Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Neuronized Priors for Bayesian Sparse Linear Regression

Minsuk Shin & Jun S Liu

To cite this article: Minsuk Shin & Jun S Liu (2021): Neuronized Priors for Bayesian Sparse Linear Regression, Journal of the American Statistical Association, DOI: 10.1080/01621459.2021.1876710

To link to this article: https://doi.org/10.1080/01621459.2021.1876710

Neuronized Priors for Bayesian Sparse Linear Regression

Minsuk Shin[1] and Jun S Liu[2]

[1]Department of Statistics, University of South Carolina

[2]Department of Statistics, Harvard University

Corresponding author Minsuk Shin mshin@fas.harvard.edu

Abstract

Although Bayesian variable selection methods have been intensively studied, their routine use in practice has not caught up with their non-Bayesian counterparts such as Lasso, likely due to difficulties in both computations and flexibilities of prior choices. To ease these challenges, we propose the neuronized priors to unify and extend some popular shrinkage priors, such as Laplace, Cauchy, horseshoe, and spike-and-slab priors. A neuronized prior can be written as the product of a Gaussian weight variable and a scale variable transformed from Gaussian via an activation function. Compared with classic spike-and-slab priors, the neuronized priors achieve the same explicit variable selection without employing any latent indicator variables, which results in both more efficient and flexible posterior sampling and more effective posterior modal estimation. Theoretically, we provide specific conditions on the neuronized formulation to achieve the optimal posterior contraction rate, and show that a broadly applicable MCMC algorithm achieves an exponentially fast convergence rate under the neuronized formulation. We also examine various simulated and real data examples and demonstrate that using the neuronization representation is computationally more or comparably efficient than its standard counterpart in all well-known cases. An `R` package `NPrior` is provided for using neuronized priors in Bayesian linear regression.

Keywords: Bayesian shrinkage; spike-and-slab prior; variable selection; scalable Bayesian computation.

1 Introduction

We consider the standard linear regression model of the form

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}, \quad (1)$$

where $\mathbf{y} = \{y_1, \ldots, y_n\}^{\mathrm{T}}$ is the vector of responses, X is the $n \times p$ covariate matrix, $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_p\}^{\mathrm{T}} \in \mathbb{R}^p$ is the coefficient vector, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathrm{I})$. To model the sparsity of $\boldsymbol{\theta}$ when p is large, one often imposes a shrinkage prior on the θ_j's. A popular choice is the one-group *continuous shrinkage prior*, which can be represented as a hierarchical scale-mixture of Gaussian distributions:

$$\begin{aligned} &\theta_j \mid \nu_w^2, \tau_j^2 \sim N(0, \nu_w^2 \tau_j^2) \\ &\tau_j^2 \sim \pi_\tau \ (\text{or } \tau_j \sim \pi_\tau') \ \text{ and } \quad \nu_w \sim \pi_g, \end{aligned} \quad (2)$$

for $j = 1, \ldots, p$, where π_τ and π_g are some distributions chosen by the user. The *local* shrinkage parameter τ_j^2 governs the shrinkage level of each individual parameter, whereas the *global* shrinkage parameter ν_w^2 controls the overall shrinkage effect (Polson and Scott, 2010). It is common that the variance of the Gaussian prior in (2) contains the unknown error variance σ^2 of the model. However, as shown in Moran et al. (2018), the inclusion of σ^2 in (2) can result in inconsistency of σ^2 under high-dimensional settings. We thus offer a choice to not mix σ^2 in the prior of $\boldsymbol{\theta}$.

A few choices of π_τ have been shown to induce desirable shrinkage on the regression parameters, including the Strawderman-Berger prior with π_τ being a mixture of gamma distributions (Berger et al., 1996), the Bayesian Lasso (Park and Casella, 2008) with π_τ being an exponential distribution, the horseshoe prior (Carvalho et al., 2010) with π_τ' being a half-Cauchy distribution, the generalized double Pareto (Armagan et al., 2013) with π_τ being a mixture of Laplace

distributions, and the Dirichlet-Laplace prior (Bhattacharya et al., 2015) with π_τ being the product of a Dirichlet and a Laplace random variables. Some recent theoretical investigations show that the marginal prior density of θ_j with a heavy tail and a sufficient mass around zero achieves the minimax optimal rate of posterior contraction (Ghosh et al., 2017; Song and Liang, 2017; van der Pas et al., 2016).

Another popular class of shrinkage priors is the class of *spike-and-slab* (SpSL) priors (George and McCulloch, 1993; Mitchell and Beauchamp, 1988), also known as two-group mixture priors, which can be written as:

$$\begin{aligned} &\theta_j \mid \gamma_j \sim (1-\gamma_j)\pi_0(\theta_j) + \gamma_j \pi_1(\theta_j) \\ &\gamma_j \sim Bernoulli(\eta), \end{aligned} \qquad (3)$$

for $j = 1, \ldots, p$. Distribution π_0 is typically chosen to be highly concentrated around zero, i.e., the "spike", whereas π_1 is relatively disperse, i.e., the "slab". Thus, when $\gamma_j = 0$, coefficient θ_j is strongly shrunk towards zero, whereas when $\gamma_j = 1$, the slab part allows θ_j to be nearly unshrunk. Parameter η controls the sparsity of the model (Scott and Berger, 2010). When a point-mass at zero is used for π_0, we call the resulting prior a *discrete SpSL* prior; otherwise we call it a *continuous SpSL* prior. Common choices of π_0 and π_1 for a continuous SpSL prior are Gaussian distributions with a small and a large variance, respectively (George and McCulloch, 1993). Under some regularity conditions, it has been shown that an appropriate choice of η leads to model selection consistency (Narisetty and He, 2014) and the optimal posterior contraction (Castillo et al., 2015; Castillo and van der Vaart, 2012) for high-dimensional linear regression and the normal means model.

With continuous shrinkage priors, MCMC sampling of θ_j given the local and global shrinkage parameters can be efficiently implemented by taking advantage of the conjugacy. However, while continuous shrinkage priors have computational advantages over discrete SpSL priors, the resulting posterior

inference does not automatically provide sparse estimates of the coefficients, so that extra and *ad hoc* steps are needed for variable selection (Hahn and Carvalho, 2015). Computational implementations of SpSL priors often employ a binary latent vector indicating which of the two components each coefficient comes from. When a discrete SpSL prior is employed, the posterior inference of θ is notoriously challenging. MCMC sampling strategies (Dellaportas et al., 2002; Guan and Stephens, 2011) and stochastic search strategies (Berger and Molina, 2005; Hans et al., 2007; Zhang et al., 2007) have been proposed to counter the computational difficulty, mostly relying on the conjugacy of each component of the prior. An MCMC strategy for non-conjugate discrete SpSL priors, such as the one that uses reversible jump proposals (Green, 1995), is rarely practical especially under high-dimensional settings.

As a computationally scalable strategy, Rockova and George (2014) proposed the Expectation Maximization Variable Selection (EMVS), which is an EM algorithm to obtain the *maximum a posteriori* (MAP) estimator of the regression coefficients under continuous SpSL priors with Gaussian components. Rockova and George (2018) further extended their idea to cases with a SpSL Lasso (SSLasso) prior by adopting Laplace distributions for π_0 and π_1. These procedures, however, provide only point estimates, and are insufficient for quantifying uncertainties in model selection and estimation.

To address these practical issues in using shrinkage priors, we propose *neuronized priors*, which provide a unified form for popular shrinkage priors such as the horseshoe, Cauchy, SpSL, and more. In the form of neuronized priors, each regression coefficient is reparameterized as a product of a weight parameter and a transformed scale parameter via an activation function, as follows:

Definition 1.1. (Neuronized prior) *For a non-decreasing activation function T and hyper-parameters α_0 and τ_w, a neuronized prior for θ_j is defined as:*

$$\theta_j := T(\alpha_j - \alpha_0) w_j, \quad (4)$$

where the scale parameter α_j follows N(0, 1) and the weight parameter w_j follows $N(0, \tau_w^2)$, all independently for $j = 1, \ldots, p$.

As the name implies, this formulation is inspired by the use of activation functions in neural network models (Rosenblatt, 1958; Rumelhart et al., 1986). Under this setting, we can write down the joint distribution as:

$$\pi(\boldsymbol{\alpha}, \boldsymbol{w} \mid \mathbf{y}, \alpha_0, \sigma^2) \propto \frac{1}{\sigma^n} \exp\left\{ -\frac{\left\| \mathbf{y} - X\boldsymbol{\theta}(\boldsymbol{\alpha}, \boldsymbol{w}, \alpha_0) \right\|_2^2}{2\sigma^2} - \frac{\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}}{2} - \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{2\tau_w^2} \right\}, \quad (5)$$

where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_p\}^{\mathrm{T}}$, $\boldsymbol{w} = \{w_1, \ldots, w_p\}^{\mathrm{T}}$, $\pi(\sigma^2)$ is the prior on σ^2, and

$$\boldsymbol{\theta}(\boldsymbol{\alpha}, \boldsymbol{w}, \alpha_0) = \{T(\alpha_1 - \alpha_0) w_1, \ldots, T(\alpha_p - \alpha_0) w_p\}^{\mathrm{T}} \overset{def}{=} D_\alpha \boldsymbol{w}, \quad (6)$$

where D_α is the diagonal matrix with diagonal elements the $T(\alpha_j - \alpha_0)$'s. We show that for most existing shrinkage priors we can find specific activation functions such that the resulting neuronized priors approximate the existing ones. Therefore, existing theoretical properties of various shrinkage priors can be directly applied to posterior behaviors based on the neuronized priors. This theoretical equivalence will be discussed in Section 2. We also show that variable selection procedures based on neuronized priors offer following advantages:

- *Unification.* Various classes of shrinkage priors can be practically implemented by just changing the activation function. This characteristic significantly reduces practical hurdles for the user to test out different priors simultaneously, which can be a valuable option. For example, we may find that horseshoe prior is appropriate for analyzing GWAS data on bipolar disorders, whereas SpSL priors work much better for Type-1 diabetes (Song et al., 2020).

- *Flexibility and efficient computation.* Without having to rely on prior-likelihood conjugacy, neuronized priors still attain comparable or better efficiency for MCMC-based posterior inference compared with the standard procedures, thus can easily accommodate non-conjugate priors. In addition, neuronized priors also enable a scalable coordinate descent optimization algorithm for posterior modal estimation, even with discrete SpSL priors.
- *Desirable theoretical properties.* We give explicit conditions on the activation function and hyperpriors so that the resulting neuronized Bayesian regression achieves the optimal posterior contraction rate (Section 5), and show that a random-walk *Metropolis-Hastings* (RWMH) algorithm converges to the target distribution at an exponential rate, even for non-conjugate priors.

The rest of the article is organized as follows. Section 2 details the neuronized counterparts of a few popular shrinkage priors for Bayesian linear regression: the discrete SpSL, the Bayesian Lasso, and the horseshoe and Cauchy priors. Section 3 shows how to manage neuronized priors to achieve one's intended goals, such as matching a target prior or controlling the sparsity level. Section 4 details main computational strategies and advantages of neuronized priors. Section 5 studies theoretical properties of the neuronized priors, including sufficient conditions for achieving an optimal posterior contraction rate and geometric ergodicity of MCMC algorithms under a simple setting. Section 6 reports simulation studies to compare the effects of different priors and their neuronized counterparts. Two real data examples are analyzed in Section 7, and a short conclusion is given in Section 8. Proofs of the main results, efficiency comparisons of some MCMC algorithms, and additional simulation studies are provided in the Supplementary Materials.

2 Neuronization of Standard Sparse Priors

2.1 Discrete and continuous SpSL priors

Let the activation function in (4) be the Rectifier Linear Unit (ReLU) function, $T(t) = \max\{0, t\}$. When $\alpha_0 = 0$, $T(\alpha_j - \alpha_0)$ follows an equal mixture of the point-mass at zero and the half standard Gaussian, as shown in Figure 1(a). This implies that the marginal density of $T(\alpha_j)w_j$ is a SpSL distribution of the form

$$\begin{aligned} &\theta \mid \gamma \sim (1-\gamma)\delta_0(\theta) + \gamma\pi(\theta), \\ &\gamma \sim Bernoulli(1/2), \end{aligned} \qquad (7)$$

where π is the marginal density of the product of two independent standard Gaussians, which is shown to have an exponential tail in Proposition 2.3. This tail behavior is desirable, since Castillo and van der Vaart (2012) and Castillo et al. (2015) showed that the optimal minimax rate of posterior contraction can be achieved when the tails of the slab part of (7) are exponential or heavier. We note that continuous SpSL priors can be obtained from formulation (4) by adopting a "leaky" ReLU activity function (Maas et al., 2013), i.e. $T(t) = \max\{ct, t\}$ for some $c < 1$.

