Scalable MCMC for Bayes Shrinkage Priors

James E. Johndrow  
Stanford University, Stanford, CA USA  
johndrow@stanford.edu

Paulo Orenstein  
Stanford University, Stanford, CA USA

and

Anirban Bhattacharya  
Texas A&M University, College Station, TX, USA

April 3, 2018

Abstract

Gaussian scale mixture priors are frequently employed in Bayesian analysis of high-dimensional models, and several members of this family have optimal risk properties when the truth is sparse. While optimization-based algorithms for the extremely popular Lasso and elastic net procedures can scale to dimension in the hundreds of thousands, corresponding Bayesian methods that use Markov chain Monte Carlo (MCMC) for computation are limited to problems at least an order of magnitude smaller. This is due to high computational cost per step of the associated Markov kernel and growth of the variance of time-averaging estimators as a function of dimension. We propose an MCMC algorithm for computation in these models that combines block updating and approximations of the Markov kernel to directly combat both of these factors. Our algorithm gives orders of magnitude speedup over the best existing alternatives in high-dimensional applications. We give theoretical guarantees for the accuracy of the kernel approximation. The scalability of the algorithm is illustrated in simulations with problem size as large as $N = 5,000$ observations and $p = 50,000$ predictors, and an application to a genome wide association study with $N = 2,267$ and $p = 98,385$. The empirical results also show that the new algorithm yields estimates with lower mean squared error, intervals with better coverage, and elucidates features of the posterior that were often missed by previous algorithms in high dimensions, including bimodality of posterior marginals indicating uncertainty about which covariates belong in the model. This latter feature is an important motivation for a Bayesian approach to testing and selection in high dimensions.

Keywords: Bayesian testing; computational complexity; global-local; horseshoe; Markov chain Monte Carlo; scalability.

1 Introduction

Modern applications in genetics and other areas of biology have stimulated considerable interest in statistical inference in the high-dimensional setting where the number of predictors $p$ is much larger than the number of observations $N$, and the truth is thought to be sparse. Regression models are frequently employed in this context. Consider a Gaussian linear model with likelihood

$$L(z | W\beta, \sigma^2) = (2\pi\sigma^2)^{-N/2}e^{-\frac{1}{2\sigma^2}(z-W\beta)'(z-W\beta)},$$

where $W$ is a $N \times p$ matrix of covariates, $\beta \in \mathbb{R}^p$ is assumed to be a sparse vector, and $z \in \mathbb{R}^N$ is an $N$-vector of response observations. A common hierarchical Bayesian approach employs a Gaussian scale-mixture prior on $\beta$ of the form

$$\beta_j \mid \sigma^2, \eta, \xi \overset{iid}{\sim} N(0, \sigma^2\xi^{-1}\eta_j^{-1}), \quad \xi \sim \pi_{\xi}(\cdot), \quad \eta \sim \pi_{\eta}(\cdot), \quad \sigma^2 \sim \pi_{\sigma^2}(\cdot), \quad j = 1, \ldots, p,$$
where \( \eta = (\eta_1, \ldots, \eta_p)' \). For appropriate choice of \( \pi_{\eta} \), this prior structure induces approximate sparsity in \( \beta \) by shrinking most components aggressively toward zero while retaining the true signals \( 25 \). In this sense, the prior \( 2 \) approximates the properties of point-mass mixture priors \( 18, 33, 11 \), which allow some components of \( \beta \) to be exactly zero a posteriori. From the point of view of testing the hypotheses \( H_{0j} : \beta_j = 0 \), the global precision parameter \( \xi \) controls how many of the hypotheses are true, while the local precisions \( \eta_j \) control which of the hypotheses are true, resulting in the designation “global-local” for this prior structure \( 25 \).

One motivation for global-local priors over point-mass mixtures is ostensibly lower computational complexity. However, the promise of scaling to large datasets has thus far not been realized, owing to the expensive linear algebra involved in MCMC algorithms for these models and slow mixing of the corresponding Markov chains. This is unfortunate, as it has kept a useful model that can provide rigorous quantification of uncertainty in selected variables out of many modern applications such as genome-wide association studies (GWAS), which often have \( N \) in the thousands and \( p \) in the hundreds of thousands or more. Here, our objective is to study the computational cost of MCMC algorithms for these models as a function of \( N \) and \( p \), and propose MCMC algorithms with lower computational cost than existing options. We focus on the motivating setting for global-local shrinkage priors where \( p \gg N \). Due to the difficulty of performing convergence rate analysis for hybrid Markov chains of the sort used in MCMC for Bayes shrinkage priors, and the looseness of most bounds found in the literature, we give a mix of empirical and theoretical results. We further show in real data applications that the proposed algorithms scale to problem sizes orders of magnitude larger than existing algorithms.

Numerous choices of \( \pi_{\eta} \) designed for high dimensions have been proposed. These priors are often independently specified on the local standard deviation \( \lambda_j = 1/\sqrt{\eta_j} \), and include the Bayes lasso \( \pi_1(x) \propto \prod_{j=1}^p x_j \exp(-x_j^2/2) \mid 21 \mid 34 \), the generalized double Pareto \( \pi_4(x) \propto \prod_{j=1}^p (2\xi)^{-1}(1 + |x_j|/(\alpha\xi))^{-1-(\alpha+1)} \), and the popular Horseshoe prior, which uses independent standard half-Cauchy priors \( \pi_2(x) \propto \prod_{j=1}^p (1 + x_j^2)^{-1} \). Several other alternatives that, like the Bayes lasso, use conditionally Gamma or exponential priors on the local variances are the R2-D2 prior of Zhang et al. \( 39 \), the Dirichlet-Laplace prior of Bhattacharya et al. \( 5 \), and the Normal-gamma prior of Griffin and Brown \( 12 \). Recent results show that, like point-mass mixtures, some priors of this class achieve the minimax-optimal rate for squared error loss in high-dimensional settings. This includes the Horseshoe \( 35 \), as well as the Dirichlet-Laplace prior \( 5 \). We consider algorithms for posterior computation using the Horseshoe prior, which has strong theoretical support, but the strategies we employ are broadly applicable.

Existing theoretical analysis of MCMC algorithms for global-local priors consists of showing a geometric ergodicity result with \( N, p \) fixed \( 20, 23, 28 \). In Khare and Hobert \( 20 \) and Pal and Khare \( 23 \), the authors consider the Gibbs sampling algorithms of Park and Casella \( 21 \) and Bhattacharya et al. \( 3 \) for the Bayes Lasso and Dirichlet-Laplace priors, respectively, and show they are geometrically ergodic. The authors in Rajaratnam et al. \( 28 \) consider a blocked Gibbs sampler for Bayes Lasso obtained by sampling \( \sigma^2 \) marginalizing \( \beta \); Polson et al. \( 20 \) Supplement §A.2 also use this update in the normal means setting. Also, in Rajaratnam et al. \( 25 \), the authors note that their bound on the geometric convergence rate tends to one at an exponential rate in \( N, p \), which would indicate it is an exponential time algorithm. This is inconsistent with empirical performance, and the authors suggest that the bound is probably quite loose, but do show a Hilbert-Schmidt condition for their kernel that does not hold when \( \beta \) is conditioned upon in updating \( \sigma^2 \). Rajaratnam and Sparks \( 27 \) show that published bounds on geometric convergence rates tending to one at an exponential rate in \( N, p \) is a general phenomenon, underscoring the inherent difficulty of obtaining sharp bounds on geometric convergence rates for many kernels used in practice.

Empirical analysis of algorithms for Bayes Lasso \( 14 \) and Horseshoe \( 26 \) Supplement] priors have been undertaken. A major factor contributing to slow convergence is high autocorrelation for the global scale parameter \( \tau = 1/\sqrt{\xi} \), and convergence rates deteriorate as \( p \) increases. Hans \( 14 \) proposes two alternatives to the sampler of Park and Casella \( 24 \) which improve the convergence rate, but come at an increased computational cost per step. Polson et al. \( 20 \) consider both slice sampling and parameter expansion for sampling \( \tau \), and find that parameter expansion offers only modest gains. In addition to the slow mixing, the cost-per-iteration presents a major bottleneck in sampling \( \beta \) when \( p \gg N \). This is substantially alleviated by a recent sampling algorithm for
multivariate Gaussians in Bhattacharya et al. [4], which, combined with the update rule of Polson et al. [20], is arguably the best option presently available for $p > N$ settings, and our main point of comparison. We mention here the recent article Hahn et al. [13] who use elliptical slice sampling rather than Gibbs and obtain a general algorithm for global-local priors. Their implementation has lower computational cost than Bhattacharya et al. [4] when $N > p$, but considerably higher cost when $p > N$. In their manuscript, a comparison to the exact version of our algorithm is undertaken that further supports this conclusion. They do not consider $p > 3,000$, whereas here we focus on much larger $p$—up to 98,385—always with $p > N$. Thus the algorithms are designed for, and perform well under, different use cases.

In this article, we provide a MCMC sampler for the Horseshoe prior with considerably lower computational cost in $p > N$ settings relative to the best available alternative. The basic ingredients of our algorithms are: extensive use of block updating in Gibbs sampling, which reduces autocorrelation; and use of highly accurate approximations to the exact kernel that massively reduce computational cost per step when the truth is sparse. We provide theoretical guarantees for the accuracy of the approximation by verifying the condition in Johndrow and Mattingly [15].

In general it is difficult to obtain useful bounds on the variance of pathwise time-averaging estimators from complicated hybrid Markov chains like the ones we study here that would enable theoretical analysis of how the variance scales in $N, p$. As such, our analysis of the contribution of the MCMC variance to the computational cost of our algorithms is mainly empirical. This analysis supports the conclusion that the variance is essentially independent of problem size. If this is indeed the case, it follows that our algorithm reduces the computational cost from $N^2 p$ to $s^2 N \lor N p$, where $s$ is the sparsity level.

The proposed algorithm also significantly improves numerical stability and inferential performance relative to the implementation of [4]. Global-local priors aggressively shrink most parameters to zero, which often results in values of the parameters that are near machine epsilon with high probability a posteriori, leading to challenging computation. The implementation of [4] relied on various truncation steps to handle numerical underflow. Our algorithms more accurately handle small values of the parameters to eliminate all truncations.

We show that algorithms that utilize truncation rules sometimes fail to converge, or converge to an apparent stationary distribution that differs from the true posterior, resulting in misleading inference. No evidence of this pathological behavior remains in our proposed algorithms, and thus they provide better statistical performance at lower computational cost. In particular, we observe bimodality of the Horseshoe posterior marginals for intermediate-sized signals in high-dimensional settings, a useful feature for quantifying uncertainty in model selection which has previously received little attention.

**Notation:** We introduce some basic notation used throughout the article. For a square matrix $A$, $\text{tr}(A)$ denotes its trace. We use $I_d$ to denote the $d \times d$ identity matrix. For an $m \times r$ matrix $A$ (with $m > r$), $s_i(A) := s_i = \sqrt{\lambda_i}$ for $i = 1, \ldots, r$ denote the singular values of $A$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \geq 0$ are the eigenvalues of $A^\top A$. The largest and smallest (non-zero) singular values are $s_{\max}(A) = s_1(A)$ and $s_{\min}(A) = s_r(A)$. Unless otherwise stated, $\|A\| := s_{\max}(A)$ denotes the operator norm of a matrix. We often make use of the facts $\|AB\| = \|BA\| \leq \|A\| \|B\|$ and $\|A + B\| \leq \|A\| + \|B\|$. For probability measures $P, Q$ on $(\mathcal{X}, \mathcal{B})$ having densities $p$ and $q$ with respect to some dominating measure $\mu$, recall the following equivalent definitions of the total variation distance

$$
\|P - Q\|_{TV} = \sup_{B \in \mathcal{B}} |P(B) - Q(B)| = \frac{1}{2} \int_{\mathcal{X}} |p - q| d\mu = \sup_{\phi \in \mathcal{C}} \int \phi(p - q) d\mu.
$$

The Kullback–Leibler (KL) divergence $\text{KL}(P \parallel Q) = \int p \log(p/q) d\mu$. From Pinsker’s inequality, $\text{KL}(P \parallel Q) \geq 2\|P - Q\|_{TV}^2$. The inverse-gamma distribution with parameters $a, b > 0$ has density proportional to $x^{-(a+1)} e^{-b/x} \mathbb{1}_{(0, \infty)}(x)$.
2 Proposed algorithms

2.1 Motivation and Summary of Findings

This article was initially motivated by the observation of [26, Supplement] that the global precision parameter $\xi$ tends to mix very slowly in MCMC algorithms for the Horseshoe, especially when $p$ is large. The approach in [26] to sample from the joint posterior of $(\beta, \sigma^2, \eta, \xi \mid z)$ is to use a Gibbs update rule of the form

$$
\begin{align*}
\sigma^2, \beta \mid \xi, \eta, z \\
\xi \mid \eta, \beta, \sigma^2 \\
\eta \mid \xi, \beta, \sigma^2.
\end{align*}
$$

The conditional posteriors of $\xi$ and $\eta$ in the second and third steps do not admit a standard form and [26] recommended slice sampling for these updates. This algorithm was found to exhibit very high autocorrelation and poor mixing for $\xi$ in the normal means setting. The authors also tried a parameter expanded Gibbs sampler, with limited success. The joint sampling of $\sigma^2, \beta \mid \xi, \eta, z$ in the first step can be carried out by first sampling $\sigma^2 \mid \xi, \eta, z$ from an inverse-gamma distribution, and then sampling $\beta \mid \sigma^2, \xi, \eta, z$ from a $p$-variate Gaussian distribution, which typically has $O(p^3)$ complexity. The overall per-iteration cost of the algorithm is dominated by the sampling step for $\beta$ when $p$ is large. Exploiting the structure of the conditional posterior of $\beta$, [4] proposed an exact algorithm to sample from such structured multivariate Gaussians with $O(N^2 p)$ complexity. While this lowers cost per step, the $O(N^2 p)$ complexity can still be prohibitive, especially when $N$ is large. This algorithm nonetheless represents the most scalable existing alternative when $p > N$, and we use the sampling strategy for $\beta$ in [4] with the update rule in (3) as our main point of comparison. Since this combination isn’t documented to best of our knowledge\(^1\), we generically refer to this as the “old algorithm.”

