Scalable Spike-and-Slab

Niloy Biswas 1   Lester Mackey 2   Xiao-Li Meng 1

Abstract

Spike-and-slab priors are commonly used for Bayesian variable selection, due to their interpretability and favorable statistical properties. However, existing samplers for spike-and-slab posteriors incur prohibitive computational costs when the number of variables is large. In this article, we propose Scalable Spike-and-Slab (S3), a scalable Gibbs sampling implementation for high-dimensional Bayesian regression with the continuous spike-and-slab prior of George & McCulloch (1993). For a dataset with \( n \) observations and \( p \) covariates, \( S^3 \) has order \( \max\{n^2p, np\} \) computational cost at iteration \( t \) where \( p_t \) never exceeds the number of covariates switching spike-and-slab states between iterations \( t \) and \( t-1 \) of the Markov chain. This improves upon the order \( n^2p \) per-iteration cost of state-of-the-art implementations as, typically, \( p_t \) is substantially smaller than \( p \). We apply \( S^3 \) on synthetic and real-world datasets, demonstrating orders of magnitude speed-ups over existing exact samplers and significant gains in inferential quality over approximate samplers with comparable cost.

1. Introduction

1.1. Bayesian computation in high dimensions

We consider linear, logistic, and probit regression in high dimensions, where the number of observations \( n \) is smaller than the number of covariates \( p \). This setting is common in modern applications such as genome-wide association studies (Guan & Stephens, 2011; Zhou et al., 2013) and astronomy (Kelly, 2007; Sereno, 2015). In the non-Bayesian paradigm, sparse point estimates such as the LASSO (Tibshirani, 1996), Elastic Net (Zou & Hastie, 2005) and SLOPE (Bogdan et al., 2015) offer a route to variable selection. These estimates are based on optimization based approaches, which are computationally efficient and scale to datasets with hundreds of thousands of covariates.

In the Bayesian paradigm, which will be our focus, one places a prior on the unknown parameters of interest and considers the corresponding posterior distribution. Sampling algorithms such as Markov chain Monte Carlo (MCMC) are then used to simulate from the posterior distribution. In modern high dimensional settings, general-purpose MCMC algorithms can have high computational cost per iteration. This has kindled a line of work on tailored algorithms for Bayesian regression (e.g., Polson et al., 2013; Yang et al., 2016; Narisetty et al., 2019; Johndrow et al., 2020; Biswas et al., 2022). Our manuscript participates in this wider effort to scale Bayesian inference to large data applications. Specifically, we propose computationally efficient MCMC algorithms for high-dimensional Bayesian linear, logistic and probit regression with spike-and-slab priors.

1.2. Variable selection with spike-and-slab priors

Consider Gaussian linear regression, logistic regression, and probit regression with \( n \) observations and \( p \) covariates. The respective likelihoods are given by

\[
L_{\text{lin}}(\beta, \sigma^2; X, y) = \frac{1}{(2\pi \sigma^2)^{n/2}} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 \right),
\]

\[
L_{\text{logit}}(\beta; X, y) = \prod_{i=1}^{n} \frac{\exp(y_i x_i^\top \beta)}{1 + \exp(x_i^\top \beta)},
\]

\[
L_{\text{prob}}(\beta; X, y) = \prod_{i=1}^{n} \Phi(x_i^\top \beta)^{y_i} (1 - \Phi(x_i^\top \beta))^{1-y_i}.
\]

Here \( X \in \mathbb{R}^{n \times p} \) is the design matrix with rows \( x_i \), \( y \in \mathbb{R}^n \) (for linear regression) or \( y \in \{0, 1\}^n \) (for logistic and probit regression) is the response vector, \( \beta \in \mathbb{R}^p \) is the unknown signal, \( \sigma^2 \in (0, \infty) \) is the unknown Gaussian noise variance, and \( \Phi \) is the cumulative density function of \( \mathcal{N}(0, 1) \).

We focus on the high-dimensional setting with \( n \ll p \), where \( \beta \in \mathbb{R}^p \) is assumed to be sparse. We use a continuous spike-and-slab prior on \( \beta \) to capture sparsity:

\[
\sigma^2 \sim \text{InvGamma}\left(\frac{a_0}{2}, \frac{b_0}{2}\right), \quad \text{(for linear regression)}
\]

\[
\sigma^2 = 1, \quad \text{(for logistic and probit regression)}
\]

\[
z_j \sim \text{Bernoulli}(q),
\]

\[
\beta_j | z_j, \sigma^2 \sim \text{Normal}(0, \sigma^2 z_j^2), \quad (1 - z_j) \mathcal{N}(0, \sigma^2 \tau^2_0) + z_j \mathcal{N}(0, \sigma^2 \tau^2_1), \quad (1)
\]

---

1Department of Statistics, Harvard University 2Microsoft Research New England. Correspondence to: Niloy Biswas <niloy.biswas@g.harvard.edu>.
where $q \in (0, 1)$, $\tau^2 > 0$, and $a_0, b_0 > 0$ are hyper-parameters. Here, $\mathcal{N}(0, \sigma^2 \tau^2_0)$ and $\mathcal{N}(0, \sigma^2 \tau^2_1)$ correspond to the spike and slab parts of the prior respectively. In the high-dimensional setting, a small constant $q \ll 1$ is often chosen, but the algorithms in this manuscript also readily extend to hierarchical variants of (1) with a hyperprior placed on $q$ (Scott & Berger, 2010; Castillo & van der Vaart, 2012).

Catalyzed by the works of George & McCulloch (1993; 1997), continuous spike-and-slab priors are now a mainstay which overcomes existing computational bottlenecks by employing certain correlated matrices (Barbieri & Berger, 2004; Tadesse & Vannucci (2021, Section I) and Banerjee et al. (2021)). The posterior probabilities tend to hierarchical variants of (1) with a hyperprior placed on $\tau$ (Scott & Berger, 2010; Castillo & van der Vaart, 2012) but can be computational cost and is given in Algorithm 1. However, this cost remains prohibitive for large $p$.

Section 2 introduces Scalable Spike-and-Slab (S3), a computationally efficient implementation of Gibbs samplers for linear and logistic regression with the prior given in (1). Section 2.1 investigates the computational bottlenecks of state-of-the-art (SOTA) implementations, which require $\Omega(n^2 p)$ computational cost per iteration for datasets with $n$ observations and $p$ covariates. Section 2.2 develops S3, which overcomes existing computational bottlenecks by employing a pre-computation based strategy and requires $\mathcal{O}(\text{max}(n^2 p n^p, n^p))$ computational cost at iteration $t$, where $p_t$ is no greater than the number of covariates switching spike-and-slab states between iterations $t$ and $t-1$ of the Markov chain. Section 2.3 analyzes the computational complexity of S3, showing that $p_t$ is typically much smaller than $p$ and that it can remain constant and even approach zero under various limiting regimes as $p$ increases.

Section 3 compares S3 with the SOTA exact MCMC sampler and a recently proposed approximate MCMC sampler, which does not converge to the posterior distribution of interest. We demonstrate that S3 offers substantially faster numerical runtimes compared to the SOTA exact MCMC sampler, reporting $50 \times$ speedups on synthetic datasets. In the same experiment, S3 and the approximate sampler have comparable runtimes, but the asymptotically exact S3 procedure provides more accurate variable selection.

Section 4 demonstrates the benefits of S3 on a diverse suite of datasets, including two synthetic datasets and eight real-world experiments. For example, on a genome-wide association study (GWAS) dataset with with $n \approx 2000$ and $p \approx 100000$, we again observe $50 \times$ computational speedups over the SOTA exact MCMC sampler. Finally, Section 5 discusses directions for future work. The open-source packages ScaleSpikeSlab in R and Python (www.github.com/niloyb/ScaleSpikeSlab) implement our methods and recreate the experiments in this paper.

2. Scalable Spike-and-Slab

2.1. Status quo and computational bottlenecks

Gibbs samplers have long been employed to sample from the posterior distributions corresponding to the prior in (1) (e.g., George & McCulloch, 1993; O’Brien & Dunson, 2004; Held & Holmes, 2006; Polson et al., 2013). The computational bottleneck of existing Gibbs samplers is linked to sampling from the full conditional of $\beta \in \mathbb{R}^p$. This is given by

$$\beta_{t+1} | z, \sigma^2_t \sim \mathcal{N}(\Sigma^{-1}_t X^\top y, \sigma^2_t \Sigma^{-1}_t)$$

for $\Sigma_t = X^\top X + D_t$, where $t$ indexes the iteration of the Markov chain, and $D_t$ is the diagonal matrix with the vector $z_t \tau^{-2}_t + (1_p - z_t) \tau^{-2}_0$ populating its diagonal elements.

Sampling from (2) using standard matrix multiplication and a generic Cholesky decomposition that ignores the specific structure of $\Sigma_t$ requires $\Omega(p^3)$ computational cost, which quickly becomes prohibitive for large $p$. Hereafter we will refer to this generic method as the Naive Sampler. Ishwaran & Rao (2005) recommend separating the components of $\beta$ into $B$ blocks of size $n/b$ each, and then updating each block using Gibbs sampling, which gives a reduced $\Omega(p^3 B^{-2})$ computational cost. However, this cost remains prohibitive for large $p$, and using a larger number of blocks $B$ induces higher auto-correlation between successive iterations of the Gibbs sampler. Recently, Bhattacharya et al. (2016) developed an algorithm based on the Woodbury matrix identity (Hager, 1989) to sample from multivariate Gaussian distributions of the form in (2), which requires a more favorable $\Omega(n^2 p)$ computational cost and is given in Algorithm 1.

1.3. Our contributions

Our contributions are summarized below. Throughout, we use $\mathcal{O}$ and $\Omega$ to respectively denote asymptotic upper and lower bounds on computational complexity growth rates.

