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* (S^3), 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 $\max\{n^2 p_t, np\}$ 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 $n^2 p$ 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 $L_{\text{lin}}(\boldsymbol{\beta}, \sigma^2; \boldsymbol{X}, \boldsymbol{y}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\big(-\frac{\sum_{i=1}^n (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2}{2\sigma^2}\big)$, $L_{\text{log}}(\boldsymbol{\beta}; \boldsymbol{X}, \boldsymbol{y}) = \prod_{i=1}^n \frac{\exp(y_i \boldsymbol{x}_i^\top \boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})}$, and $L_{\text{prob}}(\boldsymbol{\beta}; \boldsymbol{X}, \boldsymbol{y}) = \prod_{i=1}^n \Phi(\boldsymbol{x}_i^\top \boldsymbol{\beta})^{y_i} (1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\beta}))^{1-y_i}$. Here $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is the design matrix with rows $\boldsymbol{x}_i^\top$, $\boldsymbol{y} \in \mathbb{R}^n$ (for linear regression) or $\boldsymbol{y} \in \{0,1\}^n$ (for logistic and probit regression) is the response vector, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the unknown signal, $\sigma^2 \in (0, \infty)$ is the unknown Gaussian noise variance, and Φ is the cumulative density function of $\mathcal{N}(0, 1)$.

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

$$\sigma^2 \sim \text{InvGamma}\Big(\frac{a_0}{2}, \frac{b_0}{2}\Big), \quad \text{(for linear regression)}$$

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

$$z_j \overset{i.i.d.}{\underset{j=1,\dots,p}{\sim}} \text{Bernoulli}(q),$$

$$\beta_j | z_j, \sigma^2 \overset{ind}{\underset{j=1,\dots,p}{\sim}} (1-z_j)\mathcal{N}(0, \sigma^2\tau_0^2) + z_j\mathcal{N}(0, \sigma^2\tau_1^2), \quad (1)$$

[1]Department of Statistics, Harvard University [2]Microsoft Research New England. Correspondence to: Niloy Biswas <niloy_biswas@g.harvard.edu>.

where $q \in (0, 1)$, $\tau_1^2 \gg \tau_0^2 > 0$, and $a_0, b_0 > 0$ are hyper-parameters. Here, $\mathcal{N}(0, \sigma^2 \tau_0^2)$ and $\mathcal{N}(0, \sigma^2 \tau_1^2)$ 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 of Bayesian variable selection (see the recent reviews of Tadesse & Vannucci (2021, Section I) and Banerjee et al. (2021)). The posterior probabilities $\mathbb{P}(z_j = 1|\boldsymbol{y})$ provide a natural interpretable approach to variable selection. The *median probability model* selects all covariates j such that $\mathbb{P}(z_j = 1|\boldsymbol{y}) > 1/2$, and it is easily fitted using Monte Carlo samples and provides the optimal predictive model in the case with orthogonal design matrix, as well as extensions with certain correlated matrices (Barbieri & Berger, 2004; Barbieri et al., 2021). Narisetty & He (2014) have further fine-tuned the optimal scaling of q, τ_0^2, and τ_1^2 with respect to the number of covariates and sample size to establish model selection consistency for linear regression with general design matrices in high dimensions.

One could alternatively consider *point-mass* spike-and-slab priors (e.g., Mitchell & Beauchamp, 1988; Johnson & Rossell, 2012), where $\beta_j|z_j \sim (1-z_j)\delta_0(\cdot) + z_j\mathcal{N}(0, \sigma^2 \tau_1^2)$ such that a degenerate Dirac distribution about zero is chosen for the spike part. Point-mass priors have favorable statistical properties (e.g., Johnstone & Silverman, 2004; Castillo & van der Vaart, 2012) but can be computationally prohibitive (Bai et al., 2021). We hope to extend our algorithms to point-mass priors in follow-up work.

1.3. Our contributions

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

Section 2 introduces *Scalable Spike-and-Slab* (S^3), 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 S^3, which overcomes existing computational bottlenecks by employing a pre-computation based strategy and requires $\mathcal{O}(\max\{n^2 p_t, np\})$ 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 favourable computational complexity of S^3, 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 S^3 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 S^3 offers substantially faster numerical runtimes compared to the SOTA exact MCMC sampler, reporting $50\times$ speedups on synthetic datasets. In the same experiment, S^3 and the approximate sampler have comparable runtimes, but the asymptotically exact S^3 procedure provides more accurate variable selection.

Section 4 demonstrates the benefits of S^3 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 $\boldsymbol{\beta} \in \mathbb{R}^p$. This is given by

$$\boldsymbol{\beta}_{t+1}|\boldsymbol{z}_t, \sigma_t^2 \sim \mathcal{N}\left(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{X}^\top \boldsymbol{y}, \sigma_t^2 \boldsymbol{\Sigma}_t^{-1}\right) \qquad (2)$$

for $\boldsymbol{\Sigma}_t = \boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{D}_t$, where t indexes the iteration of the Markov chain, and $\boldsymbol{D}_t$ is the diagonal matrix with the vector $\boldsymbol{z}_t \tau_1^{-2} + (\mathbf{1}_p - \boldsymbol{z}_t)\tau_0^{-2}$ populating its diagonal elements.

Sampling from (2) using standard matrix multiplication and a generic Cholesky decomposition that ignores the specific structure of $\boldsymbol{\Sigma}_t$ requires $\Omega(p^3)$ computational cost, which quickly becomes prohibitive for large p. Hereafter we will refer to this generic method as the Naïve Sampler. Ishwaran & Rao (2005) recommend separating the components of $\boldsymbol{\beta}$ into B blocks of size p/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 favourable $\Omega(n^2 p)$ computational cost and is given in Algorithm 1.

Algorithm 1 An $\Omega(n^2p)$ sampler of (2) (Bhattacharya et al., 2016)

Sample $\boldsymbol{r} \sim \mathcal{N}(0, \boldsymbol{I}_p)$, $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{I}_n)$.

Set $\boldsymbol{u} = \boldsymbol{D}_t^{-\frac{1}{2}}\boldsymbol{r}$ and calculate $\boldsymbol{v} = \boldsymbol{X}\boldsymbol{u} + \boldsymbol{\xi}$.

Set $\boldsymbol{v}^* = \boldsymbol{M}_t^{-1}(\frac{1}{\sigma_t}\boldsymbol{y} - \boldsymbol{v})$ for $\boldsymbol{M}_t = \boldsymbol{I}_n + \boldsymbol{X}\boldsymbol{D}_t^{-1}\boldsymbol{X}^\top$.

Return $\boldsymbol{\beta} = \sigma_t(\boldsymbol{u} + \boldsymbol{D}_t^{-1}\boldsymbol{X}^\top\boldsymbol{v}^*)$.

For large-scale datasets with n in the thousands and p in the hundreds of thousands, as found in modern scientific applications, the $\Omega(n^2p)$ 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 (Titsias & 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 $\Omega(n^2p)$ computational cost per iteration arises from the calculation of the matrix $\boldsymbol{M}_t$. Current algorithms calculate $\boldsymbol{M}_t = \boldsymbol{I}_n + \boldsymbol{X}\boldsymbol{D}_t^{-1}\boldsymbol{X}^\top$ from scratch every iteration at $\Omega(n^2p)$ cost under standard matrix multiplication and then solve an n by n linear system to obtain v^* at $\Omega(n^3)$ cost. We propose instead to use the previous state $\boldsymbol{z}_{t-1}$ and the pre-computed matrices $\boldsymbol{M}_{t-1}$ and $\boldsymbol{M}_{t-1}^{-1}$ at each step to aid the calculation of $\boldsymbol{M}_t^{-1}$. Our strategy is given below.

Specifically, denote $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_p] \in \mathbb{R}^{n \times p}$ and its sub-matrices $\boldsymbol{X}_{A_t} \triangleq [\boldsymbol{X}_j : j \in A_t] \in \mathbb{R}^{n \times \|\boldsymbol{z}_t\|}$ and $\boldsymbol{X}_{A_t^c} \triangleq [\boldsymbol{X}_j : j \in A_t^c] \in \mathbb{R}^{n \times (p - \|\boldsymbol{z}_t\|)}$, where $A_t \triangleq \{j : z_{j,t} = 1\}$ and $A_t^c = \{j : z_{j,t} = 0\}$, 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 $t-1$; $\delta_t \triangleq \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_1 = |\Delta_t|$, the number of switches; $\boldsymbol{D}_{\Delta_t} \triangleq \text{Diag}((\boldsymbol{D}_t)_{j,j} : j \in \Delta_t) \in \text{Diag}(\mathbb{R}^{\delta_t \times \delta_t})$, the diagonal sub-matrix of $\boldsymbol{D}_t$ composed of the diagonal entries with ordered indices in Δ_t; and $\boldsymbol{C}_{\Delta_t} \triangleq \boldsymbol{D}_{\Delta_t}^{-1} - \boldsymbol{D}_{\Delta_{t-1}}^{-1}$. Finally, let $\tilde{\boldsymbol{M}}_{\tau_0} \triangleq \boldsymbol{I}_n + \tau_0^2 \boldsymbol{X}\boldsymbol{X}^T \in \mathbb{R}^{n \times n}$ and $\tilde{\boldsymbol{M}}_{\tau_1} \triangleq \boldsymbol{I}_n + \tau_1^2 \boldsymbol{X}\boldsymbol{X}^T \in \mathbb{R}^{n \times n}$, which are fixed for all iterations. Under this notation, there are three expressions for $\boldsymbol{M}_t$:

$$\boldsymbol{M}_t = \tilde{\boldsymbol{M}}_{\tau_0} + (\tau_1^2 - \tau_0^2)\boldsymbol{X}_{A_t}\boldsymbol{X}_{A_t}^\top \tag{3a}$$

$$= \tilde{\boldsymbol{M}}_{\tau_1} + (\tau_0^2 - \tau_1^2)\boldsymbol{X}_{A_t^c}\boldsymbol{X}_{A_t^c}^\top \tag{3b}$$

$$= \boldsymbol{M}_{t-1} + \boldsymbol{X}_{\Delta_t}\boldsymbol{C}_{\Delta_t}\boldsymbol{X}_{\Delta_t}^\top. \tag{3c}$$

In (3a) – (3c), calculating the matrix products $\boldsymbol{X}_{A_t}\boldsymbol{X}_{A_t}^\top$,

$\boldsymbol{X}_{A_t^c}\boldsymbol{X}_{A_t^c}^\top$, and $\boldsymbol{X}_{\Delta_t}\boldsymbol{C}_{\Delta_t}\boldsymbol{X}_{\Delta_t}^\top$ requires $\mathcal{O}(n^2\|\boldsymbol{z}_t\|_1)$, $\mathcal{O}(n^2(p - \|\boldsymbol{z}_t\|_1))$, and $\mathcal{O}(n^2\delta_t)$ cost respectively. Given $\tilde{\boldsymbol{M}}_{\tau_0}, \tilde{\boldsymbol{M}}_{\tau_1}, \boldsymbol{M}_{t-1}$, and $\boldsymbol{z}_{t-1}$, we evaluate whichever matrix product in (3a) – (3c) has minimal computational cost and thereby calculate $\boldsymbol{M}_t$ at the reduced cost of $\mathcal{O}(n^2p_t)$ where $p_t \triangleq \min\{\|\boldsymbol{z}_t\|_1, p - \|\boldsymbol{z}_t\|_1, \delta_t\}$.