In this article, we simultaneously address both slow mixing and high computational cost per step to develop improved algorithms. Specifically, we provide an alternative exact algorithm with improved mixing, and further enhance it to derive an accurate approximation thereof with much lower computational cost. Before providing the details of the proposed algorithms, we discuss some of their salient features.

Our basic approach to improving mixing is to make more extensive use of block updating, resulting in the update rule

$$
\begin{align*}
\xi, \sigma^2, \beta \mid \eta, z \\
\eta \mid \xi, \sigma^2, \beta,
\end{align*}
$$

for the Horseshoe. Figure 1 shows autocorrelations at lags 1-100 for log($\xi$) in the old algorithm compared to the algorithm in (4), which we refer to as the “new” algorithm, and the approximate algorithm for a simulation with problem size $N = 2,000, p = 20,000$. Two features are evident. First, autocorrelations at all lags are much lower in the new algorithm than the old algorithm. Second, autocorrelation is even lower in the approximate algorithm than the exact algorithm. While the new algorithm without approximation has somewhat higher computational cost than the old algorithm, the approximate algorithm has massively lower cost when the truth is sparse. Moreover, a feature not exposed by looking at autocorrelations alone is that the old algorithm actually does not converge in most cases; $\xi$ just drifts toward infinity as the chain extends. This problem is not obvious from looking only at the marginal path for $\beta$, and we discuss it in more detail throughout the paper.

While small autocorrelation lowers computational cost by effectively allowing accurate approximations of posterior quantities for much shorter path lengths, the algorithm may still be expensive because the computational cost per MCMC scan is high. As discussed, one of the factors contributing to the per-iteration complexity is sampling $\beta$ from a $p$-variate Gaussian. We identify a lingering bottleneck, more pronounced when $N$ is large, in the state-of-the-art algorithm of [4] to be due to the formation of the matrix

$$
M_\xi = I_N + \xi^{-1}W D W', \quad D = \text{diag}(\eta_j^{-1}) \in \mathbb{R}^{p \times p},
$$

\(^1\)The implementation of [4] used $[\beta \mid \sigma^2, -]$ and $[\sigma^2 \mid -]$ rather than $[\beta \mid \sigma^2, -]$ and $[\sigma^2 \mid -]$ as in (3).
and, more specifically, the matrix multiplication $WDW'$ at each scan of the MCMC. Interestingly, the matrix $M_\xi$ also appears in two other places in our blocked sampler \cite{4} for the Horseshoe, specifically, the updates for $\xi$ and $\sigma^2$, which are not present in the implementation of \cite{4} without blocking. For large $N, p$, the following approximation is accurate and provides orders of magnitude speed-ups.

The optimal behavior of continuous global-local priors like the Horseshoe in sparse situations \cite{35, 36} is a consequence of the flexibility in the corresponding choices for $\pi_\xi$ and $\pi_\eta$, which allows aggressive shrinkage of $(\xi_\eta)^{-1}$ towards zero for the null entries of the true $\beta$, thereby allowing $\beta$ to optimally concentrate around sparse truths \textit{a posteriori}. An important practical consequence, hitherto unexplored, is that once the MCMC algorithm begins to converge, the matrix $M_\xi$ will typically be well-approximated by hard-thresholding $D$ in \cite{5}, resulting in

$$M_\xi \approx M_{\xi, \delta} := I_N + \xi^{-1}WD\delta W'^t, \quad D_\delta = \text{diag}(\eta_j^{-1}1(\xi_{\max}^{-1}\eta_j^{-1} > \delta))$$

for “small” $\delta$, where $\xi_{\max}$ is the maximum of the current and proposed states of $\xi$ in our Metropolis-Hastings step. This thresholding step reduces computational cost considerably, since the columns of $W$ corresponding to the zero diagonal entries of $D_\delta$ can just be ignored. Letting $S = \{j : \xi_{\max}^{-1}\eta_j^{-1} > \delta\}$, we can also write the approximation as

$$M_{\xi, \delta} = I_N + \xi^{-1}WS_\delta W'^t,$$

where $W_S$ consists of the columns of $W$ with indices in the set $S$, and $D_S$ consists of the rows and columns of $D$ with indices in the set $S$. This makes clear the computational advantages of thresholding.

Figure 2 compares effective samples per second ($ne\{t\}$), a measure of overall computational efficiency, for the old algorithm and for our exact sampler (new algorithm) and approximate algorithm. The $ne\{t\}$ is calculated for 100 elements of the vector $\beta$, 100 elements of the vector $\eta$, for log($\sigma^2$), and for log($\xi$). The included coordinates of $\beta, \eta$ correspond to true signals that vary in size from greater than the residual standard deviation ($\sigma$) to much less than the standard deviation, as well as some coordinates corresponding to true nulls ($\beta_j = 0$). These results are based on a smaller simulation with $N = 2,000$ and $p = 20,000$. The median value of $ne\{t\}$ for the new algorithm is about half that of the old algorithm. This is because although the new algorithm mixes better, it has a higher per-iteration computational cost due to the need to compute the eigenvalues of $I + WDW'$. However, this comparison is not really valid, since the old algorithm fails to reach stationarity, so the estimates of $ne\{t\}$ are unreliable. The median value of $ne\{t\}$ for the approximate algorithm is over fifty times that of the exact algorithm. We later give a result guaranteeing a bound on the approximation error from using $D_\delta$ in lieu of $D$, and show empirically that for the values of $\delta$ we use in practice, this approximation is extremely accurate. Thus the approximate algorithm represents about a fifty-fold speedup “for free” in this simulation. For larger $N, p$, the speedup can be much greater, but at this problem size the “old” and “new” algorithms already require several hours to run on our hardware, so larger simulations for comparison are not really feasible.
Computational cost is only part of the story. The statistical performance of the proposed algorithms is also superior to that of the old algorithm. Using the approximate posterior expectation based on MCMC time averages as a point estimate, we find that the estimates output by the new and approximate algorithm both have consistently lower mean squared error than the estimates output by the old algorithm. Several illustrations of why the new algorithm performs better are shown in Figures 3 and 4. These results are based on a simulation with $N = 200$ and $p = 2,000$. Figure 3 shows trace plots from a path of length 20,000 for the tenth entry of $\beta$, whose true value of $2^{-1/4} \approx 0.84$ is slightly less than half the true residual standard deviation of $\sigma = 2$. The Horseshoe posterior for such “intermediate” sized signals is apparently bimodal, with a mode at zero and a second one away from zero, a fact that has heretofore received little attention (see e.g. the brief comment at [7, pg. 114]). This is of inferential interest, as it aptly reflects the posterior uncertainty associated with such intermediate signals, and is also a nice illustration of how well the Horseshoe posterior approximates the posterior under sharp priors that place nonzero mass on the event $\beta_j = 0$ a priori. It is not surprising that this feature has not been a focus, since the old algorithm massively underestimates the relative sizes of the two modes and places most of its mass near the origin. The bimodality creates two deep potential wells in the target, and the old algorithm gets stuck in the larger mode during the second half of the run, as is apparent from the top left panel of Figure 3. This has inferential consequences, as any thresholding procedure based on the old algorithm will unequivocally declare $\beta_{10}$ as a noise coefficient. The new and approximate algorithms apparently are more successful at crossing the potential “hill” between these wells. In our experience, this is a general phenomenon with the old algorithm that is largely gone in the new and approximate algorithms.

We suspect that this improvement is mainly due to more accurate handling of very small values of the local scales in the new algorithm, a point that is discussed more below. The old algorithm relied on various checks to control over- or under-flow of the local and global scales and the residual precision. A previously unnoticed behaviour of the old algorithm is that in many cases, the Markov chain does not converge at all in the $\xi$ coordinate, and the residual variance sometimes converges to an apparent stationary distribution that puts almost no mass near its true value. This is illustrated by the left panels in Figure 4. This problematic behaviour disappears in the new and approximate algorithms. The global precision converges rapidly to an apparent stationary distribution. The residual precision does as well, although in this example it is centered at a smaller value than the truth. This is likely due to the use of the proper Gamma(1/2, 1/2) prior on the residual precision.

We have found the use of proper priors on the residual precision important to obtaining convergence and for numerical stability of the algorithm. The common practice of using the right Haar prior $\sigma^{-1}$ on the residual standard deviation is not recommended for Bayes shrinkage priors in high dimensions, and can cause the Markov chain to fail to converge.

It is now widely recognized that the computational cost of an MCMC algorithm is the product of two factors: the cost per step and the number of steps required to achieve the desired approximation.
error. Both factors tend to increase in $N, p$, necessitating more nuanced algorithms for “big data” problems. The proposed approximation of $M$ by $M_\delta$ aims to reduce the cost per step, while the more extensive use of block updating is intended to reduce the required path length. In addition to decreasing the computational cost for any $N, p$, we also find that the computational cost of our new algorithms grow more slowly in $N, p$ than existing algorithms, implying that they are more scalable to large datasets. We first present our new algorithms, and then proceed to provide a rigorous notion of computational cost in the next section, followed by theoretical and empirical results for our algorithms.

2.2 Exact Algorithm

We first describe our exact algorithm and then detail the changes necessary for the approximate algorithm in the next subsection. Throughout, $\pi_\xi(x) = C (1 + x)^{-1} x^{-1/2} \mathbb{1}_{(0,\infty)}(x)$ for some normalizing constant $C < \infty$, which is the induced prior on $\xi$ from a half-Cauchy prior on the global standard deviation $\tau = \xi^{-1/2}$. We use an inverse gamma prior with parameters $a_0/2, b_0/2$ for $\sigma^2$, rather than the right Haar prior $\pi_{\sigma^2}(x) \propto x^{-2}$, which we have found empirically to result in numerical instability of the resulting MCMC algorithm and periodic escapes to infinity. We put $a_0 = b_0 = 1$ in simulations and applications.
As outlined in [4], our blocked Gibbs sampler cycles through sampling from \( \xi, \sigma^2, \beta \mid \eta, z \) and \( \eta \mid \xi, \sigma^2, \beta \). An exact sample from the joint distribution of \( \xi, \sigma^2, \beta \mid \eta, z \) is obtained by sampling \( \xi \mid \eta, z, \sigma^2 \mid \xi, \eta, z, \text{and } \beta \mid \sigma^2, \xi, \eta, z \), in that order. The conditional posteriors \( \sigma^2 \mid \xi, \eta, z \) and \( \beta \mid \sigma^2, \xi, \eta, z \) are respectively distributed as inverse-gamma and multivariate normal. The marginal posterior of \( \xi \mid \eta, z \) is a non-standard density and we use a Metropolis–Hastings (MH) step for it, while a standard slice sampler is used to sample \( \eta \mid \xi, \sigma^2, \beta \). The steps of the exact algorithm are detailed below.

1. Propose \( \log(\xi^*) \sim N(\log(\xi), s) \), then compute the log acceptance ratio
   \[
   \log(q) = \log \left( \frac{L(y \mid \xi^*, \eta) \pi(\xi^*) \xi^*}{L(y \mid \xi, \eta) \pi(\xi^*) \xi} \right),
   \]
   where the log-likelihood is given by
   \[
   \log\{L(z \mid \xi, \eta)\} = -\frac{1}{2} \log |M_\xi| - \frac{(n + a_0)}{2} \log(b_0 + z^T M_\xi^{-1} z/2),
   \]
   with \( M_\xi \) denoting the matrix
   \[
   M_\xi = I_N + \xi^{-1} W D W^T, \quad D = \text{diag}(\eta_j^{-1}) \in \mathbb{R}^{p \times p}.
   \]
   Then accept \( \xi^* \) with probability \( \min\{q, 1\} \). We find that taking \( s = 0.8 \) results in good performance for most values of \( N, p \) we consider.

2. Sample \( \sigma^2 \mid \xi, \eta, z \) from
   \[
   \sigma^2 \mid \xi, \eta, z \sim \text{InvGamma} \left( \frac{n + a_0}{2}, \frac{z^T M_\xi^{-1} z + b_0}{2} \right)
   \]
   where \( M_\xi \) is defined as in [4], with the current value of \( \xi \). We sometimes replace this step with a Metropolis–Hastings step, again using a random walk on \( \log(\sigma^{-2}) \) as proposal, which appears to lead to better mixing in high dimensions. We recommend using 0.1 as the default proposal standard deviation, and tuning if necessary to achieve desired acceptance rates.

3. Sample
   \[
   \beta \mid \sigma^2, \xi, \eta, z \sim N \left( (W^T W + (\xi^{-1} D)^{-1})^{-1} W^T z, \sigma^2 (W^T W + (\xi^{-1} D)^{-1})^{-1} \right)
   \]
   adopting the algorithm of [4], which requires the following steps:
   - Sample \( u \sim N(0, \xi^{-1} D) \) and \( f \sim N(0, I_N) \) independently, and set \( v = W u + f \).
   - Set \( v^* = M_\xi^{-1} (z/\sigma - v) \) by solving \( M v^* = (z/\sigma - v) \), where \( M_\xi \) is as in (7).
   - Set \( \beta = \sigma (u + \xi^{-1} D W^T v^*) \).