Section 2 introduces Scalable Spike-and-Slab (S3), a computationally efficient implementation of Gibbs samplers for linear and logistic regression with the prior given in (1).
We now develop \( \tilde{D} \) the number of switches; \( \{ t_0, \ldots, t_{\tau} \} \) index set of covariates which switch spike-and-slab states \( \{ \tilde{a}_t \} \). Current algorithms calculate \( \beta \) cost. Appendix B contains derivations and details of Algorithms 2 and 3 (the implementation of logistic regression is based on a scaled \( t \)-distribution approximation to the logistic distribution, as commonly done in the literature; see (Narisetty et al., 2019)).

For large-scale datasets with \( n \) in the thousands and \( p \) in the hundreds of thousands, as found in modern scientific applications, the \( \Omega(n^3p) \) cost per iteration is still too high. This has spurred recent work on approximate MCMC (Narisetty et al., 2019) for the continuous spike-and-slab prior on logistic regression and on variational inference methods for the point-mass spike-and-slab prior (Titiaş & Lázaro-Gredilla, 2011; Ray et al., 2020; Ray & Szabó, 2021). Such approximate samplers can provide improved computational speeds but do not converge to the posterior distribution of interest.

2.2. A scalable Gibbs sampler

We now develop \( S^3 \). Our key insight is that successive pre-computation can be used to reduce the computational cost of Algorithm 1. In Algorithm 1, the \( \Omega(n^3p) \) computational cost per iteration arises from the calculation of the matrix \( M_t \). Current algorithms calculate \( M_t = I_n + XD_t^{-1}X^T \) from scratch every iteration at \( \Omega(n^2p) \) cost under standard matrix multiplication and then solve an \( n \) by \( n \) linear system to obtain \( \nu^* \) at \( \Omega(n^3) \) cost. We propose instead to use the previous state \( z_{t-1} \) and the pre-computed matrices \( M_{t-1} \) and \( M_{t-1}^{-1} \) at each step to aid the calculation of \( M_t^{-1} \). Our strategy is given below.

Specifically, denote \( X = [X_1, \ldots, X_p] \in \mathbb{R}^{n \times p} \) and its submatrices \( X_A \triangleq [X_j : j \in A] \in \mathbb{R}^{n \times |A|} \) and \( X_{A_t} \triangleq [X_j : j \in A^I_t] \in \mathbb{R}^{n \times (p-|A|)} \), where \( A \triangleq \{ j : z_{j,t-1} = 1 \} \) and \( A^I_t \triangleq \{ j : z_{j,t} = 0 \} \), are the ordered index sets of covariates corresponding to slab states and to spike states respectively at iteration \( t \), and \( \| \cdot \|_1 \) is the \( L^1 \) norm on \( \{ 0, 1 \}^p \). Also denote \( \Delta_t \triangleq \{ j : z_{j,t} \neq z_{j,t-1} \} \), the ordered index set of covariates which switch spike-and-slab states between iterations \( t \) and \( t-1 \); \( \delta_t \triangleq \| z_t - z_{t-1} \|_1 = |\Delta_t| \), the number of switches; \( D_{\Delta_t} \triangleq \text{Diag}( (D_j)_{j \in \Delta_t} ) \in \text{Diag}(\mathbb{R}^{p \times |\Delta_t|}) \), the diagonal sub-matrix of \( D_t \) composed of the diagonal entries with ordered indices in \( \Delta_t \); and \( C_{\Delta_t} \triangleq D_{\Delta_t}^{-1} - D_{\Delta_t}^{-1} \). Finally, let \( \tilde{M}_t \triangleq I_n + \tau_t^2 X X^T \in \mathbb{R}^{n \times n} \) and \( \tilde{M}_t \triangleq I_n + \tau_t^2 X X^T \in \mathbb{R}^{n \times n} \), which are fixed for all iterations. Under this notation, there are three expressions for \( M_t \):

\[
M_t = \tilde{M}_t + (\tau_t^2 - \tau_0^2) X A_t X_{A_t}^T, \quad \text{(3a)}
\]

\[
\tilde{M}_t + (\tau_t^2 - \tau_0^2) X A_t X_{A_t}^T, \quad \text{(3b)}
\]

\[
M_{t-1} + X_{\Delta_t} C_{\Delta_t} X_{\Delta_t}^T, \quad \text{(3c)}
\]

In (3a) – (3c), calculating the matrix products \( X_A X_{A_t}^T \) and \( X_{\Delta_t} C_{\Delta_t} X_{\Delta_t}^T \) requires \( \Theta(n^2z_t, \| z_t \|_1) \), \( \Theta(n^3(p-\| z_t \|_1)) \), and \( \Theta(n^2\delta_t) \) cost respectively. Given \( \tilde{M}_{t_0}, \tilde{M}_{t_1}, M_{t-1}, \) and \( z_{t-1} \), we evaluate whichever matrix product in (3a) – (3c) has minimal computational cost and thereby calculate \( M_t \) at the reduced cost of \( \Theta(n^2p_t) \) where \( p_t \triangleq \min \{ \| z_t \|_1, p - \| z_t \|_1 \} \).

To calculate \( M_t^{-1} \), we consider the cases \( n \leq p_t \) and \( p_t < n \) separately. When \( n \leq p_t \), we calculate \( M_t^{-1} \) by directly inverting the calculated matrix \( M_t \) from (3), which requires \( \Theta(n^3) \) cost. When \( p_t < n \), we apply the Woodbury matrix identity on (3). This gives

\[
M_t^{-1} = \tilde{M}_t^{-1} - \tilde{M}_t^{-1} X A_t (\frac{1}{\tau_t^2 - \tau_0^2} I - \tau_t^2 X_{A_t} X_{A_t}^T)^{-1} X_{A_t}^T \tilde{M}_t^{-1}, \quad \text{(4a)}
\]

\[
= \tilde{M}_t^{-1} - \tilde{M}_t^{-1} X A_t (\frac{1}{\tau_t^2 - \tau_0^2} I - \tau_t^2 X_{A_t} X_{A_t}^T)^{-1} X_{A_t}^T \tilde{M}_t^{-1}, \quad \text{(4b)}
\]

\[
- \tilde{M}_t^{-1} X A_t (\frac{1}{\tau_t^2 - \tau_0^2} I - \tau_t^2 X_{A_t} X_{A_t}^T)^{-1} X_{A_t}^T \tilde{M}_t^{-1}, \quad \text{(4c)}
\]

Given \( \tilde{M}_{t_0}, \tilde{M}_{t_1}, M_{t-1}^{-1} \) and \( z_{t-1} \), we evaluate whichever expression in (4a) – (4c) has minimal computational cost to calculate \( M_t^{-1} \). Similar to (3), this requires \( \Theta(n^2p_t) \) computational cost, which arises from matrix inversion and multiplication.

Overall, this strategy of using the previous state \( z_{t-1} \) and the pre-computed matrices \( \tilde{M}_{t_0}, \tilde{M}_{t_1}, M_{t-1}^{-1} \) and \( M_{t-1}^{-1} \), reduces the computational cost of calculating the matrices \( M_t \) and \( M_{t-1}^{-1} \) from \( \Omega(n^2p) \) (as in all current implementations of Algorithm 1) to \( \Theta(n^2p_t) \). As we show in Sections 2.3 and 3, in many large-scale applications \( p_t \) is orders of magnitude smaller than both \( n \) and \( p \), yielding substantial improvements in computational efficiency. Furthermore, we emphasize that the matrices \( \tilde{M}_{t_0}, \tilde{M}_{t_1}, M_{t-1}^{-1} \) are fixed for all iterations, and the state \( z_{t-1} \) and matrices \( M_{t-1}^{-1} \) and \( M_{t-1}^{-1} \) only need to be stored temporarily to generate samples for iteration \( t \) and can be deleted after. Therefore \( S^3 \) requires minimal additional memory compared to current implementations.

The full Gibbs samplers for Bayesian linear, logistic, and probit regression which make use of this pre-computation are given in Algorithms 2 and 3. The Gibbs samplers for logistic and probit regression are based on data augmentation strategies (see, e.g., O’Brien & Dunson (2004); Narisetty et al. (2019)), and the Gibbs sampler for logistic regression requires an adjusted pre-computation strategy with \( \Theta(\max\{ n^2p_t, n^3, np \}) \) cost. Appendix B contains derivations and details of Algorithms 2 and 3. (The implementation of logistic regression is based on a scaled \( t \)-distribution approximation to the logistic distribution, as commonly done in the literature; see (Narisetty et al., 2019)).
Algorithm 2 Bayesian linear regression with S3

Input: State \( C_t \triangleq (\beta_t, z_t, \sigma_t^2) \in \mathbb{R}^p \times \{0, 1\}^p \times (0, \infty) \), state \( z_{t-1} \), and matrices \( M_{t-1}, M_{t-1}^{-1} \).

1. Calculate \( p_t \) and use (3) to calculate \( M_t \).
   if \( p_t \geq n \) then invert \( M_t \) to calculate \( M_t^{-1} \) else use
   (4) to calculate \( M_t^{-1} \).
2. Sample \( \beta_{t+1}|z_t, \sigma_t^2 \) using Algorithm 1 from \( \mathcal{N}(\Sigma_t^{-1}X^T y, \sigma_t^2 \Sigma_t^{-1}) \) for \( \Sigma_t = X^TX + D_t \).
3. Sample each \( z_{j,t+1}|\beta_{t+1}, \sigma_t^2 \) independently from
   \[ \text{Bernoulli}(q_{\mathcal{N}(\beta_{t+1};0, \sigma_t^2 \Sigma_t^{-1})}|(1-q)\mathcal{N}(\beta_{t+1};0, \sigma_t^2 \Sigma_t^{-1})) \]
   for \( j = 1, \ldots, p \).
4. Sample \( \sigma_{t+1}^2|\beta_{t+1}, z_{t+1} \) from
   \[ \text{InvGamma}(\frac{a_0+n+1}{2}, b_0 + y X \beta_{t+1} + \frac{1}{2} X D_{t-1} \beta_{t+1}) \].

