bayesian sense described above); this suggests that the data is meaningfully overwhelming the contributions of the priors. Likewise, we note that the signs on the point estimates concur across all five priors. Second, we note that different priors do make a difference, both in terms of which variables among this set are designated significant and also in terms of the magnitude of the point estimates obtained. Third, one specific difference in the results that is noteworthy is that the horseshoe prior does not flag beauty as significant, while the sharkfin prior (which asserted a prior belief that beauty was advantageous for evaluations) gives a much larger estimate.

All posterior inferences were based on effective sample sizes of approximately 2,000.

Table 3: Posterior points estimates of regression coefficients whose posterior 95% credible intervals do not include zero are shown in bold. Estimates in non-bold have 95% credible intervals that do contain zero.

variable name	horseshoe	lasso	ridge	sharkfin	non-local
class size 31 to 60	-0.1	-0.07	-0.08	**-0.15**	-0.09
class size 61 to 150	**-0.26**	**-0.18**	**-0.18**	**-0.3**	**-0.21**
class size 151 to 600	**-0.56**	**-0.40**	**-0.40**	**-0.61**	**-0.45**
tenure track	**0.68**	**0.33**	0.26	**0.98**	**0.42**
non-minority	**1.83**	**0.68**	**0.46**	**1.35**	**0.76**
highly beautiful	0.67	**0.36**	**0.48**	**1.32**	**0.37**

4 Discussion

This paper presents a new efficient sampler for the purpose of Bayesian linear regression with arbitrary priors. The new method is seen to be competitive with, or better than, the usual Gibbs samplers that are routinely used to fit such models using popular shrinkage priors. The new approach is flexible enough to handle any class of priors admitting density evaluations. We hope that our new sampling approach will foster research into interesting

classes of priors that do not have obvious latent variable representations and to encourage empirical researchers to conduct more bespoke sensitivity analysis.

References

A. Armagan. Variational Bridge Regression. In *AISTATS*, pages 17–24, 2009.

A. Armagan, M. Clyde, and D. B. Dunson. Generalized beta mixtures of Gaussians. In *Advances in neural information processing systems*, pages 523–531, 2011.

A. Armagan, D. B. Dunson, and J. Lee. Generalized double Pareto shrinkage. *Statistica Sinica*, 23(1):119, 2013.

A. Bhattacharya, D. Pati, N. S. Pillai, and D. B. Dunson. Dirichlet–Laplace Priors for Optimal Shrinkage. *Journal of the American Statistical Association*, 110(512):1479–1490, 2015.

F. Caron and A. Doucet. Sparse Bayesian nonparametric regression. In *Proceedings of the 25th international conference on Machine learning*, pages 88–95. ACM, 2008.

C. M. Carvalho, N. G. Polson, and J. G. Scott. The Horseshoe Estimator for Sparse Signals. *Biometrika*, page asq017, 2010.

D. Gamerman and H. F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.

R. B. Gramacy. *monomvn: Estimation for Multivariate Normal and Student-t Data with Monotone Missingness*, 2017. URL `https://CRAN.R-project.org/package=monomvn`. R package version 1.9-7.

J. E. Griffin and P. J. Brown. Bayesian hyper-lassos with non-convex penalization. *Australian & New Zealand Journal of Statistics*, 53(4):423–442, 2011.

J. E. Griffin, P. J. Brown, et al. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1):171–188, 2010.

D. S. Hamermesh and A. Parker. Beauty in the classroom: Instructors pulchritude and putative pedagogical productivity. *Economics of Education Review*, 24(4):369–376, 2005.

C. Hans. Bayesian lasso regression. *Biometrika*, pages 835–845, 2009.

I. Murray, R. P. Adams, and D. J. MacKay. Elliptical Slice Sampling. In *JMLR Workshop and Conference Proceedings*, volume 9, pages 541–548. JMLR, 2010.

R. M. Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

S. E. Neville, J. T. Ormerod, M. Wand, et al. Mean field variational Bayes for continuous sparse signal shrinkage: pitfalls and remedies. *Electronic Journal of Statistics*, 8(1): 1113–1151, 2014.