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

$$\boldsymbol{M}_t^{-1} = \tilde{\boldsymbol{M}}_{\tau_0}^{-1} \tag{4a}$$

$$- \tilde{\boldsymbol{M}}_{\tau_0}^{-1}\boldsymbol{X}_{A_t}\left(\frac{1}{\tau_1^2 - \tau_0^2}\boldsymbol{I}_{\|\boldsymbol{z}_t\|_1} + \boldsymbol{X}_{A_t}^\top\tilde{\boldsymbol{M}}_{\tau_0}^{-1}\boldsymbol{X}_{A_t}\right)^{-1}\boldsymbol{X}_{A_t}^\top\tilde{\boldsymbol{M}}_{\tau_0}^{-1}$$

$$= \tilde{\boldsymbol{M}}_{\tau_1}^{-1} \tag{4b}$$

$$- \tilde{\boldsymbol{M}}_{\tau_1}^{-1}\boldsymbol{X}_{A_t^c}\left(\frac{1}{\tau_0^2 - \tau_1^2}\boldsymbol{I}_{p - \|\boldsymbol{z}_t\|_1} + \boldsymbol{X}_{A_t^c}^\top\tilde{\boldsymbol{M}}_{\tau_1}^{-1}\boldsymbol{X}_{A_t^c}\right)^{-1}\boldsymbol{X}_{A_t^c}^\top\tilde{\boldsymbol{M}}_{\tau_1}^{-1}$$

$$= \boldsymbol{M}_{t-1}^{-1} \tag{4c}$$

$$- \boldsymbol{M}_{t-1}^{-1}\boldsymbol{X}_{\Delta_t}\left(\boldsymbol{C}_{\Delta_t}^{-1} + \boldsymbol{X}_{\Delta_t}^\top\boldsymbol{M}_{t-1}^{-1}\boldsymbol{X}_{\Delta_t}\right)^{-1}\boldsymbol{X}_{\Delta_t}^\top\boldsymbol{M}_{t-1}^{-1}.$$

Given $\tilde{\boldsymbol{M}}_{\tau_0}^{-1}, \tilde{\boldsymbol{M}}_{\tau_1}^{-1}, \boldsymbol{M}_{t-1}^{-1}$ and $\boldsymbol{z}_{t-1}$, we evaluate whichever expression in (4a) – (4c) has minimal computational cost to calculate $\boldsymbol{M}_t^{-1}$. Similar to (3), this requires $\mathcal{O}(n^2p_t)$ computational cost, which arises from matrix inversion and multiplication.

Overall, this strategy of using the previous state $\boldsymbol{z}_{t-1}$ and the pre-computed matrices $\tilde{\boldsymbol{M}}_{\tau_0}, \tilde{\boldsymbol{M}}_{\tau_0}^{-1}, \tilde{\boldsymbol{M}}_{\tau_1}, \tilde{\boldsymbol{M}}_{\tau_1}^{-1}, \boldsymbol{M}_{t-1}$ and $\boldsymbol{M}_{t-1}^{-1}$, reduces the computational cost of calculating the matrices $\boldsymbol{M}_t$ and $\boldsymbol{M}_t^{-1}$ from $\Omega(n^2p)$ (as in all current implementations of Algorithm 1) to $\mathcal{O}(n^2p_t)$. 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 $\tilde{\boldsymbol{M}}_{\tau_0}, \tilde{\boldsymbol{M}}_{\tau_0}^{-1}, \tilde{\boldsymbol{M}}_{\tau_1}, \tilde{\boldsymbol{M}}_{\tau_1}^{-1}$ are fixed for all iterations, and the state $\boldsymbol{z}_{t-1}$ and matrices $\boldsymbol{M}_{t-1}$ and $\boldsymbol{M}_{t-1}^{-1}$ 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 $\mathcal{O}(\max\{n^2p_t, n^3, np\})$ 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 $C_t \triangleq (\boldsymbol{\beta}_t, \boldsymbol{z}_t, \sigma_t^2) \in \mathbb{R}^p \times \{0,1\}^p \times (0,\infty)$, state $\boldsymbol{z}_{t-1}$, and matrices $\boldsymbol{M}_{t-1}, \boldsymbol{M}_{t-1}^{-1}$.

1: Calculate p_t and use (3) to calculate $\boldsymbol{M}_t$.
   if $p_t \geq n$ **then** invert $\boldsymbol{M}_t$ to calculate $\boldsymbol{M}_t^{-1}$ **else** use (4) to calculate $\boldsymbol{M}_t^{-1}$.
2: Sample $\boldsymbol{\beta}_{t+1} | \boldsymbol{z}_t, \sigma_t^2$ using Algorithm 1 from
   $$\mathcal{N}\big(\boldsymbol{\Sigma}_t^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \sigma_t^2 \boldsymbol{\Sigma}_t^{-1}\big) \text{ for } \boldsymbol{\Sigma}_t = \boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{D}_t.$$
3: Sample each $z_{j,t+1} | \boldsymbol{\beta}_{t+1}, \sigma_t^2$ independently from
   $$\text{Bernoulli}\Big(\tfrac{q\mathcal{N}(\beta_{j,t+1};0,\sigma_t^2\tau_1^2)}{q\mathcal{N}(\beta_{j,t+1};0,\sigma_t^2\tau_1^2)+(1-q)\mathcal{N}(\beta_{j,t+1};0,\sigma_t^2\tau_0^2)}\Big).$$
   for $j = 1, ..., p$.
4: Sample $\sigma_{t+1}^2 | \boldsymbol{\beta}_{t+1}, \boldsymbol{z}_{t+1}$ from
   $$\text{InvGamma}\Big(\tfrac{a_0+n+p}{2}, \tfrac{b_0+\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_{t+1}\|_2^2 + \boldsymbol{\beta}_{t+1}^\top \boldsymbol{D}_{t+1} \boldsymbol{\beta}_{t+1}}{2}\Big).$$

Output: $C_{t+1} = (\boldsymbol{\beta}_{t+1}, \boldsymbol{z}_{t+1}, \sigma_{t+1}^2), \boldsymbol{z}_t, \boldsymbol{M}_t, \boldsymbol{M}_t^{-1}$.

Algorithm 3 Bayesian logistic & probit regression with S³

Input: State $C_t \triangleq (\boldsymbol{\beta}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{\sigma}}_t^2) \in \mathbb{R}^p \times \{0,1\}^p \times \mathbb{R}^n \times (0,\infty)^n$, states $\boldsymbol{z}_{t-1}, \tilde{\boldsymbol{\sigma}}_{t-1}^2$, and matrices $\boldsymbol{M}_{t-1}, \boldsymbol{M}_{t-1}^{-1}$.

1: *Logistic regression:* Use pre-computation (see Appendix B.3) to calculate $\boldsymbol{M}_t \triangleq \boldsymbol{I}_n + \boldsymbol{W}_t^{-1/2} \boldsymbol{X} \boldsymbol{D}_t^{-1} \boldsymbol{X}^\top \boldsymbol{W}_t^{-1/2}$ for $\boldsymbol{W}_t = \text{Diag}(\tilde{\boldsymbol{\sigma}}_t^2)$. Invert $\boldsymbol{M}_t$ to calculate $\boldsymbol{M}_t^{-1}$.
   Probit regression: Calculate p_t and use (3) to calculate $\boldsymbol{M}_t \triangleq \boldsymbol{I}_n + \boldsymbol{X}\boldsymbol{D}_t^{-1}\boldsymbol{X}^\top$. **if** $p_t \geq n$ **then** invert $\boldsymbol{M}_t$ to calculate $\boldsymbol{M}_t^{-1}$ **else** use (4) to calculate $\boldsymbol{M}_t^{-1}$.
2: Sample $\boldsymbol{\beta}_{t+1} | \boldsymbol{z}_t, \tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{\sigma}}_t^2$ using Algorithm 1 from
   $$\mathcal{N}\big(\boldsymbol{\Sigma}_t^{-1} \boldsymbol{X}^\top \boldsymbol{W}_t^{-1} \tilde{\boldsymbol{y}}_t, \boldsymbol{\Sigma}_t^{-1}\big) \text{ for } \boldsymbol{\Sigma}_t = \boldsymbol{X}^\top \boldsymbol{W}_t^{-1} \boldsymbol{X} + \boldsymbol{D}_t.$$
3: Sample each $z_{j,t+1} | \boldsymbol{\beta}_{t+1}, \tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{\sigma}}_t^2$ independently from
   $$\text{Bernoulli}\Big(\tfrac{q\mathcal{N}(\beta_{j,t+1};0,\tau_1^2)}{q\mathcal{N}(\beta_{j,t+1};0,\tau_1^2)+(1-q)\mathcal{N}(\beta_{j,t+1};0,\tau_0^2)}\Big)$$
   for $j = 1, ..., p$.
4: Sample each $\tilde{y}_{i,t+1} | \boldsymbol{\beta}_{t+1}, \boldsymbol{z}_{t+1}, \tilde{\boldsymbol{\sigma}}_t^2$ independently from
   $$\begin{array}{ll} \mathcal{N}(\boldsymbol{x}_i^\top \boldsymbol{\beta}_{t+1}, \tilde{\sigma}_{i,t}^2) \text{I}_{[0,+\infty)} & \text{if } y_i = 1, \\ \mathcal{N}(\boldsymbol{x}_i^\top \boldsymbol{\beta}_{t+1}, \tilde{\sigma}_{i,t}^2) \text{I}_{(-\infty,0)} & \text{if } y_i = 0 \end{array}$$
   for $i = 1, ..., n$.
5: *Logistic regression:* Sample each $\tilde{\sigma}_{i,t+1}^2 | \boldsymbol{\beta}_{t+1}, \boldsymbol{z}_{t+1}, \tilde{\boldsymbol{y}}_{t+1}$ independently from
   $$\text{InvGamma}\Big(\tfrac{v+1}{2}, \tfrac{w^2 \nu + (\tilde{y}_{i,t+1} - \boldsymbol{x}_i^\top \boldsymbol{\beta}_{t+1})^2}{2}\Big)$$
   for $i = 1, ..., n$, where $\nu \triangleq 7.3$ and $w^2 \triangleq \pi^2(\nu-2)/(3\nu)$ are constants.
   Probit regression: Set each $\tilde{\sigma}_{i,t+1}^2 = 1$ for $i = 1, ..., n$.

Output: $C_{t+1} = (\boldsymbol{\beta}_{t+1}, \boldsymbol{z}_{t+1}, \tilde{\boldsymbol{y}}_{t+1}, \tilde{\boldsymbol{\sigma}}_{t+1}^2), \boldsymbol{z}_t, \boldsymbol{M}_t, \boldsymbol{M}_{t-1}^{-1}$

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 $\Omega(n^2 p)$ cost.

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

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

Sparse $\boldsymbol{z}_t$. Whenever $\boldsymbol{z}_t$ is sparse relative to the full dimensionality p, we have $p_t \leq \|\boldsymbol{z}_t\|_1 \ll p$. Sparsity in $\boldsymbol{z}_t$ is a common occurrence in high-dimensional regression with a sparse signal vector $\boldsymbol{\beta}^* \in \mathbb{R}^p$, as $\|\boldsymbol{z}_t\|_1$ often closely approximates the true sparsity $s \triangleq \|\boldsymbol{\beta}^*\|_0$ 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 $\min\{\|\boldsymbol{z}_t\|_1, p - \|\boldsymbol{z}_t\|_1\}$ is comparable to $p/2$, concentration of the spike-and-slab posterior targeted by the Gibbs sampler can lead to δ_t and hence p_t remaining much smaller than $p/2$. Proposition 2.2, proved in Appendix A, calculates the expectation of δ_t explicitly in terms of the posterior distribution that is targeted by our algorithms and the auto-correlation of the states $(\boldsymbol{z}_t)_{t \geq 0}$.

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

$$\mathbb{E}[\delta_t] = \sum_{j=1}^p \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 0) + \\ \mathbb{P}(z_{j,t} = 0)\mathbb{P}(z_{j,t-1} = 1) - 2\text{cov}(z_{j,t}, z_{j,t-1}). \quad (5)$$

Suppose the Markov chain generated by Algorithm 2 or 3 is at its stationary distribution π at iteration $t-1$. Then,

$$\mathbb{E}[\delta_t] = 2\sum_{j=1}^p \text{var}_\pi(z_{j,t})(1 - \text{corr}_\pi(z_{j,t}, z_{j,t-1})). \quad (6)$$

In (6), note $\mathrm{var}_\pi(z_{j,t}) = \mathbb{P}_\pi(z_{j,t} = 0)\mathbb{P}_\pi(z_{j,t} = 1) \leq 1/4$ for each component j, with equality only when $z_{j,t} \sim$ 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 δ_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 δ_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 δ_t in expectation. Proposition 2.3, proved in Appendix A, provides an additional exact expression for δ_t in terms of the empirical correlation between z_t and z_{t-1}.

