# Bayesian Generalized Horseshoe Estimation of Generalized Linear Models

Daniel F. Schmidt (✉)[1,2] and Enes Makalic[2]

[1] Faculty of Information Technology, Monash University, Clayton, Australia
daniel.schmidt@monash.edu
[2] Centre for Epidemiology and Biostatistics, The University of Melbourne, Australia
emakalic@unimelb.edu.au

**Abstract.** Bayesian global-local shrinkage estimation with the generalized horseshoe prior represents the state-of-the-art for Gaussian regression models. The extension to non-Gaussian data, such as binary or Student-$t$ regression, is usually done by exploiting a scale-mixture-of-normals approach. However, many standard distributions, such as the gamma and the Poisson, do not admit such a representation. We contribute two extensions to global-local shrinkage methodology. The first is an adaption of recent auxiliary gradient based-sampling schemes to the global-local shrinkage framework, which yields simple algorithms for sampling from generalized linear models. We also introduce two new samplers for the hyperparameters in the generalized horseshoe model, one based on an inverse-gamma mixture of inverse-gamma distributions, and the second a rejection sampler. Results show that these new samplers are highly competitive with the no U-turn sampler for small numbers of predictors, and potentially perform better for larger numbers of predictors. Results for hyperparameter sampling show our new inverse-gamma inverse-gamma based sampling scheme outperforms the standard sampler based on a gamma mixture of gamma distributions.

**Keywords:** Bayesian regression, Markov Chain Monte Carlo Sampling, Horseshoe Regression, Shrinkage

## 1 Introduction

The introduction of the horseshoe prior [5], and more generally the idea of global-local shrinkage hierarchies [16], has sparked a period of interest in heavy tailed prior distributions for coefficients in linear regression models. The Bayesian global-local shrinkage priors represent the current state-of-the-art for Gaussian linear regression models and encompass a large number of well known Bayesian shrinkage techniques, including the Bayesian ridge, the Bayesian lasso [13], the horseshoe prior, the horseshoe+ [2] and the Dirichlet-Laplace [4] prior. Let $\mathbf{y} = (y_1, \ldots, y_n)$ denote the vector of $n$ measurements of a target (dependent) variable of interest, $\bar{\mathbf{x}}_i = (x_{i,1}, \ldots, x_{i,p})$ denote the vector of of predictors (explanatory variables, covariates) associated with each target $y_i$, and let

$\mathbf{X} = (\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n)^{\mathrm{T}}$ denote the $n \times p$ matrix of explanatory variables. The global-local shrinkage (GLS) hierarchy for Gaussian linear models is given by:

$$\begin{aligned} \mathbf{y} \,|\, \boldsymbol{\beta}, \beta_0, \sigma^2 &\sim N(\mathbf{X}\boldsymbol{\beta} + \beta_0 \mathbf{1}_n, \sigma^2) \\ \beta_0 &\sim (1) d\beta_0 \\ \sigma^2 &\sim \sigma^{-2} d\sigma^2 \\ \beta_j \,|\, \lambda_j, \sigma, \tau &\sim N(0, \lambda_j^2 \tau^2 \sigma^2), \\ \lambda_j &\sim \pi(\lambda_j) d\lambda_j, \\ \tau &\sim C^+(0, 1) \end{aligned} \tag{1}$$

where $j = 1, \ldots, p$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ is the vector of model coefficients, $\beta_0$ is the intercept, $\sigma^2$ is the noise variance, $N(a, b)$ is the normal distribution with mean $a$ and variance $b$ and $C^+(0, c)$ denotes a half-Cauchy distribution with scale $c$. In hierarchy (1), the hyperparameters $\lambda_1, \ldots, \lambda_p$ are the *local shrinkage* parameters that induce shrinkage only on their corresponding coefficients; by selecting a specific prior distribution $\pi(\lambda_j)$ one can represent most standard Bayesian shrinkage procedures within this framework. The hyperparameter $\tau$ is the *global shrinkage* parameter that controls the overall level of shrinkage and ties the $p$ regression coefficients together; conditioning on $\sigma^2$ ensures that the shrinkage induced on the coefficients is not affected by scale changes of our data.