4. Update \( \eta_j \) from independent conditionals using slice sampling. Specifically, sample
   \[
   u_j \sim \text{Unif} \left( 0, \frac{1}{\eta_j + 1} \right),
   \]
   then sample \( \eta_j \) from an exponential distribution with rate
   \[
   m_j = \frac{\beta_j^2 \xi}{2\sigma^2}
   \]
   truncated to the interval \((0, r_j)\), where \( r_j = \frac{1 - u_j}{u_j} \). This is done by sampling \( v_j \sim \text{Unif}(0, 1) \) and setting
   \[
   \eta_j = -\log[1 - \{1 - \exp(-m_j r_j)\} v_j]/m_j.
   \]
Step 1. is equivalent to making a change of variables $\xi \mapsto \log(p_\xi q)$ and updating $\log(p_\xi q)$ using random-walk Metropolis. Although random walks on the logarithm are sometimes used simply to sample a strictly nonnegative parameter, Johnson and Geyer [17] suggest this as a strategy to obtain geometrically ergodic Metropolis algorithms in cases where the tails of the target on the original scale are too heavy for random walk Metropolis to be geometrically ergodic. While our algorithm is a more complicated case of Metropolis-within-Gibbs, it is worth noting that a random walk on $\sqrt{\xi}$ has very poor empirical behavior in this setting; the change of measure is essential to obtaining good performance.

The algorithm uses different linear algebra routines depending on whether $W$ is dense or sparse. When $W$ is dense, we directly solve the linear system $z = M_\xi b$ for $b$ to calculate $z' M_\xi^{-1} z$ in Step 2, and obtain the determinant of $M_\xi$ from a Cholesky decomposition. When $W$ is sparse, we use a direct sparse linear system solver. This was consistently faster than a preconditioned conjugate gradient algorithm. Also for sparse $W$, we obtain the Cholesky of $M_\xi$ after computing an approximate minimum degree permutation using the Matlab function symamd [1].

Another improvement in this algorithm is that our use of (8) appears to be more numerically stable than the reliance of [26] and [4] on built-in functions, a feature that we alluded to previously in the discussion of improved estimation from the new algorithm.

2.3 Approximate algorithm

As outlined earlier, the approximate algorithm thresholds certain entries of $D$ to replace it by $D_\delta$, where $\delta > 0$ is some a priori fixed threshold.

1. (Update of $\xi$:) Define

$$D_\delta = \text{diag} \left( \eta_j^{-1} 1(\xi_j^{-1} \eta_j^{-1} > \delta) \right),$$

where $\xi_{j_{\text{max}}}^{-1} = \max\{\xi^{-1}, (\xi^*)^{-1}\}$. In the log-likelihood calculation, replace $M_\xi$ and $M_{\xi^*}$ with $M_{\xi,\delta}$ and $M_{\xi^*,\delta}$, where

$$M_{\xi,\delta} = I_N + c^{-1} W D_\delta W' , \quad c \in \{\xi, \xi^*\}.$$

2. (Update of $\sigma^2$:) Sample $\sigma^2$ from

$$\text{InvGamma} \left( \frac{n + a_0}{2}, \frac{z^T M_{\xi,\delta}^{-1} z + b_0}{2} \right),$$

where $M_{\xi,\delta}$ is as in [1], and $\xi$ is the current value of $\xi$.

3. (Update of $\beta$:) Follow the steps

- Sample $u \sim N(0, \xi^{-1} D)$ and $f \sim N(0, I_N)$ independently, and set $v = W u + f$.
- Set $v^* = M_{\xi,\delta}^{-1} (z/\sigma - v)$ by solving $M_{\xi,\delta} v^* = (z/\sigma - v)$, where $M_{\xi,\delta}$ is as in [1].
- Set $\beta = \sigma (u + \xi^{-1} D_\delta W' v^*)$.

4. (Update of $\eta$:) Same as before.

\footnote{In reality the implementation is slightly more complicated, and we use log1p and expm1 when $m_j r_j$ is close to machine precision.}
The primary motivation behind the approximate algorithm is to improve per-iteration computational complexity without sacrificing accuracy. As discussed earlier, when the truth is sparse, we expect a large subset of \((\xi^{-1} \eta_j^{-1})\) to be small a posteriori, and hence setting the entries smaller than a small threshold \(\delta\) should not affect the accuracy of the algorithm; we make this argument concrete in the next section. Thresholding those small entries has significant computational advantages. The speedup from this approximation is best when the truth is sparse or close to sparse, so that most entries of \(\beta\) are shrunk to near zero. Critically, coordinates that are thresholded away at iteration \(k\) need not be thresholded away at iteration \(k + 1\), and in practice the set of variables that escapes the threshold does change considerably from one iteration to another. This can occur because the thresholded coordinates are never actually set to zero or omitted, but rather sampled from a Gaussian that closely approximates the exact full conditional. Thus, we are not sacrificing the primary benefit of Bayesian methods for sparse regression: estimates of uncertainty about the set of true signals are still valid.

Consider the computational cost of extending the Markov chain by a single step. The exact algorithm needs to calculate \(|M_\xi|, z' M_\xi^{-1} z\) and solve a linear system in \(M_\xi\) in each iteration, each of which requires \(O(N^3)\) operations. Further, formation of the matrix \(M_\xi\) itself requires computation of \(WDW'\), which has complexity \(O(N^2p)\). The approximate algorithm on the other hand needs to calculate \(WD_\delta W'\), with a subset of the diagonal entries of \(D_\delta\) being zero. Let \(S = \{j \in [p] : \xi^{-1} \eta_j^{-1} > \delta\}\) denote the active set of variables which escape the threshold, and set

\[
s_\delta = |S| = \sum_{j=1}^{p} \mathbf{1}(\xi^{-1} \eta_j^{-1} > \delta).
\]

Also, let \(D_S\) denote the \(s_\delta \times s_\delta\) sub-matrix of \(D\) and \(W_S\) the \(n \times s_\delta\) sub-matrix of \(W\) resulting from picking out the non-thresholded diagonal entries/columns indexed by \(S\). When the truth is sparse, \(s_\delta \approx p\) after a few iterations and \(WD_\delta W' = W_S D_S W_S'\), which costs \(N^2 s_\delta\), providing significant savings. A second level of computational savings can be made when \(s_\delta < N\), whence \(WD_\delta W'\) is a reduced-rank approximation to \(WDW'\). In such cases, our implementation altogether replaces the calculation of \(WD_\delta W'\) and the formation of \(M_{\xi, \delta}\) by directly calculating

\[
M_{\xi, \delta}^{-1} = (I_N + \xi^{-1} WD_\delta W')^{-1} = I_N - W_S (\xi D_S^{-1} + W_S W_S')^{-1} W_S',
\]

using the Woodbury matrix identity. The calculation of \(z' M_{\xi, \delta}^{-1} z\) and \(M_{\xi, \delta}^{-1}\) are performed by substituting the above expression of \(M_{\xi, \delta}^{-1}\), which only requires solving \(s_\delta \times s_\delta\) systems, and has overall complexity \(s_\delta^3 \vee s_\delta N\). The determinant of \(I + \xi^{-1} WD_\delta W'\) is then computed by (a) performing a singular value decomposition of \(W_S D_S^{1/2}\), which costs \(O(s_\delta^2 N)\), and then (b) calculating the eigenvalues as \(1 + s^2\), where \(s\) is a vector of the singular values of \(W_S D_S^{1/2}\), \((N - s_\delta)\) of which are identically zero. Accounting for the calculation of \(Wu\) performed when sampling \(\beta\), which costs \(O(Np)\), the per step computational cost of the approximate algorithm when \(s_\delta < N\) is order \((s_\delta^3 \vee p)N\). Thus by exploiting the sparse structure of the target, the algorithm achieves similar computational cost per step to coordinate descent algorithms for Lasso and Elastic Net [10] Sections 2.1, 2.2.

The threshold \(\delta\) should satisfy \(\delta \ll 1\). To see why, suppose that at most \(N\) of the diagonal entries of \(D\) are nonzero and the columns of \(W\) corresponding to nonzero diagonal entries of \(D\) are orthogonal. In this case

\[
M_\xi = I + \xi^{-1} WDW' = W(I + \xi^{-1} D)W
\]

with eigenvalues \((1 + \eta_j^{-1} \xi^{-1})\) for \(j : \lambda_j \neq 0\), so \(M^{-1}\) has eigenvalues \((1 + \eta_j^{-1} \xi^{-1})^{-1}\). In this idealized setting, an approximation \(D_\delta\) that thresholds some of the nonzero \(\eta_j\) will be very accurate precisely when \(\xi^{-1} \eta_j^{-1} \ll 1\). This heuristic will be approximately correct whenever the columns of \(W\) are not highly collinear. When highly collinear columns do exist, thresholding \(\xi^{-1} \eta_j^{-1}\) at \(\delta\) can result in a much worse approximation, since \(m\) highly collinear columns with small \(\eta_j\) correspond approximately to a single eigenvector of \(\xi^{-1} WDW'\) with eigenvalue \(m\) times what the above calculation would suggest. Thus, if we know that \(W\) has low coherence, then a value of
\[ \delta = 10^{-2} \] might be sufficient. To compensate for the likely presence of highly collinear columns in applications, we suggest taking \( \delta = 10^{-4} \). We have detected no evidence of loss of accuracy using this threshold, even with highly dependent design matrices. In practice, the overriding factor in choosing such thresholds is computational budget, so we suggest taking \( \delta \) as small as possible while satisfying computational constraints. A more rigorous discussion of approximation accuracy is given in Section 5.3.

Before we conclude this section, we provide some additional insight into the consequences of the approximation for \( \beta \). The effects of the modified updates for \( \xi \) and \( \sigma^2 \) are relatively direct to see; however those for \( \beta \) modify multiple steps of the algorithm in [4]. Define \( \Gamma := \xi^{-1}D \) and \( \Gamma_\delta = \xi^{-1}D_\delta \). The approximate algorithm for \( \beta \) sets

\[ \beta = \Gamma_\delta W'M_\delta^{-1}z + \sigma (u - \Gamma W'M_\delta^{-1}v). \]

Since \((u,v)\) is jointly Gaussian, \( \beta \) obtained above continues to have a Gaussian distribution, \( \beta \sim N(m_\delta, \sigma^2 \Sigma_\delta) \), with

\[ m_\delta := \Gamma_\delta W'M_\delta^{-1}z, \quad \Sigma_\delta := \text{cov}(u - \Gamma W'M_\delta^{-1}v). \]

Some further simplifications (see Appendix for details) yields,

\[ m_\delta = (m_S; 0_{(p - s_\delta) \times 1}), \quad m_S = (W_S'W_S + \Gamma^{-1}_S)^{-1}W_S'z, \quad (9) \]

and\(^3\)

\[ \Sigma_\delta = \begin{bmatrix} (W_S'W_S + \Gamma^{-1}_S)^{-1} & -\Gamma_S W_S'M_\delta^{-1}W_S' \Gamma_{S'c} \\ -\Gamma_{Sc} W_S'M_\delta^{-1}W_S' \Gamma_S & \Gamma_{S'} \end{bmatrix}. \quad (10) \]

Writing \( \beta = (\beta_S; \beta_{Sc}) \), we have \( E(\beta_{Sc}) = 0 \), i.e., the entries of \( \beta \) outside the active set are drawn from a zero mean distribution. Second, the marginal distribution of \( \beta_S \) is \( N((W_S'W_S + \Gamma^{-1}_S)^{-1}W_S'z, \sigma^2(W_S'W_S + \Gamma^{-1}_S)^{-1}) \), which would exactly be the full conditional distribution of \( \beta \) if the model was fitted with the current set of active variables.

### 3 Theoretical background and results

In this section we describe a notion of the computational cost of MCMC and relate this to the computational cost of algorithms for local-global models. We discuss empirical approaches for comparing the convergence properties of algorithms that are used subsequently. Although empirical analysis is no substitute for sharp bounds on convergence rates, an algorithm that performs poorly while satisfying computational constraints is unlikely to be a useful algorithm. The converse is not true, a problem that has long been observed in MCMC convergence diagnosis.

#### 3.1 A notion of computational cost

The goal of Bayesian analysis is often to compute expectations of functions with respect to the posterior

\[ m(x) = \frac{L(z \mid x)\pi(x)}{\int L(z \mid x)\pi(x)dx}, \]

where \( m(x) \) is the density of parameters conditional on observing \( z \) when our prior beliefs were expressed by the density \( \pi(x) \). We will use \( \mu(\cdot) \) to denote the measure whose density with respect to Lebesgue measure is given by \( m(x) \).

Typically expectations with respect to \( \mu \) are not available in closed form and so are approximated. Arguably the most common approximation method is to construct a Markov kernel \( \mathcal{P}(x, \cdot) \) with state space \( X = \mathbb{R}^p \) and unique invariant measure \( \mu \), and collect a path \( X_0, \ldots, X_{n-1} \) of

\(^3\)When we write \( m_\delta = (m_S; 0_{(p - s_\delta) \times 1}) \), we simply mean that the sub-vector of \( m_\delta \) corresponding to the indices in \( S \) is \( m_S \) while the rest are zero. Similarly, blocks of \( \Sigma_\delta \) are defined by \( S \) and \( S' \).
length $n$ with $X_k \sim \nu \mathcal{P}^{k-1}$ for some starting measure $\nu$. Expectations are then approximated by time averages

$$\mu \varphi = \int \varphi(x) \mu(dx) \approx \frac{1}{n} \sum_{k=0}^{n-1} \varphi(X_k).$$

Suppose there exists a nonnegative $\mu$-measurable function $C(x)$ and a nonnegative decreasing function $\gamma(n) : \mathbb{N} \rightarrow [0, \infty)$ such that

$$|\mathcal{P}^n(x, \cdot) - \mu(\cdot)|_{TV} \leq C(x) \gamma(n). \quad (11)$$

A Markov chain $X_n$ evolving according to $\mathcal{P}$ is geometrically ergodic when $\gamma(n) = \bar{\alpha}^n$ for $\bar{\alpha} \in [0, 1)$, and $X_n$ is polynomially ergodic of order $m$ when $\gamma(n) = n^{-m}$ for some $m > 0$. We refer to $\gamma(n)$ as the rate function of $\mathcal{P}$.