T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

N. G. Polson and J. G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

N. G. Polson, J. G. Scott, and J. Windle. The Bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):713–733, 2014.

A Complete simulation results

A.1 Horseshoe prior

A.1.1 Factor structure

p	n	Error				ESS per second		
		OLS	1-block	5-block	`monomvn`	1-block	5-block	`monomvn`
100	$10p$	29.498%	6.934%	6.792%	6.107%	106	107	618
100	$50p$	12.931%	4.388%	4.320%	3.610%	52	53	132
100	$100p$	9.033%	3.628%	3.609%	3.280%	40	41	67
500	$10p$	13.304%	1.616%	1.620%	1.440%	23	24	16
500	$50p$	5.689%	1.086%	1.088%	1.008%	13	14	9
500	$100p$	4.037%	1.067%	1.070%	0.955%	13	13	7
1000	$10p$	9.411%	1.057%	1.055%	0.953%	7	7	3
1000	$50p$	4.052%	0.836%	0.833%	0.762%	5	5	2
1000	$100p$	2.850%	0.618%	0.615%	0.552%	5	5	1

Table 4: Error and running time comparison, horseshoe prior, $\kappa = 1$, factor structure.

p	n	Error				ESS per second		
		OLS	1-block	5-block	`monomvn`	1-block	5-block	`monomvn`
100	$10p$	60.518%	8.273%	8.102%	8.117%	193	192	465
100	$50p$	26.070%	5.497%	5.636%	5.545%	100	102	125
100	$100p$	17.709%	3.072%	3.074%	3.090%	141	151	100
500	$10p$	26.634%	2.289%	2.284%	2.276%	43	44	14
500	$50p$	11.436%	1.292%	1.260%	1.267%	41	41	10
500	$100p$	8.051%	1.305%	1.309%	1.298%	28	29	6
1000	$10p$	18.853%	1.277%	1.358%	1.290%	22	22	3
1000	$50p$	8.110%	1.063%	1.061%	1.059%	15	15	3
1000	$100p$	5.712%	0.697%	0.698%	0.691%	15	15	1

Table 5: Error and running time comparison, horseshoe prior, $\kappa = 2$, factor structure.

A.1.2 Independent regressors

p	n	Error			ESS per second		
		OLS	slice	`monomvn`	slice	`monomvn`	ratio
100	$10p$	3.40%	2.01%	1.90%	1677	756	2.22
100	$50p$	1.47%	1.19%	1.18%	1333	266	5.01
100	$100p$	1.01%	0.83%	0.82%	1433	137	10.46
500	$10p$	1.46%	0.77%	0.68%	196	17	11.53
500	$50p$	0.64%	0.42%	0.41%	177	12	14.75
500	$100p$	0.48%	0.30%	0.29%	201	10	20.10
1000	$10p$	1.04%	0.35%	0.31%	29	1	29.00
1000	$50p$	0.45%	0.23%	0.20%	50	2	25.00
1000	$100p$	0.32%	0.21%	0.20%	46	2	23.00

Table 6: Error and running time comparison, horseshoe prior, $\kappa = 1$, independent.

p	n	Error			ESS per second		
		OLS	slice	`monomvn`	slice	`monomvn`	ratio
100	$10p$	6.89%	3.20%	3.06%	1370	761	1.80
100	$50p$	2.82%	1.62%	1.60%	1073	283	3.78
100	$100p$	2.01%	1.38%	1.37%	1302	160	8.09
500	$10p$	3.01%	1.04%	0.92%	151	17	8.89
500	$50p$	1.29%	0.58%	0.56%	146	12	12.17
500	$100p$	0.90%	0.39%	0.38%	140	9	15.56
1000	$10p$	2.10%	0.70%	0.62%	25	1	25.00
1000	$50p$	0.90%	0.42%	0.40%	49	2	24.50
1000	$100p$	0.63%	0.28%	0.27%	48	2	24.00

Table 7: Error and running time comparison, horseshoe prior, $\kappa = 2$, independent