The joint posterior distribution of a GLS hierarchy is in general intractable, so it usual to instead explore the posterior distribution by sampling. A standard approach is a Gibbs sampling procedure [9] which repeatedly iterates:

1. sample the coefficients from $p(\boldsymbol{\beta} \,|\, \beta_0, \sigma^2, \boldsymbol{\lambda}, \tau, \mathbf{y})$;
2. sample the remaining model parameters from $p(\beta_0, \sigma^2 \,|\, \boldsymbol{\beta}, \boldsymbol{\lambda}, \tau, \mathbf{y})$;
3. sample the shrinkage hyperparameters from $p(\boldsymbol{\lambda}, \tau \,|\, \boldsymbol{\beta}, \sigma^2, \mathbf{y})$.

A strength of the GLS hierarchy is that in this Gibbs sampling framework the sampling algorithms for the hyperparameters are independent from the sampling algorithm for the coefficients. This means that as long as we have algorithms for sampling the coefficients given a normal prior distribution, and an algorithm for sampling the hyperparameters, we can mix and match choices of shrinkage priors with choices for likelihoods with no additional implementation details.

Building on this idea, the aim of this article is to explore two extensions to global-local shrinkage methodology: (i) we propose to adapt recent gradient-based sampling algorithms [20] to provide simple sampling procedures for a wide-range of non-Gaussian data, and (ii) we propose two new samplers for the local shrinkage hyperparameters $\lambda_j$ under the generalized horseshoe estimator.

### 1.1 Bayesian Generalized Linear Models

One of the great successes of linear models is the ease in which they may be extended to handle data that is not typically modelled using a normal distribution

(e.g., categorical or count data) via the framework of *generalised linear models* (GLMs) introduced by Nelder and Wedderburn [11]. The GLM framework begins by defining

$$\eta_i = \bar{\mathbf{x}}_i^{\mathrm{T}} \boldsymbol{\beta} + \beta_0$$

as the linear predictor; a GLM then models the conditional mean and variance of the target $y_i$ by a suitable transformation of this linear predictor, i.e.,

$$\begin{aligned} \mathbb{E}\left[y_i \,|\, \bar{\mathbf{x}}_i\right] &= f^{-1}(\eta_i) \equiv \mu_i, \\ \mathrm{Var}\left[y_i \,|\, \bar{\mathbf{x}}_i\right] &= \sigma^2 \, v(\eta_i), \end{aligned}$$

where $\sigma^2$ is now a dispersion parameter, and $f(\mu_i) = \eta_i$ is referred to as the link function, as it links the linear predictor $\eta_i$ to the conditional mean $\mu_i$. This approach allows $y_i$ to follow many standard distributions, and with careful choice of $f(\cdot)$ the resulting GLM retains much of the computational efficiency and statistical interpretability that characterises Gaussian linear models.

The usual fashion in which the global-local hierarchy (1) is extended to non-Gaussian data is through a scale mixture of normals (SMN) representation of the desired distribution. In particular, we rewrite the data model as

$$\begin{aligned} y_i \,|\, \boldsymbol{\beta}, \beta_0, \omega_j, \sigma^2 &\sim N(\bar{\mathbf{x}}_i^{\mathrm{T}} \boldsymbol{\beta} + \beta_0, \sigma^2 \omega_j^2), \\ \omega_j &\sim \pi(\omega_j) d\omega_j. \end{aligned}$$

In this approach, the data are modelled as arising from $n$ heteroskedastic normal distributions, with an additional latent scale variable $\omega_i$ associated with each data point $y_i$. The choice of the mixing density $\pi(\omega_j)$ determines the final distribution of $y_j$. The particular advantage of this data augmentation approach is that it preserves the conditional conjugacy between the likelihood and the normal prior distribution for $\beta_j$. Therefore, efficient sampling algorithms such as those of Rue [18](for $p < n$) and Bhattarcharya [3] (for $p > n$) can be employed in a Gibbs framework to generate posterior samples for the coefficients. The SMN approach has successfully been used to represent the Laplace, Student-$t$, logistic and negative binomial distribution [17, 10].