For $\mu$-measurable functions $\varphi$ define

$$\sigma^2_{\varphi} := \text{var}_\mu(\varphi) \left(1 + 2 \sum_{k=1}^{\infty} \text{cor}(\varphi(X_k), \varphi(X_0)) \right) \quad (12)$$

for $X_0 \sim \mu$. The sum in (12) is called the integrated autocorrelation time. Under conditions that are effectively sufficient for summability, (12) will be the asymptotic variance in a central limit theorem for time averages. Specifically, suppose that either

1. $X_n$ is geometrically ergodic and $\mu|\varphi|^{2+\delta} < \infty$; or
2. $X_n$ is polynomially ergodic of order $m$, $\mu C < \infty$, and $\mu|\varphi|^{2+\delta} < \infty$ where $m \delta > 2 + \delta$,

then for any initial distribution $\nu$ such that $X_0 \sim \nu$, we have (see [19])

$$\sqrt{n} \left( n^{-1} \sum_{k=0}^{n-1} \varphi(X_k) - \mu \varphi \right) \sim \mathcal{N}(0, \sigma^2_{\varphi}).$$

A similar result can be obtained when the convergence in (11) is metrized in the Wasserstein distance, rather than total variation [21], and some form of this result holds even when the chain does not begin from the stationary measure under some conditions on the starting measure (see [21, 31]).

It follows that, in the ideal setting where the chain starts from stationarity, it is necessary to obtain a Markov chain of length at least $\epsilon^{-1} \sigma^2_{\varphi}$ to reduce the Monte Carlo variance for approximation of $\mu \varphi$ by its time-averaging estimator to a defined tolerance $\epsilon$. Similar bounds on the mean squared error of the MCMC estimate can be obtained even when the chain does not begin from stationarity. Thus, one measure of the contribution to computational cost of the MCMC path length is the rate at which $\sigma^2_{\varphi}$ grows as a function of $N, p$, since the path length must also grow at this rate in order to achieve constant Monte Carlo error.

Estimates of $\sigma^2_{\varphi}$ are referred to as the MCMC standard error, and there is a significant literature on the properties of different estimators (see [3] for a rigorous treatment). Several of these estimators are implemented in the R package mcmcse. We have consistently found the overlapping batch means estimator with the theoretically optimal $n^{1/3}$ batch size to perform the best, and we use this estimator throughout the paper. The asymptotic variance should be estimated after discarding the initial portion of the path; we discard 5,000 scans.

Using estimates $\hat{\sigma}^2_{\varphi}$ for coordinate projections, we empirically analyze the effect of problem size on the required path length as follows. Suppose that the relationship $\hat{\sigma}^2_{\varphi} = B N^{a_1} p^{a_2}$ for constants $B, a_1, a_2$ dictates the growth rate of $\hat{\sigma}^2_{\varphi}$ with $N$ and $p$; that is to say, the asymptotic variance grows like a polynomial in $N, p$. Then,

$$\log(\hat{\sigma}^2_{\varphi}) = \log(B) + a_1 \log(N) + a_2 \log(p),$$

and thus one can obtain a rough estimate of the order of $\sigma^2_{\varphi}$ in $p$ and $N$ from a regression of $\log(\hat{\sigma}^2_{\varphi})$ on $\log(N) + \log(p)$. We propose to compare estimates $\hat{a}_1, \hat{a}_2$ of $a_1, a_2$ across different algorithms.
as a way to empirically evaluate the relative computational complexity arising from the growth of the asymptotic variance.

A related pathwise quantity is the effective sample size \( n_e \), which is usually defined as

\[
n_e = \frac{\text{var}_\mu(\varphi)n}{\sigma^2_\varphi},
\]

(13)
an adjustment to the path length \( n \) to reflect how much the asymptotic variance, \( \sigma^2_\varphi \), is inflated by autocorrelation. Clearly, \( n_e \) is proportional to the reciprocal of the asymptotic variance, so larger \( n_e \) is better. To estimate \( n_e \) from paths of length \( n \), we employ the procedure in \texttt{mcmcmse}, again using the overlapping batch means estimator with \( n^{1/3} \) batch size and discarding 5,000 initial iterations.

### 3.2 Use of approximating kernels

Consider the Markov kernel defined by replacing \( WDW' \) by \( WDW' \) in the update rule in Section 2.2. Such an approximation defines a new Markov kernel \( P_\epsilon \) that differs from \( P \), with \( \epsilon \equiv \epsilon(\delta) \). Even if \( P \) has a unique invariant measure and other desirable properties, \( P_\epsilon \) may not. Johndrow and Mattingly \cite{Johndrow2016} show conditions under which time averages from \( P_\epsilon \) give useful approximations to expectations under \( P \). The basic requirement is that \( d(P(x,\cdot), P_\epsilon(x,\cdot)) \leq \epsilon(1+\delta V(x)) \), where \( V(x) \) is a Lyapunov function of \( P \), i.e. it satisfies \( (PV)(x) \leq \gamma V(x) + K \) for constants \( \gamma \in (0,1), 0 \leq K < \infty \). The metric \( d \) is a weighted total variation norm, where \( V \) is the weighting function; convergence in this weighted total variation is equivalent to \( \epsilon \rightarrow 0 \). Johndrow and Mattingly \cite{Johndrow2016} show that this weighted total variation error condition is weaker than the condition

\[
\sup_{x \in X} \|P(x,\cdot) - P_\epsilon(x,\cdot)\|_{TV} \leq \epsilon(x)
\]

where \( \epsilon(x) : X \rightarrow [0,2) \) is a state-adaptive total variation error. The exact value \( \epsilon(x) \) needed at state \( x \) depends on the spectral gap of \( P \) in a weighted total variation norm, which is typically hard to bound sharply. Instead, the basic implication is to pick a small value of \( \epsilon \) and control total variation uniformly at this value. Since paths from \( P \) of finite length initialized at, say, the origin, reside in a compact subset of \( X \) with high probability, uniform total variation control at level \( \epsilon \) will almost always be sufficient so long as we take \( \epsilon \) small enough. As such, we focus on showing that our approximate algorithm achieves uniform total variation error control. It is worth noting that when the exact kernel \( P \) is a strict contraction in a Wasserstein rather than weighted total variation metric, similar guarantees on approximation error can be achieved when \( d(P(x,\cdot), P_\epsilon(x,\cdot)) \leq \epsilon \), where \( d \) is a form of the Wasserstein metric \cite{Johndrow2016}. Wasserstein bounds can also be obtained with a Lyapunov structure, see e.g. Rudolf and Schweizer \cite{Rudolf2011}.

### 3.3 Results on approximation error

The following result shows that the Markov kernel \( P_\epsilon \) defined by the update given in 2.3 achieves uniform total variation error control.

**Theorem 3.1.** Let \( P_\epsilon(x,\cdot) \) and \( P(x,\cdot) \) respective denote the Markov operators for the approximate and exact algorithms, with \( x = (\beta, \sigma^2, \xi, \eta) \) denoting the entire state vector. Then

\[
\sup_x \|P(x,\cdot) - P_\epsilon(x,\cdot)\|_{TV} = \sqrt{\delta} |W| \sqrt{\frac{N + a_0}{b_0}} + \frac{N}{2} \frac{\|z\|^2}{b_0} + O(\delta),
\]

for sufficiently small \( \delta > 0 \).

The proof is lengthy and provided in Appendix B; we provide a high-level overview here. Substantial care is required to obtain a bound independent of the current state \( x \). We first show that the total variation distance between \( P(x,\cdot) \) and \( P_\epsilon(x,\cdot) \) is uniformly bounded by the sum of the following two quantities:

\[
\sup_{\eta, \xi} \|\pi(\beta, \sigma^2 | \eta, \xi, z) - \pi_\epsilon(\beta, \sigma^2 | \eta, \xi, z)\|_{TV},
\]

(14)
$
abla_\eta,\xi,\xi^*\alpha_\eta\xi,\xi^*|\alpha_\eta\xi,\xi^*|.$ (15)

Here, $\pi_\beta,\sigma^2|\eta,\xi,z$ denotes the full conditional distribution of $(\beta, \sigma^2)$ for the Gibbs update in the exact algorithm, and $\pi_\epsilon$ the approximate full conditional resulting from the approximations in steps (2) and (3) in [25]. Similarly, $\alpha_\eta\xi,\xi^*$ and $\alpha_\eta,\epsilon\xi,\xi^*$ respectively denote the MH acceptance probability for a move from $\xi$ to $\xi^*$ given the current value of $\eta$ in the exact and approximate algorithm.

The former bound is nontrivial to obtain, with delicate treatment of the eigenvalues of several matrices appearing in the intermediate calculations being essential to obtaining an upper bound that does not depend on the current state. Obtaining true state-independent bounds is fairly difficult, and in many cases only a state-dependent bound is available, requiring tuning of the threshold – in our case, $\delta$ – as a function of the state to achieve the theoretical conditions (see e.g. [16]). As such, the result in Theorem 3.1 is comparatively quite strong. In addition, the approach we use to obtain bounds on the total variation difference between two Normal-inverse Gamma distributions may be broadly useful, as this distribution appears frequently in Bayesian model averaging and as a conditional in Gibbs sampling.

4 Analysis of computational cost

The results in this and the following section are based on a series of simulations in which the data are generated from

\begin{align*}
    w_i & \sim w_i^{\text{id}} \sim N_p(0, \Sigma) \\
    z_i & \sim N(w_i\beta, 4) \\
    \beta_j & = \begin{cases} 
        2^{-(j/4-9/4)} & j < 24 \\
        0 & j > 23
    \end{cases}
\end{align*}

(16)

In contrast to typical simulations studies for shrinkage priors, in which signals are typically either zero or large relative to the residual variance, we use a decreasing sequence of signals. The largest signal size is 4, while 18 out of the 23 signals are smaller than the residual variance. For all of the problem sizes that we consider, this results in bimodal marginal posterior for at least some of the $\beta_j$, increasing the difficulty of sampling from the target. We consider two cases for $\Sigma$: the identity and $\Sigma_{ij} = \phi|i-j|$. The latter is the covariance matrix for an autoregressive model of order 1 with autoregressive coefficient $\phi$ and stationary variance $(1-\phi^2)^{-1}$. Throughout, we put $\phi = 0.9$ when simulating a dependent design. Because all of the nonzero signals are in the first 23 elements of $\beta$, all of the $\beta_j$ corresponding to true signals will be highly correlated a posteriori, again considerably increasing the difficulty of efficiently sampling from the target.

4.1 Cost per step

Table 1 shows estimates of coefficients from a regression of log($t$) on log($N$) + log($p$) for the old, new, and approximate algorithms. These estimates are based on 20 simulations from the model in (16) with $N$ sampled uniformly at random from integers between 200 and 1,000 and $p$ sampled uniformly at random from integers between 1,000 and 5,000. The algorithm was run for 20,000 iterations and total wall clock time recorded. Computation was performed on multicore hardware with 12 threads, so matrix multiplications contribute less to the wall clock time than do matrix decompositions, resulting in the lower than expected exponents on $N, p$. These estimates are meant to reflect the actual performance on modern multicore hardware rather than to verify the sharpness of the theoretical bounds. Moreover, the computation time of the approximate algorithm is likely non-constant in $N, p$. For larger dimensions, the initial few iterations are likely to dominate the total computation time, since the benefits do not emerge until the algorithm locates most of the true nulls. This cost could be largely eliminated by “warm starting” the algorithm at, say, the cross-validated Lasso solution, which can be computed in nearly linear time in $N, p$. 

14
| dimension | old   | new   | approximate |
|-----------|-------|-------|-------------|
| $\log(N)$ | 1.6478 | 1.6847 | 0.5449      |
| $\log(p)$ | 0.7204 | 0.6065 | 0.3392      |

Table 1: Estimates from regression of $\log(t)$ on $\log(N) + \log(p)$.

### 4.2 Cost related to variance of time averages

To assess the cost due to increased variance of the time-averaging estimator as a function of $N, p$, we conduct another set of simulations. We focus on the performance of the approximate algorithm, since its much lower computational cost per step allows a wider range of values of $N, p$ in the simulation study, improving the reliability of the results. The results that follow are based on two simulation studies from the setup in (16), each consisting of 20 independent simulations in which $N$ was sampled uniformly at random from the integers between 1,000 and 5,000 and $p$ was sampled uniformly at random from the integers between 10,000 and 50,000. In the first simulation study, we use an independent design. In the second simulation study, we use a correlated design with AR-1 structure and autocorrelation 0.9 as described above. The approximate algorithm was run for 20,000 iterations. Calculations of effective sample sizes $n_e$ and standard errors were based on the final 15,000 iterations.

Figure 5 shows the distribution of $n_e$ based on the first 100 entries of $\beta$, the corresponding entries of $\eta$, $\log(\xi)$, and $-2\log(\sigma)$ as a function of $p$; each simulation also has a different value of $N$. No variation by $p$ is evident in either the independent or correlated design case.

![Figure 5: Effective sample sizes $n_e$ for 100 entries of $\beta$, 100 entries of $\eta$, $\log(\xi)$, and $-2\log(\sigma)$. The 100 entries of $\beta,\eta$ include those corresponding to all of the true signals, and 76 null signals. The horizontal axis indicates the value of $p$ used in each of the 20 simulations.](image)

Figure 6 shows the analogous result, but as a function of $N$. A slight increase in effective sample size as $N$ increases is possible. There is apparently little difference in $n_e$ when the design matrix is correlated compared to independent design.

Tables 2 and 3 show results of a linear model with specification

$$
\log(\hat{\sigma}_j^2(i)) = a_0 + a_1 \log(N_i) + a_2 \log(p_i) + b_j + \epsilon_{ij}
$$

where $\varphi_j(x) = x_j$ is $j$th the coordinate projection of the partial state vector $x = (\beta_{1:100}, \eta_{1:100}, \log(\xi), -2\log(\sigma))$ and $i = 1, \ldots, 20$ indexes the simulation scenario. Clearly, some coordinates tend to mix better than others, and the coordinate-specific intercepts allow for this variation. Results for independent design are shown in Table 2 and for dependent design in Table 3. The small, negative coefficient estimates suggest that if anything the Markov chain actually mixes slightly more rapidly as $N, p$
Figure 6: Analogous to Figure 5, except that the horizontal axis here indicates the value of \( N \) used in each of the 20 simulations.

increase. Thus, there is little evidence that a longer path is needed to achieve fixed Monte Carlo error as \( N, p \) grow.