Figure 1 Here

More generally, hyper-parameter α_0 controls the prior probability of sparsity: $P(T(\alpha_j - \alpha_0) = 0 \mid \alpha_0) = P(\alpha_j < \alpha_0 \mid \alpha_0) = \Phi(\alpha_0)$, with Φ being the standard Gaussian CDF. Thus, setting $\gamma \sim Bernoulli(\Phi(-\alpha_0))$ in (7) leads to the same distribution as that implied by (4). Conversely, $\forall \eta \in (0,1)$, we choose $\alpha_0 = -\Phi^{-1}(\eta)$ to achieve the desired sparsity. Scott and Berger (2010) showed that, for the sparsity parameter η in model (3), the Beta hyper-prior

$$\eta \sim Beta(a_0, b_0), \qquad (8)$$

with $a_0 = b_0 = 1$ results in a strong effect on multiplicity correction. Castillo and van der Vaart (2012) and Castillo et al. (2015) found that the resulting SpSL procedure achieves model selection consistency and the optimal posterior contraction rate if one chooses $(a_0, b_0) = (1, p^a)$ for $a > 1$, under an asymptotic

regime where the number of predictors p increases at a sub-exponential rate of n, i.e., $p \asymp \exp\{n^c\}$ for $c < 1$. The neuronized priors can accommodate this Bernoulli-beta hyper-prior by adopting a hyper-prior on α_0 as below:

Proposition 2.1. *Consider (4) with $T(\cdot)$ being ReLU and a hyper-prior on α_0,*

$$\pi(\alpha_0) \propto \Phi(-\alpha_0)^{a_0-1}(1-\Phi(-\alpha_0))^{b_0-1}\phi(\alpha_0), \quad (9)$$

where ϕ and Φ are the pdf and cdf of N(0, 1), respectively. Then, the resulting prior distribution is identical to the form of (3) with the Beta prior (8) on η.

Since α_0 is highly correlated with other parameters such as α, an MCMC algorithm equipped with naive random-walk proposals would result in low sampling efficiency and poor mixing quality. Instead, we consider an efficient group-move update via a generalized Gibbs sampler (Liu and Sabatti, 2000). The details of this computational strategy is provided in Section 4.2.

As a demonstration, we analyze the Boston housing price data with linear regression. The dataset contains n = 506 median housing prices of owner-occupied homes in the Boston area, together with 10 variables that might be associated with the median prices. Under the Jeffreys prior on σ^2, which is $1/\sigma^2$, we consider the independent neuronized prior: $\theta_j = T(\alpha_j - \alpha_0)w_j$, where $\alpha_j \sim N(0,1)$ and $w_j \sim N(0, \tau_w^2)$ for $j = 1, \ldots, p$. As shown in Figure 2, the solution path resulting from the neuronized prior with the ReLU activation function is almost identical to that resulting from the standard discrete SpSL prior.

Figure 2 Here

2.2 The Bayesian Lasso

The Bayesian Lasso imposes a Laplace prior on θ_j and uses a Gaussian mixture representation to facilitate efficient MCMC computations (Park and

Casella, 2008). We shall show that the neuronized prior with $T(t) = t$ approximates the Bayesian Lasso.

Lemma 2.2. *With the activation function T(t) = t, the marginal density of θ resulting from the neuronized prior is proportional to* $\int_0^\infty z^{-1} \exp\{-\theta^2 / (2\tau_w^2 z^2) - z^2 / 2\} dz$.

Since

$\exp\{-|\theta| / \tau_w\} \propto \int_0^\infty \exp\{-\theta^2 / (2\tau_w^2 z^2) - z^2 / 2\} dz$, the Laplace density differs from the form in Lemma 2.2 only by a term z^{-1} in the integrand. Furthermore, the following proposition shows that the tail of this neuronized prior decays at an exponential rate like the Bayesian Lasso prior.

Proposition 2.3. *Let π_L be the marginal density function of θ defined in (4) with T(t) = t and $\alpha_0 = 0$. Then, $\forall \epsilon \in (0,1), \exists \theta_0$ and constants $c_1, c_2 > 0$, such that* $c_1 \exp\{-(1+\epsilon)^{1/2} |\theta| / \tau_w\} \le \pi_L(\theta) \le c_2 \exp\{-(1-\epsilon)^{1/2} |\theta| / \tau_w\}$ when $\theta > \theta_0$.

Hoff (2017) also pointed out the similarity between the Bayesian Lasso and the product representation of the parameter (i.e., the neuronized prior with an identity activation function). He showed that the MAP estimator based on the product representation of the parameter is identical to the standard Lasso.

The histogram in Figure 3(a) compares the Bayesian Lasso prior with its neuronized version, verifying that the two distributions are indeed very similar. However, the Laplace prior has slightly more density around zero than the neuronized counterpart. Figure 4 shows the solution paths of the Bayesian Lasso, the neuronized Bayesian Lasso, and the standard Lasso for the analysis of the Boston housing price data set, which are almost identical.

Figure 3 Here

Figure 4 Here

2.3 Horseshoe, Cauchy and their generalizations

We start with a simple result for transforming Normal to a heavy tail distribution. Then, we show some activation functions that can make the corresponding neuronized priors approximate the horseshoe and Cauchy priors.

Lemma 2.4. *Let* $T(t) = \exp(\lambda_1 sign(t)t^2)$ *with* $\lambda_1 \in (0,1)$, *and let* $Z \sim N(0,1)$ *and* $U = T(Z)$. *Then, the density function of U is* $f_U(u) \propto u^{-1-\frac{1}{2\lambda_1}} |\log(u)|^{-\frac{1}{2}}$, *u > 0. If* $\lambda_1 < (1+k)^{-1}$, *we have* $E(U^k) = \frac{1}{2}\left(\frac{1}{\sqrt{1-2k\lambda_1}} + \frac{1}{\sqrt{1+2k\lambda_1}} \right)$.

The proof is straightforward, and thus omitted. This lemma implies that any polynomial tails of the local shrinkage prior can be constructed by "neuronizing" a Normal random variable through an exponential function, up to a logarithmic factor. We further show that the adoption of this exponential activation function induces a marginally polynomial-tailed prior on $\theta = T(\alpha)w$, as the following result:

Proposition 2.5. *Let* π_E *be the marginal density of* θ *defined in (4) with* $T(t) = \exp(\lambda_1 sign(t)t^2)$ *for* $0 < \lambda_1 \le 1/2$. *Then, for any* $\kappa > 0$, *there exists* θ_0 *such that* $c_1(\log|\theta|)^{-\frac{1}{2}} |\theta|^{(-1-\frac{1}{2\lambda_1})(1+\kappa)} \le \pi_E(\theta) \le c_2(\log|\theta|)^{-\frac{1}{2}} |\theta|^{(-1-\frac{1}{2\lambda_1})(1-\kappa)}$ *if* $\theta > \theta_0$, *where* c_1 *and* c_2 *are some positive constants.*

As λ_1 dictates the tail behavior of a neuronized prior with an exponential activating function, we consider the following class of activating functions:

$$T(t) = \exp\{\lambda_1 \text{sign}(t)t^2 + \lambda_2 t + \lambda_3\}, \quad (10)$$

with $\lambda_1 \ge 0$, which results in a class of *generalized horseshoe priors*. We recommend to choose λ_3 so that the resulting distribution for θ_j / τ_w in (4) has a similar interquartile range as that for the standard horseshoe distribution, i.e., 1.1 ~1.5. We numerically found that, with $T(t) = \exp\{0.5\text{sign}(t)t^2 + 0.733t\}$, the neuronized prior for θ_j / τ_w approximates the horseshoe prior well (the details of

the numerical evaluation is deferred to Section 3.1). In the same sense, the neuronized prior for θ_j / τ_w approximates the standard Cauchy distribution if $T(t) = \exp\{0.5t^2 - 1.27t + 0.29\}$. We may therefore regard the neuronized priors induced by $T(t) = \exp\{\lambda_1 t^2 + \lambda_2 t + \lambda_3\}$ as generalized Cauchy priors, which differ from those induced by (10) in having weaker shrinkage effects for weak signals because $T(t)$ is bounded below by $\exp\left\{\lambda_3 + \min\left(0, \frac{\lambda_2 \mid \lambda_2 \mid}{4\lambda_1}\right)\right\}$.

Figure 3(b) and (c) show histograms contrasting the horseshoe and Cauchy priors with their corresponding neuronized versions, respectively. Figure 5 compares the solution paths under the neuronized and standard horseshoe priors for the same Boston housing price data, demonstrating their nearly identical behaviors. Table 1 summarizes the results in this Section.

Figure 5 Here
Table 1 Here

Although it covers a large class of prior densities as demonstrated, the current neuronization formulation as in (4) still has difficulties emulating some distributions. For example, nonlocal priors (Johnson and Rossell, 2010, 2012; Rossell and Telesca, 2017), which are bimodal and symmetric around zero, cannot be easily constructed using (4). However, one may still capture the bimodality of a desired prior by changing the distribution of w and α in (4) to be bimodal. Also, dependent prior densities cannot be represented by a product of neuronized priors. These examples include the Zellner's g-prior (Zellner, 1986) and the Dirichlet-Laplace prior (Bhattacharya et al., 2015). But an extension of the neuronized prior to a multivariate version may overcome this limitation.

3 Managing Neuronized Priors

3.1 Find the activation function to match a given prior

Section 2 presents neuronized formulations for some popular existing priors. More generally, if we want to find an activation function T so that the resulting neuronized prior matches a desired target distribution $\pi(\theta)$ symmetric about zero, we may consider a family of activation functions $\{T_\phi\}$ parameterized by ϕ, and then numerically find $\hat{\phi}$ so that $T_{\hat{\phi}}$ minimizes a certain discrepancy measure between the resulting neuronized prior and the target $\pi(\theta)$. For example, we can consider a family of exponential functions as in (10) by setting $\phi = \{\lambda_1, \lambda_2, \lambda_3\}$ to construct a generalized horseshoe prior. More flexibly, the function space spanned by a class of B-spline basis functions can be a reasonable choice, i.e., $T_\phi(t) = \mathbf{B}(t)\phi$, where **B** is a vector of K B-spline basis functions and $\phi \in \mathbb{R}^K$.

If we aim to match the polynomial tail of a general target prior, we can consider an additive mixture of an exponential function as in Proposition 2.5 and a basis expansion. More precisely, we define a class of activation functions parameterized by $\zeta = \{\lambda_1, \phi\}$:

$$T_\zeta(t) = \exp\{\lambda_1 \text{sign}(t) t^2\} + \mathbf{B}(t)\phi, \text{ with } \lambda_1 > 0.$$

These activation functions naturally lead to polynomial tails for the corresponding neuronized priors because the effect of B-spline bases are minimal as $|t| \to \infty$. To find an appropriate ζ, we can first find λ_1 to match the tails of the target prior based on the results of Proposition 2.5. For example, if the target prior decays at the rate of $|x|^{-b}$, we choose $\lambda_1 = \frac{1}{2(b-1)}$.

Once λ_1 is fixed, we generate a large number S of i.i.d. samples from the the neuronized prior: $\tilde{\theta}_{\zeta,i} = T_{\lambda_1,\phi}(\alpha_i - \alpha_0)w_i$, where $(\alpha_i, w_i) \sim N(0,1) \times N(0,1)$, for $i = 1, \ldots, S$; and also generate $\theta_i \overset{iid}{\sim} \pi(\theta)$ for $i = 1, \ldots, S$, where $\pi(\cdot)$ is the target prior. We measure the discrepancy between these two samples, for example, by $D(\zeta) = \sum_{i=1}^{S} |\tilde{\theta}_\zeta^{(i)} - \theta^{(i)}|$, where $\tilde{\theta}_\zeta^{(i)}$ and $\theta^{(i)}$ are the i-th largest value of the generated samples $\{\tilde{\theta}_{\zeta,i}\}_{i=1,\ldots,S}$ and $\{\theta_i\}_{i=1,\ldots,S}$, respectively. Some other attractive measures

are the *l_2 distance* or the *Wasserstein distance*. Then, we can minimize $D(\zeta)$ with respect to ζ by using a grid search algorithm or a simulated annealing algorithm (Kirkpatrick and Vecchi, 1983). This optimization is not computationally intensive as long as the dimension of ζ is moderate.

3.2 Choosing hyper-parameters

Neuronized priors have two hyper-parameters: the variance of the global shrinkage parameter τ_w^2 and the bias parameter α_0. The roles of these hyper-parameters are different according to the choice of the activation function. When we consider neuronized continuous shrinkage priors, α_0 is set at 0 by default. When we use neuronized discrete SpSL prior via the ReLU activation function, the prior probability for each coefficient to be non-zero is $\Phi(-\alpha_0)$. As shown in Proposition 2.1, we can impose a hyper-prior on α_0 so that the sparsity level is adaptively controlled by the data set. However, sampling α_0 conditional on other parameters in Gibbs sampling is not trivial and naive random-walk proposals for a MH algorithm is highly inefficient due to its high posterior correlation with other parameters. We describe an efficient group-move in the next section.

The choice of τ_w^2 is a bit complicated. When $\mathbb{E}(T^2(\alpha))$ is bounded, the prior expected signal-to-noise ratio for the regression model is $\mathbb{E}\|\boldsymbol{\theta}\|^2/\sigma^2 = p\tau_w^2\mathbb{E}[T^2(\alpha_j-\alpha_0)]/\sigma^2$. Thus, the choice of τ_w needs to reflect our prior knowledge about the signal strength in the data. Although some theoretical analysis has been attempted on the normal means model (van der Pas et al., 2014) under a fixed τ_w^2, a theoretically justified selection of the hyper-prior for τ_w^2 has not been found.

When $\mathbb{E}(T^2(\alpha))$ does not exist as in the horseshoe and Cauchy cases, the signal strength interpretation is not valid. As noted by Carvalho et al. (2010), for horseshoe priors the shrinkage factor $\kappa_j = (1+T^2(\alpha_j)\tau_w^2)^{-1}$ determines the shrinkage level of θ_j and can be interpreted as an approximation of $\mathbb{E}(1-\gamma_j)$ in (3). We thus numerically search τ_w^2 so that $1-\mathbb{E}(\kappa_j)=\pi_0$ for some prior belief on

the proportion of non-zero parameters $\pi_0 \in (0,1)$, which is set at $\min\{0.01, 0.1 \times n/p\}$ by default. We subsequently use this setting and show that the empirical performance of the resulting procedure is promising in various simulation and real data examples.