Proposition 2.3 (Swap count decomposition). *Let* $\tau_t = \sqrt{\|z_t\|_1(p - \|z_t\|_1)}$ *and* ρ_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* δ_t *as given in Proposition 2.1,*

$$\delta_t = \|z_t\|_1 + \|z_{t-1}\|_1 - \frac{2\|z_t\|_1\|z_{t-1}\|_1 + 2\rho_t\tau_t\tau_{t-1}}{p}. \tag{7}$$

Since $|\rho_t| \leq 1$, Proposition 2.3 implies

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

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

The lower bound in (8) is a good approximation to δ_t when ρ_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, τ_t exhibits similar "stable/stuck" behavior, implying that the lower bound itself will be close to zero. This suggests δ_t and hence p_t is close to zero when ρ_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 the mean and one standard error bars of $(p_t)_{t=5000}^{10000}$ are plotted. For Figure 1 (Left), we consider datasets with $n = 100$, varying p with
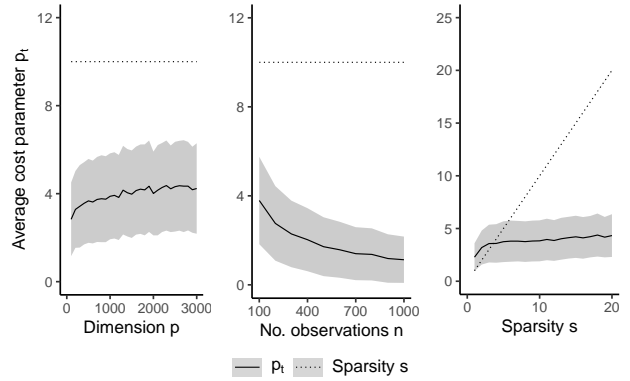

Figure 1. The S^3 cost parameter p_t for averaged over iterations $5000 < t \leq 10000$ with one standard error bars, for synthetic linear regression datasets with varying number of covariates p (**Left**), varying number of observations n (**Center**), and varying sparsity s (**Right**). The ground truth sparsity s is also plotted for comparison. See Section 2.3 for details.

$p \geq n$, and a sparse true signal $\beta^* \in \mathbb{R}^p$ with components $\beta_j^* = 2\mathrm{I}\{j \leq s\}$ for sparsity $s = 10$, and noise standard deviation $\sigma^* = 2$. For Figure 1 (Center), we consider datasets with $p = 1000$, varying n with $n \leq p$, $s = 10$, and $\sigma^* = 2$. For Figure 1 (Right), we consider datasets with $n = 10s$, $p = 1000$, and $\sigma^* = 2$ 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 δ_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 showcases the substantially lower computational cost of S^3 compared to current implementations which cost $\Omega(n^2 p)$ 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 $\tilde{M}_{\tau_0}^{-1}$ and $\tilde{M}_{\tau_1}^{-1}$ 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 $\mathcal{O}(np^2)$ computational cost and $\mathcal{O}(p^2)$ memory. Then the matrices $X_{A_t}^\top X_{A_t}$ and $X_{A_t^c}^\top X_{A_t^c}$ in (3a) – (3b) correspond to pre-computed sub-matrices of $X^\top X$, and calculating M_t using (3a) – (3b) at iteration t involves matrix addition which only requires $\mathcal{O}(n^2)$ cost. Similarly, matrices $X_{A_t}^\top \tilde{M}_{\tau_0}^{-1} X_{A_t}$ and

MCMC SAMPLER	COST	CONVERGES TO POSTERIOR
NAÏVE	$\Omega(p^3)$	$\checkmark$
STATE-OF-THE-ART	$\Omega(n^2 p)$	$\checkmark$
SKINNY GIBBS	$\Omega(\max\{n\|z_t\|_1^2, np\})$	$\times$
S^3 (LINEAR AND PROBIT)	$\mathcal{O}(\max\{n^2 p_t, np\})$	$\checkmark$
S^3 (LOGISTIC)	$\mathcal{O}(\max\{n^2 p_t, n^3, np\})$	$\checkmark$

Figure 2. Comparison of time per iteration between S^3, state-of-the-art (SOTA) exact MCMC sampler and the Skinny Gibbs approximate sampler of Narisetty et al. (2019) on synthetic binary classification datasets. See Section 3 for details.

$\boldsymbol{X}_{A_t^c}^\top \tilde{\boldsymbol{M}}_{\tau_1}^{-1} \boldsymbol{X}_{A_t^c}$ in (4a) – (4b) correspond to pre-computed sub-matrices of $\boldsymbol{X}^\top \tilde{\boldsymbol{M}}_{\tau_0}^{-1} \boldsymbol{X}$ and $\boldsymbol{X}^\top \tilde{\boldsymbol{M}}_{\tau_1}^{-1} \boldsymbol{X}$ respectively and do not need to be recalculated at iteration t. Therefore calculating $\left((\tau_1^2 - \tau_0^2)^{-1} \boldsymbol{I}_{\|\boldsymbol{z}_t\|_1} + \boldsymbol{X}_{A_t}^\top \tilde{\boldsymbol{M}}_{\tau_0}^{-1} \boldsymbol{X}_{A_t}\right)^{-1}$ or $\left((\tau_0^2 - \tau_1^2)^{-1} \boldsymbol{I}_{p - \|\boldsymbol{z}_t\|_1} + \boldsymbol{X}_{A_t^c}^\top \tilde{\boldsymbol{M}}_{\tau_1}^{-1} \boldsymbol{X}_{A_t^c}\right)^{-1}$ in (4a) – (4b) at each iteration t only requires $\mathcal{O}(\|z_t\|_1^3)$ or $\mathcal{O}((p - \|z_t\|_1)^3)$ cost respectively.

To sample from (2), consider the cases $n \le \min\{\|z_t\|_1, p - \|z_t\|_1\}$ and $\min\{\|z_t\|_1, p - \|z_t\|_1\} < n$ separately. When $n \le \min\{\|z_t\|_1, p - \|z_t\|_1\}$, we calculate $\boldsymbol{M}_t^{-1}$ by directly inverting the calculated matrix $\boldsymbol{M}_t$ from (3), which requires $\mathcal{O}(n^3)$ cost. When $\min\{\|z_t\|_1, p - \|z_t\|_1\} < n$, we avoid calculating M_t^{-1} explicitly and instead calculate the matrix vector product $M_t^{-1}(\frac{1}{\sigma_t}\boldsymbol{y} - \boldsymbol{v})$ in Algorithm 1 right-to-left, using whichever expression in (4a) – (4b) has minimal computational cost. Overall, now the Gibbs samplers for linear and probit regression require only $\mathcal{O}(\max\{\min\{\|z_t\|_1, p - \|z_t\|_1, n\}^3, np\})$ computational cost at iteration t. This provides lower computational cost for S^3 linear and probit regression whenever $\min\{\|z_t\|_1, p - \|z_t\|_1, n\}^3 < n^2 p_t$. A similar extension for logistic regression requires only $\mathcal{O}(\max\{n^3, np\})$ 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 $\|\boldsymbol{z}_t\|_1^2 \le np_t$ or not. However, unlike S^3, the Skinny Gibbs sampler does not converge to the correct posterior distribution.

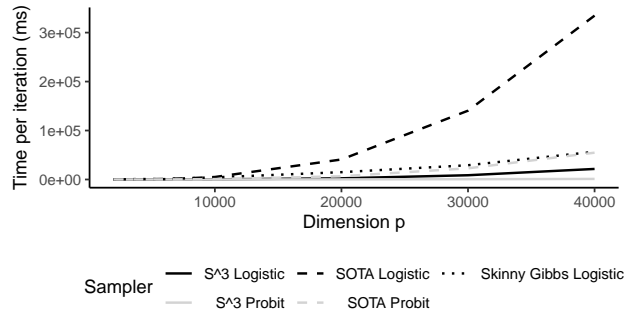To assess the practical impact of computational cost and asymptotic bias, Figures 2 and 3 compares the numerical runtimes and statistical performance of S^3 with the SOTA sampler and the Skinny Gibbs sampler. For the Skinny Gibbs sampler, we use the skinnybasad R package of Narisetty et al. (2019), which implements only logistic regression. We consider synthetically generated datasets with a true signal $\boldsymbol{\beta}^* \in \mathbb{R}^p$ where $\beta_j^* = 2\mathrm{I}\{j \le s\}$ for sparsity s. We consider datasets with $n = 10s$ observations and $p = 100s$ covariates for varying sparsity $s \ge 1$. 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, $p = 40000$ 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 1100ms 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
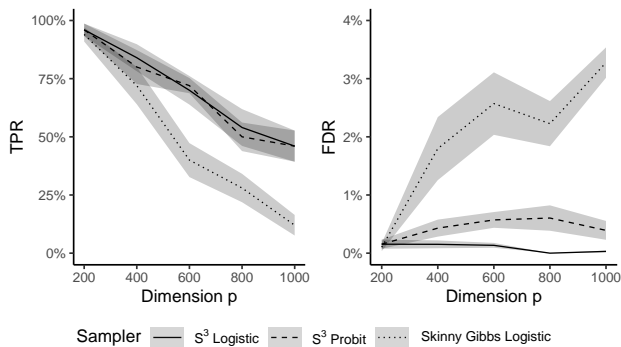
Figure 3. Average true positive rate (TPR) and false discovery rate (FDR) of S^3 and the Skinny Gibbs approximate sampler (Narisetty et al., 2019) across 20 independently generated datasets with one standard error bars. See Section 3 for details.

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

DATASET	n	p	RESPONSE TYPE
BOROVECKI	31	22283	BINARY
CHIN	118	22215	BINARY
CHOWDARY	104	22283	BINARY
GORDON	181	12533	BINARY
LYMPH	148	4514	BINARY
MAIZE	2266	98385	CONTINUOUS
MALWARE	373	503	BINARY
PCR	60	22575	CONTINUOUS
SYNTHETIC BINARY	1000	50000	CONTINUOUS
SYNTHETIC CONTINUOUS	1000	50000	BINARY

from S^3 for logistic and probit regression and Skinny Gibbs on synthetic binary classification datasets. To assess variable selection for signals of varying magnitude, we consider an exponentially decaying sparse true signal $\boldsymbol{\beta}^* \in \mathbb{R}^p$ such that $\beta_j^* = 2^{\frac{9-j}{4}}$ for $j \leq s$ and $\beta_j^* = 0$ for $j > s$ for sparsity s. The corresponding synthetically generated datasets have $n = 100$ observations, varying number of covariates p with $p \geq n$, and sparsity $s = 5$. For each synthetic dataset, we implement S^3 for logistic and probit regression and the Skinny Gibbs sampler for 5000 iterations with a burn-in of 1000 iterations and calculate the TPR and FDR from the samples. We use the same prior hyperparameters for all the algorithms, which are chosen according to Narisetty et al. (2019). Additional experimental details are included in Appendices C. The SOTA sampler is not shown in Figure 3, as SOTA and S^3 are alternative implementations of the same Gibbs sampler and by definition have the same statistical performance. Figure 3 shows that in higher dimensions, samples from S^3 yield significantly higher TPR and lower FDR than the Skinny Gibbs sampler. We observe similar results for other choices of prior hyperparameters, which give S^3 to either have comparable or more favorable statistical performance to the Skinny Gibbs sampler.

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.

4. Applications

We now examine the benefits of S^3 on a diverse suite of regression and binary classification datasets. Table 2 sum-

marizes 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 $\pi_j \triangleq \mathbb{P}_\pi(z_j = 1)$ are estimated by $\hat{\pi}_j = \frac{1}{I(T-S)} \sum_{i=1}^{I} \sum_{t=S+1}^{T} z_{j,t}^{(i)}$, where $(\boldsymbol{z}_t^{(i)})_{t \geq 0}$ 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 (Left) plots $\hat{\pi}_j$ against j in the decreasing order of $\hat{\pi}_j$s. It shows $\hat{\pi}_j$s 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 (Left) shows that the estimates using S^3 and the SOTA sampler are indistinguishable. Furthermore, in this example all components of z_t 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.

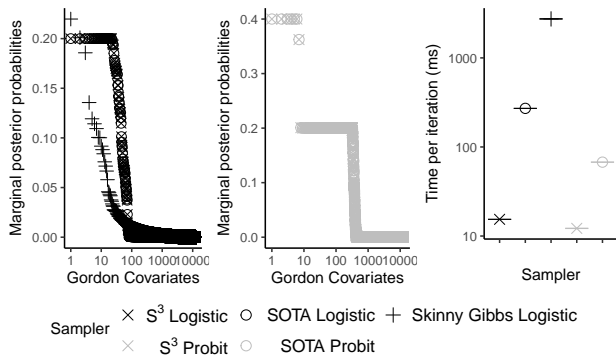Figure 5, plotted with the y-axis on the log-scale, compares the runtimes of S^3 and the SOTA sampler for linear and

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 $\mathbb{P}_\pi(z_j = 1)$ 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.

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 650ms per iteration on average, which is 48 times faster than the SOTA sampler requiring 31300ms 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 S^3 applied to these datasets including effective sample size (ESS) calculations, marginal posterior probabilities, and performance under 10-fold cross-validation.