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -7.65 | 0.32 | -23.65 | 0.00 |
| log(N) | -0.17 | 0.03 | -5.39 | 0.00 |
| log(p) | -0.03 | 0.02 | -1.83 | 0.07 |

Table 2: estimated parameters from regression of \(- \log(n_e) \) on \( \log(N) + \log(p) \), independent design

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -7.56 | 0.54 | -14.03 | 0.00 |
| log(N) | -0.15 | 0.03 | -5.04 | 0.00 |
| log(p) | -0.06 | 0.04 | -1.47 | 0.14 |

Table 3: estimated parameters from regression of \(- \log(n_e) \) on \( \log(N) + \log(p) \), dependent design

5 Statistical performance

Because the old algorithm often fails to converge, computational cost is not a complete measure of the difference between the old and new algorithms. In this section, we analyze the performance of the three algorithms in the estimation of \( \beta \), which is typically the focus of inference. We again use the simulation setup in (16) with \( N \) sampled uniformly at random from the integers between 200 and 1,000, and \( p \) sampled uniformly at random from the integers between 1,000 and 5,000. Mean squared error (MSE) for estimation of \( \beta \) by MCMC time averages is shown in the left panel of Figure 7. There is no discernible difference between the performance of the new and approximate algorithms, but the old algorithm has about double the MSE at the median over the 20 simulations. Similarly, median empirical coverage of 95 percent credible intervals is about 90 percent for the old algorithm, and in only one case did the empirical coverage achieve 95 percent. In contrast, the new and approximate algorithms have median empirical coverage of about 93 percent, and never exhibited empirical coverage below 90 percent. We know from van der Pas et al. [37] that credible intervals for intermediate-sized signals cannot achieve the nominal coverage, even asymptotically. Since our simulation involves a sequence of decreasing signals, undoubtedly some of them fall into this “intermediate” categorization. As such, the performance of the new and approximate algorithms with respect to empirical coverage is probably near optimal.
As described in Section 2, we conjecture that the poor performance of the old algorithm is due to numerical truncations that were used to prevent underflow, but inadvertently resulted in non-convergence. While it is primarily the global scale that fails to converge, this appears to cause the residual variance to settle around a value far from the truth, and components of \( \beta \) corresponding to intermediate sized true signals to eventually “stick” in the zero-centered mode and never re-emerge. This latter symptom is most likely the cause of poor performance in estimation of \( \beta \). These failure modes are illustrated quite dramatically in Figures 3 and 4.

Figures 12, 13, and 11 in the Appendix also evaluate statistical performance of the approximate algorithm. Figures 12 and 13 show posterior marginals for the first 25 entries of \( \beta \) for simulations with \( N = 1,000, p = 5,000 \) and \( N = 5,000, p = 50,000 \), respectively, along with the true values of \( \beta \). In general, the marginals have single modes centered near the truth for larger true signals, two modes with one centered near the truth and one centered at zero for intermediate sized true signals, and single modes at zero when the true signal is small or identically zero. This is consistent with the expected behavior of the Horseshoe. Figure 11 shows violin plots with indicated 95 percent credible intervals for \( \sigma^2 \) over 20 independent simulations each with 1,000 \( \leq N \leq 5,000 \) and 5,000 \( \leq p \leq 50,000 \). All but two of the intervals cover the true value of 2. Overall, the approximate algorithm has exhibited excellent statistical performance by every metric we have considered.

6 GWAS Application

We use the Horseshoe with computation by the approximate algorithm to analyze a Genome-Wide Association Study (GWAS) dataset. The data consist of \( N = 2,267 \) observations and \( p = 98,385 \) single nucleotide polymorphisms (SNPs) in the genome of maize. These data have been previously studied by [22] and [38]. Each observation corresponds to a different inbred maize line from the USDA Ames seed bank [30]. As the response, we use growing degree days to silking, a measure of the average number of days exceeding a certain temperature that are necessary for the maize to “silk.” Maize is typically ready to harvest about 60 days after silking, so this is a measure of the length of the growth cycle for a particular line of maize, crudely controlling for temperature. This response is also considered by [38].

We run the approximate algorithm for 30,000 iterations, discarding 5,000 iterations as burn-in. Figure 8 shows histograms of \( n_e \) and \( n_e/t \) for 200 entries of \( \beta \), the corresponding 200 entries of \( \eta \), \( \log(\xi) \), and \( -2\log(\sigma) \). The 200 entries of \( \beta, \eta \) includes the 100 entries for which the posterior mean is largest in absolute value, as well as 100 other entries. The smallest value of \( n_e \) observed was 893, and the smallest value of \( n_e/t \) 0.05. The median values were 4531 and 0.24, respectively. Thus the algorithm remains quite efficient, even on a fairly large, real dataset.

Figure 9 shows density plots of samples for the nine entries of \( \beta \) with largest estimated absolute posterior mean, as well as the estimated posterior mean. The Lasso estimates for these parameters, with the penalty chosen by 10-fold cross-validation, are also indicated. It is clear that, even for the entries of \( \beta \) for which the signal strength is largest, the Horseshoe marginals are typically bimodal, with the weight in the mode centered at zero increasing with decreasing signal strength. This
suggests that the bimodal shape of the marginals may be quite common in applications, and gives some sense of the level of uncertainty about which entries correspond to true signals. The Lasso estimates for these relatively large parameters are typically shrunken toward zero relative to the Horseshoe posterior mean, a behavior that has been observed previously (see [3]).

Figure 9 plots the number of entries of $\beta$ for which the absolute value of the corresponding lasso or Horseshoe point estimates exceed a threshold between 0.0005 and 0.1. Also shown is the size of the intersection of these two sets. For larger thresholds, the number of Horseshoe point estimates exceeding the threshold is typically larger than that for Lasso, while for smaller thresholds, this trend is reversed. This is again consistent with the tendency of Lasso to overshrink large signals and undershrink small signals [3]. The size of the intersection closely tracks the minimum size of the two sets, suggesting that Lasso and Horseshoe largely agree as to which coefficients represent signals, but disagree somewhat about their magnitude. Of course, Lasso provides no notion of uncertainty in the selected variables such as that conveyed by the posterior marginals of Horseshoe.

7 Discussion

It is now ten years since the Bayesian Lasso and the associated Gibbs sampling algorithm were proposed in [24], eight years since the Horseshoe prior appeared in [6], and 22 years since the
landmark Lasso paper of Tibshirani [34]. While the introduction of the least angle regression
algorithm [5], and, more recently, coordinate descent algorithms [10], have made $L_1$ regularized
regression with hundreds of thousands of predictors possible on standalone computing hardware,
no existing implementation of Bayes Lasso, Horseshoe, or any other Bayesian global-local shrinkage
prior scales to this problem size. This has probably limited the adoption of these attractive Bayesian
methods by practitioners, especially in the biological sciences where large $p$ is common. Regardless
of the virtues of a statistical procedure, it is of little use practically if it is not computable.

Here we have for the first time offered a computational algorithm for Horseshoe that can scale
to hundreds of thousands of predictors. Our algorithm has the same computational cost per step
as coordinate descent for elastic net and Lasso when the truth is sparse, though naturally more
computation time is required to obtain a Markov chain of the requisite length than to obtain a
single path of Lasso solutions. However, one gains more information from the Horseshoe, perhaps
most critically some measure of uncertainty in which $\beta_j$ correspond to true signals. The Bayesian
community has long recommended against selecting single models without reporting uncertainty in
which model is selected, but has often not provided algorithms that implement the recommended
methods in large $p$ problems. This has perhaps contributed to the growing importance of selective
inference over Bayesian methods, as practitioners have mostly adopted the strategy of selecting a
single model. We hope that the computational strategies outlined here will contribute to growth
in the use of Bayesian methods in high-dimensional settings, and that exploiting sparsity and
other special structure of the target will be more widely adopted as a means to develop accurate
approximate MCMC algorithms for modern applications.

Acknowledgements

The authors thank Professors Xiaolei Liu and Xiang Zhou for sharing the Maize GWAS data. Prof.
Bhattacharya’s research is supported by the National Science Foundation (NSF DMS 1613156) and
an NSF CAREER award (DMS 1653404).
Appendix

A Derivation of $m_\delta$ and $\Sigma_\delta$ in (9) and (10)

Let us first derive $m_\delta$. Recalling the definition of $D_\delta$, we have $\Gamma S W' = (\Gamma S W'_S; 0_{p-\delta \times N})$ and $\Omega T S W' = W_S \Gamma S W'_S$. Thus, $m_\delta = (m_S; 0_{p-\delta \times 1})$, with

$$m_S = \Gamma S W'_S (I_N + W_S \Gamma S W'_S)^{-1} z = (W'_S W_S + \Gamma S^{-1})^{-1} W'_S z.$$  

A proof of the second equality can be found in the proof of Proposition 1 in [4].

We now derive $\Sigma_\delta$. Again, using the definition of $D_\delta$, we have $u = \Gamma W'M_\delta^{-1} v = (u_S - \Gamma S W'_S M_\delta^{-1} v; u_{S^c})$, where $M_S = (I_N + W_S \Gamma S W'_S)$. Also, recall that $v = W' u + f = W_S u_S + W_{S^c} u_{S^c} + f$, and $u_S \perp u_{S^c}$ since $\Gamma$ is diagonal, which together imply $cov(u_S, v) = \Gamma S W'_S$ and $cov(u_{S^c}, v) = \Gamma S W'_S$. We now derive the blocks of $\Sigma_\delta$.

1. We have,

$$cov(u_S - \Gamma S W'_S M_\delta^{-1} v) = \Gamma S - \Gamma S W'_S M_\delta^{-1} W_S \Gamma S = (W'_S W_S + \Gamma S^{-1})^{-1},$$

where the proof of the second equality can be found in the proof of Proposition 1 in [4].

2. Next, using $u_S \perp u_{S^c}$,

$$cov(u_S - \Gamma S W'_S M_\delta^{-1} v, u_{S^c}) = -\Gamma S W'_S M_\delta^{-1} W_S \Gamma S^{-1}.$$  

3. Finally, $cov(u_{S^c}) = \Gamma S_{\delta}.$

B Proof of Theorem 3.1

Recall we denote $x = (\beta, \sigma^2, \xi, \eta)$ for the entire state vector. We shall also call $\theta = (\beta, \sigma^2)$ and $x_{\eta} = (\beta, \sigma^2, \xi)$. The Markov kernel $P(x, \cdot)$ for the exact algorithm has a transition density

$$p(x' | x) = \pi(\eta' | x_{\eta}, z) \pi(\theta' | \xi', \eta, z) p_{\text{sm}}(\xi' | \xi, \eta), \quad x' \in X$$

where $\pi(\cdot | \cdot, z)$ denotes the conditional posterior of a subset of parameters given others, and $p_{\text{sm}}(\xi' | \xi, \eta)$ denotes the transition kernel of the MH-within-Gibbs update for $\xi$, which can be written as

$$p_{\text{sm}}(\xi' | \xi, \eta) = \alpha_{\eta}(\xi, \xi') h(\xi' | \xi) + r_{\eta}(\xi) \delta_{\xi}(\xi'),$$

where $h(\cdot | \cdot)$ is the log-normal proposal kernel for $\xi$, $\alpha_{\eta}(\xi, \xi') = \min \{1, q_{\eta}(\xi, \xi')\}$ is the probability of accepting a move to $\xi'$ from $\xi$,

$$r_{\eta}(\xi) = 1 - \int \alpha_{\eta}(\xi, \xi') h(\xi | \xi') d\xi$$

is the probability of staying at $\xi$, and $\delta_{\xi}(\cdot)$ denotes a point-mass at $\xi$. Noting that the update for $\eta$ remains the same in the approximate algorithm, the approximate Markov kernel $P_\epsilon(x, \cdot)$ has a transition density

$$p_\epsilon(x' | x) = \pi(\eta' | x_{\eta}, z) \pi_\epsilon(\theta' | \xi', \eta, z) p_{\text{sm}, \epsilon}(\xi' | \xi, \eta), \quad x' \in X,$$

where $\pi_\epsilon(\cdot | \xi', \eta, z)$ denotes the approximate full conditional of $\theta = (\beta, \sigma^2)$ resulting from the approximations in steps (2) and (3) in 2.3

and

$$p_{\text{sm}, \epsilon}(\xi' | \xi, \eta) = \alpha_{\eta, \epsilon}(\xi, \xi') h(\xi' | \xi) + r_{\eta, \epsilon}(\xi) \delta_{\xi}(\xi'),$$

is the approximate MH-within-Gibbs transition kernel obtained by the approximation in the acceptance probability $\alpha_{\eta, \epsilon}(\xi, \xi') = \min \{1, q_{\eta, \epsilon}(\xi, \xi')\}$ in step (1) of 2.3.
We now separate to bound the total variation distance between $P(x, \cdot)$ and $P_\epsilon(x, \cdot)$. We have, for a fixed $x \in X$,

$$2\|P(x, \cdot) - P_\epsilon(x, \cdot)\|_{TV} = \int |p(x' | x) - p_\epsilon(x' | x)| dx'$$

$$= \int |\pi(\theta' | \xi', \eta, z) p_{\text{MH}}(\xi' | \xi, \eta) - \pi_\epsilon(\theta' | \xi', \eta, z) p_{\text{MH}, \epsilon}(\xi' | \xi, \eta)| \, dx'_{\theta}$$

$$(\iota) \leq \int \left[ \int |\pi(\theta' | \xi', \eta, z) - \pi_\epsilon(\theta' | \xi', \eta, z)| \, d\theta' \right] p_{\text{MH}}(\xi' | \xi, \eta) \, d\xi' + \int \left[ |p_{\text{MH}}(\xi' | \xi, \eta) - p_{\text{MH}, \epsilon}(\xi' | \xi, \eta)| \right] d\xi'$$

$$(\iota) \leq 2 \sup_{\xi', \eta} \|\pi(\cdot | \xi, \eta, z) - \pi_\epsilon(\cdot | \xi, \eta, z)\|_{TV} + 2 \sup_{\xi', \eta} |\alpha(\xi, \xi') - \alpha_\epsilon(\xi, \xi')|.$$}