As shown in Moran et al. (2018), the traditional conjugate prior for linear models, i.e., $\boldsymbol{\theta} \mid \sigma^2 \sim N(\mathbf{0}, \lambda_0 \sigma^2 I_p)$ and $\sigma^2 \sim \text{Inv-Chisq}(\nu_0)$ *a priori*, can lead to inconsistency in high-dimensional problems. To avoid this undesirable situation, we assume that $w_j \sim N(0, \tau_w^2)$ and $\sigma^2 \sim$ Inv-Gam(*a*, *b*) are independent *a priori*.

4 Sampling and Optimization with Neuronized Priors

4.1 MCMC sampling with neuronized priors

Consider the linear regression model in (1) and the unnormalized joint distribution of $\boldsymbol{\alpha}$, $\boldsymbol{w}$, and σ^2 as in (5). The conditional posterior distribution of $\boldsymbol{w}$ given $\boldsymbol{\alpha}$ and other hyper-parameters is Gaussian:

$$\boldsymbol{w} \mid \mathbf{y}, \boldsymbol{\alpha}, \sigma^2, \tau_w^2 \sim N(\tilde{\mu}, \sigma^2 \tilde{\Sigma}), \quad (11)$$

where $\tilde{\Sigma} = (D_\alpha X^{\mathrm{T}} X D_\alpha + \sigma^2 \tau_w^{-2} \mathrm{I})^{-1}$ and $\tilde{\mu} = \tilde{\Sigma} D_\alpha X^{\mathrm{T}} \mathbf{y}$, with D_α as defined in (6). When an Inv-Gam(*a*, *b*) is imposed on σ^2, the conditional distribution of σ^2 given other parameters is Inv-Gam$(n/2 + a, \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 / 2 + b)$. When *p* is large relative to *n*, the numerical calculation of $(D_\alpha X^{\mathrm{T}} X D_\alpha + \sigma^2 \tau_w^{-2} I)^{-1}$ is highly expensive. Bhattacharya et al. (2016) proposed a fast sampling procedure that reduces the computational complexity from $O(p^3)$ to $O(n^2 p)$, which is employed here. Conditional on $\boldsymbol{w}$ and $\boldsymbol{\alpha}_{(-j)}$, each α_j can be sampled by a naive RWMH algorithm, for $j = 1, \ldots, p$. Since w_j and α_j tend to be highly correlated *a posteriori*, a better strategy is to integrate out w_j so as to draw α_j^* from $\pi(\alpha_j \mid \mathbf{y}, \boldsymbol{w}_{(-j)}, \boldsymbol{\alpha}_{(-j)})$, and then draw w_j from $\pi(w_j \mid \mathbf{y}, \boldsymbol{w}_{(-j)}, \boldsymbol{\alpha}_{(-j)}, \alpha_j^*)$.

Algorithm 1 Here

The RWMH step in Algorithm 1 is local and cheap, and is thus iterated M times for sampling each α_j. We set M = 10 in all our numerical examples and find the resulting algorithm to perform well. We use $N(\alpha_j^{(t)}, 2^2)$ as the proposal distribution, which enables α_j to propose efficiently between the regions $\{\alpha_j : \alpha_j < \alpha_0\}$ and $\{\alpha_j : \alpha_j \geq \alpha_0\}$. We subsequently use Algorithm 1 as the default to implement the posterior inference based on the neuronized prior. Parameter α_0 is set at 0 for neuronized continuous shrinkage priors, but will follow a hyper-prior distribution as in (9) for neuronized SpSL priors, whose MCMC update is detailed next.

4.2 Sampling α_0 efficiently

For neuronized discrete SpSL priors, we may want to impose a prior distribution on α_0 to accommodate some vague prior knowledge of the sparsity level as in Proposition 2.1. Due to high correlation between α_0 and the α_j's, however, a naive MH approach in which α_0 is updated by a MH step conditioned on $\boldsymbol{\alpha}$ is highly inefficient. To overcome this difficulty, we consider a group-move via the generalized Gibbs sampling formulation (Liu and Sabatti, 2000): update $\boldsymbol{\alpha}$ and α_0 simultaneously by a common shift $\delta \in \mathbb{R}$. More precisely, $(\boldsymbol{\alpha}, \alpha_0)$ is updated as

$$(\boldsymbol{\alpha}, \alpha_0) \to (\boldsymbol{\alpha} + \delta \mathbf{1}, \alpha_0 + \delta),$$

where δ is drawn from the distribution $g(\delta) \propto \pi^*(\boldsymbol{\alpha} + \delta \mathbf{1}, \alpha_0 + \delta)$, where $\pi^*(\boldsymbol{\alpha}, \alpha_0)$ is the conditional posterior density of $\boldsymbol{\alpha}$ and α_0. After this group-move, it is necessary to update each α_j conditionally to distinguish the individual posterior behavior, but we do not need to consider an extra step to update α_0 individually.

When the prior is of the form (9) with $a_0 = b_0 = 1$, $g(\delta)$ is simply Gaussian:

$$N((\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha} + \alpha_0)/(p+1), (p+1)^{-1}). \qquad (12)$$

However, when $a_0 \neq 1$ or $b_0 \neq 1$, the distribution $g(\delta)$ is non-standard, and an extra approximation step is needed for updating δ. To this end, we propose a multiple-try MH independence sampler (MTM-IS) to sample δ, following the ideas in Liu et al. (2000). This algorithm proposes multiple candidates $\delta_1,\ldots,\delta_m$ drawn independently from a proposal distribution (such as the Gaussian distribution in (12)), and then chooses one from them with probability proportional to their importance weights. The acceptance-rejection ratio is adjusted to account for this selection effect. The detailed algorithm is as follow.

Algorithm 2 Here

A proof of the correctness of this algorithm follows immediately the approach in Liu et al. (2000) and thus omitted.

4.3 MCMC strategies for discrete SpSL priors

A most direct and effective approach for conducting sparse Bayesian linear regression is to employ a discrete SpSL prior for the coefficients. When the continuous component of this prior is conjugate to the Gaussian likelihood, a well-known computational strategy is the *collapsed Gibbs sampler* (Liu, 1994), which integrates out all the continuous parameters (e.g., regression coefficients) and samples, via MCMC, the binary indicator vector γ defined in (3) from the posterior distribution $\pi(\gamma \mid \mathbf{y}, \sigma^2) = \dfrac{m_\gamma(\mathbf{y} \mid \sigma^2)h(\gamma)}{\sum_{\gamma'} m_{\gamma'}(\mathbf{y} \mid \sigma^2)h(\gamma')}$, where $m_\gamma(\mathbf{y})$ is the marginal likelihood of γ and $h(\cdot)$ is the model prior mass function. Note that σ^2 is still present in the marginal likelihood because our prior is not fully conjugate with respect to the error variance. This collapsed sampler can become highly inefficient if one calculates the marginal likelihood by brute force at every iteration. A more efficient strategy is to update the required matrix inversion and determinant incrementally. For example, to add or remove a variable from the current model, we need to modify the sample covariance matrix by adding or

deleting one row and one column. The corresponding inverse and determinant can be updated using the formulas in Section B of Supplementary Materials. However, even with this efficient implementation, the fully collapsed sampler is still rather slow.

Alternatively, we can consider a half-collapsed sampling strategy, which appears to be computationally more efficient. Instead of integrating out all the θs, at each iteration we sample γ_j from the conditional distribution $[\gamma_j \mid \boldsymbol{\theta}_{(-j)}, \mathbf{y}]$, with θ_j integrated out, and then update θ_j conditional on γ_j. Although each iteration step of this half-collapsed sampler is less efficient than the fully-collapsed one, a major advantage of this approach is that every step is much faster to compute. A comparison between the fully-collapsed and the half-collapsed Gibbs sampler is provided in the Supplementary Materials, suggesting that the half-collapsed Gibbs sampler is ten times or more efficient than the fully-collapsed one for the examined examples.

However, both collapsing approaches become unavailable if one cannot analytically integrate out the continuous parameters. In such cases, either a crude and/or time-consuming approximation strategy, or a cleverly designed, yet case-specific, data augmentation strategy (Polson et al., 2013), or a much less efficient reversible-jump scheme (Green, 1995), has to be employed. In contrast, the neuronized priors can achieve the same effect as standard discrete SpSL priors while permitting more efficient computation even if one cannot marginalize out continuous components in the joint posterior distribution. When a ReLU activation function is adopted, the result below further shows that conditional distribution $\pi(\alpha_j \mid \mathbf{y}, \boldsymbol{w}_{(-j)}, \boldsymbol{\alpha}_{(-j)})$ is a mixture of two truncated Gaussians and can be sampled exactly.

Proposition 4.1. *Let* $r_j = \mathbf{y} - \sum_{k \neq j} X_k \theta_k$ *and* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$*, and let* $N_{tr}(a, b; c, d)$ *denote the truncated Gaussian with mean a and variance b on (c, d). The conditional distribution* $[\alpha_j \mid \boldsymbol{\alpha}_{-j}, \boldsymbol{w}, \mathbf{y}, \sigma^2]$ *based on the posterior distribution (5)*

with the ReLU activation function is $\kappa N_{tr}(0,1;-\infty,\alpha_0)+(1-\kappa)N_{tr}(\tilde{\alpha}_j,\tilde{\sigma}_j^2;\alpha_0,\infty)$, *where* $\tilde{\alpha}_j=\frac{(r_j+X_j\alpha_0 w_j)^{\mathrm{T}}X_j w_j}{X_j^{\mathrm{T}}X_j w_j^2+\sigma^2}$, $\tilde{\sigma}_j^2=\sigma^2\left(X_j^{\mathrm{T}}X_j w_j^2+\sigma^2\right)^{-1}$, *and*

$$\kappa=\frac{\Phi(\alpha_0)\exp\left\{-\frac{\|r_j\|_2^2}{2\sigma^2}\right\}}{\Phi(\alpha_0)\exp\left\{-\frac{\|r_j\|_2^2}{2\sigma^2}\right\}+\left\{1-\Phi\left(\frac{\alpha_0-\tilde{\alpha}_j}{\tilde{\sigma}_j}\right)\right\}\tilde{\sigma}_j\exp\left\{\frac{\tilde{\alpha}_j^2}{2\tilde{\sigma}_j^2}-\frac{\|r_j+X_j\alpha_0 w_j\|_2^2}{2\sigma^2}\right\}}.$$

There is another computational advantage of using the ReLU activation function. When sampling $\boldsymbol{w}$ in a Gibbs step, the conditional posterior distribution can be decomposed as a product of independent Gaussian densities so that the numerical inversion of the $p \times p$ matrix $\tilde{\Sigma}$ in (11) can be avoided. We can rewrite that

$$\tilde{\Sigma}=\begin{pmatrix}\tilde{\Sigma}^* & 0\\ 0 & \sigma^{-2}\tau_w^2\mathrm{I}\end{pmatrix},\qquad \tilde{\mu}=\begin{pmatrix}\tilde{\mu}^*\\ 0\end{pmatrix},$$

in (11), where $\tilde{\Sigma}^*=(D_\alpha^* X^{*\mathrm{T}}X^* D_\alpha^*+\sigma^2\tau_w^{-2}\mathrm{I})^{-1}$, $\tilde{\mu}^*=\tilde{\Sigma}^* D_\alpha^* X^{*\mathrm{T}}\mathbf{y}$, and D_α^* and X^* are the sub-matrices induced by the index of the nonzero regression coefficients. This expression means that for those *j* with $\alpha_j<\alpha_0$, coefficient θ_j is set to zero and the sampling of w_j follows $N(0,\sigma^2\tau_w^2)$ independently. The conditional distribution of the sub-vector $\boldsymbol{w}^*=\{w_j:\alpha_j>\alpha_0\}$ is $N(\tilde{\mu}^*,\sigma^2\tilde{\Sigma}^*)$. To sample $\boldsymbol{w}^*$, we only need to compute $\tilde{\Sigma}^*$, which has a much smaller size than the $p \times p$ matrix $\tilde{\Sigma}$, reducing computational complexity from $O(p^3\wedge np)$ to $O\left(|\boldsymbol{w}^*|^3\wedge p\wedge n|\boldsymbol{w}^*|\right)$, where $a\wedge b$ is the minimum operator between *a* and *b*.

4.4 A scalable algorithm for finding posterior modes

For massive-sized data sets, MCMC algorithms may not be practical and one needs to consider optimization-based algorithms. We here propose the Coordinate-Ascent Algorithm for Neuronized priors (CAAN) to find the MAP

estimator. CAAN adopts a warm start strategy as in Rockova and George (2018) by initiating with a hyper-parameter that results in a weak shrinkage and increasing gradually the strength of the shrinkage. While this warm start strategy requires multiple implementations of the optimization with various hyper-parameters, it reduces the chance of being trapped in a local optimum. Although it cannot be guaranteed to converge to a global optimum, empirical results in Sections 6 and 7 show that CAAN performs similarly as SSLasso and significantly better than other considered methods.

Algorithm 3 Here

A key to the success of CAAN is the optimization with respect to α_j while fixing other parameters, $\boldsymbol{\alpha}_{(-j)}$ and $\boldsymbol{w}$. Because the function of α_j in ($\Diamond$) of Algorithm 3 is a linear combination of a quadratic function and a function of $T(\alpha_j - \alpha_0)$, we divide the optimization space into two parts: $\{\alpha_j : \alpha_j > \alpha_0\}$ and $\{\alpha_j : \alpha_j \le \alpha_0\}$, and find a local maximum from each part. Then, we update α_j to the best of the two local maxima. This one-dimensional optimization problem can be easily solved by many existing algorithms and we adopt the secant algorithm of Brent (1973). Vector $\boldsymbol{w}$ is updated jointly conditioning on $\boldsymbol{\alpha}$ by taking advantage of the Gaussian conjugacy.