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 *p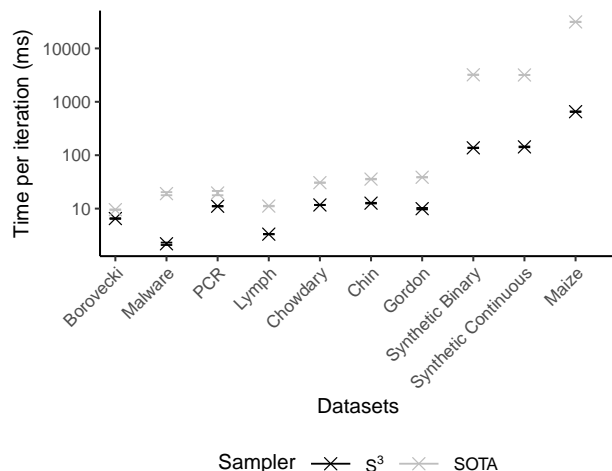oint-mass* spike-and-slab priors (e.g., Mitchell & Beauchamp, 1988; Johnson & Rossell, 2012), where a degenerate Dirac distribution about zero is chosen for the spike part, and extensions which consider the heavier-tailed Laplace distribution for the slab part instead of a Gaussian distribution (Castillo 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 alleviate 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.

References

Bai, R., Ročková, V., and George, E. I. Spike-and-slab meets LASSO: A review of the spike-and-slab LASSO. *Handbook of Bayesian Variable Selection*, 2021.

Banerjee, S., Castillo, I., and Ghosal, S. Bayesian inference in high-dimensional models. *Springer volume on Data Science*, 2021.

Barbieri, M. M. and Berger, J. O. Optimal predictive model selection. *Annals of Statistics*, 32(3):870 – 897, 2004. doi: 10.1214/009053604000000238. URL https://doi.org/10.1214/009053604000000238.

Barbieri, M. M., Berger, J. O., George, E. I., and Ročková, V. The Median Probability Model and Correlated Variables. *Bayesian Analysis*, 16(4):1085 – 1112, 2021. doi: 10.1214/20-BA1249. URL https://doi.org/10.1214/20-BA1249.

Bhadra, A., Datta, J., Polson, N. G., and Willard, B. Lasso Meets Horseshoe: A Survey. *Statistical Science*, 34 (3):405 – 427, 2019. doi: 10.1214/19-STS700. URL https://doi.org/10.1214/19-STS700.

Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. Dirichlet–Laplace Priors for Optimal Shrinkage. *Journal of the American Statistical Association*, 110(512):1479–1490, 2015. doi: 10.1080/01621459.2014.960967. URL https://doi.org/10.1080/01621459.2014.960967.

Bhattacharya, A., Chakraborty, A., and Mallick, B. K. Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika*, 103 (4):985–991, 2016. ISSN 0006-3444. doi: 10.1093/biomet/asw042. URL https://doi.org/10.1093/biomet/asw042.

Bhattacharya, S., Khare, K., and Pal, S. Geometric ergodicity of Gibbs samplers for the Horseshoe and its regularized variants. *Electronic Journal of Statistics*, 16(1):1 – 57, 2022. doi: 10.1214/21-EJS1932. URL https://doi.org/10.1214/21-EJS1932.

Biswas, N., Jacob, P. E., and Vanetti, P. Estimating convergence of markov chains with l-lag couplings. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/aec851e565646f6835e915293381e20a-Paper.pdf.

Biswas, N., Bhattacharya, A., Jacob, P. E., and Johndrow, J. E. Coupling-based convergence assessment of some gibbs samplers for high-dimensional bayesian regression with shrinkage priors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2022. doi: 10.1111/rssb.12495. URL https://doi.org/10.1111/rssb.12495.

Bogdan, M., van den Berg, E., Sabatti, C., Su, W., and Candès, E. J. SLOPE—Adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9 (3):1103 – 1140, 2015. doi: 10.1214/15-AOAS842. URL https://doi.org/10.1214/15-AOAS842.

Carvalho, C. M., Polson, N. G., and Scott, J. G. The horseshoe estimator for sparse signals. *Biometrika*, 97 (2):465–480, 04 2010. ISSN 0006-3444. doi: 10.1093/biomet/asq017. URL https://doi.org/10.1093/biomet/asq017.

Castillo, I. and van der Vaart, A. Needles and Straw in a Haystack: Posterior concentration for possibly sparse sequences. *Annals of Statistics*, 40(4):2069 – 2101, 2012. doi: 10.1214/12-AOS1029. URL https://doi.org/10.1214/12-AOS1029.

Castillo, I., Schmidt-Hieber, J., and van der Vaart, A. Bayesian linear regression with sparse priors. *Annals of Statistics*, 43(5):1986–2018, 2015. doi: 10.1214/15-AOS1334. URL https://doi.org/10.1214/15-AOS1334.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Flegal, J. M., Hughes, J., Vats, D., Dai, N., Gupta, K., and Maji, U. *mcmcse: Monte Carlo Standard Errors for MCMC*. Riverside, CA, and Kanpur, India, 2021. R package version 1.5-0.

George, E. I. and McCulloch, R. E. Variable Selection via Gibbs Sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993. doi: 10.1080/01621459.1993.10476353. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1993.10476353.

George, E. I. and McCulloch, R. E. Approaches for Bayesian Variable Selection. *Statistica Sinica*, 7(2): 339–373, 1997. ISSN 10170405, 19968507. URL http://www.jstor.org/stable/24306083.

Gordon, G. J. G., Jensen, R. V. R., Hsiao, L.-L. L., Gullans, S. R. S., Blumenstock, J. E. J., Ramaswamy, S. S., Richards, W. G. W., Sugarbaker, D. J. D., and Bueno, R. R. Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Research*, 62(17):4963–4967, September 2002.

Guan, Y. and Stephens, M. Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *The Annals of Applied Statistics*, 5 (3):1780 – 1815, 2011. doi: 10.1214/11-AOAS455. URL https://doi.org/10.1214/11-AOAS455.

Hager, W. W. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, 1989. ISSN 00361445. URL http://www.jstor.org/stable/2030425.

Hans, C., Dobra, A., and West, M. Shotgun Stochastic Search for "Large p" Regression. *Journal of the American Statistical Association*, 102(478):507–516, 2007. doi: 10.1198/016214507000000121. URL https://doi.org/10.1198/016214507000000121.

Held, L. and Holmes, C. C. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145 – 168, 2006. doi: 10.1214/06-BA105. URL https://doi.org/10.1214/06-BA105.

Ishwaran, H. and Rao, J. S. Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730 – 773, 2005. doi: 10.1214/009053604000001147. URL https://doi.org/10.1214/009053604000001147.

Johndrow, J., Orenstein, P., and Bhattacharya, A. Scalable Approximate MCMC Algorithms for the Horseshoe Prior. *Journal of Machine Learning Research*, 21(73):1–61, 2020. URL http://jmlr.org/papers/v21/19-536.html.

Johnson, V. E. A coupling-regeneration scheme for diagnosing convergence in Markov chain Monte Carlo algorithms. *Journal of the American Statistical Association*, 93(441):238–248, 1998.

Johnson, V. E. and Rossell, D. Bayesian Model Selection in High-Dimensional Settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012. doi: 10.1080/01621459.2012.682536. URL https://doi.org/10.1080/01621459.2012.682536.

Johnstone, I. M. and Silverman, B. W. Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Annals of Statistics*, 32(4):1594 – 1649, 2004. doi: 10.1214/009053604000000030. URL https://doi.org/10.1214/009053604000000030.

Kelly, B. C. Some Aspects of Measurement Error in Linear Regression of Astronomical Data. *The Astrophysical Journal*, 665(2):1489–1506, aug 2007. doi: 10.1086/519947. URL https://doi.org/10.1086/519947.

Liang, F., Song, Q., and Yu, K. Bayesian Subset Modeling for High-Dimensional Generalized Linear Models. *Journal of the American Statistical Association*, 108(502):589–606, 2013. doi: 10.1080/01621459.2012.761942. URL https://doi.org/10.1080/01621459.2012.761942.

Liu, X., Huang, M., Fan, B., Buckler, E. S., and Zhang, Z. Iterative Usage of Fixed and Random Effect Models for Powerful and Efficient Genome-Wide Association Studies. *PLOS Genetics*, 12(2):1–24, 2016. doi: 10.1371/journal.pgen.1005767. URL https://doi.org/10.1371/journal.pgen.1005767.

Mitchell, T. J. and Beauchamp, J. J. Bayesian Variable Selection in Linear Regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988. ISSN 01621459. URL http://www.jstor.org/stable/2290129.

Narisetty, N. N. and He, X. Bayesian variable selection with shrinking and diffusing priors. *Annals of Statistics*, 42 (2):789 – 817, 2014. doi: 10.1214/14-AOS1207. URL https://doi.org/10.1214/14-AOS1207.

Narisetty, N. N., Shen, J., and He, X. Skinny Gibbs: A Consistent and Scalable Gibbs Sampler for Model Selection. *Journal of the American Statistical Association*, 114(527):1205–1217, 2019. doi: 10.1080/01621459.2018.1482754. URL https://doi.org/10.1080/01621459.2018.1482754.

O'Brien, S. M. and Dunson, D. B. Bayesian multivariate logistic regression. *Biometrics*, 60(3):739–746, 2004. doi: 10.1111/j.0006-341X.2004.00224.x. URL https://doi:10.1111/j.0006-341X.2004.00224.x.

Pal, S. and Khare, K. Geometric ergodicity for Bayesian shrinkage models. *Electronic Journal of Statistics*, 8(1): 604–645, 2014. doi: 10.1214/14-EJS896. URL https://doi.org/10.1214/14-EJS896.

Polson, N. G., Scott, J. G., and Windle, J. Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013. doi: 10.1080/01621459.2013.829001. URL https://doi.org/10.1080/01621459.2013.829001.

Qin, Q. and Hobert, J. P. Convergence complexity analysis of Albert and Chib's algorithm for Bayesian probit regression. *Annals of Statistics*, 47(4):2320–2347, 2019. doi:

10.1214/18-AOS1749. URL https://doi.org/10.1214/18-AOS1749.

Ray, K. and Szabó, B. Variational bayes for high-dimensional linear regression with sparse priors. *Journal of the American Statistical Association*, 0(0):1–12, 2021. doi: 10.1080/01621459.2020.1847121. URL https://doi.org/10.1080/01621459.2020.1847121.

Ray, K., Szabo, B., and Clara, G. Spike and slab variational bayes for high dimensional logistic regression. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14423–14434. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/a5bad363fc47f424ddf5091c8471480a-Paper.pdf.

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., McMullen, M. D., Holland, J. B., Buckler, E. S., and Gardner, C. A. Comprehensive genotyping of the USA national maize inbred seed bank. *Genome Biology*, 14(6):R55, 2013. doi: 10.1186/gb-2013-14-6-r55. URL https://doi.org/10.1186/gb-2013-14-6-r55.

Ročková, V. Bayesian estimation of sparse signals with a continuous spike-and-slab prior. *Annals of Statistics*, 46 (1):401 – 437, 2018. doi: 10.1214/17-AOS1554. URL https://doi.org/10.1214/17-AOS1554.

Scott, J. G. and Berger, J. O. Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *Annals of Statistics*, 38(5):2587 – 2619, 2010. doi: 10.1214/10-AOS792. URL https://doi.org/10.1214/10-AOS792.

Sereno, M. A Bayesian approach to linear regression in astronomy. *Monthly Notices of the Royal Astronomical Society*, 455(2):2149–2162, 11 2015. ISSN 0035-8711. doi: 10.1093/mnras/stv2374. URL https://doi.org/10.1093/mnras/stv2374.

Tadesse, M. G. and Vannucci, M. *Handbook of Bayesian Variable Selection*. Chapman and Hall/CRC, 2021. doi: 10.1201/9781003089018. URL https://doi.org/10.1201/9781003089018.

Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1): 267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Titsias, M. and Lázaro-Gredilla, M. Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/b495ce63ede0f4efc9eec62cb947c162-Paper.pdf.

Vats, D. and Knudson, C. Revisiting the Gelman–Rubin Diagnostic. *Statistical Science*, 36(4):518 – 529, 2021. doi: 10.1214/20-STS812. URL https://doi.org/10.1214/20-STS812.

Vats, D., Flegal, J. M., and Jones, G. L. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337, 2019. ISSN 0006-3444. doi: 10.1093/biomet/asz002. URL https://doi.org/10.1093/biomet/asz002.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, 16 (2):667 – 718, 2021. doi: 10.1214/20-BA1221. URL https://doi.org/10.1214/20-BA1221.