For (i), we used triangle inequality and that $\int \pi(\theta' | \xi', \eta, z) \, d\theta' = 1$. For (ii), we used that

$$\int |\pi(\theta' | \xi', \eta, z) - \pi_\epsilon(\theta' | \xi', \eta, z)| \, d\theta' \leq \int |\alpha(\xi, \xi') - \alpha_\epsilon(\xi, \xi')| \, d\xi' + |r(\xi) - r_\epsilon(\xi)|$$

$$\leq 2 \sup_{\xi', \eta} |\alpha(\xi, \xi') - \alpha_\epsilon(\xi, \xi')|.$$

Since the bound in (ii) is independent of $x$, we conclude that

$$\sup_{x \in X} \|P(x, \cdot) - P_\epsilon(x, \cdot)\|_{TV} \leq \sup_{\xi, \eta} \|\pi(\cdot | \xi, \eta, z) - \pi_\epsilon(\cdot | \xi, \eta, z)\|_{TV} + \sup_{\xi', \eta} |\alpha(\xi, \xi') - \alpha_\epsilon(\xi, \xi')| \tag{21}$$

We now separately bound $TV_1$ and $TV_2$. We show that

$$TV_1^2 = 4\|W\|^2 \delta + \frac{N + a_0}{b_0} \|W\|^2 \delta + \frac{N}{2} \|z\|^2 \|W\|^2 \delta + O(\delta^2),$$

$$TV_2 = N \|W\|^2 (1 + \|z\|^2 / b_0) \delta + O(\delta^2),$$

for sufficiently small $\delta$, which produce the desired bound. Since the derivations to obtain these bounds are somewhat lengthy, we split them into two different sections.

### B.1 Bounding $TV_2$: MH ratio approximations for $\xi$

We first record a couple of useful auxiliary results. The first result is a well-known eigenvalue perturbation bound due to Weyl.

**Lemma B.1 (Weyl).** Let $A, E$ be an $n \times n$ Hermitian matrices. Then, for $i = 1, \ldots, n$,

$$|\nu_i(A + E) - \nu_i(A)| \leq \|E\|,$$

where $\nu_i(A)$ denotes the $i$th eigenvalue of $A$, and $\|\cdot\|$ denotes the operator norm of a matrix.

Next, we present a simple yet useful result to bound the difference between MH acceptance probabilities.

**Lemma B.2.** For any $a, b > 0$,

$$|\min(a, 1) - \min(b, 1)| \leq \max \{|(a/b) - 1|, |(b/a) - 1|\} \leq e^{\Delta}|1 - 1,$$

where $\Delta = \log(a/b)$.

**Proof.** First observe that $|\min(a, 1) - \min(b, 1)| \leq |a - b|$, which can be verified by enumerating the 4 different cases (i) $a < b$, (ii) $a < b$, (iii) $b < a$, and (iv) $a > b$. In case (iv), the left hand side is 0 and the claimed bound is trivially satisfied. In the remaining cases, bound

$$|a - b| = |\max(a, b)/\min(a, b) - 1| \min(a, b) \leq \{|\max(a, b)/\min(a, b) - 1| \leq \max \{|(a/b) - 1|, |(b/a) - 1|\}.$$

This proves the first part. The second part simply follows from the monotonicity of $x \mapsto e^x$. \qed
It follows from \[\text{(2.2)}\] that
\[
\alpha_\eta(x, y) = \min\{1, q_\eta(x, y)\}, \quad \alpha_{\eta, c}(x, y) = \min\{1, q_{\eta, \delta}(x, y)\}
\]
with
\[
q_\eta(x, y) = \frac{|M_y|^{-1/2} (b_0 + z'M_y^{-1} z)^{-\frac{(N+\alpha_\eta)}{2}}}{|M_x|^{-1/2} (b_0 + z'M_x^{-1} z)^{-\frac{(N+\alpha_\eta)}{2}}} y\sqrt{x} (1 + x) \frac{1}{x\sqrt{y} (1 + y)},
\]
and \(q_{\eta, \delta}(x, y)\) is obtained by replacing \(M_t\) by \(M_{t, \delta}\), where, recall that
\[
M_t = I_N + t^{-1} WDW', \quad M_{t, \delta} = I_N + t^{-1} WD_{\delta} W', \quad t \in \{x, y\}.
\]
It then follows from Lemma \[\text{(B.2)}\] that
\[
|\alpha_\eta(x, y) - \alpha_{\eta, c}(x, y)| \leq \exp(|\Delta|) - 1,
\]
where
\[
\Delta = \log \frac{q_{\eta, \delta}(x, y)}{q_\eta(x, y)} = \Delta_1 + \Delta_2,
\]
\[
\Delta_1 = \Delta_{1, y} - \Delta_{1, x}, \quad \Delta_1,t = -\frac{1}{2} \left[ \log |M_{t, \delta}| - \log |M_t| \right], \quad t \in \{x, y\},
\]
\[
\Delta_2 = \Delta_{2, y} - \Delta_{2, x}, \quad \Delta_2,t = -\frac{n + 6\alpha}{2} \left[ \log(1 + z'M_{t, \delta}^{-1} z/b_0) - \log(1 + z'M_t^{-1} z/b_0) \right], \quad t \in \{x, y\}.
\]
We shall prove below that
\[
|\Delta| \leq N \|W\|^2 (1 + \|z\|^2/b_0)\delta.
\]
Observe the right hand side is independent of \(\xi\) and \(\eta\).

To establish \[\text{(22)}\], we bound
\[
|\Delta| \leq \sum_{t \in \{x, y\}} [ |\Delta_{1, t}| + |\Delta_{2, t}|].
\]
We now proceed to individually bound \(|\Delta_{1, t}|\) and \(|\Delta_{2, t}|\) for \(t \in \{x, y\}.

For \(t \in \{x, y\}\), we have
\[
|\log |M_t| - \log |M_{t, \delta}|| = \left| \sum_{i=1}^N \left[ \log\{1 + t^{-1} \nu_i(W DW')\} - \log\{1 + t^{-1} \nu_i(W D_{\delta} W')\} \right] \right|
\leq \sum_{i=1}^N \left| \log\{1 + t^{-1} \nu_i(W DW')\} - \log\{1 + t^{-1} \nu_i(W D_{\delta} W')\} \right|
\leq \sum_{i=1}^N |t^{-1} \nu_i(W DW') - t^{-1} \nu_i(W D_{\delta} W')|,
\]
where the last step uses the fact that the map \(u \mapsto \log(1 + u)\) for \(u > 0\) is Lipschitz. Write
\[
t^{-1} WDW' = t^{-1} WD_{\delta} W' + t^{-1} WD_{c, \delta} W',
\]
where \(D_{c, \delta} = \text{diag}\left(\nu_j^{-1}\right) \mathbf{1}(j \in \mathbb{Z}_+^n)\) retains the entries of \(D\) which are thresholded. By Weyl’s perturbation bound (see Lemma \[\text{(B.1)}\]), for any \(i = 1, \ldots, N\),
\[
|t^{-1} \nu_i(W DW') - t^{-1} \nu_i(W D_{\delta} W')| \leq t^{-1} \|W D_{c, \delta} W'\| \leq \delta\|W\|^2,
\]
and
where we use the fact that, given our thresholding rule, all non-zero diagonal entries of the matrix $t^{-1}D_{<\delta}$ is bounded by $\delta$ for $t \in \{x, y\}$. Substituting the bound, we obtain,
\begin{equation}
\sum_{t \in \{x, y\}} |\Delta_{1,t}| \leq N\|W\|^2 \delta. \tag{24}
\end{equation}

Next, we bound $|\Delta_{2,t}|$ for $t \in \{x, y\}$. To that end, one again using that the map $u \mapsto \log(1+u)$ is Lipschitz, bound
\[
|\log(1 + z'M_t^{-1}z/b_0) - \log(1 + z'M_{t,\delta}^{-1}z/b_0)| \\
\leq |z'(M_t^{-1} - M_{t,\delta}^{-1})z/b_0| \\
\leq (\|z\|^2/b_0) \|M_t^{-1} - M_{t,\delta}^{-1}\| \\
\leq (\|z\|^2/b_0) \|M_t^{-1}(M_{t,\delta} - M_t)M_{t,\delta}^{-1}\| \\
\leq (\|z\|^2/b_0) \|M_{t,\delta} - M_t\|,
\]
where we have used the identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$, the bound $\|AB\| \leq \|A\|\|B\|$, and the fact that both $\|M_t^{-1}\|$ and $\|M_{t,\delta}^{-1}\|$ are bounded above by 1. Continuing from the last line of the display, $\|M_t - M_{t,\delta}\| = \|t^{-1}W D_{<\delta}W\| \leq \delta\|W\|^2$ using the same argument as in the bound for $\Delta_{1,t}$. Substituting this bound, we obtain,
\begin{equation}
\sum_{t \in \{x, y\}} |\Delta_{2,t}| \leq (N + a_0) (\|z\|^2/b_0) \|W\|^2 \delta. \tag{25}
\end{equation}

Substituting (24) and (25) in (23), we obtain (22). Now, making a Taylor expansion of $e^x - 1$ about zero, we obtain for $0 < x < 1$
\[e^x - 1 = (1 + x + O(x^2)) - 1 = x + O(x^2),\]
which gives
\begin{equation}
TV_2 = N \|W\|^2 (1 + \|z\|^2/b_0)\delta + O(\delta^2) \tag{26}
\end{equation}
for sufficiently small $\delta$.

**B.2 Bounding $TV_1$**

Define the multivariate normal inverse-gamma (MNIG) distribution to be the joint distribution of $(\beta, \sigma^2) \in \mathbb{R}^p \otimes (0, \infty)$ defined by the hierarchy
\begin{equation}
\beta \mid \sigma^2 \sim N(m, \sigma^2 \Sigma), \quad \sigma^2 \sim IG(a, b). \tag{27}
\end{equation}

We denote the above distribution by MNIG($m, \Sigma, a, b$).

We first record a lemma which calculates the KL divergence between two MNIG distributions with the same shape parameter; a proof is provided in Appendix B.3.1.

**Lemma B.3.** Suppose $p_i \sim$ MNIG($m_i, \Sigma_i, a_i, b_i$) for $i = 0, 1$, with $a_0 = a_1$. Then,
\[
KL(p_0 \mid p_1) = \frac{1}{2} \left[ \text{tr}(\Sigma_1^{-1}\Sigma_0 - I_p) - \log |\Sigma_1^{-1}\Sigma_0| + (\mu_1 - \mu_0)'\Sigma_1^{-1}(\mu_1 - \mu_0) \frac{a_0}{b_0} \right] + a_0 \log(b_0/b_1) + \frac{(b_1 - b_0)a_0}{b_0}.
\]

The full conditional for $(\beta, \sigma^2)$ from the exact algorithm, $\pi(\cdot \mid \xi, \eta, z)$, is distributed as MNIG($\mu, \Sigma, a, b$), with
\begin{align*}
\mu &= \Gamma W'M^{-1}z, \quad \Sigma = (W'W + \Gamma^{-1})^{-1} = \Gamma - \Gamma W'M^{-1}\Phi\Gamma, \\
a &= (N + a_0)/2, \quad b = (z'M^{-1}z + b_0)/2. \tag{28}
\end{align*}
The full conditional for \((\beta, \sigma^2)\) from the approximate algorithm, \(\pi_x(\cdot \mid \xi, \eta, z)\), is distributed as \(\text{MNIG}(\mu_0, \Sigma_0, a_0, b_0)\), with

\[
\begin{align*}
\mu_0 &= \Gamma_0 W' M_0^{-1} z, \\
\Sigma_0 &= \Gamma - (2 \Gamma W' M_0^{-1} W \Gamma - \Gamma_0 W' M_0^{-1} M M_0^{-1} W \Gamma), \\
a_0 &= (N + a_0)/2, \\
b_0 &= (s' M_0^{-1} z + b_0)/2.
\end{align*}
\] (29)

To bound the total variation distance between \(\pi_x(\cdot \mid \xi, \eta, z)\) and \(\pi_x(\cdot \mid \xi, \eta, z)\), we use Pinsker’s inequality,

\[
\|\pi_x(\cdot \mid \xi, \eta, z) - \pi_x(\cdot \mid \xi, \eta, z)\|_{TV} \leq \frac{1}{2} \text{KL}(\pi_x(\cdot \mid \xi, \eta, z) \mid \pi(\cdot \mid \xi, \eta, z)),
\] (30)

and subsequently use the expression for \(\text{KL}\) between two MNIGs derived in Lemma B.3; note that the shape parameters \(a_0 = a = (N + a_0)/2\) and hence the Lemma applies.

Let us define

\[
\begin{align*}
\text{KL}_1 &= \text{tr}(\Sigma^{-1} \Sigma_0 - I_p) - \log |\Sigma^{-1} \Sigma_0|, \\
\text{KL}_2 &= (\mu - \mu_0)' \Sigma^{-1} (\mu - \mu_0) a_0 b_0^{-1}, \\
\text{KL}_3 &= a_0 \log(b_0/b) + (b - b_0) a_0 b_0^{-1}
\end{align*}
\] (31)

so that

\[
\text{KL}(\pi_x(\cdot \mid \xi, \eta, z) \mid \pi(\cdot \mid \xi, \eta, z)) = 0.5(\text{KL}_1 + \text{KL}_2) + \text{KL}_3.
\]

We now proceed to bound each of the terms subsequently.