The algorithm employs a temperature scheme to help with the optimization task. With t taking values in an $(2L+1)$-level schedule, $t_0 \ge \cdots \ge t_{2L} = 1$, CAAN maximizes the objective function

$$-\frac{1}{2t\sigma^2}\|\mathbf{y} - X\boldsymbol{\theta}(\boldsymbol{\alpha}, \boldsymbol{w})\| - \frac{\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}}{2t} - \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{2t\tau_w^2} - \frac{n}{2}\log\sigma^2$$

with respect to $\boldsymbol{\alpha}$, $\boldsymbol{w}$, and σ^2. At each temperature t_k, we conduct coordinate ascent iterations M times. This approach is different from simulated annealing (Kirkpatrick and Vecchi, 1983) in that (a) the term $\frac{n}{2}\log\sigma^2$ is free of the temperature, (b) temperature t is bounded below by one, and (c) we do

coordinate-ascending instead of MCMC sampling at each iteration. Consequently, at a warm temperature the solution tends to select a large-sized model, and irrelevant features get eliminated as the temperature decreases. At default, we set M = 20, L = 10, N = 20, $t_k = \left(3 - \frac{2k}{L}\right)^2$ for $k = 0, \ldots, L$, and $t_{L+1} = \cdots = t_{2L}$ =1. To reduce the chance of getting trapped in a local optima, in the first L levels of schedule, we add a random noise $\xi \sim \text{Exp}(1)$ to σ^2 after every N iterations.

For the ReLU activation function, α_0 affects the sparsity level since it sets the prior probability for each coefficient to be non-zero as $\psi_0 = \Phi(-\alpha_0)$. By using Proposition 2.1, we deploy a hyper-prior on α_0 so that the induced prior on ψ_0 is $Beta(a_0, b_0)$. As a default in all SpSL procedures, we set $(a_0, b_0) = (1,1)$, and this beta-binomial prior on the sparsity has been shown to have a strong effect on multiplicity control (Scott and Berger, 2010).

4.5 Comparisons with other posterior optimization procedures

We consider four optimization procedures for the Bayesian SpSL variable selection problem: a Majorization-Minimization (MM) algorithm (Yen et al., 2011), EMVS (Rockova and George, 2014), SSLasso (Rockova and George, 2018), and our CAAN. To compare the algorithms and track their solution paths, we adopt as goodness measures the mean-squared error (MSE) and the *Extended Bayesian information criterion* (EBIC; Chen and Chen (2008)), i.e.,

$$\text{EBIC}(\mathbf{k}) = \text{BIC} + \zeta |\mathbf{k}| \log p, \quad (13)$$

where $\mathbf{k}$ denote the set of selected variables ζ is a tuning parameter, and BIC is the *Bayesian information criterion* (Schwarz et al., 1978). We set ζ = 1 as suggested by Chen and Chen (2008).

MM finds the MAP estimator of our problem by approximating the l_0-norm by a continuous function: $\|\boldsymbol{\theta}\|_0 = \lim_{\tau_3 \to 0} \sum_{j=1}^{p} \log(1+\tau_3^{-1}|\theta_j|)/(\log(1+\tau_3^{-1}))$. In practice, we need to choose τ_3 in advance, which strongly affects the performance of the approximation. While a smaller τ_3 leads to a better approximation to the original posterior distribution, the resulting target function becomes highly non-concave and is much more difficult to optimize.

EMVS and SSLasso were proposed to evaluate the MAP estimator based on an EM formulation when using a continuous SpSL prior as in (3). The prior for EMVS is a mixture of $\pi_0 = N(0, \nu_0)$ and $\pi_1 = N(0, \nu_1)$, and that for SSLasso is a mixture of $\pi_0 = Laplace(\lambda_0)$ and $\pi_1 = Laplace(\lambda_1)$, where $\nu_0 \ll \nu_1$ and $\lambda_0 \gg \lambda_1$. Since the spike prior part is not a point mass, ν_0 (or λ_0) needs to be carefully chosen to control how much the spike prior density is concentrated around zero. We impose a uniform prior on η in (3). We choose $\nu_1 = 100$ and $\nu_0^{-1} \in (1,1000)$ for EMVS; and choose $\lambda_1 = 1$ and let λ_0 vary in (5,50) for SSLasso. To implement EMVS, we use the `EMVS` library in `R`. For SSLasso, we follow the recommendations in Rockova and George (2018). At the beginning, we fix $\lambda_0 = \lambda_1 (= 1)$; then, we increase the value of λ_0 by 1 after the convergence of the optimization step and use the solution of the previous evaluation as the initial point for the following optimization. At the end, we track the solutions of SSLasso with varying λ_0.

We generate synthetic data based on the Bardet-Biedel data set (Scheetz et al., 2006) to be detailed in Section 7. Specifically, we retain the original predictors, set the error variance $\sigma^2 = 1$, and let the first ten elements of the coefficient vector be ±2 with random signs and the rest zero. EBIC and log-MSE paths for each procedure are examined as iteration increases. For each procedure, we consider ten initial points randomly generated from i.i.d. standard Gaussian.

Figures 6 displays the optimization paths of MM, EMVS, SSLasso, and CAAN. We observe that the optimization paths of MM and EMVS quickly converged to some sub-optimal models, corresponding to different solutions when started with different random initializations. Although all procedures failed to provide consistent results when initialized with different starting configurations, CAAN and SSLasso showed similar behaviors and were more stable than EMVS and MM in that the searched models of CAAN and SSLasso tend to have smaller EBIC values. In the Supplementary Materials, we also provide an additional example where the true model size is five, and all methods performed better. In particular, CAAN and SSLasso consistently chose the same model with ten different initializations.

Figure 6 Here

5 Theoretical Properties of Neuronized Priors

5.1 Posterior contraction rates

Because neuronized priors are naturally related to standard ones as demonstrated in Section 2, existing theoretical results for standard frameworks can also be applied to their neuronized counterparts. In this section, we formalize more specific conditions on neuronized priors to achieve optimal theoretical properties as with standard Bayesian sparse regression procedures in high-dimensions.

We first introduce some notations. For two sequences a_n and b_n, $a_n \succ b_n$ means that $a_n / b_n \to \infty$ as $n \to \infty$, $a_n \succeq b_n$ indicates that $b_n = O(a_n)$, and $a_n \asymp b_n$ denotes that the asymptotic rates of a_n and b_n are the same. For a symmetric matrix A, $\lambda_{min}(A)$ and $\lambda_{max}(A)$ denote the minimum and maximum eigenvalues of A, respectively. We assume that the true regression coefficient vector $\boldsymbol{\theta}_0 \in \mathbb{R}^p$ is indeed sparse, and we denote the corresponding set of relevant variables as

$\mathbf{t} = \{j : \theta_{0,j} \neq 0\}$. The size of a finite set $\mathbf{k}$ is denoted by $|\mathbf{k}|$, the sub-matrix of A implicated by the index set $\mathbf{k}$ is $A_{\mathbf{k}}$, and the corresponding sub-vector of $\boldsymbol{\theta}$ is $\boldsymbol{\theta}_{\mathbf{k}}$.

We say that the posterior contraction rate of a parameter $\boldsymbol{\theta} \in \mathbb{R}^p$ is ϵ_n, if for any constant M, $\sup_{\boldsymbol{\theta}_0} \mathbb{E}_{\boldsymbol{\theta}_0} \left\{ \pi\left[d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) > M\epsilon_n \mid \mathbf{y}, X \right] \right\} \to 0$, where $\mathbb{E}_{\boldsymbol{\theta}_0}$ is the expectation with respect to the sampling distribution of the data under the true parameter $\boldsymbol{\theta}_0$, and d is a discrepancy measure, such as the l_1 or l_2 distance. It has been shown that the minimax optimal contraction rate can be achieved for linear regression coefficients under discrete SpSL priors (Castillo et al., 2015), continuous SpSL priors (Narisetty and He, 2014; Ročková et al., 2018; Rockova and George, 2018), and continuous shrinkage priors (Bhattacharya et al., 2015; Ghosh et al., 2017; Song and Liang, 2017). To obtain sufficient conditions for neuronized priors to achieve desirable theoretical properties, we consider the following conditions.

Regularity conditions: There exist constants $C_1, C_2, C_3, C_4 > 0$ such that

(A1) Sparsity: $|\mathbf{t}|^2 (\log p)/n = o(1)$.

(A2) Feature magnitudes: $C_1\sqrt{n} \leq \min_{1\leq j\leq p} \left\|X_j\right\|_2 \leq \max_{1\leq j\leq p} \left\|X_j\right\|_2 \leq C_2\sqrt{n}$.

(A3) Eigenvalues of the design matrix: $\inf_{\mathbf{k}:|\mathbf{k}|\leq|\mathbf{t}|\log n} \lambda_{min}(X_{\mathbf{k}}^{\mathrm{T}} X_{\mathbf{k}}) > C_3 n$.

(A4) Signal strength: $\min_{j\in\mathbf{t}} \theta_{0,j}^2 \succ |\mathbf{t}| \log p / n$ and $\max_{j\in\mathbf{t}} \theta_{0,j}^2 < C_4$.

Condition (A3) is commonly considered in recovering the true model (Bühlmann and van de Geer, 2011; Kim et al., 2012; Narisetty and He, 2014; Shin et al., 2018; Song and Liang, 2017) when p increases much faster than n. Condition (A4) is imposed to prevent degenerating situations where the true coefficients decay or diverge at an extremely fast rate.

Theorem 5.1. *Assume that (A1) – (A4) hold and σ^2 is known. Suppose, for the neuronized prior defined in Definition 1.1 with T be the ReLU function, $(n\log p)^{-1}/16 \le \tau_w^2 \le n^{-1}p^2$ and α_0 follows the distribution in (9) with $(a_0, b_0) = (1, p^u)$ for some constant u > 1. Then, the posterior distribution based on this neuronized prior achieves the optimal posterior contraction rate ϵ_n, i.e.,*

$$\epsilon_n = \begin{cases} |\mathbf{t}|\sqrt{\log p/n}, & \text{under } l_1 \text{ norm}, \\ \sqrt{|\mathbf{t}|\log p/n}, & \text{under } l_2 \text{ norm}. \end{cases} \quad (14)$$

Song and Liang (2017) investigated a similar posterior contraction problem under standard continuous shrinkage priors. They showed that when the tails of a prior decay at a polynomial rate and the prior possesses enough density around the true regression coefficients, the resulting posterior distribution contracts to the true coefficient at the optimal minimax rate. Following their approach, we show that the same claim can be applied to the neuronized version of continuous shrinkage priors as follows:

Theorem 5.2. *Assume that (A1) – (A4) hold and σ^2 is known. Suppose that $T(t) = \exp\{t^2/\{2(r-1)\}\}$ for $r \ge 2$, and let $\tau_w \quad p^{-(u+1)/(r-1)}|\mathbf{t}|\log p/n$ and $-\log\tau_w = O(\log p)$ for some u > 0, and $\alpha_0 = 0$. Then, the posterior distribution of θ based on the corresponding neuronized prior achieves the optimal contraction rate in (14).*

Two practical implications follow immediately from these theorems: for discrete neuronized priors, cares are required for specifying a hyper-prior on α_0 (in particular, the choice of b_0) to control the asymptotic sparsity level; for continuous neuronized priors, the choice of the activation function is important.

5.2 Convergence of naive MCMC algorithms

Convergence properties of MCMC algorithms have been of interest to many researchers. In particular, *geometric ergodicity* of the Markov chain underlying a practical MCMC algorithm has been deemed necessary (Jarner and

Hansen, 2000; Johnson et al., 2013; Roberts et al., 2004; Roberts and Tweedie, 1996). A Markov chain with a transition kernel P and the target distribution $\pi(\cdot \mid \mathbf{y})$ is said to be geometrically ergodic if, $\forall\, \boldsymbol{\theta}^{(0)} \in \Theta$ and $\forall t = 1, 2, \ldots,\ \left\| P^t(\boldsymbol{\theta}^{(0)}, \cdot) - \pi(\cdot \mid \mathbf{y}) \right\|_{TV} \leq C(\boldsymbol{\theta}^{(0)}) \rho^t$, for some $\rho \in (0,1)$ and a finite function $C(\cdot)$, where $\left\| W - G \right\|_{TV} = \sup_{A \in \mathcal{F}} | W(A) - G(A) |$, with $\mathcal{F}$ being a Borel σ-algebra of subsets of Θ, is called the total variation between distributions W and G. A geometrically ergodic Markov chain is called *uniformly ergodic* if C is uniformly bounded on Θ. Geometric ergodicity implies that a generalized central limit theorem is valid for estimates based on MCMC samples (Atchadé et al., 2011; Flegal and Jones, 2011; Jones et al., 2006).

Tan et al. (2013) investigated convergence behaviors of MH algorithms with different proposal distributions and showed that a Gibbs sampler and its MH-within-Gibbs algorithm either are both geometrically ergodic or are both not. By using this fact, we show geometric ergodicity of Algorithm 1 for a wide class of neuronized priors characterized by activation functions with *stable tables*, including all cases discussed previously.