Yang, Y., Wainwright, M. J., and Jordan, M. I. On the computational complexity of high-dimensional Bayesian variable selection. *Annals of Statistics*, 44(6):2497–2532, 2016. doi: 10.1214/15-AOS1417. URL https://doi.org/10.1214/15-AOS1417.

Zeng, P. and Zhou, X. Non-parametric genetic prediction of complex traits with latent Dirichlet process regression models. *Nature Communications*, 8(1):456, 2017. doi: 10.1038/s41467-017-00470-2. URL https://doi.org/10.1038/s41467-017-00470-2.

Zhou, X., Carbonetto, P., and Stephens, M. Polygenic Modeling with Bayesian Sparse Linear Mixed Models. *PLOS Genetics*, 9(2):1–14, 02 2013. doi: 10.1371/journal.pgen.1003264. URL https://doi.org/10.1371/journal.pgen.1003264.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67 (2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x. URL https://doi.org/10.1111/j.1467-9868.2005.00503.x.

A. Proofs

Proof of Proposition 2.1. Consider Step 1 of Algorithm 2 and Algorithm 3 for probit regression. Given pre-computed matrices $\tilde{\boldsymbol{M}}_{\tau_0}, \tilde{\boldsymbol{M}}_{\tau_0}, \boldsymbol{M}_{t-1}, \boldsymbol{M}_{t-1}^{-1}$ and state $\boldsymbol{z}_{t-1}$, calculating $\boldsymbol{M}_t, \boldsymbol{M}_t^{-1}$ requires $\mathcal{O}(n^2 p_t)$ cost by (3) and (4), where $p_t = \min\{\|\boldsymbol{z}_t\|_1, p - \|\boldsymbol{z}_t\|_1, \delta_t\}$ for $\delta_t = \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_1$.

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

Given $\boldsymbol{M}_t^{-1}$, Step 2 of Algorithms 2 and 3 then requires $\mathcal{O}(np)$ cost, which arises from the matrix vector product $\boldsymbol{X}\boldsymbol{D}_t^{-\frac{1}{2}} r$ for $r \sim \mathcal{N}(0, \boldsymbol{I}_p)$ in Algorithm 1. By component-wise independence, Step 3 of Algorithms 2 and 3 costs $\mathcal{O}(p)$ and Step 4 of Algorithm 3 cost $\mathcal{O}(n)$. Step 4 of Algorithm 3 and Step 5 of Algorithm 3 for probit regression both cost $\mathcal{O}(1)$. Step 5 of Algorithm 3 for logistic regression both costs $\mathcal{O}(n)$.

This gives an overall computational cost of $\mathcal{O}(\max\{n^2 p_t, np\})$ for Algorithm 2 and Algorithm 3 for probit regression at iteration t, and a cost of $\mathcal{O}(\max\{n^2 p_t, n^3, np\})$ 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 j the random variables $z_{j,t}$ and $z_{j,t-1}$ are on $\{0,1\}$. For each j, we obtain

$$
\begin{aligned}
\mathbb{P}(z_{j,t} \neq z_{j,t-1}) &= \mathbb{P}(z_{j,t} = 1, z_{j,t-1} = 0) + \mathbb{P}(z_{j,t} = 0, z_{j,t-1} = 1) \\
&= \big(\mathbb{P}(z_{j,t} = 1) - \mathbb{P}(z_{j,t} = 1, z_{j,t-1} = 1)\big) + \big(\mathbb{P}(z_{j,t-1} = 1) - \mathbb{P}(z_{j,t} = 1, z_{j,t-1} = 1)\big) \\
&= \big(\mathbb{P}(z_{j,t} = 1) - \text{cov}(z_{j,t}, z_{j,t-1}) - \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 1)\big) + \\
&\quad \big(\mathbb{P}(z_{j,t-1} = 1) - \text{cov}(z_{j,t}, z_{j,t-1}) - \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 1)\big) \\
&= \mathbb{P}(z_{j,t} = 1)\mathbb{P}(z_{j,t-1} = 0) + \mathbb{P}(z_{j,t} = 0)\mathbb{P}(z_{j,t-1} = 1) - 2\text{cov}(z_{j,t}, z_{j,t-1}).
\end{aligned}
$$

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

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

□

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

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

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

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

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

$$
\mathbb{E}_J(z_{J,t}z_{J,t-1}) = \text{Cov}_J(z_{J,t}, z_{J,t-1}) + \mathbb{E}_J(z_{J,t})\mathbb{E}_J(z_{J,t-1}) = \frac{\rho_t \tau_t \tau_{t-1} + \|\boldsymbol{z}_t\|_1 \|\boldsymbol{z}_{t-1}\|_1}{p^2}, \tag{12}
$$

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 $(\boldsymbol{\beta}, \boldsymbol{z}, \sigma^2) \in \mathbb{R}^p \times \{0,1\}^p \times (0,\infty)$ is given by

$$
\pi(\boldsymbol{\beta}, \boldsymbol{z}, \sigma^2 | \boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y}; \boldsymbol{X}\boldsymbol{\beta}, \sigma^2)\text{InvGamma}\left(\sigma^2; \frac{a_0}{2}, \frac{b_0}{2}\right) \tag{13}
$$

$$
\prod_{j=1}^p \big(q\mathcal{N}(\beta_j; 0, \sigma^2 \tau_1^2)\big)^{z_j} \big((1-q)\mathcal{N}(\beta_j; 0, \sigma^2 \tau_0^2)\big)^{1-z_j}. \tag{14}
$$

From (13), we can calculate the conditional distributions. We have

$$\pi(\boldsymbol{\beta}|\boldsymbol{z}, \sigma^2, \boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y}; \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n) \prod_{j=1}^{p} \mathcal{N}(\beta_j; 0, \sigma^2 \tau_1^2)^{z_j} \mathcal{N}(\beta_j; 0, \sigma^2 \tau_0^2)^{1-z_j}$$

$$\propto \mathcal{N}(\boldsymbol{y}; \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n) \mathcal{N}(\boldsymbol{\beta}; 0, \sigma^2 \boldsymbol{D}^{-1}) \text{ for } \boldsymbol{D} \triangleq \operatorname{Diag}(\boldsymbol{z}\tau_1^{-2} + (\mathbf{1}_p - \boldsymbol{z})\tau_0^{-2})$$

$$\propto \exp\left\{ -\frac{1}{2\sigma^2 \boldsymbol{I}_n} \left(\boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{\beta}^\top \boldsymbol{D} \boldsymbol{\beta} \right) \right\}$$

$$\propto \mathcal{N}(\boldsymbol{\beta}; \boldsymbol{\Sigma}^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \sigma^2 \boldsymbol{\Sigma}^{-1}) \text{ for } \boldsymbol{\Sigma} = \boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{D},$$

$$\pi(\boldsymbol{z}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{y}) \propto \prod_{j=1}^{p} \left(q\mathcal{N}(\beta_j; 0, \sigma^2 \tau_1^2) \right)^{z_j} \left((1-q)\mathcal{N}(\beta_j; 0, \sigma^2 \tau_0^2) \right)^{1-z_j}$$

$$\propto \prod_{j=1}^{p} \operatorname{Bernoulli}\left(z_i; \frac{q\mathcal{N}(\beta_j; 0, \sigma^2 \tau_1^2)}{q\mathcal{N}(\beta_j; 0, \sigma^2 \tau_1^2) + (1-q)\mathcal{N}(\beta_j; 0, \sigma^2 \tau_0^2)} \right), \text{ and}$$

$$\pi(\sigma^2|\boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y}; \boldsymbol{X}\boldsymbol{\beta}, \sigma^2) \operatorname{InvGamma}\left(\sigma^2; \frac{a_0}{2}, \frac{b_0}{2} \right) \mathcal{N}(\boldsymbol{\beta}; 0, \sigma^2 \boldsymbol{D}^{-1})$$

$$\propto \left(\frac{1}{\sigma^2} \right)^{\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} \left(\frac{1}{\sigma^2} \right)^{\frac{a_0}{2}} \exp\left\{ -\frac{1}{2\sigma^2} b_0 \right\} \left(\frac{1}{\sigma^2} \right)^{\frac{p}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \boldsymbol{\beta}^\top \boldsymbol{D} \boldsymbol{\beta} \right\}$$

$$\propto \operatorname{InvGamma}\left(\sigma^2; \frac{a_0 + n + p}{2}, \frac{b_0 + \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \boldsymbol{\beta}^\top \boldsymbol{D} \boldsymbol{\beta}}{2} \right)$$

as given in Algorithm 2.

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

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

$$z_j \overset{i.i.d.}{\sim} \operatorname{Bernoulli}(q) \quad \text{for all } j = 1, ..., p$$

$$\beta_j|z_j \overset{ind}{\sim} (1 - z_j)\mathcal{N}(0, \tau_0^2) + z_j \mathcal{N}(0, \tau_1^2) \quad \text{for all } j = 1, ..., p \tag{15}$$

$$\tilde{y}_i|\boldsymbol{\beta} \overset{ind}{\sim} \mathcal{N}(\boldsymbol{x_i}^\top \boldsymbol{\beta}, 1) \quad \text{for all } i = 1, ..., n$$

$$y_i = \mathrm{I}\{\tilde{y}_i > 0\} \quad \text{for all } i = 1, ..., n.$$

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

$$\pi(\boldsymbol{\beta}, \boldsymbol{z}, \tilde{\boldsymbol{y}}|\boldsymbol{y}) \propto \left(\prod_{i=1}^{n} \mathrm{I}\{\mathrm{I}\{\tilde{y}_i > 0\} = y_i\} \mathcal{N}(\tilde{y}_i; \boldsymbol{x_i}^\top \boldsymbol{\beta}, 1) \right)$$

$$\prod_{j=1}^{p} \left(q\mathcal{N}(\beta_j; 0, \tau_1^2) \right)^{z_j} \left((1-q)\mathcal{N}(\beta_j; 0, \tau_0^2) \right)^{1-z_j}. \tag{16}$$

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

$$\pi(\boldsymbol{\beta}|\boldsymbol{z}, \tilde{\boldsymbol{y}}, \boldsymbol{y}) \propto \mathcal{N}(\tilde{\boldsymbol{y}}; \boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{I}_n) \mathcal{N}(\boldsymbol{\beta}; 0, \boldsymbol{D}^{-1}) \text{ for } \boldsymbol{D} \triangleq \operatorname{Diag}(\boldsymbol{z}\tau_1^{-2} + (\mathbf{1}_p - \boldsymbol{z})\tau_0^{-2})$$

$$\propto \mathcal{N}(\boldsymbol{\beta}; \boldsymbol{\Sigma}^{-1} \boldsymbol{X}^\top \tilde{\boldsymbol{y}}, \boldsymbol{\Sigma}^{-1}) \text{ for } \boldsymbol{\Sigma} = \boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{D},$$

$$\pi(\boldsymbol{z}|\boldsymbol{\beta}, \tilde{\boldsymbol{y}}, \boldsymbol{y}) \propto \prod_{j=1}^{p} \operatorname{Bernoulli}\left(z_i; \frac{q\mathcal{N}(\beta_j; 0, \tau_1^2)}{q\mathcal{N}(\beta_j; 0, \tau_1^2) + (1-q)\mathcal{N}(\beta_j; 0, \tau_0^2)} \right), \text{ and}$$

$$\pi(\tilde{\boldsymbol{y}}|\boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{y}) \propto \prod_{i=1}^{n} \mathcal{N}(\tilde{y}_i; \boldsymbol{x_i}^\top \boldsymbol{\beta}, 1) \mathrm{I}\{\mathrm{I}\{\tilde{y}_i > 0\} = y_i\}.$$