However, not all distributions utilised in standard generalized linear modelling have known or convenient SMN representations; examples include the Poisson, gamma, Weibull and inverse-Gaussian distributions. In such cases, one must usually resort to alternative sampling techniques. One of the earliest approaches was utilisation of adaptive rejection sampling to implement one-at-a-time sampling of the coefficients. Such an approach is potentially slow and can result in a chain that mixes poorly, particularly if the predictors are correlated. More recent sampling techniques that can be utilised to handle Bayesian generalized linear models include the Hamiltonian MCMC no U-turn sampler (NUTS)[8] implemented in the Stan tool; generalized elliptical slice sampling [12]; and the Metropolis adjusted preconditioned Crank-Nicolson (pCNL) Langevin-based approach [6].

### 1.2 Generalized Horseshoe Priors

A particular important prior is the so-called generalized horseshoe (GHS, also known as the generalized beta mixture of Gaussians and the inverse-gamma-gamma prior). The generalized horseshoe [1] places a beta prior distribution over the coefficient of shrinkage, i.e., $\lambda_j^2(1+\lambda_j^2)^{-1} \sim \text{Beta}(a,b)$. This induces the following distribution over $\lambda_j$:

$$\pi(\lambda_j \,|\, a, b) = \frac{2\lambda_j^{2a-1}(1+\lambda_j^2)^{-a-b}}{B(a,b)}. \tag{2}$$

The well known horseshoe arises if we set $a = b = 1/2$; in this case the beta distribution has a 'U'-shape from which the horseshoe prior obtains its distinctive name, and (2) reduces to the unit half-Cauchy. To gain an understanding of the effect that the hyperparameters $a$ and $b$ have on inference we can examine the corresponding marginal prior distribution over $\beta_j$:

$$\pi(\beta_j \,|\, a, b) = \int_0^\infty \phi(\beta/\lambda_j)/\lambda_j \pi(\lambda_j \,|\, a, b) d\lambda_j$$

where $\phi(\cdot)$ denotes the standard normal density function. Appealing to Proposition 1 and Thereoms 2 and 3 from [19], we have for $a < 1/2$

$$\pi(\beta_j \,|\, a, b) = O(|\beta|^{-1+2a})$$

as $|\beta| \to 0$, and for all $b > 0$

$$\pi(\beta_j \,|\, a, b) = O\left(|\beta|^{-1-2b}\right)$$

as $|\beta| \to \infty$. Therefore, the hyperparameter $a$ controls the degree to which prior probability is concentrated around $\beta = 0$; smaller values indicate an *a priori* belief in great underlying sparsity of the coefficients vector. The hyperparameter $b$ controls the rate at which the tail of the marginal prior decays; smaller values result in a slower decay, which indicates an *a priori* belief that some of the coefficients may be substantially greater in magnitude than others.

**Sampling generalized horseshoe hyperparameters** Most MCMC implementations of (a variant of) the generalized horseshoe are based on Gibbs sampling. For the particular case of the horseshoe (i.e., $a = b = 1/2$) there exist a number of approaches to sampling the conditional posterior $p(\lambda_j \,|\, \beta_j, \tau, \sigma^2)$. These include slice sampling [15], an inverse-gamma inverse-gamma scale mixture representation [10] and a gamma-gamma scale mixture representation [1]. Of these methods, only the gamma-gamma mixture currently handles the generalized horseshoe; it utilises the fact that if

$$x^2 \,|\, c \sim \text{Ga}(a, 1/c), \text{ and } c \sim \text{Ga}(b, 1)$$

then $x$ follows the distribution (2), where $\text{Ga}(a, c)$ denotes a gamma distribution with shape $a$ and scale $c$. Introducing a set of latent variables $\nu_1, \ldots, \nu_p$, this augmentation leads to the full conditionals

$$\lambda_j^2 \mid \cdots \sim \text{GIG}(a - 1/2, 2\nu_j, 2m_j) \text{ and } \nu_j \mid \cdots \sim \text{Ga}\left(a + b, (1 + \lambda_j^2)^{-1}\right), \quad (3)$$

where $m_j = \beta_j^2/(2\sigma^2\tau^2)$ and GIG denotes a generalized inverse Gaussian distribution. Implementation within a Gibbs framework therefore requires sampling from the GIG distribution, which is potentially troublesome. Algorithms for generating GIG random variates are not distributed by default in packages such as MATLAB and R, and the best implementations are slower than generating random variates from standard distributions such as the gamma.