Output: \( C_{t+1} = (\beta_{t+1}, z_{t+1}, \sigma_{t+1}^2) \), \( z_t \), \( M_t \), \( M_t^{-1} \).

Algorithm 3 Bayesian logistic & probit regression with S3

Input: State \( C_t \triangleq (\beta_t, z_t, \gamma_t, \tilde{\sigma}_t^2) \in \mathbb{R}^p \times \{0, 1\}^p \times \mathbb{R}^n \times (0, \infty) \), states \( z_{t-1} \), \( \tilde{\sigma}_{t-1}^2 \), and matrices \( M_{t-1}, M_{t-1}^{-1} \).

1. Logistic regression: Use pre-computation (see Appendix B.3) to calculate \( M_t \triangleq I_n + W_t^{-1/2} X D_t^{-1} X^T W_t^{-1/2} \) for \( W_t = \text{Diag}(\tilde{\sigma}_t^2) \).
   Invert \( M_t \) to calculate \( M_t^{-1} \).
2. Probit regression: Calculate \( p_t \) and use (3) to calculate \( M_t \triangleq I_n + X D_t^{-1} X^T \).
   if \( p_t \geq n \) then invert \( M_t \) to calculate \( M_t^{-1} \) else use
   (4) to calculate \( M_t^{-1} \).
3. Sample \( \beta_{t+1}|z_t, \gamma_t, \tilde{\sigma}_t^2 \) using Algorithm 1 from \( \mathcal{N}(\Sigma_t^{-1}X^T y, \tilde{\sigma}_t^2 \Sigma_t^{-1}) \) for \( \Sigma_t = X^TX^{-1}X+D_t \).
4. Sample each \( z_{j,t+1}|\beta_{t+1}, \gamma_t, \tilde{\sigma}_t^2 \) independently from
   \[ \text{Bernoulli}(q_{\mathcal{N}(\beta_{t+1};0, \tilde{\sigma}_t^2 \Sigma_t^{-1})}|(1-q)\mathcal{N}(\beta_{t+1};0, \tilde{\sigma}_t^2 \Sigma_t^{-1})) \]
   for \( j = 1, \ldots, p \).
5. Logistic regression: Sample each \( \tilde{\sigma}_{i,t+1}^2|\beta_{t+1}, z_{i,t+1}, \gamma_t \) independently from
   \[ \text{InvGamma}(\frac{a_0+n+1}{2}, \frac{y^2 + x_i^T \beta_{t+1}}{2}) \]
   for \( i = 1, \ldots, n \) where \( \nu \triangleq 7.3 \) and \( w_i^2 \triangleq \pi^2(\nu - 2)/(3\nu) \)
   are constants.
6. Probit regression: Set each \( \tilde{\sigma}_{i,t+1}^2 = 1 \) for \( i = 1, \ldots, n \).

Output: \( C_{t+1} = (\beta_{t+1}, z_{t+1}, \gamma_t, \tilde{\sigma}_{t+1}^2) \), \( z_t \), \( M_t \), \( M_t^{-1} \).

2.3. Analysis of computational complexity

We now investigate the favorable computational complexity of Algorithms 2 and 3. Proposition 2.1, proved in Appendix A, gives the computational cost of these Gibbs samplers for linear and logistic regression, showing an improvement over existing implementations which have \( \Omega(n^2p) \) cost.

Proposition 2.1 (Computational cost). Algorithm 2 and Algorithm 3 for probit regression both have a computational cost of \( \mathcal{O}(\max\{n^2p, n^3\}) \) at iteration \( t \), and Algorithm 3 for logistic regression has a computational cost of \( \mathcal{O}(\max\{n^2p, n^3\}) \) at iteration \( t \), where \( p_t = \min\{||z_t||_1, p - ||z_t||_1, \delta_t\} \) for \( \delta_t = ||z_t - z_{t-1}||_1 \).

In Proposition 2.1, \( ||z_t||_1 \) and \( p - ||z_t||_1 \) are the number of slab covariates and the number of spike covariates respectively at iteration \( t \) of the Markov chain, and \( \delta_t \) is the number of covariates switching spike-and-slab states between iterations \( t \) and \( t - 1 \) of the Markov chain. Note that \( p_t \leq \min\{||z_t||_1, p - ||z_t||_1\} \leq p/2 \) directly. In practice, there are a variety of scenarios under which \( p_t \) is significantly smaller than \( p/2 \).

Sparse \( z_t \). Whenever \( z_t \) is sparse relative to the full dimensionality \( p \), we have \( p_t \leq ||z_t||_1 \ll p \). Sparsity in \( z_t \) is a common occurrence in high-dimensional regression with a sparse signal vector \( \beta^* \in \mathbb{R}^p \), as \( ||z_t||_1 \) often closely approximates the true sparsity \( s \triangleq ||\beta^*||_0 \) with high probability. This occurs, for instance, in the settings of Narisetty & He (2014) and Narisetty et al. (2019), where strong model selection consistency of the continuous spike-and-slab posterior is established for linear and logistic regression respectively.

Posterior concentration. Even when \( \min\{||z_t||_1, p - ||z_t||_1\} \) is comparable to \( p/2 \), concentration of the spike-and-slab posterior targeted by the Gibbs sampler can lead to \( \delta_t \) and hence \( p_t \) remaining much smaller than \( p/2 \). Proposition 2.2, proved in Appendix A, calculates the expectation of \( \delta_t \) explicitly in terms of the posterior distribution that is targeted by our algorithms and the auto-correlation of the states \( (z_t)_{t\geq0} \).

Proposition 2.2 (Expected spike-and-slab swap count). For \( \delta_t \) as given in Proposition 2.1,

\[
\mathbb{E}[\delta_t] = \sum_{j=1}^{p} \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 0) + \mathbb{P}(z_{j,t} = 0)\mathbb{P}(z_{j,t-1} = 1) - 2\text{cov}(z_{j,t}, z_{j,t-1}).
\]

Suppose the Markov chain generated by Algorithm 2 or 3 is at its stationary distribution \( \pi \) at iteration \( t - 1 \). Then,

\[
\mathbb{E}[\delta_t] = 2\sum_{j=1}^{p} \text{var}_\pi(z_{j,t})(1 - \text{corr}_\pi(z_{j,t}, z_{j,t-1})).
\]
In (6), note \( \text{var}_\pi (z_{j,t}) = \mathbb{P}_\pi (z_{j,t} = 0) \mathbb{P}_\pi (z_{j,t} = 1) \leq 1/4 \) for each component \( j \), with equality only when \( z_{j,t} \sim \text{Bernoulli}(1/2) \). Therefore all components \( j \) with \( \mathbb{P}_\pi (z_{j,t} = 1) \) close to 0 or 1 do not contribute significantly towards \( \delta_t \) in expectation. Such posterior concentration is guaranteed whenever \( z_t \) is convergent, be it to the true model selection vector as in Narisetty & He (2014) and Narisetty et al. (2019) or to any other value. In such circumstances we can have \( \delta_t = \| z_t - z_{t-1} \|_1 = o(p) \) and even \( \delta_t = o(\|z_t\|_1) \), regardless of the magnitude of \( \|z_t\|_1 \).

**High auto-correlation.** High auto-correlation of the Gibbs sampler can also lead to smaller values of \( \delta_t \) and hence \( p_t \). We already see from Proposition 2.2 that, even for components \( j \) with bimodal marginal posterior distributions, high auto-correlation between successive states \( z_{j,t-1} \) and \( z_{j,t} \) can yield lower \( \delta_t \) in expectation. Proposition 2.3, proved in Appendix A, provides an additional exact expression for \( \delta_t \) in terms of the empirical correlation between \( z_t \) and \( z_{t-1} \).

**Proposition 2.3 (Swap count decomposition).** Let \( \tau_t = \sqrt{\|z_t\|^1_1 (p - \|z_t\|_1)} \) and \( p_t \) be the empirical correlation between \( z_t \) and \( z_{t-1} \) (that is, the correlation between \( z_{j,t} \) and \( z_{j,t-1} \) when \( J \) is uniform on \( \{1, \ldots, p\} \)). Then, for \( \delta_t \) as given in Proposition 2.1,

\[
\delta_t = \|z_t\|_1 + \|z_{t-1}\|_1 - 2\|z_t\|_1 \|z_{t-1}\|_1 + 2p_t \tau_t \tau_{t-1}.
\]

Since \( |p_t| \leq 1 \), Proposition 2.3 implies

\[
\delta_t \geq \frac{(\|z_t\|_1 - \|z_{t-1}\|_1)^2}{p} + \frac{(\tau_t - \tau_{t-1})^2}{p} \quad \text{and} \quad (8)
\]

\[
\delta_t \leq \frac{(\|z_t\|_1 - \|z_{t-1}\|_1)^2}{p} + \frac{(\tau_t + \tau_{t-1})^2}{p}. \quad (9)
\]

The lower bound in (8) is a good approximation to \( \delta_t \) when \( p_t \) is close to one, which is the case either when the Gibbs sampler is converging (such that \( z_t \) becomes stable) or when it gets stuck (such that \( z_t \) changes slowly with \( t \)). In either case, \( \tau_t \) exhibits similar “stable/stuck” behavior, implying that the lower bound itself will be close to zero. This suggests \( \delta_t \) and hence \( p_t \) is close to zero when \( p_t \) is close to 1, even if \( \min\{\|z_t\|_1, p - \|z_t\|_1\} \) is not negligible.