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 $i = 1, ..., n$, $\mathbb{P}(y_i = 1|\boldsymbol{x_i}, \boldsymbol{\beta}) = 1 - \mathbb{P}(y_i = 0|\boldsymbol{x_i}, \boldsymbol{\beta}) = \frac{\exp(\boldsymbol{x_i}^\top \boldsymbol{\beta})}{1 + \exp(\boldsymbol{x_i}^\top \boldsymbol{\beta})}$ for $\boldsymbol{x_i}^\top$ the i-th row of the design matrix $\boldsymbol{X}$. We obtain $y_i = \mathrm{I}\{\tilde{y}_i > 0\}$ where $\tilde{y}_i \overset{ind}{\sim} \text{Logistic}(\boldsymbol{x_i}^\top \boldsymbol{\beta}, 1)$, corresponding to the logistic distribution centered about $\boldsymbol{x_i}^\top \boldsymbol{\beta}$ 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 $\text{Logistic}(\boldsymbol{x_i}^\top \boldsymbol{\beta}, 1)$ with $\boldsymbol{x_i}^\top \boldsymbol{\beta} + w t_\nu$, where t_ν denotes a t-distribution with ν degrees of freedom and w is a multiplicative factor. The constants $\nu \triangleq 7.3$ and $w^2 \triangleq \frac{\pi^2(\nu-2)}{3\nu}$ 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

$$\tilde{y}_i|\boldsymbol{x_i}, \boldsymbol{\beta}, \tilde{\sigma}_i \sim \mathcal{N}(\boldsymbol{x_i}^\top \boldsymbol{\beta}, \tilde{\sigma}_i^2), \quad \tilde{\sigma}_i^2 \sim \text{InvGamma}\left(\frac{v}{2}, \frac{w^2 v}{2}\right), \tag{17}$$

where each $\tilde{\sigma}_i^2$ is an augmented variable. The Bayesian logistic regression model is then given by

$$
\begin{aligned}
z_j &\overset{i.i.d.}{\sim} \text{Bernoulli}(q) \quad \text{for all } j = 1, ..., p \\
\beta_j|z_j &\overset{ind}{\sim} (1 - z_j)\mathcal{N}(0, \tau_0^2) + z_j\mathcal{N}(0, \tau_1^2) \quad \text{for all } j = 1, ..., p \\
\tilde{\sigma}_i^2 &\overset{i.i.d.}{\sim} \text{InvGamma}(\frac{\nu}{2}, \frac{w^2\nu}{2}) \\
\tilde{y}_i|\boldsymbol{\beta}, \tilde{\sigma}_i^2 &\overset{ind}{\sim} \mathcal{N}(\boldsymbol{x_i}^\top \boldsymbol{\beta}, \tilde{\sigma}_i^2) \quad \text{for all } i = 1, ..., n \\
y_i &= \mathrm{I}\{\tilde{y}_i > 0\} \quad \text{for all } i = 1, ..., n.
\end{aligned}
\tag{18}
$$

Let $\tilde{\boldsymbol{\sigma}}^2$ denote the vector with entries $\tilde{\sigma}_i^2$ for $i = 1, ..., n$. For the prior and likelihood in (18), the posterior density of $(\boldsymbol{\beta}, \boldsymbol{z}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\sigma}}^2)$ on $\mathbb{R}^p \times \{0, 1\}^p \times \mathbb{R}^n \times (0, \infty)^n$ is given by

$$
\begin{aligned}
\pi(\boldsymbol{\beta}, \boldsymbol{z}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\sigma}}^2|\boldsymbol{y}) &\propto \prod_{j=1}^p \left(q\mathcal{N}(\beta_j; 0, \tau_1^2)\right)^{z_j} \left((1-q)\mathcal{N}(\beta_j; 0, \tau_0^2)\right)^{1-z_j} \\
&\quad \prod_{i=1}^n \mathrm{I}\{\mathrm{I}\{\tilde{y}_i > 0\} = y_i\}\mathcal{N}(\tilde{y}_i; \boldsymbol{x_i}^\top \boldsymbol{\beta}, \tilde{\sigma}_i^2)\text{InvGamma}\left(\tilde{\sigma}_i^2; \frac{\nu}{2}, \frac{w^2\nu}{2}\right).
\end{aligned}
\tag{19}
$$

From (19), we can calculate the conditional distributions. Let $\boldsymbol{W} = \text{Diag}(\tilde{\boldsymbol{\sigma}}^2)$. We obtain

$$
\begin{aligned}
\pi(\boldsymbol{\beta}|\boldsymbol{z}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\sigma}}^2, \boldsymbol{y}) &\propto \mathcal{N}(\tilde{\boldsymbol{y}}; \boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{W})\mathcal{N}(\boldsymbol{\beta}; 0, \boldsymbol{D}^{-1}) \text{ for } \boldsymbol{D} \triangleq \text{Diag}(\boldsymbol{z}\tau_1^{-2} + (\mathbf{1}_p - \boldsymbol{z})\tau_0^{-2}) \\
&\propto \mathcal{N}(\boldsymbol{\beta}; \boldsymbol{\Sigma}^{-1}\boldsymbol{X}^\top \boldsymbol{W}^{-1}\tilde{\boldsymbol{y}}, \boldsymbol{\Sigma}^{-1}) \text{ for } \boldsymbol{\Sigma} = \boldsymbol{X}^\top \boldsymbol{W}^{-1}\boldsymbol{X} + \boldsymbol{D}, \\
\pi(\boldsymbol{z}|\boldsymbol{\beta}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\sigma}}^2, \boldsymbol{y}) &\propto \prod_{j=1}^p \text{Bernoulli}\left(z_i; \frac{q\mathcal{N}(\beta_j; 0, \tau_1^2)}{q\mathcal{N}(\beta_j; 0, \tau_1^2) + (1-q)\mathcal{N}(\beta_j; 0, \tau_0^2)}\right), \\
\pi(\tilde{\boldsymbol{y}}|\boldsymbol{\beta}, \boldsymbol{z}, \tilde{\boldsymbol{\sigma}}^2, \boldsymbol{y}) &\propto \prod_{i=1}^n \mathcal{N}(\tilde{y}_i; \boldsymbol{x_i}^\top \boldsymbol{\beta}, \tilde{\sigma}_i^2)\mathrm{I}\{\mathrm{I}\{\tilde{y}_i > 0\} = y_i\}, \text{ and} \\
\pi(\tilde{\boldsymbol{\sigma}}^2|\boldsymbol{\beta}, \boldsymbol{z}, \tilde{\boldsymbol{y}}, \boldsymbol{y}) &\propto \prod_{i=1}^n \mathcal{N}(\tilde{y}_i; \boldsymbol{x_i}^\top \boldsymbol{\beta}, \tilde{\sigma}_i^2)\text{InvGamma}\left(\tilde{\sigma}_i^2; \frac{\nu}{2}, \frac{w^2\nu}{2}\right) \\
&\propto \prod_{i=1}^n \text{InvGamma}\left(\tilde{\sigma}_i^2; \frac{\nu+1}{2}, \frac{w^2\nu + (\tilde{y}_i - \boldsymbol{x_i}^\top \boldsymbol{\beta})^2}{2}\right).
\end{aligned}
$$

as required for logistic regression in Algorithm 3.

Algorithm 4 An $\Omega(n^2 p)$ sampler of (20) (Bhattacharya et al., 2016)

Sample $\boldsymbol{r} \sim \mathcal{N}(0, \boldsymbol{I}_p)$, $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{I}_n)$.

Set $\boldsymbol{u} = \boldsymbol{D}_t^{-\frac{1}{2}} \boldsymbol{r}$ and calculate $\boldsymbol{v} = \boldsymbol{W}_t^{-1/2} \boldsymbol{X} \boldsymbol{u} + \boldsymbol{\xi}$.

Set $\boldsymbol{v}^* = \boldsymbol{M}_t^{-1}(\boldsymbol{W}_t^{-1/2}\tilde{\boldsymbol{y}} - \boldsymbol{v})$ for $\boldsymbol{M}_t = \boldsymbol{I}_n + \boldsymbol{W}_t^{-1/2} \boldsymbol{X} \boldsymbol{D}_t^{-1} \boldsymbol{X}^\top \boldsymbol{W}_t^{-1/2}$.

Return $\boldsymbol{\beta} = \boldsymbol{u} + \boldsymbol{D}_t^{-1} \boldsymbol{X}^\top \boldsymbol{W}_t^{-1/2} \boldsymbol{v}^*$.

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 $\boldsymbol{\beta} \in \mathbb{R}^p$. This is given by

$$\boldsymbol{\beta}_{t+1}|\boldsymbol{z}_t, \tilde{\boldsymbol{\sigma}}_t^2 \sim \mathcal{N}\big(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{X}^\top \boldsymbol{W}_t^{-1}\tilde{\boldsymbol{y}}, \boldsymbol{\Sigma}_t^{-1}\big) \quad \text{for} \quad \boldsymbol{\Sigma}_t = \boldsymbol{X}^\top \boldsymbol{W}_t^{-1}\boldsymbol{X} + \boldsymbol{D}_t, \tag{20}$$

where t indexes the iteration of the Markov chain, $\boldsymbol{W}_t$ is the diagonal matrix with the vector $\tilde{\boldsymbol{\sigma}}_t^2$ populating its diagonal elements, and $\boldsymbol{D}_t$ is the diagonal matrix with the vector $\boldsymbol{z}_t \tau_1^{-2} + (\boldsymbol{1}_p - \boldsymbol{z}_t)\tau_0^{-2}$ 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, S³ for logistic regression uses pre-computation to reduce the computational cost of Algorithm 4. Using the notation from Section 2.2 with $\boldsymbol{M}_t \triangleq \boldsymbol{I}_n + \boldsymbol{W}_t^{-1/2}\boldsymbol{X}\boldsymbol{D}_t^{-1}\boldsymbol{X}^\top\boldsymbol{W}_t^{-1/2}$, we note

$$\boldsymbol{M}_t = \boldsymbol{I}_n + \boldsymbol{W}_t^{-1/2}\big(\tilde{\boldsymbol{M}}_{\tau_0} - \boldsymbol{I}_n + (\tau_1^2 - \tau_0^2)\boldsymbol{X}_{A_t^c}\boldsymbol{X}_{A_t^c}^T\big)\boldsymbol{W}_t^{-1/2} \tag{21a}$$

$$= \boldsymbol{I}_n + \boldsymbol{W}_t^{-1/2}\big(\tilde{\boldsymbol{M}}_{\tau_1} - \boldsymbol{I}_n + (\tau_0^2 - \tau_1^2)\boldsymbol{X}_{A_t^c}\boldsymbol{X}_{A_t^c}^T\big)\boldsymbol{W}_t^{-1/2} \tag{21b}$$

$$= \boldsymbol{I}_n + \boldsymbol{W}_t^{-1/2}\big(\boldsymbol{W}_{t-1}^{1/2}(\boldsymbol{M}_{t-1} - \boldsymbol{I}_n)\boldsymbol{W}_{t-1}^{1/2} + \boldsymbol{X}_{\Delta_t}\boldsymbol{C}_{\Delta_t}\boldsymbol{X}_{\Delta_t}^T\big)\boldsymbol{W}_t^{-1/2}. \tag{21c}$$

In (21a) – (21c), calculating the matrix products $\boldsymbol{X}_{A_t}\boldsymbol{X}_{A_t}^\top$, $\boldsymbol{X}_{A_t^c}\boldsymbol{X}_{A_t^c}^\top$, and $\boldsymbol{X}_{\Delta_t}\boldsymbol{C}_{\Delta_t}\boldsymbol{X}_{\Delta_t}^\top$ requires $\mathcal{O}(n^2\|\boldsymbol{z}_t\|_1)$, $\mathcal{O}(n^2(p - \|\boldsymbol{z}_t\|_1))$, and $\mathcal{O}(n^2\delta_t)$ cost respectively. Given $\tilde{\boldsymbol{M}}_{\tau_0}$, $\tilde{\boldsymbol{M}}_{\tau_1}$, $\boldsymbol{M}_{t-1}$, and $\boldsymbol{z}_{t-1}$, we evaluate whichever matrix product in (21a) – (21c) has minimal computational cost and thereby calculate $\boldsymbol{M}_t$ at the reduced cost of $\mathcal{O}(n^2 p_t)$ where $p_t \triangleq \min\{\|\boldsymbol{z}_t\|_1, p - \|\boldsymbol{z}_t\|_1, \delta_t\}$. To calculate $\boldsymbol{M}_t^{-1}$, we calculate $\boldsymbol{M}_t^{-1}$ by directly inverting the calculated matrix $\boldsymbol{M}_t$, which requires $\mathcal{O}(n^3)$ cost. Overall, this strategy reduces the computational cost of calculating the matrices $\boldsymbol{M}_t$ and $\boldsymbol{M}_t^{-1}$ from $\Omega(n^2 p)$ to $\mathcal{O}(\max\{n^2 p_t, n^3\})$.