**B.2.1 Bounds for KL**

The matrix \(\Sigma^{-1} \Sigma_0\) is similar to the positive definite matrix \(\Sigma^{-1/2} \Sigma_0 \Sigma^{-1/2}\), and hence its eigenvalues \(\{\zeta_j\}_{j=1}^p\) are all positive. Expressing the trace and determinant in terms of the eigenvalues, we obtain,

\[
\text{KL}_1 = \sum_{j=1}^p (\zeta_j - 1 - \log \zeta_j).
\] (32)

Now, write

\[
\Sigma_0 = \Sigma + \Delta, \quad \Delta = \Gamma W' M_0^{-1} \Phi \Gamma - (2 \Gamma W' \Gamma_0 W M_0^{-1} M M_0^{-1} W \Gamma),
\]

and

\[
\Sigma^{-1} \Sigma_0 = I_p + \Sigma^{-1} \Delta.
\]

Using \(\text{rank}(B_1 B_2) \leq \min\{\text{rank}(B_1), \text{rank}(B_2)\}\), \(\Delta\) is the difference of two matrices with rank at most \(N\) each, and using \(\text{rank}(B_1 + B_2) \leq \text{rank}(B_1) + \text{rank}(B_2)\), we can bound \(\text{rank}(\Delta) \leq 2N\), which then implies \(\text{rank}(\Sigma^{-1} \Delta) \leq 2N\). Letting \(\{\tilde{\zeta}_j\}_{j=1}^p\) denote the eigenvalues of \(\Sigma^{-1} \Delta\), it then follows that \(\tilde{\zeta}_j = 0\) for \(j \geq 2N\). Since \(\zeta_j = 1 + \tilde{\zeta}_j\), we conclude that \(\zeta_j = 1\) for \(j \geq 2N\), and

\[
\text{KL}_1 = \sum_{j=1}^{2N} (\zeta_j - 1 - \log \zeta_j) = \sum_{j=1}^{2N} [\tilde{\zeta}_j - \log(1 + \tilde{\zeta}_j)].
\] (33)

Observe that the right hand side is a positive quantity, since \(\log(1 + x) \leq x \leq 1\) for \(x > -1\) and \(\tilde{\zeta}_j > -1\) for all \(j\) (since \(\zeta_j > 0\) for all \(j\)). Using Taylor expansion, it can be further shown that \(x - \log(1 + x) < x^2\) whenever \(|x| \leq 1/2\). Using that the magnitude of the eigenvalues of a matrix are bounded by its operator norm, we have \(|\tilde{\zeta}_j| \leq \|\Sigma^{-1} \Delta\|\) for all \(j = 1, \ldots, 2N\). Hence, if we can show that \(\|\Sigma^{-1} \Delta\|\) is small, we can bound

\[
\text{KL}_1 \leq \sum_{j=1}^{2N} |\tilde{\zeta}_j|^2 \leq 2N \|\Sigma^{-1} \Delta\|^2.
\] (34)
With this motivation, we now proceed to bound \(|\Sigma^{-1}\Delta|\). To facilitate our bounds, we decompose
\[
(S_{\delta} - \Sigma) = (S_{\delta} - S_{\star}) + (S_{\star} - \Sigma),
\]
where
\[
S_{\star} = \Gamma - \Gamma W' (2M_{\delta}^{-1} - M_{\delta}^{-1} M M_{\delta}^{-1}) W \Gamma.
\]

\(S_{\star}\) itself is a covariance matrix, although this isn’t used in the subsequent analysis. Letting \(A = M_{\delta}^{-1} M M_{\delta}^{-1}\),
\[
\Gamma \delta W' A W \delta - \Gamma W' A W \Gamma = \Gamma \delta W' A W (\Gamma \delta - \Gamma) + (\Gamma \delta - \Gamma) W' A W \Gamma.
\]

Hence,
\[
S_{\delta} - S_{\star} = \frac{2 \Gamma W' M_{\delta}^{-1} W (\Gamma - \Gamma \delta)}{\bar{r}_{1}} + \frac{\Gamma \delta W' A W (\Gamma \delta - \Gamma)}{\bar{r}_{2}} + \frac{(\Gamma \delta - \Gamma) W' A W \Gamma}{\bar{r}_{3}}.
\]

Recall that \(\Sigma^{-1} = (W' W + \Gamma^{-1})\). Let us now calculate
\[
\Sigma^{-1} (S_{\delta} - S_{\star}) = (W' W + \Gamma^{-1}) (T_{1} + T_{2} + T_{3}) = H_{1} + H_{2} + H_{3},
\]
with
\[
\begin{align*}
H_{1} &= 2 \left[ W' W T W' M_{\delta}^{-1} W (\Gamma - \Gamma \delta) + W' M_{\delta}^{-1} W (\Gamma - \Gamma \delta) \right] \\
H_{2} &= \left[ W' W T \delta W' A W (\Gamma - \Gamma \delta) + \Gamma^{-1} \Gamma \delta W' A W (\Gamma - \Gamma \delta) \right] \\
H_{3} &= \left[ W' W (\Gamma \delta - \Gamma) W' A W \Gamma + \Gamma^{-1} (\Gamma \delta - \Gamma) W' A W \Gamma \right].
\end{align*}
\]

Next,
\[
S_{\star} - S_{\star} = \Gamma W' \left[ M^{-1} + M_{\delta}^{-1} M M_{\delta}^{-1} - 2 M_{\delta}^{-1} \right] W \Gamma.
\]

Hence,
\[
H_{4} := \Sigma^{-1} (S_{\delta} - S_{\star}) = (W' W + \Gamma^{-1}) \Gamma W' E W T. \tag{36}
\]

Combining \((35)\) and \((36)\) and using the triangle inequality for the operator norm,
\[
\|\Sigma^{-1}\Delta\| = \|\Sigma^{-1} (S_{\delta} - S_{\star})\| = \|H_{1} + H_{2} + H_{3} + H_{4}\| \leq \sum_{i=1}^{4} \|H_{i}\|.
\]

We now record a Lemma which collects various results required to bound the operator norms of the \(H_{i}\); a proof is provided in Appendix B.3.2.

Lemma B.4. The following inequalities hold:
(i) \(\max \{\|M^{-1}\|, \|M_{\delta}^{-1}\|\} \leq 1\).
(ii) \(\max \{\|M - M_{\delta}\|, \|M^{-1} M_{\delta} - I_{N}\|, \|M_{\delta} M^{-1} - I_{N}\|, \|M_{\delta}^{-1} M - I_{N}\|, \|M M_{\delta}^{-1} - I_{N}\|\} \leq \|W\|^{2} \delta\).
(iii) \(\max \{\|W T W' M^{-1}\|, \|W T W' M_{\delta}^{-1}\|\} \leq 1\).
(iv) \(\|W T W' M_{\delta}^{-1}\| \leq 1 + \|W\|^{2} \delta\).
(v) Recall that \(A = M_{\delta}^{-1} M M_{\delta}^{-1}\), we have \(\|A\| \leq (1 + \|W\|^{2} \delta)\). Further, \(\|\Gamma \delta W' A\| \leq (1 + \|W\|^{2} \delta)\) and \(\|A W T W'\| \leq (1 + \|W\|^{2})^{2}\).

Using Lemma B.4 we now proceed to bound the \(\|H_{i}\|\); that \(|\Gamma - \Gamma \delta| < \delta\) is used throughout, along with the facts \(\|B_{1} B_{2}\| = \|B_{2} B_{1}\|\) and \(\|B_{1} + B_{2}\| \leq \|B_{1}\| + \|B_{2}\|\).

Bound for \(\|H_{1}\|\). We obtain, using (i) and (iv) in Lemma B.4,
\[
\|H_{1}\| \leq 2 \|W\|^{2} \delta \left[\|W T W' M_{\delta}^{-1}\| + \|M_{\delta}^{-1}\|\right] \leq 2 \|W\|^{2} \delta \left[2 + \|W\|^{2}\delta\right].
\]
Bound for $\|H_2\|$. We obtain, using (v) in Lemma B.4 and the fact that $\|\Gamma^{-1}\Gamma_3\| \leq 1$,
$$\|H_2\| \leq \|W\|^2 \delta \left(\|WT_5 W' A\| + \|A\|\right) \leq \|W\|^2 \delta \left(1 + \|W\|^2 \delta\right).$$

Bound for $\|H_3\|$. We obtain, using (v) in Lemma B.4 and the fact that $\|\Gamma^{-1}\Gamma_3\| \leq 1$,
$$\|H_3\| \leq \|W\|^2 \delta \left(\|AWTW'\| + \|A\|\right) \leq \|W\|^2 \delta \left(1 + \|W\|^2 \delta\right)^2 + \left(1 + \|W\|^2 \delta\right).$$

Bound for $\|H_4\|$. We have,
$$\|H_4\| = \|WT(W'W + \Gamma^{-1})GW' E\|
= \|WTW' (I_N + WTW') E\|
= \|WTW' M E\|.$$
Now, $ME = I_N + MM_3^{-1}MM_3^{-1} - 2MM_3^{-1} = (MM_3^{-1} - I_N)^2$. Substituting in the above display, and once again invoking Lemma B.4,
$$\|H_4\| = \|MM_3^{-1}WTW' (M - M_3)M_3^{-1}(M - M_3)\|
\leq \|WTW' M_3^{-1}\| \|M - M_3\|^2
\leq (1 + \|W\|^2 \delta) (\|W\|^2 \delta)^2.$$

Bound for $\|\Sigma^{-1} \Delta\|$. Collecting the bounds for $\|H_i\|$ and substituting in the display before Lemma B.4 plus some simplifying algebra yields,
$$\|\Sigma^{-1} \Delta\| \leq (\|W\|^2 \delta) \left(3 + 3(1 + \|W\|^2 \delta) + 2(1 + \|W\|^2 \delta)^2\right) = 8\|W\|^2 \delta + O(\delta^2) \quad (37)$$
for sufficiently small $\delta$.

B.2.2 Bound for KL2

Focus first on $(\mu - \mu_\delta)\Sigma^{-1}(\mu - \mu_\delta)$. We have $\mu - \mu_\delta = (\Gamma W' M^{-1} - \Gamma_\delta W'M_3^{-1})z$. Write
$$\Gamma W' M^{-1} - \gamma_\delta W'M_3^{-1} = \left(\Gamma - \Gamma_\delta\right) W'M^{-1} + \Gamma_\delta W'M^{-1} (I_N - MM_3^{-1}).$$

We can now write
$$\left(\mu - \mu_\delta\right)\Sigma^{-1}(\mu - \mu_\delta) = \left(\mu - \mu_\delta\right)\Sigma^{-1}(\mu - \mu_\delta) z
\leq (\|U + V\|^2) \|\Sigma^{-1}(U + V)\| \|z\|^2
\leq (\|U + V\|^2) \|z\|^2
\leq 2(\|U\|^2 \delta + \|V\|^2 \delta) \|z\|^2
= 2(\|U\|^2 \delta + \|V\|^2 \delta) \|z\|^2,$$
where we used the inequality $\|B_1 + B_2\|^2 \leq 2(\|B_1\|^2 + \|B_2\|^2)$. Next,
$$\|U\|^2 \delta = \left\|M^{-1}W(\Gamma - \Gamma_\delta)(W' W + \Gamma^{-1})(\Gamma - \Gamma_\delta) W'M^{-1}\right\|
\leq \|\Gamma - \Gamma_\delta\| \|W' W + \Gamma^{-1}\| \|\Gamma - \Gamma_\delta\| \|W'M^{-2} W\|
\leq \|\Gamma - \Gamma_\delta\| \|W' W + \Gamma^{-1}\| \|W\|^2 \delta
\leq \|W\|^2 \delta (1 + \|W\|^2 \delta) = \|W\|^2 \delta + O(\delta^2),$$
for sufficiently small $\delta$, where we have used conclusions of Lemma B.4 in multiple places and in the last step, we used $\|\Gamma - \Gamma_\delta\| \|\Gamma^{-1}\| \leq 1$ since it is a diagonal matrix with zeros and ones on the diagonal. Similarly, it can be verified that $\|V\|^2 \delta \leq \|W\|^2 \delta (1 + \|W\|^2 \delta)$. So then it follows
$$\text{KL}_2 \leq 2(\|W\|^2 \delta + O(\delta^2)) \frac{N + a_0}{b_0} = 2 \frac{N + a_0}{b_0} \|W\|^2 \delta + O(\delta^2).$$
where the last factor is an upper bound on $a_\delta/b_\delta$ which originates from bounding $b_\delta$ below by $b_0/2$. 

26
B.2.3 Bound for KL₃

Using \(\log(x) \leq (x - 1)\) for \(x > 0\), we have,

\[
KL₃ \leq a₃ \left\{ (b₇/b₆ - 1) + (b/b₃ - 1) \right\}
\leq 2a₃|b - b₆|,
\]

since \(b, b₃ > 1\). Since \(|b - b₆| = |z'(M⁻¹ - M₈⁻1)z|/b₀ \leq (\|z\|^2/b₀) \|W\|^2\delta\), we have

\[
KL₃ \leq N (\|z\|^2/b₀) \|W\|^2\delta.
\]

B.2.4 Summing up

We now combine the bounds to obtain the final result. We have

\[
KL₁ + KL₂ + KL₃ \leq 8\|W\|^2\delta + 2N + a₀/b₀ \|W\|^2\delta + N (\|z\|^2/b₀) \|W\|^2\delta + O(\delta^2)
\]

so by Pinsker’s inequality

\[
TV₁^2 \leq 4\|W\|^2\delta + N + a₀/b₀ \|W\|^2\delta + N \frac{\|z\|^2}{b₀} \|W\|^2\delta + O(\delta^2)
\]

and so finally, combining with \(\boxed{26}\) – which contributes only factors of order \(\delta^2\) or smaller after squaring – via \(\boxed{21}\), we obtain

\[
\sup_x |\delta_x P - \delta_x Pᵩ|_{TV} = \sqrt{\frac{4\|W\|^2\delta + N + a₀/b₀ \|W\|^2\delta + N \frac{\|z\|^2}{b₀} \|W\|^2\delta + O(\delta)}{N + a₀/b₀}}.
\]

since none of the bounds depend upon the remaining state variable \(\eta\), giving the result.