### 1.3 Our contributions

In Section 2 of this paper, we adapt the recently proposed class of auxiliary gradient-based sampling algorithms [20] to the hierarchy (1). While these algorithms were designed for Gaussian process regression, they are perfectly positioned for application to GLMs and global-local shrinkage hierarchies. In Section 3 of this paper we present two new samplers for $\lambda_j$ in the case of the GHS. One is a generalization of the inverse-gamma inverse-gamma approach proposed in [10]; the other is a new rejection sampler that exploits the log-concavity of the conditional distribution of $\log \lambda_j$.

Results in Section 4 demonstrate that despite their simplicity, the new gradient based sampling algorithms are competitive with alternative non-specialized sampling algorithms in terms of effective samples per second, and can potentially outperform them. Experimental results also show that the new inverse-gamma inverse-gamma sampler for the generalized horseshoe leads to a Gibbs simpler that is frequently more efficient in terms of effective samples per second than (3), while remaining substantially simpler in terms of implementation.

## 2 Gradient-based samplers for Bayesian GLMs

We propose to utilise the recently developed auxiliary gradient-based sampling algorithms [20]. These algorithms work by first augmenting the target density with auxiliary random variables, and using this in conjunction with a first-order Taylor series expansion of the likelihood and a marginalisation step to build a Metropolis-Hastings proposal density that is both likelihood and prior informed. Specifically, they were designed to target densities of the form

$$p(\boldsymbol{\beta}) \propto \exp(f(\boldsymbol{\beta}); \beta_0, \sigma^2) N(\boldsymbol{\beta} \mid \mathbf{0}, \mathbf{C})$$

where $f(\boldsymbol{\beta}, \beta_0, \sigma^2)$ denotes the log-likelihood and $\mathbf{C}$ denotes the prior covariance matrix for the coefficients $\boldsymbol{\beta}$. The posterior distribution for the coefficients $\boldsymbol{\beta}$ of a generalized linear model with global-local shrinkage priors, conditional on the shrinkage hyperparameters $\boldsymbol{\lambda}$ and $\tau$, directly matches this class of problems. This

facilitates application of these auxilary gradient-based samplers within the usual Gibbs sampling framework. Furthermore, in the case of a GLM with a standard link function, the log-likelihood $f(\boldsymbol{\beta})$ is a convex function of the coefficients $\boldsymbol{\beta}$. As our starting point we consider the general pre-conditioned marginal gradient sampler (eq. (8) in [20]), which uses as a proposal

$$\bar{\boldsymbol{\beta}} \,|\, \boldsymbol{\beta} \sim N\left(\mathbf{A}\left(\bigtriangledown f(\boldsymbol{\beta}) + \left(\frac{2}{\delta}\right)\mathbf{S}^{-1}\boldsymbol{\beta}^{\mathrm{T}}\right), \left(\frac{2}{\delta}\right)\mathbf{A}\mathbf{S}^{-1}\mathbf{A} + \mathbf{A}\right) \tag{4}$$

where $\delta > 0$ is the MH step-size, $\mathbf{S}$ is the pre-conditioning matrix, $\bigtriangledown f(\boldsymbol{\beta}) = (\partial f(\boldsymbol{\beta}, \beta_0, \sigma^2)/\partial \boldsymbol{\beta})$ denotes the gradient, and

$$\mathbf{A} = \left(\mathbf{C}^{-1} + \left(\frac{2}{\delta}\right)\mathbf{S}^{-1}\right)^{-1}.$$

The step-size $\delta$ should be chosen such that $50\% - 60\%$ of samplers are accepted. We discuss a robust fully automatic method for doing this in Section 2.4.

### 2.1 Algorithm 1: mGrad-1

The first variant of the algorithm we will consider uses $\mathbf{S} = \mathbf{I}_p$. In the case of the global-local shrinkage hierarchy (1) the prior covariance matrix $\mathbf{C}$ is simply a diagonal matrix with entries

$$C_{j,j} = \tau^2\sigma^2\lambda_j^2, \quad j = 1, \ldots, p.$$

Combined with the choice $\mathbf{S} = \mathbf{I}_p$, the proposal (4) and the Metropolis-Hastings acceptance step dramatically simplify. We call this algorithm 'mGrad-1', as it uses only first order information. The mGrad-1 algorithm works as follows:

1. Generate proposals for coefficients using

$$\bar{\beta}_j \sim N\left(\frac{C_{j,j}(\delta\,[\bigtriangledown f(\boldsymbol{\beta})]_j + 2\beta_j)}{2C_{j,j} + \delta}, \frac{\delta\, C_{j,j}(4C_{j,j} + \delta)}{(2C_{j,j} + \delta)^2}\right)$$

2. Generate $u \sim U(0, 1)$, and accept the new proposal if

$$u < \exp\left(f(\bar{\boldsymbol{\beta}}, \beta_0, \sigma^2) - f(\boldsymbol{\beta}, \beta_0, \sigma^2) + h_1(\boldsymbol{\beta}, \bar{\boldsymbol{\beta}}) - h_1(\bar{\boldsymbol{\beta}}, \boldsymbol{\beta})\right),$$

$$h_1(\boldsymbol{\beta}, \bar{\boldsymbol{\beta}}) = \sum_{j=1}^{p}\left(\beta_j - \frac{C_{j,j}(4\bar{\beta}_j + \delta\,[\bigtriangledown f(\bar{\boldsymbol{\beta}})]_j)}{2(2C_{j,j} + \delta)}\right)\left(\frac{2C_{j,j} + \delta}{4C_{j,j} + \delta}\right)[\bigtriangledown f(\bar{\boldsymbol{\beta}})]_j$$

Due to the nature of the global-local shrinkage hierarchy, the mGrad-1 algorithm has a total computational complexity of only order $O(pn)$ for a GLM. This gives it potential for application to large $p$ regression problems.

### 2.2 Algorithm 2: mGrad-2

A potential problem with the mGrad-1 algorithm is that it only utilises first order likelihood information when generating the proposal; therefore, potential exists to improve mixing in the face of correlation between predictors by utilising second-order information. A second-order Taylor series expansion method is proposed in the supplementary material of [20], but was found to perform poorly, and is slow to implement as the proposal distribution depends on state-dependent second-order information. Given the nature of GLMs, we instead propose to set $\mathbf{S} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$, i.e., to use the correlation matrix of the predictors as the preconditioner. This keeps the covariance of the proposal independent of the state and allows for pre-computation of $\mathbf{S}^{-1}$. We call this the 'mGrad-2' algorithm, as it utilises second-order information. The computational effort of mGrad-2 is $O(p^3)$, which can be substantially higher than the computational complexity of mGrad-1.

### 2.3 Sampling the intercept

The mGrad-1 and mGrad-2 algorithms provide us with a way to sample the coefficients $\boldsymbol{\beta}$. We observed that using a single MH step for both $\beta_0$ and $\boldsymbol{\beta}$ led to reduced mixing, so we instead sample the intercept separately, using a simple proposal that does not depend on a step size parameter. To sample $\beta_0$ for a GLM we use the following procedure:

1. Generate a proposal from
$$\bar{\beta}_0 \,|\, \beta_0 \sim N\left(\beta, \frac{2.5}{H(\beta_0)}\right)$$
where $H(\beta_0)$ is the second-derivative of the negative log-likelihood with respect to $\beta_0$.
2. Generate $u \sim U(0,1)$ and accept $\bar{\beta}_0$ if $u < \exp(f(\boldsymbol{\beta}, \bar{\beta}_0, \sigma^2) - f(\boldsymbol{\beta}, \beta_0), \sigma^2)$.

We find this choice leads to acceptance rates in the range $50\% - 60\%$ for all experiments we considered.

### 2.4 Tuning the step size $\boldsymbol{\delta}$

Both the mGrad-1 and mGrad-2 algorithms are Metropolis-Hastings based approaches and require the selection of an appropriate step-size. For the base algorithm from which these methods are derived it is recommended that the optimal step-size $\delta$ should yield an acceptance rate in the range of $50\% - 60\%$. The step-size that achieves this rate will depend crucially on the particular problem, so it must be chosen adaptively. During the initial burn-in period we use the following procedure to estimate an appropriate value for $\delta$.

We divide the burn-in period into windows of size $w$; then, every $w$ iterations we record the step-size $\delta$ used in window $j$ as $\delta_j$, and the observed acceptance

rate for the window as $p_j$. We first find values of $\delta$ such that our algorithm never accepts samples, and accepts all samples; call our initial guesses at these two quantities $\delta_{\max}$ and $\delta_{\min}$, respectively. We continually increase $\delta$ by a factor of $k > 0$ every window, starting from $\delta = \delta_{\max}$, until we observe an acceptance rate of zero and update our value of $\delta_{\max}$. We then continually decrease $\delta$ by a factor of $k$ every window, starting from $\delta = \delta_{\min}$, until we observe an acceptance rate of one and update our value of $\delta_{\min}$.