A.2 Laplace prior

A.2.1 Factor structure

p	n	Error					ESS per second			
		OLS	1-block	5-block	`monomvn`	Hans	1-block	5-block	`monomvn`	Hans
100	$10p$	29.080%	8.515%	8.516%	8.517%	8.711%	118	552	360	35
100	$50p$	12.968%	7.206%	7.204%	7.206%	7.334%	102	1952	329	23
100	$100p$	9.041%	6.228%	6.222%	6.236%	6.274%	83	2490	189	19
500	$10p$	13.419%	3.832%	3.827%	3.827%	3.853%	17	64	16	7
500	$50p$	5.753%	3.194%	3.190%	3.190%	3.210%	7	104	11	2
500	$100p$	3.746%	2.637%	2.637%	2.646%	2.627%	6	178	14	2
1000	$10p$	9.390%	2.675%	2.674%	2.674%	2.679%	7	47	4	4
1000	$50p$	6.357%	2.579%	2.579%	2.578%	2.583%	6	105	9	3
1000	$100p$	4.042%	2.428%	2.419%	2.417%	2.433%	8	185	13	3

Table 8: Error and running time comparison, Laplace prior, $\kappa = 1$, factor structure. "Hans" refers to an alternative Gibbs sampler from Hans [2009].

p	n	Error					ESS per second			
		OLS	1-block	5-block	`monomvn`	Hans	1-block	5-block	`monomvn`	Hans
100	$10p$	59.978%	8.952%	8.960%	8.963%	9.549%	547	958	827	103
100	$50p$	24.961%	8.366%	8.380%	8.360%	9.163%	224	1509	314	38
100	$100p$	17.872%	7.912%	7.907%	7.907%	8.409%	71	681	97	17
500	$10p$	26.550%	3.983%	3.983%	3.983%	4.012%	39	60	16	17
500	$50p$	11.440%	3.788%	3.783%	3.780%	3.863%	16	72	12	5
500	$100p$	8.269%	3.514%	3.511%	3.507%	3.635%	13	124	14	3
1000	$10p$	28.838%	3.808%	3.807%	3.807%	3.820%	19	31	5	6
1000	$50p$	8.314%	3.604%	3.604%	3.604%	3.675%	10	45	7	4
1000	$100p$	8.188%	3.547%	3.540%	3.545%	3.746%	12	126	14	3

Table 9: Error and running time comparison, Laplace prior, $\kappa = 2$, factor structure. "Hans" refers to an alternative Gibbs sampler from Hans [2009].

A.2.2 Independent regressors

p	n	Error				ESS per second		
		OLS	slice	`monomvn`	Hans	slice	`monomvn`	Hans
100	$10p$	3.381%	3.112%	3.112%	3.120%	1243	478	663
100	$50p$	1.396%	1.374%	1.375%	1.375%	1938	220	788
100	$100p$	1.023%	1.014%	1.015%	1.015%	2165	135	802
500	$10p$	1.487%	1.378%	1.378%	1.379%	343	25	150
500	$50p$	0.739%	0.628%	0.628%	0.628%	416	20	158
500	$100p$	0.751%	0.633%	0.633%	0.633%	434	20	138
1000	$10p$	1.065%	0.972%	0.972%	0.972%	70	5	65
1000	$50p$	1.061%	0.980%	0.980%	0.980%	95	4	70
1000	$100p$	1.059%	0.986%	0.986%	0.986%	75	4	65

Table 10: Error and running time comparison, Laplace prior, $\kappa = 1$, independent regressors.

p	n	Error				ESS per second		
		OLS	slice	`monomvn`	Hans	slice	`monomvn`	Hans
100	$10p$	6.617%	5.216%	5.224%	5.314%	1389	531	827
100	$50p$	2.885%	2.722%	2.721%	2.730%	1988	270	1004
100	$100p$	1.990%	1.937%	1.937%	1.938%	4369	378	1754
500	$10p$	2.995%	2.398%	2.398%	2.412%	281	22	150
500	$50p$	1.266%	1.194%	1.194%	1.196%	220	13	112
500	$100p$	0.928%	0.903%	0.903%	0.903%	348	17	134
1000	$10p$	2.112%	1.684%	1.684%	1.688%	65	4	67
1000	$50p$	2.030%	1.683%	1.683%	1.681%	349	17	134
1000	$100p$	2.090%	1.691%	1.691%	1.695%	75	4	72

Table 11: Error and running time comparison, Laplace prior, $\kappa = 2$, independent regressors.