Extensions to Scalable Spike-and-Slab for logistic regression. Suppose the matrices $\boldsymbol{X}^\top \boldsymbol{X}$ is pre-computed. This initial step requires $\mathcal{O}(np^2)$ computational cost and $\mathcal{O}(p^2)$ memory. Then the matrices $\boldsymbol{X}_{A_t}^\top \boldsymbol{X}_{A_t}$ and $\boldsymbol{X}_{A_t^c}^\top \boldsymbol{X}_{A_t^c}$ in (21a) – (21b) correspond to pre-computed sub-matrices of $\boldsymbol{X}^\top \boldsymbol{X}$, and calculating $\boldsymbol{M}_t$ using (21a) – (21b) each iteration t involves matrix addition and diagonal matrix multiplication which only requires $\mathcal{O}(n^2)$ cost. To sample from (20), we calculate $\boldsymbol{M}_t^{-1}$ by directly inverting the calculated matrix $\boldsymbol{M}_t$ from (3), which requires $\mathcal{O}(n^3)$ cost. Overall, now the Gibbs samplers for logistic regression requires $\mathcal{O}(\max\{n^3, np\})$ computational cost at iteration t, which is an improvement compared to S³.

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 $\tau_0^2 = \frac{1}{n}$, $\tau_1^2 = \max\{\frac{p^{2.1}}{100n}, 1\}$ and $q = \mathbb{P}(z_j = 1)$ such that $\mathbb{P}(\sum_{j=1}^p \mathrm{I}\{z_j = 1\} > K) = 0.1$ 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 β_j^* that are correctly selected respectively. They are calculated as $\frac{1}{s}\sum_{j=1}^s \mathrm{I}\{\mathbb{P}_\pi(z_j = 1) > 0.5\}$ and $\frac{1}{p-s}\sum_{j=s+1}^p \mathrm{I}\{\mathbb{P}_\pi(z_j = 1) > 0.5\}$ respectively, where the marginal posterior probabilities $\pi_j \triangleq \mathbb{P}_\pi(z_j = 1)$ are estimated by $\hat{\pi}_j \triangleq \frac{1}{4000}\sum_{t=1001}^{5000} z_{j,t}$ for sample points $(\boldsymbol{z}_t)_{t \geq 0}$ generated using S³ 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 $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ such that each

$[\boldsymbol{X}]_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ 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 $\boldsymbol{\beta}^* \in \mathbb{R}^p$ such that $\beta_j^* = 2\mathrm{I}\{j \leq s\}$, where s is the sparsity parameter corresponding to the number of non-zero components. Given $\boldsymbol{X}$ and $\boldsymbol{\beta}^*$, we generate $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^* + \sigma^*\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}_n)$, where $\sigma^* = 2$ is the Gaussian noise standard deviation.

Synthetic binary response dataset in Section 3. In Figures 2 and 3, synthetic binary classification datasets are considered. For number of observations n and number of covariates p, we generate a design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ such that each $[\boldsymbol{X}]_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ 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 $\boldsymbol{\beta}^* \in \mathbb{R}^p$ such that $\beta_j^* = 2^{\frac{9-j}{4}}\mathrm{I}\{j \leq s\}$, where s is the sparsity parameter corresponding to the number of non-zero components. Given $\boldsymbol{X}$ and $\boldsymbol{\beta}^*$, we generate $y_i = \mathrm{I}\{\tilde{y}_i > 0\}$ for $\tilde{y}_i \sim \mathrm{Logistic}(\boldsymbol{x}_i^\top \boldsymbol{\beta}^*, 1)$ for $i = 1..., n$, where $\boldsymbol{x}_i^\top$ is the i-th row of $\boldsymbol{X}$ and $\mathrm{Logistic}(\boldsymbol{x}_i^\top \boldsymbol{\beta}^*, 1)$ is the Logistic distribution with mean $\boldsymbol{x}_i^\top \boldsymbol{\beta}^*$ and scale parameter 1.

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 Borovecki, Chowdary, Chin and Gordon datasets are all high-dimensional microarray datasets. They are publicly available on the `datamicroarray` package in R. The Borovecki dataset has $n = 31$ observations with binary responses and $p = 22283$ covariates. The Chowdary dataset has $n = 104$ observations with binary responses and $p = 22283$ covariates. The Chin dataset has $n = 118$ observations with binary responses and $p = 22215$ covariates. The Gordon dataset has $n = 181$ observations with binary responses and $p = 12533$ covariates.

The PCR GWAS dataset has $n = 60$ observations with continuous responses and $p = 22575$ covariates, and is publicly available on `www.ncbi.nlm.nih.gov/geo` (accession number $GSE3330$). The Lymph Node GWAS dataset has $n = 148$ observations with binary responses and $p = 4514$ covariates, and has been previously considered (Hans et al., 2007; Liang et al., 2013; Narisetty et al., 2019). The Maize GWAS dataset has $n = 2266$ observations with continuous responses and $p = 98385$ covariates, and has been previously considered (Romay et al., 2013; Liu et al., 2016; Zeng & Zhou, 2017). The Lymph Node GWAS and the Maize GWAS datasets are not publicly available.

The synthetic continuous dataset has $n = 1000$ observations and $p = 50000$ covariates. The design matrix $\boldsymbol{X}$ is generates such that each $[\boldsymbol{X}]_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ 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. The true signal $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is chosen such that $\beta_j^* = 2^{\frac{9-j}{4}}\mathrm{I}\{j \leq s\}$, where s is the sparsity parameter corresponding to the number of non-zero components. Given $\boldsymbol{X}$ and $\boldsymbol{\beta}^*$, we generate $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^* + \sigma^*\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}_n)$, where $\sigma^* = 2$ is the Gaussian noise standard deviation. The synthetic binary classification dataset is generated as in Section 3, with $n = 1000$ observations and $p = 50000$ covariates.

E. Additional Experiments

Variable selection performance as a function of time or number of iterations. Figure 6 plots the average true positive rate (TPR) and the false discovery rate (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 $\boldsymbol{\beta}^* \in \mathbb{R}^p$ such that $\beta_j^* = 2^{\frac{9-j}{4}}$ 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 $\pi_j \triangleq \mathbb{P}_\pi(z_j = 1)$ now estimated by $\hat{\pi}_j \triangleq \frac{1}{T-999}\sum_{t=1000}^{T} z_{j,t}$ for a burn-in of 1000, a varying chain length $T \geq 1000$, and sample points $(\boldsymbol{z}_t)_{t \geq 0}$ generated using S^3 or Skinny Gibbs. We use the same prior hyperparameters for all the algorithms, which are chosen according to Narisetty 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
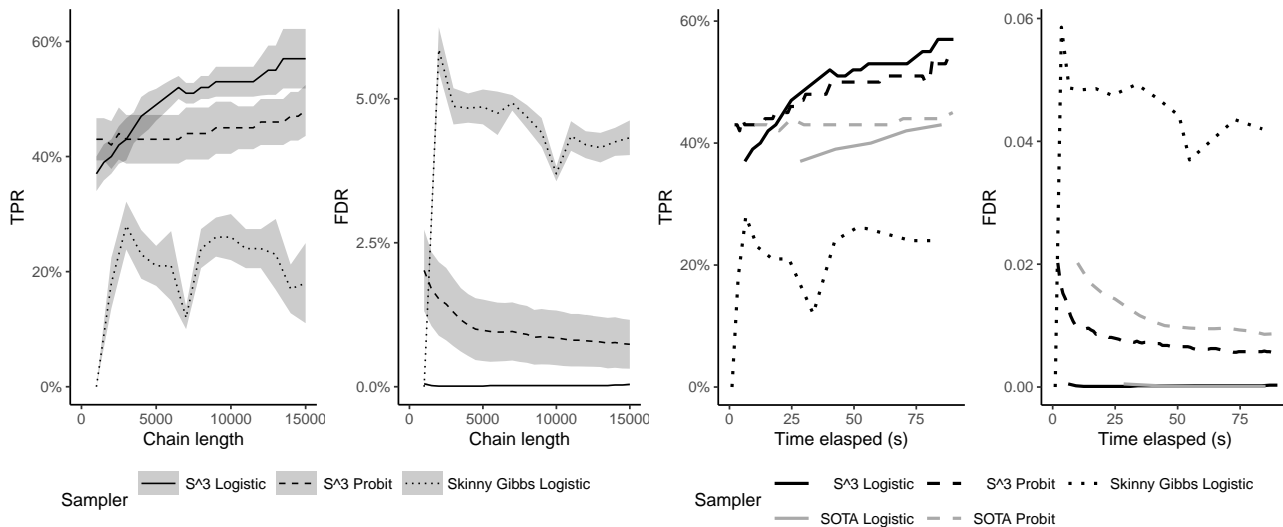
Figure 6. Avrage rue positive rate (TPR) and false discovery rate (FDR) plotted against the number of iterations and the total time elapsed in seconds. We consider S^3, SOTA, and the Skinny Gibbs approximate sampler (Narisetty et al., 2019) applied to a synthetic binary classification dataset with $n = 200$ observations and $p = 1000$ covariates. The TPR and FDR are averaged over 10 independent chains, and one standard error bars are shown on the left and center-left plots and omitted on the right and center-right plots for visibility. 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. See Section E for details.

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 mcmcse 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 β 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 $\hat{\pi}_j$s, 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 (other than possible numerical discrepancies, as discussed in Section 4).

Figures 8 – 14 (Center) show the average time taken per iteration with one standard error bars for S^3, the SOTA sampler, and the Skinny Gibbs sampler based on 5 independent chains of length 10000 iterations.

Figures 8 – 14 (Right) show the 10-fold cross-validation average root-mean-square error (RMSE) against the total time elapsed to run one S^3 and one SOTA chain. To compute this evaluation, we partition the observed dataset into 10 folds uniformly at random and, for each fold k, 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 $\frac{1}{10} \sum_{k=1}^{10} r_k$, where r_k is the RMSE for the kth 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 kth fold, $\hat{y}_i \triangleq \frac{1}{T-1000} \sum_{t=1001}^{T} x_i^T \beta_t$ are the predicted responses, and $(\beta_t)_{t \geq 0}$ are samples from S^3 and SOTA targeting the posterior distribution of the kth training set. For datasets with binary valued responses, the quantities
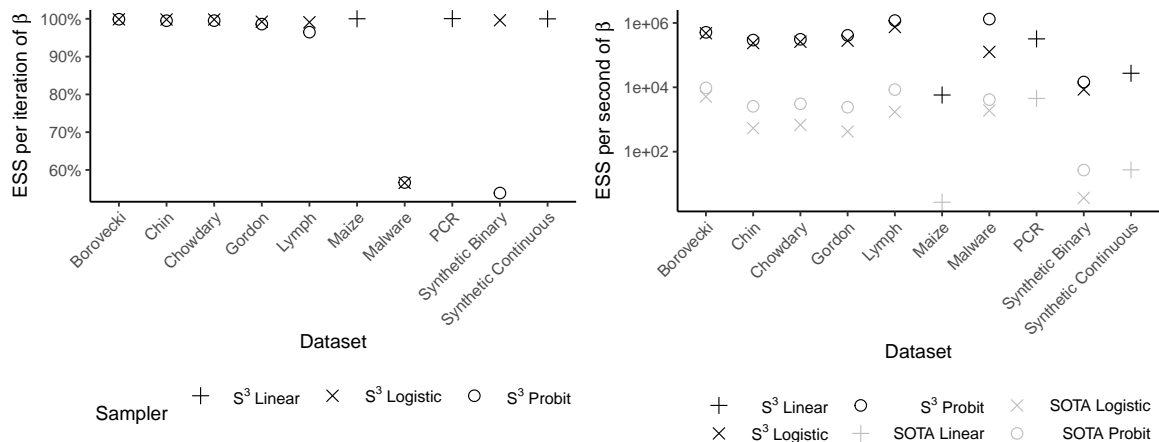
Figure 7. Effective sample size (ESS) per iteration and per second of S^3 and the SOTA sampler for some of the datasets in Section 4. The ESS is calculated using one S^3 chain of length 10000 iterations with a burn-in of 1000 iterations. The ESS per iteration of the SOTA sampler is omitted from the Left plot as it implements the same Gibbs sampler as S^3.
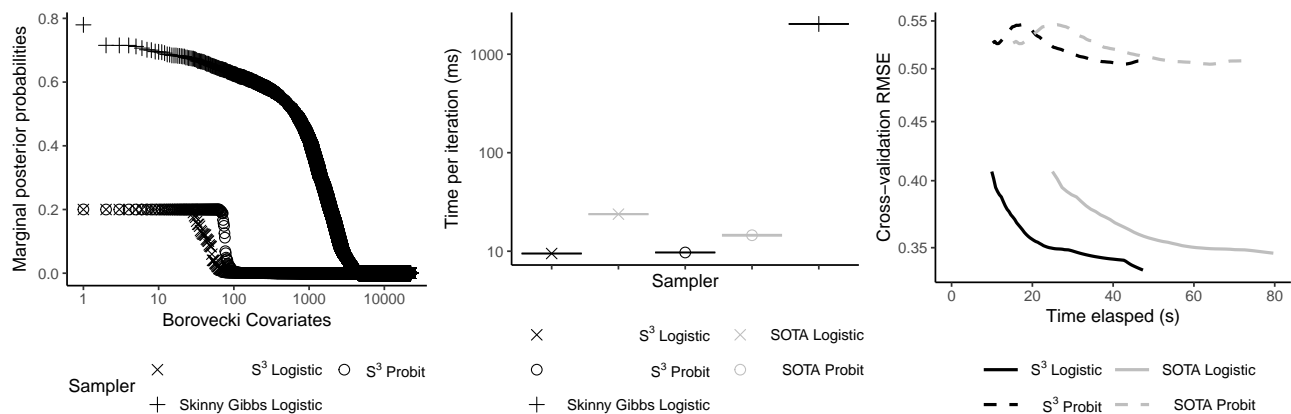