Definition 5.3. *Function T(x), $x \in \mathbb{R}$, is said to have stable tails if there exist constants $C_1, C_2, C_3 > 0$ such that (a) when $x < -C_3$, either $| T'(x) | \leq C_1$ or $| T'(x) | \geq C_2$ and the sign of $T'(x)$ does not change; and (b) when $x > C_3$, either $| T'(x) | \leq C_1$ or $| T'(x) | \geq C_2$ and the sign of $T'(x)$ does not change.*

Theorem 5.4. *Consider the case with X being orthogonal, σ^2 known, and α_0 fixed. Suppose the activation function T for a neuronized prior has stable tails. Then, Algorithm 1 is geometrically ergodic.*

Theorem 5.5. *Under the standard Bayesian linear regression setting, suppose we employ a standard continuous shrinkage prior as in (2) with a heavy-tailed distribution π_τ such that $\pi_\tau(x) \succeq \exp\{-cx^\kappa\}$, x > 0, for some constants c > 0 and*

$0 < \kappa < 1$. *Then, the corresponding MCMC algorithm cannot achieve geometric ergodicity if one updates τ_j conditional on other variables by a RWMH algorithm.*

Theorem 5.4 implies that a naive MH algorithm can be practical for neuronized priors provided that the activation function is not too erratic. All activation functions in Table 1 attain stable tails, so the considered neuronized Bayesian shrinkage procedures achieve a fast convergence of their MCMC. In contrast, Theorem 5.5 shows that, under the conventional setting, geometric ergodicity cannot be achieved by a RWMH algorithm under a heavy-tailed prior on τ_j, e.g., the horseshoe prior. In this setting, the conditional posterior distribution of τ_j, used in the Gibbs sampler, is also heavy-tailed (at least sub-exponential). As shown in Mengersen and Tweedie (1996), when the target distribution of a RWMH algorithm is heavy-tailed, the resulting MCMC algorithm cannot be geometrically ergodic.

To attain an optimal rate of posterior contraction, however, we need to choose a heavy-tailed prior on τ_j's as discussed in Section 5.1. Thus, some clever, but case-specific, MCMC moves need to be designed. For example, using a slice sampler for updating τ_j in horseshoe priors can be shown to be geometrically ergodic (Roberts and Rosenthal, 1999). Even so, empirical results in Sections 6 and 7 show that employing the neuronized horseshoe prior with Algorithm 1 is computationally more efficient than an efficient MCMC algorithm using the slice sampling under the conventional framework, which may be due to high correlations between the τ_j's and θ_j's when using representation (2) for such a prior. The form of the neuronized prior can be viewed as a transformed parameter expansion (Liu and Wu, 1999) via an activation function, which improves the mixing property of Algorithm 1. This advantage of parameter expansion is also discussed in Scott (2010).

6 Simulation Studies

6.1 Simulation setups and evaluation criteria

Under the Bayesian regression framework, we compare the effect of some standard priors, such as Bayesian Lasso, the horseshoe, and the discrete SpSL as in (3), with that of their neuronized counterparts. We also include a scalable approximation algorithm called Skinny Gibbs (SkG; Narisetty et al. (2019)) for continuous SpSL priors. By ignoring the correlation between selected variables and the other variables, SkG improves computational efficiency.

Among the optimization-based algorithms in comparison, we include two penalized likelihood procedures, Lasso (Tibshirani, 1996) and SCAD (Fan and Li, 2001). Cross-validations (CV) and either BIC (when $n > p$) or EBIC (when $n < p$) are used to select tuning parameters for both LASSO and SCAD. As a calibration, we provide the oracle estimate, i.e., the OLS estimate under the true model. Posterior mode-finding algorithms includes MM, EMVS, SSLasso and CAAN under two neuronized SpSL priors, in which the slab part matches either Laplace or Cauchy (denoted as N-SpSL-L and N-SpSL-C, respectively; see Table 1). We impose $a_0 = b_0 = 1$ in (8), $v_1 = 10$ for EMVS, and $\lambda_1 = 0.1$ for SSLasso. Then, we evaluate the MAP estimators based on different choices of v_0 for EMVS and λ_0 for SSLasso and select a value that minimizes BIC for low-dimensions and EBIC for high-dimensions. `R` packages `EMVS` and `SSLASSO` (available on the CRAN) are used for the implementation.

To evaluate the estimation performances, we report both the *Mean Squared Error* (MSE) and the cosine of the angle between the true coefficient vector $\boldsymbol{\theta}_0$ and its estimate $\boldsymbol{\theta}$, i.e., $\frac{\boldsymbol{\theta}_0^{\mathrm{T}}\boldsymbol{\theta}}{\|\boldsymbol{\theta}_0\|\,\|\boldsymbol{\theta}\|}$, for each method. The angle measure is more stringent as it cannot benefit from a simple shrinkage. To measure model selection performances, we examine the *Matthews correlation coefficient* (MCC; Matthews (1975)) defined as $\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$, where TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. The value of MCC is bounded by one, and the closer to one MCC is, the better a model selection procedure is.

The *Effective Sample Size* (ESS) is adopted as an efficiency measure for a MCMC procedure, which is defined as $\mathrm{ESS} = \dfrac{N}{1 + 2\sum_{t}^{\infty} \rho(t)}$, where N is number of MCMC samples and $\rho(t)$ is the lag-t autocorrelation. We report the average of the ESS (per second) of the ten "most significant" coefficients, i.e., with the largest posterior variances.

We consider a Toeplitz design (i.e., AR(1) dependence) to generate the covariates: $X_i \sim N(0, \Sigma)$ for $i = 1, \ldots, n$, where $\Sigma = (\sigma_{lk})$ with $\sigma_{lk} = 0.7^{|l-k|}$ for $1 \le l, k \le p$. Additional simulation settings, such as one with i.i.d. standard Gaussian covariates, can be found in Supplementary Materials. Two "low"-dimensional cases are tested: (a) $n = 100, p = 50$; and (b) $n = 400, p = 100$. The number of nonzero β_j's is $p_1 = p/10$, with each taking ±0.2 randomly. Another two "high"-dimensional cases (with $n < p$) are also tested: (c) $n = 100, p = 300$; and (d) $n = 150, p = 1000$. We let the coefficient vector be $\beta_0 = \{\pm 0.4, \pm 0.45, \pm 0.5, \pm 0.55, \pm 0.6, 0, \ldots, 0\}$. The error variance is set at $\sigma^2 = 1$ for all scenarios.

6.2 Technicalities about computational strategies

For using regular SpSL priors in (3), we impose a uniform distribution on η. For its neuronized version, we impose a hyper-prior on α_0 as in (9). We consider the Jeffrey's prior on σ^2 for all Bayesian procedures; i.e. $\pi(\sigma^2) \propto 1/\sigma^2$. For the horseshoe prior and its neuronized version, we numerically find a proper τ_w^2 as discussed in Section 3.2. For Bayesian Lasso and its neuronized version, we choose the global shrinkage parameter that matches the tuning parameter value λ_{CV} determined by cross-validations for the standard Lasso procedure.

For standard discrete SpSL priors, we examine both the Gaussian and Cauchy distributions for the slab part. We employ the half-collapsed Gibbs sampler as discussed in Section 4.3, denoted as SpSL-G(HCG) and SpSL-C(HCG) for Gaussian slabs and Cauchy slabs, respectively. Note that the use of a Gaussian

slab does not match the neuronized SpSL prior with a ReLU activation function since the product of two independent Gaussians in the neuronization formulation results in a Laplace-like slab distribution. Nevertheless, we use the standard Gaussian SpSL prior to sustain computational efficiency.

We let "N-SpSL-L(Exact)" denote the neuronized SpSL prior implemented via the exact Gibbs sampler as in Proposition 4.1, and use "N-SpSL-L(RW)" and "N-SpSL-C(RW)", corresponding to a Laplace-like and a Cauchy slab, respectively, to denote that implementation via Algorithm 1, which uses RWMH to update a_0. Since "N-SpSL-L(RW)" produces identical results as "N-SpSL-L(Exact)" but is 60% - 80% less efficient (see the Supplementary Materials for a detailed comparison), we omit its results from the comparison tables. For the standard SpSL prior with a Cauchy slab (i.e., "SpSL-C"), we lose the conjugacy and need to use numerical integration (a trapezoidal rule) to marginalize out each coefficient in a Gibbs sampler. In contrast, its neuronized version N-SpSL-C(RW) can be implemented by Algorithm 1 directly, only requiring one to choose an appropriate activation function as in Table 1. We note that, due to the existence of a location-shift by a_0, the resulting neuronized prior differs slightly from the standard SpSL prior with a Cauchy slab, although they share the same behavior at tails, i.e., decaying at the rate of x^{-2}.

The Bayesian Lasso is implemented by an efficient Gibbs sampler as in Park and Casella (2008). For the standard horseshoe prior, we use a slice sampler to sample each local shrinkage parameter. For both procedures, since the posterior distribution does not provide a sparse solution, we set a threshold of $0.1\times\hat{\sigma}$, where $\hat{\sigma}^2$ is the posterior mean of the regression error variance, and select only those predictors whose posterior mean estimates of the coefficients have a magnitude higher than the threshold. For all procedures, we generate 10,000 MCMC samples after 2,000 burn-in iterations, replicate 100 data sets, and average the results over the replications.

Tables 2, 3 Here

6.3 Results discussion

Tables 2 and 3 summarize low-dimensional and high-dimensional simulation results, respectively. In general, we observe that (a) no procedure clearly dominate others in all situations for all criteria; (b) Bayesian averaging results in a better performance than the corresponding MAP estimator; (c) the Lasso-based procedures typically show the best estimation performance under the low-dimensional settings, but they tends to select more false positives; (d) the SpSL-based procedures attain competitive model selection performances under high-dimensional settings.

SpSL-G(HCG) shows the most efficient performance in terms of ESS because it takes advantage of the conjugacy to marginalize continuous components, which, however, is also restrictive. For example, with a Cauchy slab, SpSL-C(HCG) has a much reduced ESS because it has to employ a numerical integration method for marginalization. In contrast, its neuronized counterpart N-SpSL-C(RW), which is implemented via a single unified algorithm that can accommodate any activation function, obtained an ESS 80% larger than that of SpSL-C(HCG).

In general, neuronized priors performed robustly throughout all situations, with improved computational efficiency in comparison with their standard counterparts for most cases. In particular, the N-HS was at least two times more efficient than the HS in terms of ESS in all simulation scenarios, which might be due to the highly correlated latent structure between τ_j's and θ_j's in the standard horseshoe prior. We verified via very long MCMC iterations that our implementations of the horseshoe prior and its neuronized counterpart indeed produce identical posterior inference results (more details are given in Supplementary Materials). Their differences shown in the tables are due to numerical approximation errors.

The tables also list the performances of optimization-based SpSL procedures including the CAAN, the MM algorithm, the EMVS, and the SSLasso. The results show that, overall, the CAAN and the SSLasso significantly outperformed the MM and the EMVS algorithms in terms of estimation and model selection.

7 Real Data Examples

We analyze both the Boston housing data set introduced in Section 2 and the Bardet-Biedl data set available in the `R` package `flare`. The Bardet-Biedl data set contains mRNA expression values of 31,042 probe sets in eye tissues of 120 twelve-week old male rats, normalized by the robust multi-chip averaging method (Irizarry et al., 2003). This data set has been analyzed previously (Fan et al., 2011; Huang et al., 2008; Kim et al., 2008). As with those papers, our goal is to find a subset of probe sets that are associated with the probe set *1389163_at*, corresponding to gene *TRIM32*, which is linked to the Bardet-Biedl syndrome. All probe sets are ranked according to the magnitudes of their marginal correlations with *1389163_at*, and the top 200 are retained for the regression analysis (n = 120 and p = 200).

Figure 7 shows the ESS obtained at 5 seconds, 10 seconds, and 20 seconds, respectively, by the MCMC algorithms corresponding to different priors for the Boston housing data set and the Bardet-Biedl data set. The efficiency comparison results are consistent with those in the simulation study. In (a), we observe that SpSL-G(HCG) obtained the largest ESS, and N-SpSL-L(Exact) was about 50% less efficient. With the Cauchy slab, SpSL-C(HCG) and N-SpSL-C(RW) performed similarly. For (b), the advantage of SpSL-G(HCG) over N-SpSL-L(Exact) appeared to have shrunk, and N-SpSL-C(RW) attained 50% more ESS than SpSL-C(HCG) does under the same time unit. In (c) and (d), we see clear evidences that the neuronized horseshoe formulation results in significantly more efficient computation than the standard one.

Figure 7 Here

Table 4 Here

We employ the out-of-sample mean squared prediction error (MSPE) to measure the prediction performance of each procedure by setting aside a randomly selected 10% of the samples for testing. We also consider the cosine angle between the test responses and the corresponding predicted values; i.e., $\mathbf{y}_{\text{test}}^{\text{T}}\mathbf{y} / (\|\mathbf{y}_{\text{test}}\|_2 \cdot \|\mathbf{y}\|_2)$. This measure is useful in cases where people care more about how correlated the prediction is with the observation, such as in financial market forecasting. The process is replicated 100 times and the averages are reported in Table 4, which shows that the neuronized priors performed comparably with their standard counterparts. In particular, N-SpSL(MAP) achieve the smallest MSPE for both data sets. For the Boston housing data set, the sizes of the models selected by different approaches are comparable. For the Bardet-Biedl data set, however, the Bayesian Lasso and its neuronized version N-BL(RW) selected much larger models than other methods. We also noticed that both EMVS and SkG selected the null model but had different prediction results, which is due to their adoption of different non-degenerate priors for the model parameters.