A.3 Ridge regression

A.3.1 Factor structure

p	n	Error				ESS per second		
		OLS	1-block	5-block	`monomvn`	1-block	5-block	`monomvn`
100	$10p$	29.695%	8.669%	8.697%	8.688%	305	1588	1416
100	$50p$	13.058%	7.516%	7.516%	7.508%	108	2366	862
100	$100p$	9.273%	6.596%	6.595%	6.595%	80	2787	498
500	$10p$	13.465%	3.845%	3.843%	3.844%	45	206	42
500	$50p$	5.795%	3.308%	3.307%	3.307%	14	246	31
500	$100p$	4.044%	2.854%	2.854%	2.853%	8	254	22
1000	$10p$	9.416%	2.712%	2.710%	2.711%	8	42	6
1000	$50p$	4.083%	2.339%	2.331%	2.334%	2	42	5
1000	$100p$	2.838%	1.936%	1.936%	1.936%	2	42	4

Table 12: Error and running time comparison, ridge prior, $\kappa = 1$, factor structure.

p	n	Error				ESS per second		
		OLS	1-block	5-block	`monomvn`	1-block	5-block	`monomvn`
100	$10p$	59.265%	8.890%	8.882%	8.887%	653	1663	1228
100	$50p$	24.914%	8.545%	8.547%	8.540%	284	1573	859
100	$100p$	17.648%	8.158%	8.155%	8.154%	185	1852	468
500	$10p$	26.125%	3.997%	3.997%	3.997%	133	200	43
500	$50p$	11.413%	3.779%	3.779%	3.779%	36	182	30
500	$100p$	11.505%	3.799%	3.800%	3.801%	40	197	30
1000	$10p$	18.886%	2.768%	2.769%	2.769%	22	39	6
1000	$50p$	8.139%	2.684%	2.684%	2.683%	7	31	5
1000	$100p$	6.256%	2.237%	2.237%	2.237%	6	34	5

Table 13: Error and running time comparison, ridge prior, $\kappa = 2$, factor structure.

A.3.2 Independent regressors

p	n	Error			ESS per second		
		OLS	slice	`monomvn`	slice	`monomvn`	ratio
100	$10p$	3.328%	3.106%	3.102%	865	263	3.29
100	$50p$	1.447%	1.434%	1.435%	1199	112	10.71
100	$100p$	1.039%	1.036%	1.036%	1301	69	18.87
500	$10p$	1.486%	1.419%	1.415%	113	10	11.30
500	$50p$	0.643%	0.638%	0.638%	163	9	18.11
500	$100p$	0.448%	0.446%	0.446%	605	22	27.5
1000	$10p$	1.054%	1.004%	1.001%	127	7	18.14
1000	$50p$	0.450%	0.446%	0.446%	142	6	23.67
1000	$100p$	0.319%	0.317%	0.317%	117	5	23.40

Table 14: Error and running time comparison, ridge prior, $\kappa = 1$, independent regressors.

p	n	Error			ESS per second		
		OLS	slice	`monomvn`	slice	`monomvn`	ratio
100	$10p$	6.705%	5.624%	5.499%	1433	442	3.23
100	$50p$	2.882%	2.760%	2.761%	1092	111	9.84
100	$100p$	2.011%	1.984%	1.986%	1993	107	18.63
500	$10p$	3.020%	2.633%	2.465%	111	10	11.10
500	$50p$	1.271%	1.233%	1.230%	148	9	16.44
500	$100p$	0.899%	0.884%	0.881%	594	22	27
1000	$10p$	2.096%	1.801%	1.792%	142	8	17.75
1000	$50p$	0.906%	0.868%	0.870%	140	6	23.33
1000	$100p$	0.631%	0.619%	0.622%	120	5	24.00

Table 15: Error and running time comparison, ridge prior, $\kappa = 2$, independent regressors.