Motivated by such discussions and theoretical analysis, we now empirically examine how \( p_t \) grows as the number of observations \( n \), the number of covariates \( p \), and the sparsity \( s \) of the true signal varies. Figure 1 is based on synthetic linear regression datasets. For each dataset, one Markov chain is generated using Algorithm 2 to target the corresponding spike-and-slab posterior, from which we sample 10,000 iterations and take every 5th iteration. For Figure 1 (Left), we consider datasets with \( n = 100 \), varying \( p \) with \( p \geq n \), and a sparse true signal \( \beta^* \in \mathbb{R}^p \) with components \( \beta^*_j = 0 \) for sparsity \( s = 10 \), and \( \sigma^2 = 2 \). For Figure 1 (Center), we consider datasets with \( p = 1000 \), varying \( n \) with \( n \leq p \), \( s = 10 \), and \( \sigma^2 = 2 \). For Figure 1 (Right), we consider datasets with \( n = 10s \), \( p = 1000 \), and \( \sigma^2 = 2 \) for varying \( s \geq 1 \). Details of the synthetically generated datasets are in Appendix D.

Figure 1 (Left) shows that both \( p_t \) is substantially smaller than both \( p \) and \( n \) and that it does not increase with the number of covariates \( p \). Figure 1 (Center) shows that \( p_t \) tends to zero as \( n \) increases. Figure 1 (Right) shows that \( p_t \) decreases as the sparsity \( s \) increases. All figures suggest that \( p_t \) is controlled by \( \delta_t \) in these settings, because \( \|z_t\|_1 \) takes values close to \( s \), and \( p - \|z_t\|_1 \) tends to be much larger than \( \|z_t\|_1 \). Overall, Figure 1 highlights that not only does \( p_t \) tend to be substantially smaller than \( p \), but it also tends to be smaller than both \( n \) and \( s \). By Proposition 2.1, this shows the substantially lower computational cost of \( S^3 \) compared to current implementations which cost \( O(n^2 p) \) per iteration.

### 2.4 Extensions to Scalable Spike-and-Slab

With additional memory capacity and pre-computation, we can further improve the per-iteration costs of \( S^3 \).

For the matrices \( \tilde{M}_{\tau_1} \) and \( \tilde{M}_{\tau_2} \) in Section 2.2, suppose the matrices \( X^\top X \), \( X^\top \tilde{M}_{\tau_0}^{-1} X \), and \( X^\top \tilde{M}_{\tau_1}^{-1} X \) are pre-computed. This initial step requires \( O(np^2) \) computational cost and \( O(p^2) \) memory. Then the matrices \( X_{A_t}^\top X_{A_t} \) in (3a) – (3b) correspond to pre-computed sub-matrices of \( X^\top X \), and calculating \( M_t \) using (3a) – (3b) at each iteration involves matrix addition which only requires \( O(n^2) \) cost. Similarly, matrices \( X_{A_t}^\top \tilde{M}_{\tau_0}^{-1} X_{A_t} \) and
Table 1. Comparison of $S^3$ with alternatives ($p_t = \min\{\|z_t\|_1, p - \|z_t\|_1, \delta_t\}$ for $\delta_t = \|z_t - z_{t-1}\|_1$.

<table>
<thead>
<tr>
<th>MCMC Sampler</th>
<th>Cost</th>
<th>Converges to Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve</td>
<td>$\Omega(n^2p)$</td>
<td>✓</td>
</tr>
<tr>
<td>State-of-the-art</td>
<td>$\Omega(n^2p)$</td>
<td>✓</td>
</tr>
<tr>
<td>Skinny Gibbs</td>
<td>$\Omega(nnz_j, np)\setminus n$</td>
<td>✓</td>
</tr>
<tr>
<td>$S^3$ (Linear and Probit)</td>
<td>$O(\max{n^2p, np})$</td>
<td>✓</td>
</tr>
<tr>
<td>$S^3$ (Logistic)</td>
<td>$O(\max{n^2p, n^3, np})$</td>
<td>✓</td>
</tr>
</tbody>
</table>

$X^\top A_t^i \tilde{M}^{-1} A_t^i X^\top$ in (4a)–(4b) correspond to pre-computed sub-matrices of $X^\top \tilde{M}^{-1} X$ and $X^\top \tilde{M}^{-1} X$ respectively and do not need to be recalculated at iteration $t$. Therefore calculating $((\tau_0^2 - \tau_t^2)^{-1} I_{\|z_t\|_1} + X^\top A_i \tilde{M}^{-1} X A_i)^{-1}$ or $((\tau_0^2 - \tau_t^2)^{-1} I_{p - \|z_t\|_1} + X^\top A_i \tilde{M}^{-1} X A_i)^{-1}$ in (4a)–(4b) at each iteration $t$ only requires $O(\|z_t\|_1^3)$ or $O((p - \|z_t\|_1)^3)$ cost respectively.

To sample from (2), consider the cases $n < \min\{\|z_t\|_1, p - \|z_t\|_1\}$ and $\min\{\|z_t\|_1, p - \|z_t\|_1\} = n$ separately. When $n < \min\{\|z_t\|_1, p - \|z_t\|_1\}$, we calculate $M^{-1}_t$ by directly inverting the calculated matrix $M_t$ from (3), which requires $O(n^3)$ cost. When $\min\{\|z_t\|_1, p - \|z_t\|_1\} = n$, we avoid calculating $M^{-1}_t$ explicitly and instead calculate the matrix vector product $M^{-1}_t(\frac{1}{\tau_t}y - v)$ in Algorithm 1 right-to-left, using whichever expression in (4a)–(4b) has minimal computational cost. Overall, the Gibbs samplers for linear and probit regression require only $O(\max\{\min\{\|z_t\|_1, p - \|z_t\|_1\}, n^3, np\})$ computational cost at iteration $t$. This provides lower computational cost for $S^3$ linear and probit regression whenever $\min\{\|z_t\|_1, p - \|z_t\|_1\} \geq n^2p_t$. A similar extension for logistic regression requires only $O(\max\{n^3, np\})$ computational cost at iteration $t$ and is given in Appendix B.3.

### 3. Comparison with Alternatives

In this section we compare $S^3$ with the naïve sampler, the SOTA exact MCMC sampler based on the sampling algorithm of Bhattacharya et al. (2016), and the Skinny Gibbs approximate MCMC sampler of Narisetty et al. (2019) for logistic regression. Table 1 highlights the favorable computational cost of $S^3$ compared to the naïve and SOTA samplers. The Skinny Gibbs sampler typically has lower computational cost compared to $S^3$ for logistic regression, and can have lower or higher computational cost than $S^3$ for probit regression depending on whether $\|z_t\|_1^2 \leq np_t$ or not. However, unlike $S^3$, the Skinny Gibbs sampler does not converge to the correct posterior distribution.

To assess the practical impact of computational cost and asymptotic bias, Figures 2 and 3 compares the numerical runtimes and statistical performance of $S^3$ with the SOTA sampler and the Skinny Gibbs sampler. For the Skinny Gibbs sampler, we use the skinnybase package of Narisetty et al. (2019), which implements only logistic regression. We consider synthetically generated datasets with a true signal $\beta^* \in \mathbb{R}^p$ where $\beta_j^* = 2I\{j \leq s\}$ for sparsity $s$. We consider datasets with $n = 10^4$ observations and $p = 100$ covariates for varying sparsity $s \geq 1$. Details of the synthetically generated dataset are in Appendix D. For each synthetic dataset, we run $S^3$ for logistic and probit regression and the Skinny Gibbs sampler for 1000 iterations, run the SOTA sampler for 100 iterations, and record the average time taken per iteration. All timings were obtained using a single core of an Apple M1 chip on a Macbook Air 2020 laptop with 16 GB RAM.

Figure 2 highlights that the numerical runtimes of $S^3$ are orders of magnitude faster than the SOTA sampler and comparable to the Skinny Gibbs sampler. For example, for $n = 4000$ observations, $p = 4000$ covariates, and sparsity $s = 400$, $S^3$ for logistic regression requires 21500ms per iteration on average, which is approximately 15 times faster than the SOTA sampler for probit regression (which requires 335000ms per iteration on average) and 2.5 times faster than the Skinny Gibbs sampler (which requires 55600ms per iteration on average), and $S^3$ for probit regression requires 11000ms per iteration on average, which is approximately 50 times faster than the SOTA sampler for probit regression (which requires 55800ms per iteration on average). For larger real-life datasets with hundreds of thousands of covariates, the numerical runtimes of $S^3$ are similarly favorable compared to the SOTA sampler. This is showcased in Section 4, where for a genetics dataset, $S^3$ is 50 times faster than the SOTA sampler.

Figure 3 plots the true positive rate (TPR) and the false discovery rate (FDR) of variable selection based on samples...
Figure 3. Average true positive rate (TPR) and false discovery rate (FDR) of S^3 and the Skinny Gibbs approximate sampler (Narisetty et al., 2019) across 20 independently generated datasets with one standard error bars. See Section 3 for details.

Figure 4 (Left) plots the marginal posterior probabilities π_j = P(z_j = 1) are estimated by \( \hat{\pi}_j = \frac{1}{I} \sum_{i=1}^{I} \sum_{t=s+1}^{T} z_{j,t} \), where \( z_{j,t} \geq 0 \) are samples from \( i = 1, ..., I \) independent Markov chains generated using S^3. We sample \( I = 5 \) independent chains of length \( T = 5000 \) iterations with a burn-in of 1000 iterations for both S^3 and the SOTA sampler. The average runtime per iteration with one standard error bars are calculated based on these independent chains.