8 Discussion

Inspired by the idea of neuron activation, which is central to all neural network-based methods, we propose to use an activation function and a product representation to unify and extend shrinkage priors employed in high-dimensional Bayesian regression analyses. By simply changing the activation function, our unified framework (together with its companion software package) enables practitioners to easily test out effects of different classes of priors for a regression model. We show that the neuronization procedure can be efficiently implemented to emulate a wide class of distributions including many non-conjugate and mixture priors, which is a clear advantage over existing Bayesian regression frameworks. The neuronization formulation can also be easily

extended to a broad class of nonlinear models (such as logistic regression), where the lack of prior conjugacy may hinder the applicability and scalability of conventional Bayesian regression procedures, especially when one wants to employ discrete SpSL priors.

Furthermore, the neuronization idea can be applied to construct structured sparsity priors for more complicated models. For example, some sparsity patterns may be spatially correlated, which is computationally challenging if one directly imposes spatial correlations among the latent indicator variables that underlie either a discrete or a continuous multivariate SpSL prior. In contrast, a multivariate structure can be easily imposed on the α_j's in the neuronized prior setting (4). Because all parameters in such a setting are continuous and non-latent, a Hamiltonian Monte Carlo algorithm can be used to efficiently sample from the posterior distribution.

All introduced algorithms are coded in the `R` package `NPrior` available on a Github repository `github.com/rabbitinasubmarine/NPrior`.

Acknowledgment

This research is supported in part by the NSF grants DMS-1903139, DMS-2015528, and DMS-2015411.

References

Armagan, A., Dunson, D., and Lee, J. (2013). Generalized double Pareto shrinkage. *Statistica Sinica*, 23(1):119–143.

Atchadé, Y. F. et al. (2011). Kernel estimators of asymptotic variance for adaptive Markov chain Monte Carlo. *The Annals of Statistics*, 39(2):990–1011.

Berger, J. O. and Molina, G. (2005). Posterior model probabilities via path-based pairwise priors. *Statistica Neerlandica*, 59(1):3–15.

Berger, J. O., Strawderman, W. E., et al. (1996). Choice of hierarchical priors: admissibility in estimation of normal means. *The Annals of Statistics*, 24(3):931–951.

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

Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015). Dirichlet-Laplace priors for optimal shrinkage. *J. Am. Statist. Ass.*, 110(512):1479–1490.

Brent, R. P. (1973). *Algorithms for minimization without derivatives.* Prentice-Hall.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

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

Castillo, I., Schmidt-Hieber, J., Van der Vaart, A., et al. (2015). Bayesian linear regression with sparse priors. *Ann. Statist.*, 43(5):1986–2018.

Castillo, I. and van der Vaart, A. (2012). Needles and straw in a haystack: Posterior concentration for possibly sparse sequences. *Ann. Statist.*, 40(4):2069–2101.

Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Dellaportas, P., Forster, J. J., and Ntzoufras, I. (2002). On Bayesian model and variable selection using mcmc. *Statistics and Computing*, 12(1):27–36.

Diaconis, P., Khare, K., Saloff-Coste, L., et al. (2008). Gibbs sampling, exponential families and orthogonal polynomials. *Statistical Science*, 23(2):151–178.

Fan, J., Feng, Y., and Song, R. (2011). Nonparametric independence screening in sparse ultra-high-dimensional additive models. *J. Am. Statist. Ass.*, 106(494):544–557.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Ass.*, 96(456):1348–1360.

Flegal, J. M. and Jones, G. L. (2011). Implementing MCMC: estimating with confidence. *Handbook of Markov Chain Monte Carlo*, pages 175–197.

George, E. and McCulloch, R. (1993). Variable selection via Gibbs sampling. *J. Am. Statist. Ass.*, 88(423):881–889.

Ghosh, P., Chakrabarti, A., et al. (2017). Asymptotic optimality of one-group shrinkage priors in sparse high-dimensional problems. *Bayesian Analysis*, 12(4):1133–1161.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, pages 711–732.

Guan, Y. and Stephens, M. (2011). Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *Ann. Appl. Statist.*, pages 1780–1815.

Hahn, P. R. and Carvalho, C. M. (2015). Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *J. Am. Statist. Ass.*, 110(509):435–448.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. *J. Am. Statist. Ass.*, 102(478):507–516.

Hoff, P. D. (2017). Lasso, fractional norm and structured sparse estimation using a hadamard product parametrization. *Computational Statistics & Data Analysis*, 115:186–198.

Huang, J., Ma, S., and Zhang, C.-H. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pages 1603–1618.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264.

Jarner, S. F. and Hansen, E. (2000). Geometric ergodicity of metropolis algorithms. *Stochastic processes and their applications*, 85(2):341–361.

Ji, C. and Schmidler, S. C. (2013). Adaptive Markov chain Monte Carlo for bayesian variable selection. *Journal of Computational and Graphical Statistics*, 22(3):708–728.

Johnson, A. A., Jones, G. L., and Neath, R. C. (2013). Component-wise markov chain monte carlo: Uniform and geometric ergodicity under mixing and composition. *Statistical Science*, 28(3):360–375.

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

Johnson, V. E. and Rossell, D. (2010). On the use of non-local prior densities in Bayesian hypothesis tests. *J. R. Statist. Soc. B*, 72(2):143–170.

Johnson, V. E. and Rossell, D. (2012). Bayesian model selection in high-dimensional settings. *J. Am. Statist. Ass.*, 107(498):649–660.

Jones, G. L., Haran, M., Caffo, B. S., and Neath, R. (2006). Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101(476):1537–1547.

Kim, Y., Choi, H., and Oh, H.-S. (2008). Smoothly clipped absolute deviation on high dimensions. *J. Am. Statist. Ass.*, 103(484):1665–1673.

Kim, Y., Kwon, S., and Choi, H. (2012). Consistent model selection criteria on high dimensions. *The Journal of Machine Learning Research*, 13:1037–1057.

Kirkpatrick, S. and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Liu, J. S. (1994). The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966.

Liu, J. S., Liang, F., and Wong, W. H. (2000). The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134.

Liu, J. S. and Sabatti, C. (2000). Generalised gibbs sampler and multigrid monte carlo for bayesian computation. *Biometrika*, 87(2):353–369.

Liu, J. S. and Wu, Y. N. (1999). Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448):1264–1274.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. Citeseer.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Mengersen, K. L. and Tweedie, R. L. (1996). Rates of convergence of the hastings and metropolis algorithms. *The Annals of Statistics*, 24(1):101–121.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *J. Am. Statist. Ass.*, 83(404):1023–1032.

Moran, G. E., Ročková, V., George, E. I., et al. (2018). Variance prior forms for high-dimensional bayesian variable selection. *Bayesian Analysis*, pages 1091–1119.

Narisetty, N. N. and He, X. (2014). Bayesian variable selection with shrinking and diffusing priors. *Ann. Statist.*, 42(2):789–817.

Narisetty, N. N., Shen, J., and He, X. (2019). Skinny Gibbs: A consistent and scalable Gibbs sampler for model selection. *Journal of the American Statistical Association*, 114(527):1205–1217.

Park, T. and Casella, G. (2008). The Bayesian lasso. *J. Am. Statist. Ass.*, 103(482):681–686.

Polson, N. and Scott, J. (2010). Shrink globally, act locally: sparse Bayesian regularization and prediction. In *Proceedings of the 9th Valencia World Meeting on Bayesian Statistics*, volume 9, pages 501–538. Oxford University Press.

Polson, N., Scott, J., and Windle, J. (2013). Bayesian inference for logistic models using polya–gamma latent variables. *J. Am. Statist. Ass.*, 108(504):1339–1349.

Robert, C. P. (1995). Convergence control methods for Markov chain Monte Carlo algorithms. *Statistical Science*, pages 231–253.

Roberts, G. O. and Rosenthal, J. S. (1999). Convergence of slice sampler markov chains. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):643–660.

Roberts, G. O., Rosenthal, J. S., et al. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, 16(4):351–367.

Roberts, G. O., Rosenthal, J. S., et al. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71.

Roberts, G. O. and Tweedie, R. L. (1996). Geometric convergence and central limit theorems for multidimensional hastings and metropolis algorithms. *Biometrika*, 83(1):95–110.

Ročková, V. et al. (2018). Bayesian estimation of sparse signals with a continuous spike-and-slab prior. *The Annals of Statistics*, 46(1):401–437.

Rockova, V. and George, E. I. (2014). EMVS: The EM approach to Bayesian variable selection. *J. Am. Statist. Ass.*, 109(506):828–846.

Rockova, V. and George, E. I. (2018). The spike-and-slab lasso. *J. Am. Statist. Ass.*, 113(521):431–444.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rossell, D. and Telesca, D. (2017). Non-local priors for high-dimensional estimation. *J. Am. Statist. Ass.*, (just-accepted).

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., et al. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464.

Scott, J. G. (2010). Parameter expansion in local-shrinkage models. *arXiv preprint arXiv:1010.5265*.

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

Shin, M., Bhattacharya, A., and Johnson, V. E. (2018). Scalable Bayesian variable selection using nonlocal prior densities in ultrahigh-dimensional settings. *Statistica Sinica*, 28(2):1053–1081.

Song, Q. and Liang, F. (2017). Nearly optimal bayesian shrinkage for high dimensional regression. *arXiv preprint arXiv:1712.08964*.

Song, S., Hou, L., and Liu, J. S. (2020). A fexible bayesian regression approach for accurate polygenic risk prediction. *Technical Report*, in preparation.

Tan, A., Jones, G. L., and Hobert, J. P. (2013). On the geometric ergodicity of two-variable Gibbs samplers. In *Advances in Modern Statistical Theory and Applications: A Festschrift in Honor of Morris L. Eaton*, pages 25–42. Institute of Mathematical Statistics.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, pages 267–288.

van de Geer, S. A., Bühlmann, P., et al. (2009). On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392.

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

van der Pas, S., Salomond, J.-B., Schmidt-Hieber, J., et al. (2016). Conditions for posterior contraction in the sparse normal means problem. *Electronic Journal of Statistics*, 10(1):976–1000.

Yen, T.-J. et al. (2011). A majorization–minimization approach to variable selection using spike and slab priors. *Ann. Statist.*, 39(3):1748–1775.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In *Bayesian inference and decision techniques: Essays in Honor of Bruno de Finetti*, pages 233–243. North Holland, Amsterdam.

Zhang, J. L., Lin, M. T., Liu, J. S., and Chen, R. (2007). Lookahead and piloting strategies for variable selection. *Statistica Sinica*, 17(3):985–1003.

Algorithms

Algorithm 1 A general MCMC algorithm for neuronized priors

Initialize the parameters $\boldsymbol{\alpha}$, α_0, $\boldsymbol{w}$, σ^2. For $i = 1, \ldots N$

- Sample $\boldsymbol{w}$ conditional on $\mathbf{y}, \boldsymbol{\alpha}, \sigma^2$ from (11).
- Set $\mathbf{r} = \mathbf{y} - X\boldsymbol{\theta}(\boldsymbol{\alpha}, \boldsymbol{w})$.

For $j = 1, \ldots, p$

- Update $\mathbf{r} = \mathbf{r} + X_j T(\alpha_j - \alpha_0) w_j$.

Repeat M times

- Sample α_j from $[\alpha_j \mid \mathbf{y}, \boldsymbol{\alpha}_{(-j)}, \boldsymbol{w}_{(-j)}, \sigma^2, \tau_w^2]$ by using a RWMH step for

the log-target function $-\log(v_j)/2 - \alpha_j^2/2 + v_j m_j^2/(2\sigma^2)$, —(*)

where $v_j = X_j^{\mathrm{T}} X_j T^2 (\alpha_j - \alpha_0) + \sigma^2/\tau_w^2$ and $m_j = \mathbf{r}^{\mathrm{T}} X_j T(\alpha_j - \alpha_0)/v_j$.

- Sample w_j from $[w_j \mid \mathbf{y}, \boldsymbol{\alpha}_{(-j)}, \alpha_j, \boldsymbol{w}_{(-j)}, \sigma^2, \tau_w^2]$, which is $N(m_j, \sigma^2 v_j^{-1})$.

End.

- Update $\mathbf{r} = \mathbf{r} - X_j T(\alpha_j - \alpha_0) w_j$.

End.

- Sample σ^2 from $[\sigma^2 \mid \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{w}, \tau_w^2]$, which is an inverse Gamma.

- When $a_0 = b_0 = 1$, sample δ from (12). In case where $a_0 \neq 1$ or $b_0 \neq 1$,

draw δ via Algorithm 2. Then, update $\boldsymbol{\alpha} = \boldsymbol{\alpha} + \delta \mathbf{1}$ and $\alpha_0 = \alpha_0 + \delta$.

End.

Algorithm 2 The Multiple-Try-Metropolized Independence Sampler (MTM-IS)

Let the target density be $g(\delta)$, and let the trial/proposal density be $h(\delta)$ (a default is (12)). We define $w(\delta) = g(\delta)/h(\delta)$. Let $\delta^{(t)}$ be the sample at step t. Then, at step t + 1,

- Draw $\delta_1, \ldots, \delta_m$ i.i.d. from the trial density $h()$;
- Select $\delta' = \delta_j$ from $\{\delta_1, \ldots, \delta_m\}$ with probability $\propto w(\delta_j)$

- Compute $p^{(t)} = \min\left\{1, \frac{\sum_{k=1}^{m} w(\delta_k)}{w(\delta^{(t)}) + \sum_{k \neq j} w(\delta_k)}\right\}$; let $\delta^{(t+1)} = \delta'$ with probability $p^{(t)}$, and let $\delta^{(t+1)} = \delta^{(t)}$ with probability $1 - p^{(t)}$.