Figure 8. Borovecki dataset with $n = 31$ observations, $p = 22283$ covariates and binary valued responses.

r_k are calculated as $\left(\frac{1}{|D_k|} \sum_{i \in D_k} (y_i - \hat{p}_i)^2\right)^{1/2}$, where D_k is the kth 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 kth 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.
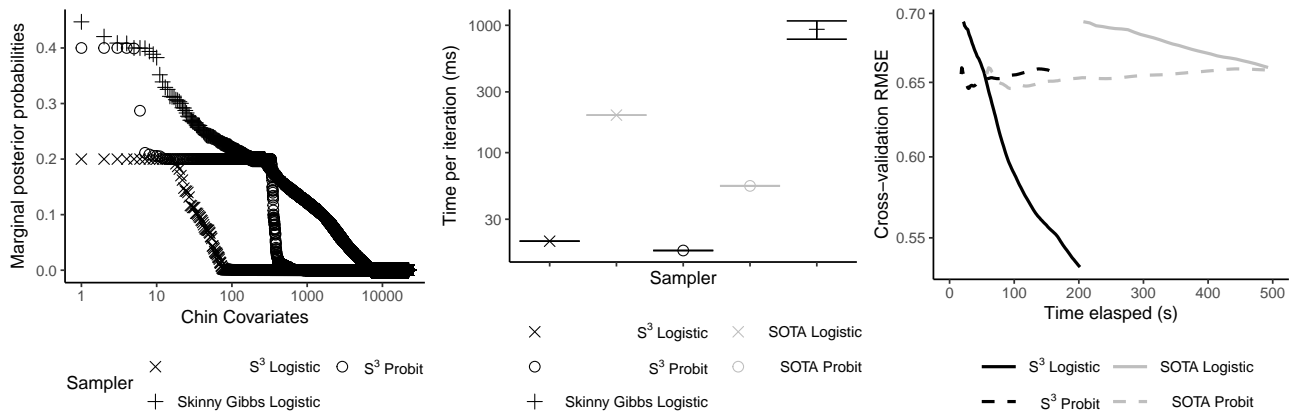
Figure 9. Chin dataset with $n = 118$ observations, $p = 22215$ covariates and binary valued responses.
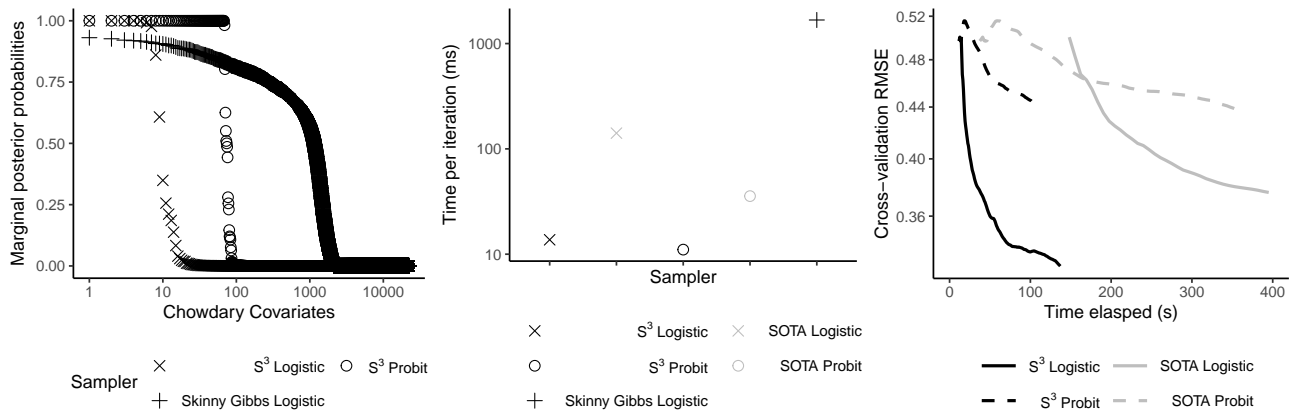

Figure 10. Chowdary dataset with $n = 104$ observations, $p = 22283$ covariates and binary valued responses.
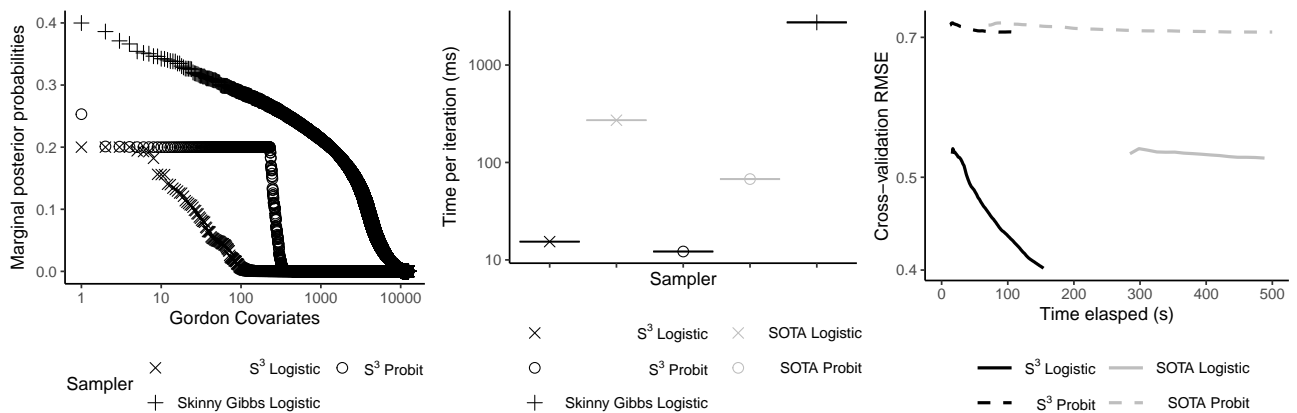

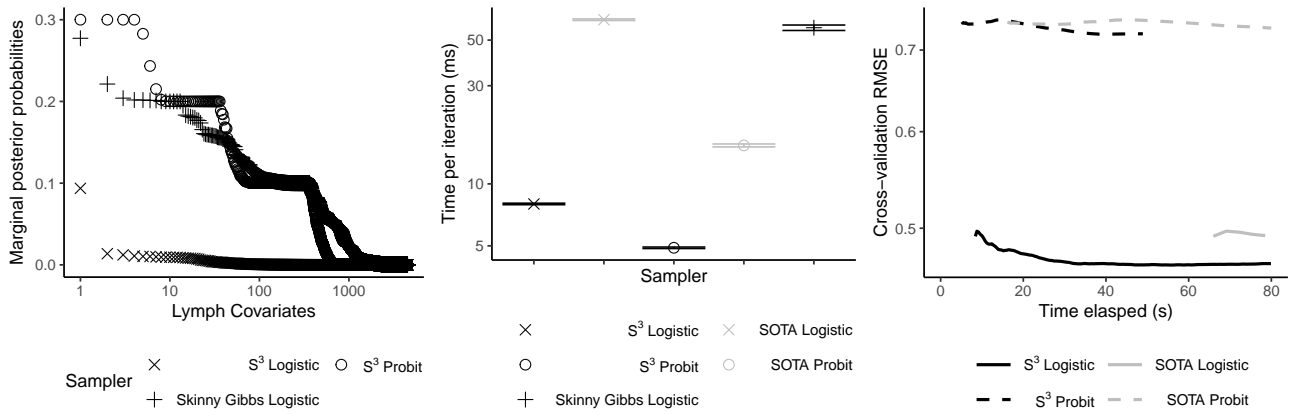*Figure 11.* Gordon dataset with $n = 181$ observations, $p = 12533$ covariates and binary valued responses.

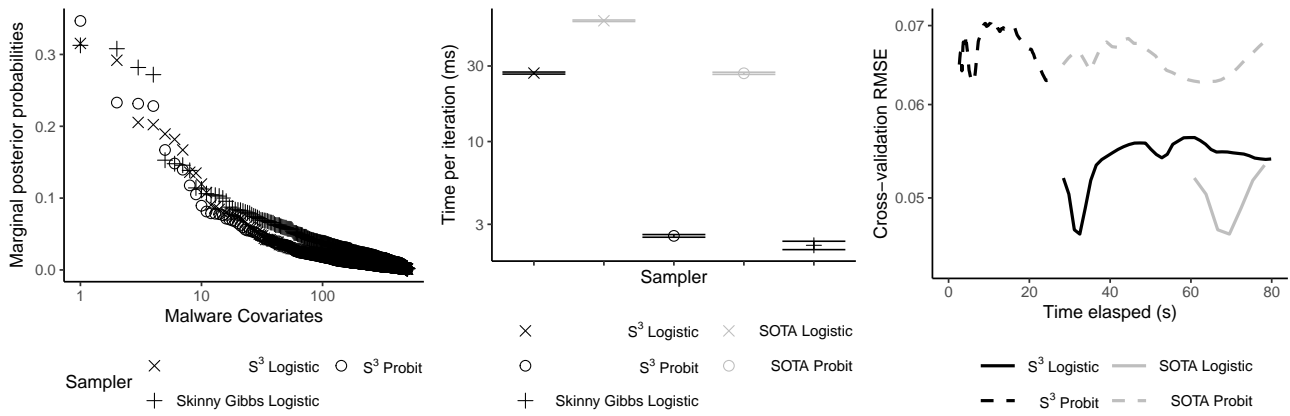*Figure 12.* Lymph dataset with $n = 148$ observations, $p = 4514$ covariates and binary valued responses.

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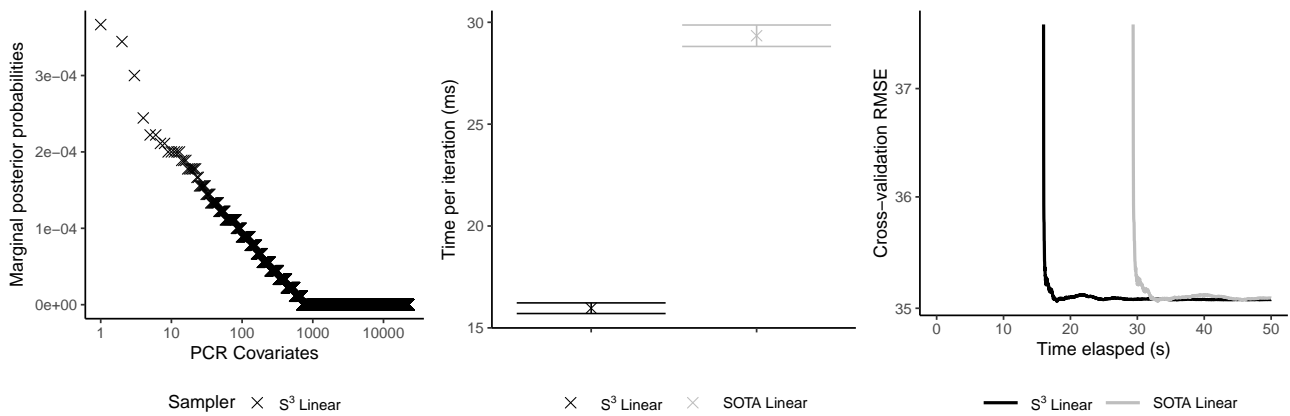

Figure 14. PCR dataset with $n = 60$ observations, $p = 22575$ covariates and continuous valued responses.