Figure 4 (Right) plots \( \hat{\pi}_j \) against \( j \) in the decreasing order of \( \hat{\pi}_j \). It shows \( \hat{\pi}_j \) based on samples from both S^3 and the SOTA sampler. We simulate both S^3 and the SOTA sampler with the same random numbers at each iteration, so that any differences will be due to numerical imprecision. Figure 4 (Right) shows that the estimates using S^3 and the SOTA sampler are indistinguishable. Furthermore, in this example all components of \( z_j \) are identical between S^3 and the SOTA sampler chains for all iterations \( t \). Despite producing Markov chains with indistinguishable marginal distributions and hence statistical properties, Figure 4 (Right) shows that S^3 has approximately 20 and 6 times faster runtime per iteration than SOTA for logistic and probit regression respectively. Furthermore, S^3 for logistic regression has approximately 100 times faster runtime per iteration than the Skinny Gibbs sampler. Overall, Figure 4 highlights the practical value of S^3 over the SOTA sampler and the Skinny Gibbs sampler.

Table 2. Synthetic and real-life datasets considered in Section 4.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>n</th>
<th>p</th>
<th>RESPONSE TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOROVECKI</td>
<td>31</td>
<td>22283</td>
<td>BINARY</td>
</tr>
<tr>
<td>CHIN</td>
<td>118</td>
<td>22215</td>
<td>BINARY</td>
</tr>
<tr>
<td>CHOWDARY</td>
<td>104</td>
<td>22282</td>
<td>BINARY</td>
</tr>
<tr>
<td>GORDON</td>
<td>181</td>
<td>12533</td>
<td>BINARY</td>
</tr>
<tr>
<td>LYMPH</td>
<td>148</td>
<td>4514</td>
<td>BINARY</td>
</tr>
<tr>
<td>MAIZE</td>
<td>2266</td>
<td>98385</td>
<td>CONTINUOUS</td>
</tr>
<tr>
<td>MALWARE</td>
<td>373</td>
<td>503</td>
<td>BINARY</td>
</tr>
<tr>
<td>PCR</td>
<td>60</td>
<td>22575</td>
<td>CONTINUOUS</td>
</tr>
<tr>
<td>SYNTHETIC BINARY</td>
<td>1000</td>
<td>50000</td>
<td>CONTINUOUS</td>
</tr>
<tr>
<td>SYNTHETIC CONTINUOUS</td>
<td>1000</td>
<td>50000</td>
<td>BINARY</td>
</tr>
</tbody>
</table>

4. Applications

We now examine the benefits of S^3 on a diverse suite of regression and binary classification datasets. Table 2 summarizes the two synthetic and eight real-world datasets considered, with further details in Appendix D.

We first consider the Gordon microarray dataset (Gordon et al., 2002) with \( n = 181 \) observations (corresponding to a binary response vector indicating presence of lung cancer) and \( p = 12533 \) covariates (corresponding to genes expression levels). Figure 4 shows the marginal posterior probabilities estimated using samples from S^3 and the SOTA sampler for logistic and probit regression and the Skinny Gibbs sampler for logistic regression, as well as the corresponding average runtimes per iteration. The marginal posterior probabilities \( \pi_j = P(z_j = 1) \) are estimated by \( \hat{\pi}_j = \frac{1}{I} \sum_{i=1}^{I} \sum_{t=s+1}^{T} z_{j,t} \). We sample \( I = 5 \) independent chains of length \( T = 5000 \) iterations with a burn-in of 1000 iterations for both S^3 and the SOTA sampler. The average runtime per iteration with one standard error bars are calculated based on these independent chains.

Overall, Figures 2 and 3 highlight that S^3 can have comparable or even favorable computational cost to the Skinny Gibbs sampler, whilst having the correct stationary distribution and more favorable statistical properties in higher dimensions. Appendix E contains additional simulation results showcasing S^3 performance for individual datasets as the chain length and the total time elapsed varies.
probit regression on ten regression and binary classification datasets respectively. It plots the average runtimes with one standard error bars based on 10 independent chains each of length 1000 and 100 for S^3 and the SOTA sampler respectively. Figure 5 shows that S^3 has lower runtimes per iteration compared to the SOTA sampler for all the datasets considered, with the most substantial speedups for larger datasets. For example, for the Maize GWAS dataset (Romay et al., 2013; Liu et al., 2016; Zeng & Zhou, 2017) with n = 2266 observations (corresponding to average number of days taken for silk emergence in different maize lines) and p = 98385 covariates (corresponding to single nucleotide polymorphisms (SNPs) in the genome), S^3 requires 650 ms per iteration on average, which is 48 times faster than the SOTA sampler requiring 31300 ms per iteration. For researchers, such speedups can reduce algorithm runtime from days to hours, giving substantial time and computational cost savings at no compromise to inferential quality. Appendix E contains additional results of Section 4 applied to the ten continuous and binary response datasets. See Section 4 for details.

5. Further Work

The following questions arise from our work.

Extensions of S^3 to point-mass spike-and-slab priors, as well as to non-Gaussian tails. Whilst priors given in (1) are one of the most common formulations employed in practice, a number of alternatives are available. This includes point-mass spike-and-slab priors (e.g., Mitchell & Beauchamp, 1988; Johnson & Rossell, 2012), where a degenerate Dirac distribution about zero is chosen for the spike part, and extensions which consider the heavier-tailed Laplace distribution for the slab part instead of a Gaussian distribution (Castillo et al., 2015; Ročková, 2018; Ray et al., 2020; Ray & Szabó, 2021). An extension of S^3 would to be employ similar pre-computation based strategy of Section 2 to MCMC samplers for these alternative formulations.

Convergence complexity analysis of S^3. An important question that is not addressed in this article is the number of iterations required for S^3 or other similar samplers to converge to the their target posterior distributions. For Gibbs samplers targeting posteriors corresponding to continuous shrinkage priors (e.g., Carvalho et al., 2010; Bhattacharya et al., 2015; Bhadra et al., 2019), much theoretical progress has been made (Pal & Khare, 2014; Qin & Hobert, 2019; Bhattacharya et al., 2022; Biswas et al., 2022). Convergence of Gibbs samplers targeting spike-and-slab posteriors has been less extensively studied and requires more attention.

Diagnostics to assess the convergence of and asymptotic variance of S^3. Given some time and computational budget constraints, an immediate benefit of S^3 is the ability to run longer Markov chains targeting spike-and-slab posteriors. This can allievate some concerns linked to burn-in and asymptotic variance, but convergence and effective sample size diagnostics (Johnson, 1998; Biswas et al., 2019; Vats & Knudson, 2021; Vehtari et al., 2021) remain an important consideration particularly in high-dimensional settings. We hope to investigate convergence diagnostics in future work.

Figure 4. Comparing Bayesian logistic and probit regression with S^3, SOTA, and Skinny Gibbs on the Gordon microarray dataset with n = 181 observations and p = 12533 covariates. (Left and Middle) Marginal posterior probabilities P(\pi_j = 1) estimated using samples from each chain: the recovered S^3 and SOTA probabilities are indistinguishable but differ significantly from the Skinny Gibbs probabilities. (Right) Average runtime per sampler iteration with one standard error bars. See Section 4 for details.

Figure 5. Average runtime per iteration with one standard error bars for S^3 and the SOTA sampler for linear and probit regression applied to the ten continuous and binary response datasets. See Section 4 for details.
Acknowledgments. We thank Juan Shen for sharing the PCR and the Lymph Node datasets, as well as Xiaolei Liu and Xiang Zhou for sharing the Maize GWAS dataset. NB was supported by the NSF grant DMS-1844695, a GSAS Merit Fellowship, and a Two Sigma Fellowship Award. XLM was partially supported by the NSF grant DMS-1811308.

References


A. Proofs

Proof of Proposition 2.1. Consider Step 1 of Algorithm 2 and Algorithm 3 for probit regression. Given pre-computed matrices $M_{n_0}$, $\tilde{M}_{n_0}$, $M_{n_1}$, $M_{n_1}^{-1}$ and state $z_{t-1}$, calculating $M_t$ requires $O(n^3p_t)$ cost by (3) and (4), where $p_t = \min\{\|z_t\|_1, p - \|z_t\|_1, \delta_t\}$ for $\delta_t = \|z_t - z_{t-1}\|_1$.

Consider Step 1 of Algorithm 3 for logistic regression. This requires $O(\max\{n^2p_t, n^3\})$ cost, where the $O(n^2p_t)$ cost arises from the calculation of $M_t$ using (21) and the $O(n^3)$ cost arises from inverting $M_t^{-1}$.

Given $M_t^{-1}$, Step 2 of Algorithms 2 and 3 then requires $O(np_t)$ cost, which arises from the matrix vector product $XD_z^{-\frac{1}{2}}r$ for $r \sim N(0, I_p)$ in Algorithm 1. By component-wise independence, Step 3 of Algorithms 2 and 3 costs $O(p)$ and Step 4 of Algorithm 3 cost $O(n)$. Step 4 of Algorithm 3 and Step 5 of Algorithm 3 for probit regression both cost $O(1)$. Step 5 of Algorithm 3 for logistic regression both costs $O(n)$.

This gives an overall computational cost of $O(\max\{n^2p_t, np_t\})$ for Algorithm 2 and Algorithm 3 for probit regression at iteration $t$, and a cost of $O(\max\{n^2p_t, n^3, np_t\})$ for 3 for logistic regression. 