B.3 Proof of auxiliary lemmata

B.3.1 Proof of Lemma B.3

We have

\[
\int p₀(\beta, \sigma^2) \log \frac{p₀(\beta, \sigma^2)}{π₁(\beta, \sigma^2)} d\beta d\sigma^2
= \int p₀(\beta | \sigma^2) p₀(\sigma^2) \left[ \log \frac{p₀(\beta | \sigma^2)}{π₁(\beta | \sigma^2)} + \log \frac{p₀(\sigma^2)}{π₁(\sigma^2)} \right] d\beta d\sigma^2
= \int KL\left( p₀(\cdot | \sigma^2) || p₁(\cdot | \sigma^2) \right) p₀(\sigma^2) dσ^2 + \int p₀(\sigma^2) \log \frac{p₀(\sigma^2)}{π₁(\sigma^2)} dσ^2.
\]

First, using normality of \(p₁(\cdot | \sigma^2)\) and the standard expression for the KL divergence between two multivariate normals,

\[
KL\left( p₀(\cdot | \sigma^2) || p₁(\cdot | \sigma^2) \right) = \frac{1}{2} \left[ \text{tr}(Σ⁻¹₁Σ₀ - Iₚ) - \log |Σ⁻¹₁Σ₀| + \frac{(μ₁ - μ₀)Σ⁻¹₁(μ₁ - μ₀)}{σ^2} \right].
\]

Thus,

\[
\int KL\left( p₀(\cdot | \sigma^2) || p₁(\cdot | \sigma^2) \right) p₀(\sigma^2) dσ^2
= \frac{1}{2} \left[ \text{tr}(Σ⁻¹₁Σ₀ - Iₚ) - \log |Σ⁻¹₁Σ₀| + \frac{(μ₁ - μ₀)Σ⁻¹₁(μ₁ - μ₀)a₀}{b₀} \right].
\]
Next, using that $a_0 = a_1$,
\[
\int p_0(\sigma^2) \log \frac{p_0(\sigma^2)}{p_1(\sigma^2)} \, d\sigma^2 = \int p_0(\sigma^2) [ (a_0 \log b_0 - a_0 \log b_1) + (b_1 - b_0)(\sigma^2)^{-1} ] \, d\sigma^2 = a_0 \log(b_0/b_1) + \frac{(b_1 - b_0)a_0}{b_0}.
\]

### B.3.2 Proof of Lemma B.4

We make multiple usage of the following facts. For matrices $A$ and $B$ of compatible size, $\|AB\| \leq \|A\|\|B\|$ and $\|A + B\| \leq \|A\| + \|B\|$. For invertible $A$, $\|A^{-1}\| = 1/s_{\min}(A)$. For a symmetric p.d. matrix $A$, its eigenvalues and singular values are identical.

(i) Follows since $s_{\min}(M)$ and $s_{\min}(M_\delta)$ are both bounded below by 1.

(ii) First, $\|M - M_\delta\| = \|W(W^{-1}\Gamma_\delta)W\| \leq \|W\|^2\delta$ since $\Gamma_\delta - \Gamma_\delta$ is a diagonal matrix with the non-zero entries bounded by $\delta$. The remaining 4 inequalities have near identical proofs so we only prove one of them. We have $\|M_\delta^{-1} - I_N\| = \|M_\delta^{-1}(M_\delta - M)\| \leq \|M - M_\delta\|$ by (i).

(iii) Writing $WTW' M^{-1} = I_N - M^{-1}$, all its eigenvalues are bounded above by 1. Similarly for the second part.

(iv) Bound $\|WTW'M^{-1}\| \leq \|WTW'M^{-1}\| + \|WTW^\prime M^{-1}(I_N - MM_\delta^{-1})\| \leq \|WTW'M^{-1}\| (1 + \|I_N - MM_\delta^{-1}\|)$. Conclude from (ii) and (iii).

(v) First, bound $\|A\| \leq \|M_\delta^{-1}M\| \|M_\delta^{-1}M_\delta^{-1}\| \|M_\delta^{-1}\|$. The bound then follows from (i) and (ii) and noting that $(M_\delta^{-1}M_\delta^{-1}) = M_\delta^{-1}(M_\delta - M_\delta)M_\delta^{-1}$. For the third bound, write $\|AWTW'\| = \|M!^{-1}_{\delta}MM_\delta^{-1}WTW'\| \leq \|M_\delta^{-1}M\| \|M_\delta^{-1}WTW\|$. The bound then follows from (ii) and (iii). The bound for $\|WTW'A\|$ follows similarly.

### C Derivation of MCMC algorithm

In this section we derive the update rule for the exact algorithm.

#### C.1 Derivation $p(\xi \mid \eta, z)$

To find the update for $p(\xi \mid \eta, z)$, first note that the half-Cauchy prior on $\xi^{-1/2}$ induces the prior
\[
p(\xi) = \frac{1}{1 + \xi^{-1}} = \frac{1}{\sqrt{\xi(1 + \xi)}}
\]
on $\xi$, with the term $\xi^{-3/2}$ resulting from the change of variables. Then
\[
p(\xi \mid \eta, z) = \int p(\xi, \sigma^2 \mid \eta, z) \, d\sigma^2 \propto \int p(z \mid \eta, \xi, \sigma^2) p(\sigma^2) \, d\sigma^2 \propto p(\xi) \int p(z \mid \eta, \xi, \sigma^2) (2\pi)^{-n/2} \sigma^{-a_0/2-1} e^{-b_0/(2\sigma^2)} \, d\sigma^2
\]
\[
= \frac{1}{\sqrt{\xi(1 + \xi)}} \left| M_\xi \right| \xi^{-1/2} \int (2\pi)^{-n/2} \sigma^{-a_0/2-1} e^{-\frac{1}{2\sigma^2} \left( (z' M_\xi^{-1} z + b_0) \right)} \, d\sigma^2.
\]

#### C.2 Derivation $p(\sigma^2 \mid \xi, \eta, z)$

Note we can write the model as
\[
\beta = \varepsilon_1, \quad \varepsilon_1 \sim N(0, \sigma^2 \xi^{-1} D)
\]
\[ z = W\beta + \varepsilon_2, \quad \varepsilon_2 \sim N(0, \sigma^2 I), \]

so

\[ z = W\varepsilon_1 + \varepsilon_2 \sim N(0, \sigma^2 M_\xi). \]

Thus,

\[
p(\sigma^2 | \xi, \eta, z) \propto \frac{1}{(\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2} (\varepsilon^T M^{-1}_\xi z)} \frac{1}{(\sigma^2)^{a_0/2}} e^{-b_0/(2\sigma^2)} \]

\[
\sigma^2 | \xi, \eta, z \sim \text{InvGamma} \left( \frac{n + a_0}{2} - \frac{1}{2} \left[ \varepsilon^T M_\xi^{-1} z + b_0 \right] \right) \]

### C.3 Remaining full conditionals

The conditional \( \beta | \sigma^2, \xi, \eta \) as given in [4] as

\[
\beta | \sigma^2, \xi, \eta \sim N \left( (W'W + \xi^{-1} D)^{-1} W'z, \sigma^2 (W'W + \xi^{-1} D)^{-1} \right),
\]

and we sample \( \eta | \xi, \beta, \sigma^2 \) exactly as in [26] Section 2.2, Supplement.

### D Extra Figures

Here we provide additional figures relevant to the statistical performance of time-averaging estimators from the Approximate algorithm.

![Figure 11](image-url)

**Figure 11:** Marginals for the residual standard deviation \( \sigma \) over 20 values of \( N, p \) using the approximate algorithm. The small horizontal lines indicate the 0.025 and 0.975 approximate posterior quantiles. The true value is 2 in all cases.
Figure 12: Marginals for the first 25 entries of $\beta$ for $N = 1000, p = 5000$, true value indicated with red line. Approximate algorithm.
Figure 13: Marginals for the first 25 entries of $\beta$ for $N = 5000, p = 50,000$, true value indicated with red line. Approximate algorithm.
References

[1] Amestoy, P. R., Davis, T. A., and Duff, I. S. (2004). Algorithm 837: AMD, an approximate minimum degree ordering algorithm. ACM Trans. Math. Softw., 30(3):381–388.

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

[3] Bhadra, A., Datta, J., Polson, N. G., and Willard, B. T. (2017). Lasso meets horseshoe. arXiv preprint arXiv:1706.10179.

[4] Bhattacharya, A., Chakraborty, A., and Mallick, B. K. (2016). Fast sampling with Gaussian scale mixture priors in high-dimensional regression. Biometrika, 103(4):985–991.

[5] Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015). Dirichlet–Laplace priors for optimal shrinkage. Journal of the American Statistical Association, 110(512):1479–1490.

[6] Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The horseshoe estimator for sparse signals. Biometrika, 97(2):465–480.

[7] Datta, J. and Ghosh, J. K. (2013). Asymptotic properties of Bayes risk for the horseshoe prior. Bayesian Analysis, 8(1):111–132.

[8] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. The Annals of statistics, 32(2):407–499.

[9] Flegal, J. M. and Jones, G. L. (2010). Batch means and spectral variance estimators in Markov chain Monte Carlo. The Annals of Statistics, 38(2):1034–1070.

[10] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22.

[11] George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. Statistica sinica, 7:339–373.

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

[13] Hahn, P. R., He, J., and Lopes, H. F. (2017). Efficient sampling for gaussian linear regression with arbitrary priors. Technical Report, https://math.la.asu.edu/prhahn/slicepaper.pdf.

[14] Hans, C. (2009). Bayesian lasso regression. Biometrika, 96(4):835–845.

[15] Johndrow, J. E. and Mattingly, J. C. (2017). Error bounds for approximations of Markov chains. arXiv preprint arXiv:1711.05382.

[16] Johndrow, J. E., Mattingly, J. C., Mukherjee, S., and Dunson, D. (2015). Approximations of Markov chains and Bayesian inference. arXiv preprint arXiv:1508.03387.

[17] Johnson, L. T. and Geyer, C. J. (2012). Variable transformation to obtain geometric ergodicity in the random-walk Metropolis algorithm. The Annals of Statistics, 40(6):3050–3076.

[18] Johnson, V. E. and Rossell, D. (2012). Bayesian model selection in high-dimensional settings. Journal of the American Statistical Association, 107(498):649–660.

[19] Jones, G. L. (2004). On the Markov chain central limit theorem. Probability surveys, 1:299–320.

[20] Khare, K. and Hobert, J. P. (2013). Geometric ergodicity of the Bayesian lasso. Electronic Journal of Statistics, 7:2150–2163.

[21] Komorowski, T. and Walczuk, A. (2012). Central limit theorem for Markov processes with spectral gap in the Wasserstein metric. Stochastic Processes and their Applications, 122(5):2155–2184.
[22] Liu, X., Huang, M., Fan, B., Buckler, E. S., and Zhang, Z. (2016). Iterative usage of fixed and random effect models for powerful and efficient genome-wide association studies. *PLoS genetics*, 12(2):e1005767.

[23] Pal, S. and Khare, K. (2014). Geometric ergodicity for Bayesian shrinkage models. *Electronic Journal of Statistics*, 8(1):604–645.

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

[25] Polson, N. G. and Scott, J. G. (2010). Shrink globally, act locally: Sparse Bayesian regularization and prediction. In *Bayesian Statistics*, volume 9, pages 501–538. Oxford University Press.

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

[27] Rajaratnam, B. and Sparks, D. (2015). MCMC-based inference in the era of big data: A fundamental analysis of the convergence complexity of high-dimensional chains. *arXiv preprint arXiv:1508.00947*.

[28] Rajaratnam, B., Sparks, D., Khare, K., and Zhang, L. (2017). Scalable Bayesian shrinkage and uncertainty quantification in high-dimensional regression. *arXiv preprint arXiv:1703.09163*.

[29] Roberts, G. and Rosenthal, J. (1997). Geometric ergodicity and hybrid Markov chains. *Electronic Communications in Probability*, 2:13–25.

[30] Romay, M. C., Millard, M. J., Glaubitz, J. C., Peiffer, J. A., Swarts, K. L., Casstevens, T. M., Elshire, R. J., Acharya, C. B., Mitchell, S. E., Flint-Garcia, S. A., et al. (2013). Comprehensive genotyping of the USA national maize inbred seed bank. *Genome Biology*, 14(6):R55.

[31] Rudolf, D. (2011). Explicit error bounds for Markov chain Monte Carlo. *arXiv preprint arXiv:1108.3201*.

[32] Rudolf, D. and Schweizer, N. (2018). Perturbation theory for Markov chains via Wasserstein distance. *Bernoulli*, 24(4):2610–2639.

[33] Scott, J. G. and Berger, J. O. (2010). Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, 38(5):2587–2619.

[34] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[35] van der Pas, S., Kleijn, B., and van der Vaart, A. (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8(2):2585–2618.

[36] van der Pas, S., Szabó, B., and van der Vaart, A. (2017a). Adaptive posterior contraction rates for the horseshoe. *Electronic Journal of Statistics*, 11(2):3196–3225.

[37] van der Pas, S., Szabó, B., and van der Vaart, A. (2017b). Uncertainty quantification for the horseshoe (with discussion). *Bayesian Analysis*, 12(4):1221–1274.

[38] Zeng, P. and Zhou, X. (2017). Non-parametric genetic prediction of complex traits with latent dirichlet process regression models. *Nature communications*, 8(1):456.

[39] Zhang, Y., Reich, B. J., and Bondell, H. D. (2016). High dimensional linear regression via the R2-D2 shrinkage prior. *arXiv preprint arXiv:1609.00046*.