Algorithm 3 The Coordinate-Ascent Algorithm for Neuronized prior (CAAN)

- Initialize the parameters $\boldsymbol{\alpha}$, α_0, $\boldsymbol{w}$, σ^2.

- Set a candidate set of temperature, $\{t_{(1)}, \ldots, t_{(2L+1)}\}$, where $t_{(l)} > t_{(l+1)}$ and $t_{(2L+1)} = 1$.

For $l = 1, \ldots, 2L+1$

- Set $t = t_{(l)}$.

- Set $\mathbf{r} = \mathbf{y} - X\boldsymbol{\theta}(\boldsymbol{\alpha}, \boldsymbol{w})$.

For M iterations

For $j = 1, \ldots, p$

- Update $\mathbf{r} = \mathbf{r} + X_j T(\alpha_j - \alpha_0) w_j$.

- Update α_j by optimizing the logarithm of the marginalized posterior density function $-\log(v_j)/2 - \alpha_j^2/2 + v_j m_j^2/2$ with respect to α_j,

where $v_j = X_j^{\mathrm{T}} X_j T^2 (\alpha_j - \alpha_0) + \sigma^2 / \tau_w^2$ and $m_j = \mathbf{r}^{\mathrm{T}} X_j T(\alpha_j - \alpha_0) / v_j$. — (◊)

- Update w_j by m_j.

- Update $\mathbf{r} = \mathbf{r} - X_j T(\alpha_j - \alpha_0) w_j$.

End.

Every N iterations,

- Update $\sigma^2 = (\|\mathbf{y} - X\boldsymbol{\theta}(\boldsymbol{\alpha}, \boldsymbol{w})\|_2^2 / t + 2b_1) / \{n + 2a_1 + 2\}$.

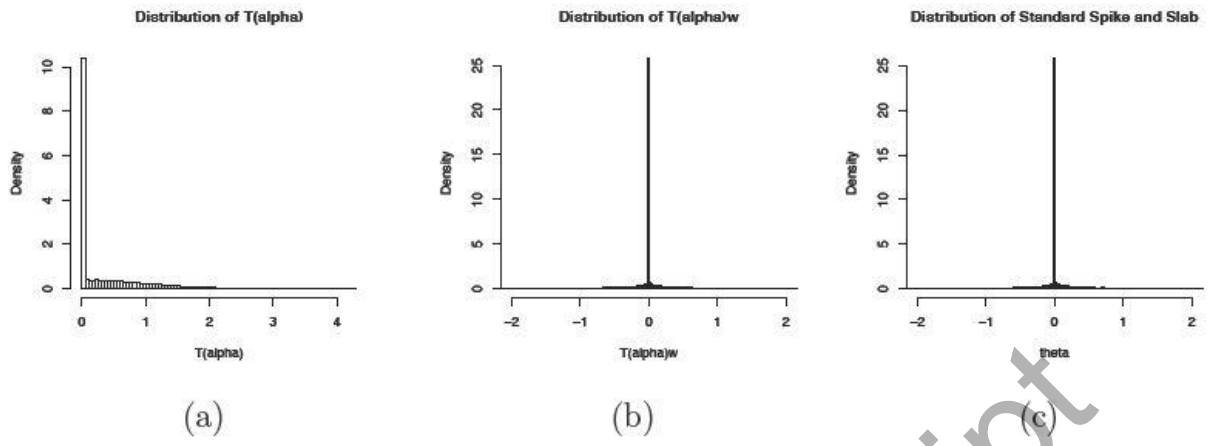
If $l < L$

- Set $\sigma^2 = \sigma^2 + z$, where $z \sim \exp(1)$.

- Update $\alpha_0 = \{\sum_{j=1}^{p} \mathbb{I}(\alpha_j > \alpha_0) + a_0 - 1\} / (p + b_0 - 2)$.

End.

End.



(a) (b) (c)

Fig. 1 (a) histogram of $T(\alpha)$; (b) histogram of $T(\alpha)w$; (c) histogram of the standard SpSL prior in (7).

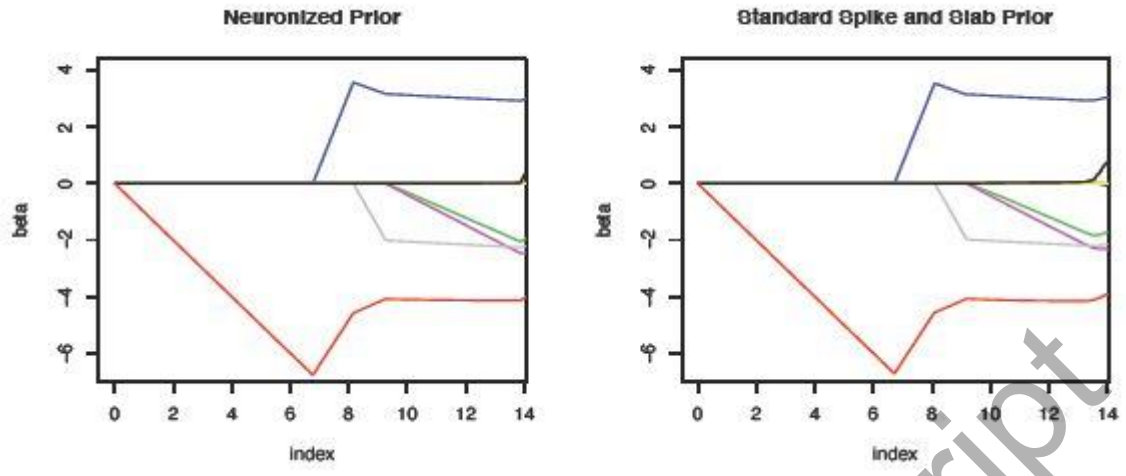
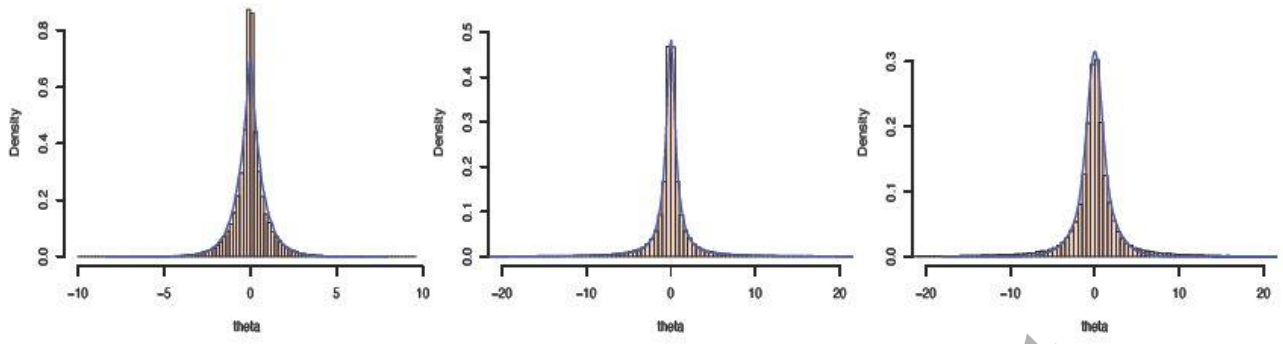


Fig. 2 Solution paths of the neuronized prior and the discrete SpSL prior.



(a) Bayesian Lasso (b) Horseshoe prior (c) Cauchy prior

Fig. 3 Histograms of some prior distributions. The blue lines indicate the density functions of their neuronized counterpart.

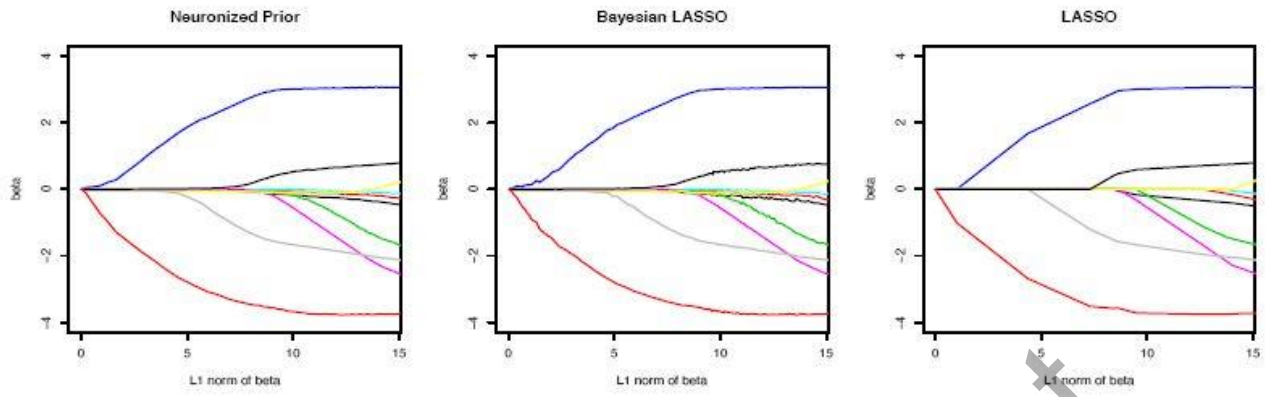


Fig. 4 Solution paths of the neuronized prior, the Bayesian Lasso and the Lasso.

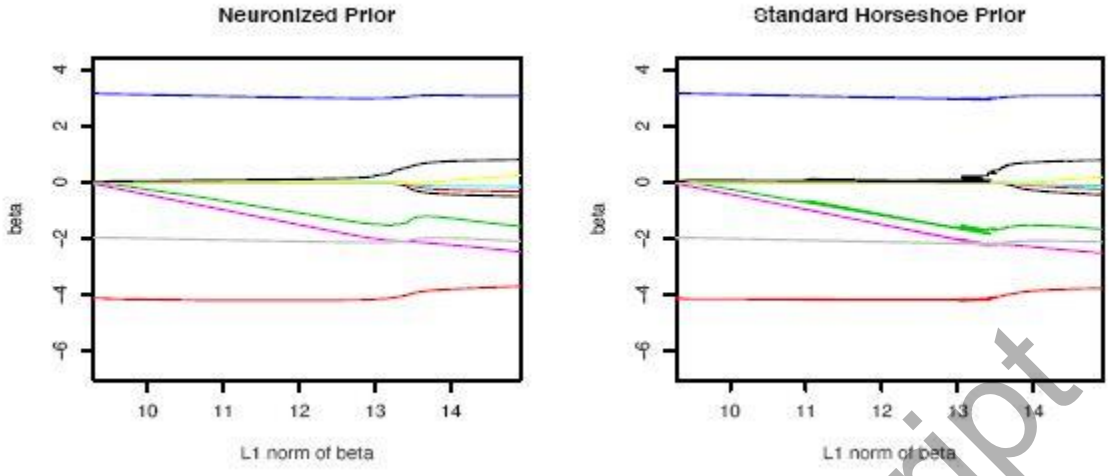


Fig. 5 Solution paths of the neuronized prior and the horseshoe prior.

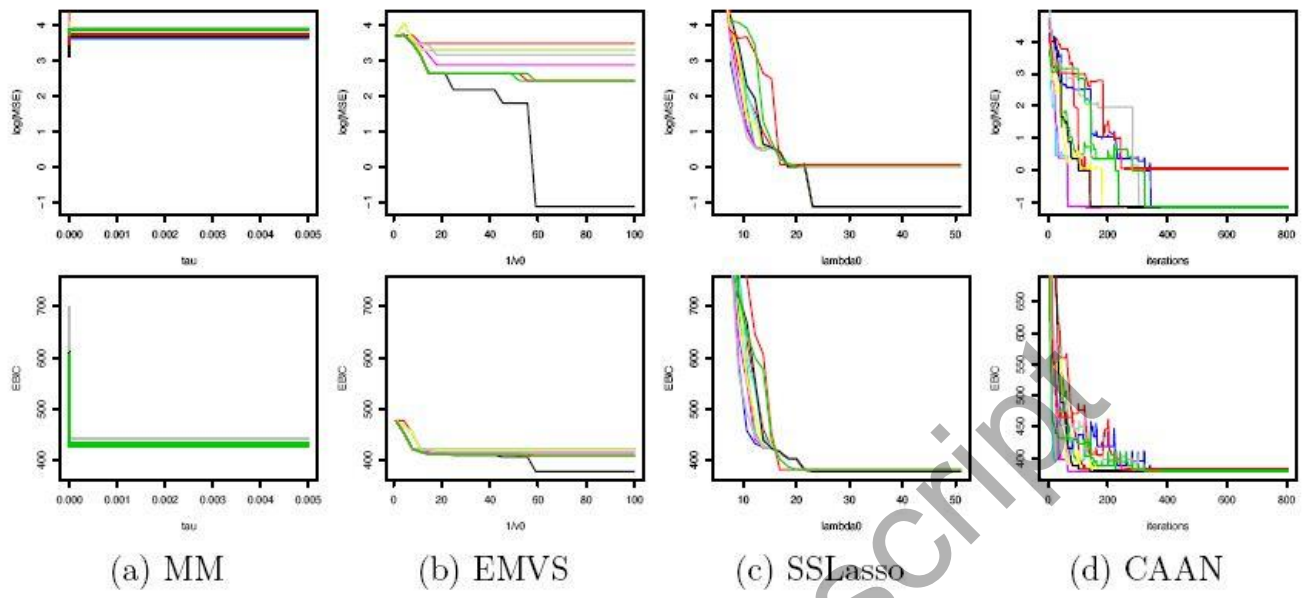


Fig. 6 Trace plots of the log-MSE (top row) and EBIC (bottom row) paths from 10 different initial points for the four optimization algorithms, based on a synthetic data set generated from the Bardet-Biedl dataset (n = 120 and $p = 200$) with the true model 10. The MM procedure used $\tau_3 = 10^{-2}$.