Proof of Proposition 2.2. By linearity, $\mathbb{E}[\delta_t] = \sum_{j=1}^{p} \mathbb{P}(z_{j,t} \neq z_{j,t-1})$ where for each component $z_{j,t}$ the random variables $z_{j,t}$ and $z_{j,t-1}$ are in $\{0, 1\}$. For each $j$, we obtain

$$
\mathbb{P}(z_{j,t} \neq z_{j,t-1}) = \mathbb{P}(z_{j,t} = 1, z_{j,t-1} = 0) + \mathbb{P}(z_{j,t} = 0, z_{j,t-1} = 1) 
= (\mathbb{P}(z_{j,t} = 1) - \mathbb{P}(z_{j,t} = 1, z_{j,t-1} = 1)) + (\mathbb{P}(z_{j,t-1} = 1) - \mathbb{P}(z_{j,t} = 1, z_{j,t-1} = 1))
= \mathbb{P}(z_{j,t} = 1) - \mathbb{P}(z_{j,t-1} = 1) - \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 1) + \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 1) - \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 1) - \mathbb{P}(z_{j,t} = 0)\mathbb{P}(z_{j,t-1} = 1).
$$

When $z_{j,t-1}$ follows the stationary $\pi$, $z_{j,t} \sim z_{j,t-1}$ and var$(z_{j,t}) = \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 0)$. Consequently,

$$
\mathbb{P}(z_{j,t} \neq z_{j,t-1}) = 2\text{var}_\pi(z_{j,t}) - \text{cov}(z_{j,t}, z_{j,t-1}) = 2\text{var}_\pi(z_{j,t})(1 - \text{corr}_\pi(z_{j,t}, z_{j,t-1})).
$$

Proof of Proposition 2.3. Note that $a^2 = a$ if $a$ only takes the value 0 and 1. This gives

$$
\mathbb{I}\{z_{j,t} \neq z_{j,t-1}\} = (z_{j,t} - z_{j,t-1})^2 = z_{j,t} + z_{j,t-1} - 2z_{j,t}z_{j,t-1}\tag{10}
$$

Let $J$ be the uniform random variable on the integers $\{1, \ldots, p\}$. Then,

$$
\delta_t = p(\mathbb{E}_J(z_{J,t}) + \mathbb{E}_J(z_{J,t-1}) - 2\mathbb{E}_J(z_{J,t}z_{J,t-1}))\tag{11}
$$

where the expectation is taken with respect to the random index $J$. Note that $\mathbb{E}_J(z_{J,t}) = \|z_t\|_1/p$ and $\text{Var}_J(z_{J,t}) = (\|z_t\|_1/p)(1 - \|z_t\|_1/p) = \tau_t/p^2$. We obtain

$$
\mathbb{E}_J(z_{J,t}z_{J,t-1}) = \text{Cov}_J(z_{J,t}, z_{J,t-1}) + \mathbb{E}_J(z_{J,t})\mathbb{E}_J(z_{J,t-1}) = \rho_t\tau_t - \|z_t\|_1\|z_{t-1}\|_1/p^2, \tag{12}
$$

where $\rho_t = \text{corr}_J(z_{J,t}, z_{J,t-1})$. Combining (11)-(12) yields (7).

B. Algorithm Derivations

B.1. Linear regression with spike-and-slab priors

For linear regression with the continuous spike-and-slab priors in (1), the posterior density of $(\beta, z, \sigma^2) \in \mathbb{R}^p \times \{0, 1\}^p \times (0, \infty)$ is given by

$$
\pi(\beta, z, \sigma^2 | y) \propto \mathcal{N}(y; X\beta, \sigma^2) \text{InvGamma}(\sigma^2; \frac{a_0}{2}, \frac{b_0}{2}) \prod_{j=1}^{p} \left(q \mathcal{N}(\beta_j; 0, \sigma^2\tau_j^2)\right)^{z_j} \left((1 - q) \mathcal{N}(\beta_j; 0, \sigma^2\tau_0^2)\right)^{1 - z_j}. \tag{13}
$$

$$
\prod_{j=1}^{p} \left(q \mathcal{N}(\beta_j; 0, \sigma^2\tau_j^2)\right)^{z_j} \left((1 - q) \mathcal{N}(\beta_j; 0, \sigma^2\tau_0^2)\right)^{1 - z_j}. \tag{14}
$$
From (13), we can calculate the conditional distributions. We have

\[
\pi(\beta|z, \sigma^2, y) \propto N(y; X\beta, \sigma^2 I_n) \prod_{j=1}^{p} N(\beta_j; 0, \sigma^2 \tau_1^2)\ N(\beta_j; 0, \sigma^2 \tau_0^2)^{1-z_j} \\
\propto N(y; X\beta, \sigma^2 I_n) N(\beta; 0, \sigma^2 D^{-1}) \text{ for } D \triangleq \text{Diag}(z\tau_1^{-2} + (1-p-z)\tau_0^{-2}) \\
\propto \exp \left\{ -\frac{1}{2\sigma^2 I_n} (\beta^T X^T X \beta - 2\beta^T X^T y + \beta^T D\beta) \right\} \\
\propto N(\beta; \Sigma^{-1} X^T y, \sigma^2 \Sigma^{-1}) \text{ for } \Sigma = X^T X + D, \\
\pi(z|\beta, \sigma^2, y) \propto \prod_{j=1}^{p} \left( qN(\beta_j; 0, \sigma^2 \tau_1^2) \right)^{z_j} \left( (1-q)N(\beta_j; 0, \sigma^2 \tau_0^2) \right)^{1-z_j} \\
\propto \prod_{j=1}^{p} \text{Bernoulli} \left( z_j; \frac{qN(\beta_j; 0, \sigma^2 \tau_1^2)}{qN(\beta_j; 0, \sigma^2 \tau_1^2) + (1-q)N(\beta_j; 0, \sigma^2 \tau_0^2)} \right), \
\pi(\sigma^2|\beta, z, y) \propto N(y; X\beta, \sigma^2) \text{InvGamma} \left( \sigma^2; \frac{a_0}{2}, \frac{b_0}{2} \right) N(\beta; 0, \sigma^2 D^{-1}) \\times \frac{1}{\sigma^2} \exp \left\{ -\frac{1}{2\sigma^2} \|y - X\beta\|_2^2 \right\} \frac{1}{\sigma^2} \exp \left\{ -\frac{1}{2\sigma^2} \beta^T D\beta \right\} \\
\propto \text{InvGamma} \left( \sigma^2; \frac{a_0 + n + p}{2}, \frac{b_0 + \|y - X\beta\|_2^2 + \beta^T D\beta}{2} \right)
\]
as given in Algorithm 2.

B.2. Probit regression with spike-and-slab priors

Consider the probit regression likelihood, where for each observation \( i = 1, ..., n \), \( P(y_i = 1|x_i, \beta) = 1 - P(y_i = 0|x_i, \beta) = \Phi(x_i^T \beta) \) for \( x_i \) the \( i \)-th row of the design matrix \( X \), and \( \Phi \) that cumulative density function of a univariate Normal distribution. We obtain \( y_i = I\{\tilde{y}_i > 0\} \) for \( \tilde{y}_i \sim N(x_i^T \beta, 1) \). The Bayesian probit regression model is then given by

\[
\begin{align*}
  z_j &\sim \text{Bernoulli}(q) \quad \text{for all } j = 1, ..., p \\
  \beta_j|z_j &\sim (1-z_j)N(0, \tau_0^2) + z_jN(0, \tau_1^2) \quad \text{for all } j = 1, ..., p \\
  \tilde{y}_i &\sim N(x_i^T \beta, 1) \quad \text{for all } i = 1, ..., n \\
  y_i &= \{\tilde{y}_i > 0\} \quad \text{for all } i = 1, ..., n.
\end{align*}
\]

For the prior and likelihood in (15), the posterior density of \((\beta, z, \tilde{y}) \in \mathbb{R}^p \times \{0, 1\}^p \times \mathbb{R}^p\) is given by

\[
\pi(\beta, z, \tilde{y}|y) \propto \left( \prod_{i=1}^{n} \left( \{y_i > 0\} = y_i \right) N(\tilde{y}_i; x_i^T \beta, 1) \right) \\
\prod_{j=1}^{p} \left( qN(\beta_j; 0, \tau_1^2) \right)^{z_j} \left( (1-q)N(\beta_j; 0, \tau_0^2) \right)^{1-z_j}.
\]

From (16), we can calculate the conditional distributions. We obtain

\[
\begin{align*}
  \pi(\beta|z, \tilde{y}, y) &\propto N(\tilde{y}; X\beta, I_n) N(\beta; 0, D^{-1}) \text{ for } D \triangleq \text{Diag}(z\tau_1^{-2} + (1-p-z)\tau_0^{-2}) \\
  &\propto N(\beta; \Sigma^{-1} X^T \tilde{y}, \Sigma^{-1}) \text{ for } \Sigma = X^T X + D, \\
  \pi(z|\beta, \tilde{y}, y) &\propto \prod_{j=1}^{p} \text{Bernoulli} \left( z_j; \frac{qN(\beta_j; 0, \tau_1^2)}{qN(\beta_j; 0, \tau_1^2) + (1-q)N(\beta_j; 0, \tau_0^2)} \right), \
  \pi(\tilde{y}|\beta, z, y) &\propto \prod_{i=1}^{n} N(\tilde{y}_i; x_i^T \beta, 1) I\{\tilde{y}_i > 0\} = y_i.
\end{align*}
\]
as required for probit regression in Algorithm 3.
B.3. Logistic regression with spike-and-slab priors

We first describe the Bayesian logistic regression model considered. Consider the logistic regression likelihood, where for each observation \(i = 1, \ldots, n\), 
\[
\mathbb{P}(y_i = 1|x_i, \beta) = \frac{1}{1 + \exp(-x_i^\top \beta)}
\]
and to minimize the integrated squared distance between the respective densities. The Gaussian scale representation of this is given by 
\[
\tilde{y}_i \sim \text{Logistic}(x_i^\top \beta, 1),
\]
corresponding to the logistic distribution centered about \(x_i^\top \beta\) and scale parameter 1.

B.3.1. STUDENT’S t-DISTRIBUTION BASED APPROXIMATION OF THE LOGISTIC REGRESSION LIKELIHOOD.

Following O’Brien & Dunson (2004) and Narisetty et al. (2019), we can approximate \(\text{Logistic}(x_i^\top \beta, 1)\) with \(x_i^\top \beta + w t_\nu\), where \(t_\nu\) denotes a t-distribution with \(\nu\) degrees of freedom and \(w\) is a multiplicative factor. The constants \(\nu \equiv 7.3\) and \(w^2 \equiv \frac{\nu - 2}{\nu}\) are chosen following O’Brien & Dunson (2004), in order to match the variance of the logistic distribution and to minimize the integrated squared distance between the respective densities. The Gaussian scale representation of this t-distribution is
\[
\tilde{y}_i | x_i, \beta, \tilde{\sigma}_i \sim \mathcal{N}(x_i^\top \beta, \tilde{\sigma}_i^2), \quad \tilde{\sigma}_i^2 \sim \text{InvGamma}(\frac{\nu}{2}, \frac{w^2 \nu}{2}),
\]
where each \(\tilde{\sigma}_i^2\) is an augmented variable. The Bayesian logistic regression model is then given by
\[
\begin{align*}
&z_j \overset{i.i.d.}{\sim} \text{Bernoulli}(q) \quad \text{for all } j = 1, \ldots, p \\
&\beta_j | z_j \overset{i.i.d.}{\sim} (1 - z_j) \mathcal{N}(0, \tau_0^2) + z_j \mathcal{N}(0, \tau_1^2) \quad \text{for all } j = 1, \ldots, p \\
&\tilde{\sigma}_i^2 \overset{i.i.d.}{\sim} \text{InvGamma}(\frac{\nu}{2}, \frac{w^2 \nu}{2}) \\
&\tilde{y}_i | \beta, \tilde{\sigma}_i^2 \overset{i.i.d.}{\sim} \mathcal{N}(x_i^\top \beta, \tilde{\sigma}_i^2) \quad \text{for all } i = 1, \ldots, n \\
y_i = \frac{\tilde{y}_i > 0}{i} \quad \text{for all } i = 1, \ldots, n.
\end{align*}
\]

Let \(\tilde{\sigma}^2\) denote the vector with entries \(\tilde{\sigma}_i^2\) for \(i = 1, \ldots, n\). For the prior and likelihood in (18), the posterior density of \((\beta, z, \tilde{y}, \tilde{\sigma}^2)\) on \(\mathbb{R}^p \times \{0, 1\}^p \times \mathbb{R}^n \times (0, \infty)^n\) is given by
\[
\pi(\beta, z, \tilde{y}, \tilde{\sigma}^2) \propto \prod_{j=1}^{p} (q \mathcal{N}(\beta_j; 0, \tau_1^2))^{z_j} ((1 - q) \mathcal{N}(\beta_j; 0, \tau_0^2))^{1 - z_j} \\
\prod_{i=1}^{n} \mathcal{I}\{\tilde{y}_i > 0\} = y_i \mathcal{N}(\tilde{y}_i; x_i^\top \beta, \tilde{\sigma}_i^2) \text{InvGamma}\left(\frac{\nu}{2}, \frac{w^2 \nu}{2}\right).
\]
From (19), we can calculate the conditional distributions. Let \(W = \text{Diag}(\tilde{\sigma}^2)\). We obtain
\[
\begin{align*}
\pi(\beta | z, \tilde{y}, \tilde{\sigma}^2, y) &\propto \mathcal{N}(\tilde{y}_i; X \beta, W) \mathcal{N}(\beta; 0, D^{-1}) \quad \text{for } D \equiv \text{Diag}(z_1 \tau_1^{-2} + (1 - z) \tau_0^{-2}) \\
&\propto \mathcal{N}(\beta; \Sigma^{-1} X^\top W^{-1} \tilde{y}, \Sigma^{-1}) \quad \text{for } \Sigma = X^\top W^{-1} X + D, \\
\pi(z | \beta, \tilde{y}, \tilde{\sigma}^2, y) &\propto \prod_{j=1}^{p} \text{Bernoulli}\left(z_j; q \mathcal{N}(\beta_j; 0, \tau_1^2) + (1 - q) \mathcal{N}(\beta_j; 0, \tau_0^2)\right), \\
\pi(\tilde{y}_i | z, \beta, \tilde{\sigma}^2, y) &\propto \prod_{i=1}^{n} \mathcal{N}(\tilde{y}_i; x_i^\top \beta, \tilde{\sigma}_i^2) \mathcal{I}\{\tilde{y}_i > 0\} = y_i, \quad \text{and} \\
\pi(\tilde{\sigma}^2 | z, \beta, \tilde{y}, y) &\propto \prod_{i=1}^{n} \mathcal{N}(\tilde{y}_i; x_i^\top \beta, \tilde{\sigma}_i^2) \text{InvGamma}\left(\frac{\nu}{2}, \frac{w^2 \nu}{2}\right) \\
&\propto \prod_{i=1}^{n} \text{InvGamma}\left(\tilde{\sigma}_i^2; \frac{\nu + 1}{2}, \frac{w^2 \nu + (\tilde{y}_i - x_i^\top \beta)^2}{2}\right),
\end{align*}
\]
as required for logistic regression in Algorithm 3.
Algorithm 4 An $O(n^2 p)$ sampler of (20) (Bhattacharya et al., 2016)

Sample $r \sim N(0, I_p)$, $\xi \sim N(0, I_n)$.

Set $u = D_t^{-1/2} r$ and calculate $v = W_t^{-1/2} Xu + \xi$.

Set $v^* = M_t^{-1}(W_t^{-1/2} \tilde{y} - v)$ for $M_t = I_n + W_t^{-1/2} XD_t^{-1} X^\top W_t^{-1/2}$.

Return $\beta = u + D_t^{-1} X^\top W_t^{-1/2} v^*$.

A scalable Gibbs sampler for logistic regression. The computational bottleneck of existing Gibbs samplers for logistic regression is linked to sampling from the full conditional of $\beta \in \mathbb{R}^p$. This is given by

$$\beta_{t+1} | z_t, \tilde{\sigma}_t^2 \sim N(\Sigma_t^{-1} X^\top W_t^{-1/2} \tilde{y}, \Sigma_t^{-1}) \text{ for } \Sigma_t = X^\top W_t^{-1/2} X + D_t, \quad (20)$$

where $t$ indexes the iteration of the Markov chain, $W_t$ is the diagonal matrix with the vector $\tilde{\sigma}_t^2$ populating its diagonal elements, and $D_t$ is the diagonal matrix with the vector $\sigma_t^2 + (1_p - z_t) \tau_0^{-2}$ populating its diagonal elements. To sample from (20), we can use the $\Omega(n^2 p)$ sampler of Bhattacharya et al. (2016), which is given in Algorithm 4.

Following the strategy in Section 2.2, $S$ for logistic regression uses pre-computation to reduce the computational cost of Algorithm 4. Using the notation from Section 2.2 with $M_t \triangleq I_n + W_t^{-1/2} XD_t^{-1} X^\top W_t^{-1/2}$, we note

$$M_t = I_n + W_t^{-1/2}(\tilde{M}_t - I_n + (\tau_1^2 - \tau_0^2)X_A^\top X_A)W_t^{-1/2} = I_n + W_t^{-1/2}(\tilde{M}_t - I_n + (\tau_0^2 - \tau_1^2)X_A^\top X_A^c)W_t^{-1/2} = I_n + W_t^{-1/2}(W_t^{-1/2}(M_t - I_n)W_t^{-1/2} + X_{\Delta_c} C_{\Delta_c} X_{\Delta_c}^\top)W_t^{-1/2}. \quad (21a) - (21c)$$

In (21a) – (21c), calculating the matrix products $X_A X_A^\top$, $X_{A^c} X_{A^c}^\top$, and $X_{\Delta_c} C_{\Delta_c} X_{\Delta_c}^\top$ requires $O(n^2 \|z_t\|_1)$, and $O(n^2 (p - \|z_t\|_1))$, and $O(n^2 \delta_t)$ cost respectively. Given $\tilde{M}_t$, $\tilde{M}_t$, $M_t - I_n$, and $z_t$, we evaluate whichever matrix product in (21a) – (21c) has minimal computational cost and thereby calculate $M_t$ at the reduced cost of $O(n^2 p_t)$ where $p_t \triangleq \min\{\|z_t\|_1, p - \|z_t\|_1, \delta_t\}$. To calculate $M_t^{-1}$, we calculate $M_t^{-1}$ by directly inverting the calculated matrix $M_t$, which requires $O(n^3)$ cost. Overall, this strategy reduces the computational cost of calculating the matrices $M_t$ and $M_t^{-1}$ from $\Omega(n^2 p_t)$ to $O(\max\{n^3 p_t, n^2\})$.

Extensions to Scalable Spike-and-Slab for logistic regression. Suppose the matrices $X^\top X$ is pre-computed. This initial step requires $O(np^2)$ computational cost and $O(p^2)$ memory. Then the matrices $X_A X_A^\top$, $X_A X_{A^c}^\top$, and $X_{\Delta_c} C_{\Delta_c} X_{\Delta_c}^\top$ in (21a) – (21b) correspond to pre-computed sub-matrices of $X^\top X$, and calculating $M_t$ using (21a) – (21b) each iteration $t$ involves matrix addition and diagonal matrix multiplication which only requires $O(n^2)$ cost. To sample from (20), we calculate $M_t^{-1}$ by directly inverting the calculated matrix $M_t$ from (3), which requires $O(n^3)$ cost. Overall, now the Gibbs samplers for logistic regression requires $O(\max\{n^3, np\})$ computational cost at iteration $t$, which is an improvement compared to $S$.