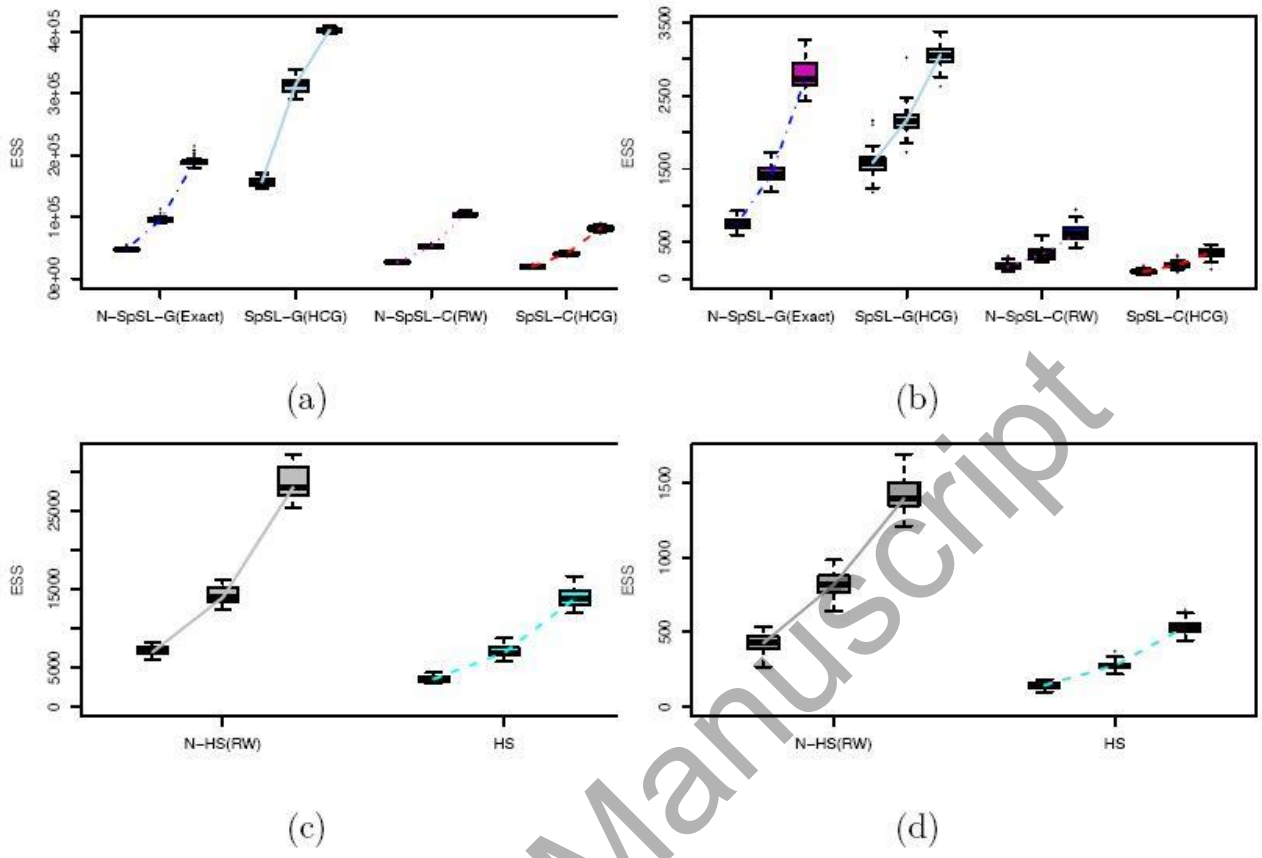


Fig. 7 Effective samples size versus actual computation time for the Boston housing data set (the first column) and the Bardet-Biedl data set (the second column). For each procedure, the first, second, and third boxplot indicates the ESS evaluated at 5 seconds, 10 seconds, and 20 seconds, respectively.

Table 1 The choice of T for neuronized priors and the corresponding existing Bayesian priors. The default value of τ_w^2 is set to be one, if it is not specified.

Activation function $T(t)$	Target Prior
$\max\{0,t\}$ (ReLU)	Discrete SpSL with Laplace slab
t (linear)	Bayesian Lasso
$\exp\{0.5\text{sign}(t)t^2+0.733t\}$	Horseshoe
$T(t)=\exp\{0.5t^2-1.27t+0.29\}$	Cauchy

Table 2 Results for the low-dimensional setting with dependent covariates. SpSL, HS, and BL indicate the procedure based on the discrete SpSL, the horseshoe, and Bayesian Lasso priors, respectively. The sign "N" stands for the neuronized version of the corresponding prior.

	$(n = 200, p = 50)$					$(n = 400, p = 100)$				
Method	MSE	Cos	MCC	FP	ESS	MSE	Cos	MCC	FP	ESS
Oracle	0.069(0.008)	0.870				0.069(0.013)	0.929			
SpSL-G(HCG)	0.167(0.009)	0.558	0.48(0.02)	0.01	15242.5	0.295(0.014)	0.581	0.49(0.01)	0.03	2594.8
N-SpSL-L(Exact)	0.150(0.009)	0.592	0.53(0.04)	0.03	5826.2	0.261(0.014)	0.630	0.53(0.01)	0.07	1089.6
SpSL-C(HCG)	0.159(0.009)	0.582	0.51(0.03)	0.02	1023.6	0.275(0.014)	0.613	0.51(0.01)	0.05	277.7
N-SpSL-C(RW)	0.168(0.009)	0.554	0.48(0.02)	0.01	1747.8	0.299(0.014)	0.574	0.48(0.01)	0.03	422.2
HS	0.142(0.008)	0.594	**0.55**(0.03)	0.04	610.1	0.240(0.012)	0.658	0.56(0.01)	0.04	91.1
N-HS(RW)	0.143(0.008)	0.594	**0.55**(0.03)	0.03	1357.0	0.243(0.012)	0.653	0.55(0.01)	0.04	217.3
BL	0.198(0.011)	0.569	0.51(0.01)	2.94	2798.2	0.273(0.010)	0.669	0.60(0.02)	3.25	397.5
N-BL(RW)	0.157(0.008)	0.601	0.53(0.01)	1.49	1152.8	**0.218**(0.009)	**0.698**	**0.62**(0.02)	0.99	361.2
SkG	0.159(0.008)	0.573	0.50(0.01)	0.02	9961.8	0.276(0.010)	0.614	0.51(0.01)	0.06	1189.6
SpSL(MM)	0.193(0.012)	0.481	0.41(0.04)	1.27		0.310(0.012)	0.582	0.48(0.02)	2.77	
EMVS	0.225(0.010)	0.436	0.45(0.02)	0.00		0.412(0.013)	0.419	0.40(0.02)	0.00	

	$(n = 200, p = 50)$					$(n = 400, p = 100)$				
SSLasso	0.208(0.009)	0.449	0.41(0.02)	0.80		0.355(0.010)	0.510	0.49(0.01)	1.26	
N-SpSL(MAP)	0.222(0.010)	0.537	0.48(0.02)	1.03		0.310(0.013)	0.624	0.57(0.02)	1.12	
N-BL(MAP)	0.152(0.009)	**0.616**	0.54(0.02)	1.70		0.226(0.011)	0.684	0.61(0.01)	1.86	
Lasso(CV)	**0.134**(0.009)	0.608	0.48(0.02)	5.34		0.228(0.011)	0.668	0.45(0.01)	13.61	
SCAD(CV)	0.222(0.009)	0.528	0.39(0.02)	3.72		0.339(0.011)	0.572	0.38(0.02)	8.86	
Lasso(BIC)	0.174(0.010)	0.465	0.48(0.02)	0.44		0.307(0.013)	0.516	0.57(0.01)	0.56	
SCAD(BIC)	0.187(0.010)	0.465	0.45(0.02)	0.69		0.361(0.014)	0.475	0.50(0.02)	1.14	

Table 3 Results for the high-dimensional setting with dependent covariates.

	$(n = 100, p = 300)$					$(n = 150, p = 1000)$				
Method	MSE	Cos	MCC	FP	ESS	MSE	Cos	MCC	FP	ESS
Oracle	0.141(0.045)	0.956				0.084	0.977			
SpSL-G(HCG)	0.872(0.059)	0.630	0.56(0.04)	0.11	3123.3	0.759(0.052)	0.675	0.58(0.02)	0.09	709.9
N-SpSL-L(Exact)	0.824(0.057)	0.641	0.56(0.02)	0.26	837.8	0.709(0.051)	0.693	0.62(0.02)	0.24	185.3
SpSL-C(HCG)	0.829(0.057)	**0.647**	0.58(0.02)	0.23	113.8	**0.699**(0.046)	**0.705**	0.63(0.01)	0.16	30.9
N-SpSL-C(RW)	0.882(0.059)	0.625	0.56(0.04)	0.06	261.3	0.759(0.055)	0.673	0.58(0.02)	0.10	66.4
HS	0.820(0.054)	0.629	0.57(0.02)	0.88	15.5	0.765(0.049)	0.655	0.56(0.01)	3.66	3.5
N-HS(RW)	0.813(0.054)	0.636	**0.58**(0.02)	0.82	122.8	0.738(0.049)	0.670	0.57(0.01)	3.70	8.1
BL	1.055(0.134)	0.455	0.53(0.11)	8.84	78.6	0.984(0.058)	0.255	0.57(0.18)	0.43	19.1
N-BL(RW)	0.902(0.065)	0.536	**0.58**(0.08)	5.74	124.2	0.967(0.061)	0.451	0.59(0.17)	0.09	15.4
SkG	0.939(0.063)	0.585	0.54(0.02)	0.00	1950.7	0.924(0.050)	0.587	0.53(0.03)	0.01	530.7
SpSL(MM)	1.022(0.069)	0.519	0.45(0.12)	1.05		1.282(0.078)	0.380	0.32(0.16)	2.77	
EMVS	1.283(0.073)	0.385	0.48(0.10)	0.00		1.327(0.083)	0.339	0.49(0.12)	0.00	
SSLasso	0.965(0.057)	0.587	0.56(0.03)	0.00		0.752(0.047)	0.672	**0.67**(0.02)	0.00	

	$(n=100, p=300)$					$(n=150, p=1000)$				
N-SpSL(MAP)	1.057(0.064)	0.538	0.51(0.05)	0.32		0.999(0.062)	0.554	0.55(0.02)	0.00	
N-BL(MAP)	0.786(0.053)	0.619	0.29(0.19)	19.63		0.727(0.052)	0.636	0.25(0.23)	35.48	
Lasso(CV)	0.782(0.051)	0.636	0.43(0.04)	10.81		0.717(0.047)	0.664	0.39(0.08)	16.52	
SCAD(CV)	1.027(0.070)	0.575	0.34(0.08)	9.80		1.000(0.065)	0.587	0.34(0.10)	13.69	
Lasso(EBIC)	1.186(0.078)	0.476	0.50(0.06)	0.04		1.165(0.076)	0.497	0.55(0.03)	0.04	
SCAD(EBIC)	1.183(0.079)	0.476	0.49(0.05)	0.06		1.178(0.074)	0.493	0.54(0.03)	0.05	

Table 4 Results for the real data sets. MSPE and MS stand for the mean squared prediction error (out-of-sample) and the model size (number of selected variables), respectively.

	Boston housing			Bardet-Biedl		
Method	MSPE	Cos(Angle)	MS	MSPE	Cos(Angle)	MS
SpSL-G(HCG)	25.246(0.914)	0.841	6.82	0.425(0.026)	0.697	2.64
N-SpSL-L(Exact)	25.288(0.903)	0.841	6.98	0.421(0.024)	0.701	2.28
SpSL-C(HCG)	25.252(0.907)	0.841	6.18	0.421(0.026)	0.689	2.36
N-SpSL-C(RW)	25.203(0.892)	0.841	6.08	0.452(0.038)	0.696	2.28
HS	25.479(0.927)	0.839	5.56	0.375(0.020)	0.697	8.56
N-HS(RW)	25.461(0.924)	0.840	5.60	0.378(0.020)	0.696	8.12
BL	25.448(0.938)	0.829	6.10	0.357(0.021)	0.642	80.97
N-BL(RW)	25.411(0.903)	0.829	6.10	0.364(0.015)	0.661	94.53
SkG	25.332(0.891)	0.841	8.00	0.766(0.047)	0.653	0.00
SpSL(MM)	27.341(1.115)	0.826	4.81	0.502(0.035)	0.669	5.63
EMVS	25.385(0.923)	0.840	6.00	0.697(0.040)	0.685	0.00
SSLasso	25.058(0.897)	0.842	6.00	0.491(0.038)	0.648	2.90
N-SpSL(MAP)	24.043(0.871)	0.848	6.90	**0.355**(0.017)	0.689	2.76
N-BL(MAP)	25.192(0.890)	0.842	6.59	0.432(0.029)	0.705	12.00
Lasso(CV)	25.196(0.886)	0.842	8.60	0.424(0.031)	**0.707**	22.59
SCAD(CV)	25.111(0.894)	0.842	7.31	0.491(0.037)	0.694	9.77
Lasso(BIC)	26.833(0.954)	0.833	7.09	1.176(0.047)	0.665	2.07
SCAD(BIC)	25.515(0.926)	0.839	6.34	1.157(0.052)	0.655	2.25