C. Experiment Details

Figure 3 of Section 3. In Figure 3, we use the same prior hyperparameters for all the algorithms. Following Narisetty et al. (2019), we choose $\tau_0^2 = \frac{1}{n}$, $\tau_1^2 = \max\{\frac{p}{2mn}, 1\}$ and $q = \mathbb{P}(z_j = 1)$ such that $\mathbb{P}(\sum_{j=1}^p I[z_j = 1] > K) = 0.1$ for $K = \max\{10, \log n\}$. The true positive rate (TPR) and the false discovery rate (FDR) correspond to the proportion of non-zero and zero components of $\beta^*_j$ that are correctly selected respectively. They are calculated as $\frac{1}{p} \sum_{j=1}^p I[\mathbb{P}^*(z_j = 1) > 0.5]$ and $\frac{1}{p} \sum_{j=1}^p I[\mathbb{P}^*(z_j = 1) < 0.5]$ respectively, where the marginal posterior probabilities $\pi_j \triangleq \mathbb{P}(z_j = 1)$ are estimated by $\hat{\pi}_j \triangleq \frac{1}{100} \sum_{i=1}^{1000} z_{j,t}$ for sample points $(z_t)_{t \geq 0}$ generated using $S$ or Skinny Gibbs. The lines in Figure 3 correspond to the average TPR and FDR across 20 independently generated datasets, and the grey bands correspond to one standard error of the averages.

D. Dataset Details

Synthetic continuous response dataset in Section 2.3. In Figure 1, synthetic linear regression datasets are considered. For number of observations $n$ and number of covariates $p$, we generate a design matrix $X \in \mathbb{R}^{n \times p}$ such that each
The PCR GWAS dataset has \( n = 118 \) observations with binary responses and \( p = 22283 \) covariates. The Chowdary dataset has \( n = 104 \) observations with binary responses and \( p = 22283 \) covariates. The Gordon dataset has \( n = 118 \) observations with binary responses and \( p = 22215 \) covariates. The Borovecki dataset has \( n = 373 \) observations with binary responses and \( p = 104 \) covariates.

Datasets in Section 4. The Malware detection dataset from the UCI machine learning repository (Dua & Graff, 2017) has \( n = 373 \) observations with binary responses and \( p = 503 \) covariates, and is publicly available on www.kaggle.com/piyushrumao/malware-executable-detection.

The synthetic continuous dataset has \( n = 1000 \) observations and \( p = 50000 \) covariates. The design matrix \( X \) is generated such that each \( [X]_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1) \) for all \( 1 \leq i \leq n \) and \( 1 \leq j \leq p \), which is then scaled to ensure each column has a mean of 0 and a standard error of 1. We choose the true signal \( \beta^* \in \mathbb{R}^p \) such that \( \beta^*_j = 2I\{j \leq s\} \), where \( s \) is the sparsity parameter corresponding to the number of non-zero components. Given \( X \) and \( \beta^* \), we generate \( y = X \beta^* + \sigma^* \epsilon \) for \( \epsilon \sim \mathcal{N}(0, I_n) \), where \( \sigma^* = 2 \) is the Gaussian noise standard deviation.

E. Additional Experiments

Variable selection performance as a function of time or number of iterations. Figure 6 plots the average true positive rate \( \text{TPR} \) and the false discovery rate \( \text{FDR} \) of variable selection based on samples from \( S^3 \) and Skinny Gibbs as the length of the chains are varied. The TPR and FDR are averaged over 10 independent chains, and one standard error bars are shown. We consider a synthetic binary classification dataset generated using a logistic regression model as in Section 3, with \( n = 200 \) observations, \( p = 1000 \) covariates, sparsity \( s = 10 \), and an exponentially decaying sparse true signal \( \beta^* \in \mathbb{R}^p \) such that \( \beta^*_j = 2e^{-j/50} \) for \( j \leq s \) and \( \beta^*_j = 0 \) for \( j > s \). The TPR and FDR are calculated as in Section 3 with the marginal posterior probabilities \( \pi_j : \equiv \mathbb{P}(z_j = 1) \) now estimated by \( \hat{\pi}_j = \frac{1}{T-999} \sum_{t=1000}^T z_{j,t} \) for a burn-in of 1000, a varying chain length \( T \geq 1000 \), and sample points \( \{z_t\}_{t \geq 0} \) generated using \( S^3 \) or Skinny Gibbs. We use the same prior hyperparameters for all the algorithms, which are chosen according to Narisetti et al. (2019).

Figure 6 Left and Center-Left plot the TPR and FDR against the chain length \( T \). It shows that \( S^3 \) for both logistic and probit regression have higher TPR and lower FDR than Skinny Gibbs for all chain lengths. Furthermore, \( S^3 \) for logistic regression has higher TPR and lower FDR than \( S^3 \) for probit regression, which is expected as the synthetic dataset for this example is generated using a logistic regression model. The SOTA sampler is omitted from the Left and Center-Left plots as its output has the same marginal distribution and statistical performance as \( S^3 \). Figure 6 Center-Right and Right plot the TPR and FDR against total time elapsed in seconds to generate samples using \( S^3 \), SOTA, or Skinny Gibbs chains with a burn-in of 1000.
iterations. The standard error bars are now omitted for better visibility. For each time budget, we observe better variable selection performance from $S^3$ when compared with the slower SOTA implementation or with Skinny Gibbs.

**Effective Sample Size of $S^3$ for the datasets in Section 4.** Figure 7 shows the Effective Sample Size per iteration and per unit of time (in seconds) of $S^3$ and the SOTA sampler for the datasets in Section 4. The ESS is calculated using the mcmccse package (Flegal et al., 2021; Vats et al., 2019) for one $S^3$ chain of length 10000 iterations with a burn-in of 1000 iterations for each dataset. The average ESS of the $\beta$ components are then plotted. Figure 7 Right shows that $S^3$ has significantly higher ESS per second compared to the corresponding SOTA sampler for all the datasets considered.

**Performance metrics for the datasets in Section 4.** Figures 8 – 14 show various performance metrics of $S^3$ for some of the datasets considered in Section 4. Figures 8 – 14 (Left) plot the marginal posterior probability estimates $\hat{\pi}_j$ against $j$ in the decreasing order of $\hat{\pi}_j$, following the setup in Figure 4. For datasets with continuous valued responses, $\hat{\pi}_j$s are based on samples from $S^3$ for linear regression. For datasets with binary valued responses, $\hat{\pi}_j$s are based on samples from $S^3$ for logistic and probit regression, and the Skinny Gibbs sampler from logistic regression. We use samples from 5 independent chains of length 10000 iterations with a burn-in of 1000 iterations. Estimates based on samples from the SOTA sampler are not shown, as they implement the same Gibbs sampler as $S^3$ and one SOTA chain. To compute this evaluation, we partition the observed dataset into 3 folds, run a chain conditioned on all data outside of fold $k$, and evaluate its performance on the held-out data in the $k$-th fold. The average RMSE is calculated as $\frac{1}{|D|} \sum_{k=1}^{10} r_k$, where $r_k$ is the RMSE for the $k^{th}$ fold. For datasets with continuous valued responses, the quantities $r_k$ for linear regression are calculated as $\left(\frac{1}{|D_k|} \sum_{i \in D_k} (y_i - \hat{y}_i)^2\right)^{1/2}$, where $D_k$ is the $k^{th}$ fold, $\hat{y}_i \equiv \frac{1}{T=1000} \sum_{t=1001}^{T} x_i^T \beta_t$ are the predicted responses, and $(\beta_t)_{t \geq 0}$ are samples from $S^3$ and SOTA targeting the posterior distribution of the $k^{th}$ training set. For datasets with binary valued responses, the quantities...
**Figure 7.** Effective sample size (ESS) per iteration and per second of $S^3$ and the SOTA sampler for some of the datasets in Section 4. The ESS is calculated using one $S^3$ chain of length 10000 iterations with a burn-in of 1000 iterations. The ESS per iteration of the SOTA sampler is omitted from the Left plot as it implements the same Gibbs sampler as $S^3$.

**Figure 8.** Borovecki dataset with $n = 31$ observations, $p = 22283$ covariates and binary valued responses.

$r_k$ are calculated as $(\frac{1}{|D_k|} \sum_{i \in D_k} (y_i - \hat{p}_i)^2)^{1/2}$, where $D_k$ is the $k^{th}$ fold, $\hat{p}_i \triangleq \frac{1}{T-1000} \sum_{t=1001}^{T} \text{Logistic}(x_i^T \beta_t)$ and $\hat{p}_i \triangleq \frac{1}{T-1000} \sum_{t=1001}^{T} \Phi(x_i^T \beta_t)$ are the predicted probabilities for logistic and probit regression respectively, and $(\beta_t)_{t \geq 0}$ are samples from $S^3$ and SOTA targeting the posterior distribution of the $k^{th}$ training set. Figures 8 – 14 (Right) plot the average RMSE against total time elapsed in seconds to generate samples using $S^3$ or SOTA chains with a burn-in of 1000 iterations. The RMSE of the Skinny Gibbs sampler is not available, as the skinnybasad package does not output the full chain trajectories required for RMSE calculations.
Figure 9. Chin dataset with \( n = 118 \) observations, \( p = 22215 \) covariates and binary valued responses.

Figure 10. Chowdary dataset with \( n = 104 \) observations, \( p = 22283 \) covariates and binary valued responses.

Figure 11. Gordon dataset with \( n = 181 \) observations, \( p = 12533 \) covariates and binary valued responses.
Figure 12. Lymph dataset with \( n = 148 \) observations, \( p = 4514 \) covariates and binary valued responses.

Figure 13. Malware dataset with \( n = 373 \) observations, \( p = 503 \) covariates and binary valued responses.

Figure 14. PCR dataset with \( n = 60 \) observations, \( p = 22575 \) covariates and continuous valued responses.