Once this is done we set $\delta \leftarrow (\delta_{\min}\delta_{\max})^{1/2}$, and begin 'probing' to learn the relationship between $\delta$ and the acceptance probability. For every window $j$ thereafter, we fit a logistic regression of $(\log \delta_1, \ldots, \log \delta_j)$ to the acceptance probabilities $(p_1, \ldots, p_j)$; call the fitted slope $\hat{\alpha}_1$ and intercept $\hat{\alpha}_0$. We then update the step-size for the next window by first generating $u \sim U(0.45, 0.65)$, and then setting $\delta = d(u, \hat{\alpha}_0, \hat{\alpha}_1)$ where

$$d(u, \alpha_0, \alpha_1) = \exp\left[-\frac{1}{\alpha_1}\left(-\log\left(\frac{1}{1/u - 1}\right) + \alpha_0\right)\right].$$

solves the equation

$$\log\left(\frac{u}{1-u}\right) = \alpha_1 \log \delta + \alpha_0$$

for $\delta$. Once the burn-in phase is complete, we choose the final step-size as $\delta = d(0.55, \hat{\alpha}_0, \hat{\alpha}_1)$. In this way we are using the estimated relationship between the step-size and acceptance probability to select an appropriate value for $\delta$. In our implementation we took $w = 75$, $\delta_{\max} = 100$, $\delta_{\min} = 10^{-7}$ and $k = 10$, though our experiments show the algorithm is almost completely insensitive to the particular values chosen. In all cases we observed that a burn-in period of $5,000$ samples usually provided an estimate of $\delta$ that achieved an acceptance rate between 0.5 and 0.6 for the remaining samples.

### 2.5 Implementation details

Implementation of the mGrad-1 and mGrad-2 algorithms require only knowledge of the log-likelihood and the gradient of the log-likelihood. For convenience, these quantities are presented in Table 1 for a number of distributions frequently used in GLMs. Both algorithms require computation of the likelihood for the acceptance step. By careful implementation the number of computations can be reduced to one additional computation per sample being simulated.

While the computation of the likelihood is not required by the SMN technique, it is common to compute a diagnostic statistic such as the widely applicable information criterion (WAIC) from MCMC output, for which computation of the likelihood is required for every sample. In this case, our samplers effectively provide the likelihood information 'for free' which improves their competitiveness in comparison to SMN approaches.

| | Log-likelihood, $f(\boldsymbol{\beta}, \beta_0, \sigma^2)$ | $[\bigtriangledown f(\boldsymbol{\beta})]_j$ | $\sigma^2 v(\mu_i)$ |
|---|---|---|---|
| Normal | $-\frac{1}{2\sigma^2}\sum_{i=1}^{n} e_i^2$ | $\frac{1}{\sigma^2}\sum_{i=1}^{n} e_i X_{i,j}$ | $\sigma^2$ |
| Binomial | $\sum_{i=1}^{n} y_i \log \mu_i + (1-y_i)\log(1-\mu_i)$ | $\sum_{i=1}^{n} e_i X_{i,j}$ | $\mu_i(1-\mu_i)$ |
| Poisson | $\left[\sum_{i=1}^{n} y_i \eta_i - \mu_i\right]$ | $\sum_{i=1}^{n} e_i X_{i,j}$ | $\mu_i$ |
| Geometric | $\left[\sum_{i=1}^{n} \eta_i y_i - (y_i+1)\log(\mu_i+1)\right]$ | $\sum_{i=1}^{n}\left(y_i - \frac{\mu_i(y_i+1)}{\mu_i+1}\right) X_{i,j}$ | $\mu(\mu_i+1)$ |
| Gamma | $-\frac{1}{\kappa}\sum_{i=1}^{n}\left[\log \mu_i + \frac{y_i}{\mu_i}\right]$ | $\frac{1}{\kappa}\sum_{i=1}^{n}\left(\frac{y_i}{\mu_i} - 1\right) X_{i,j}$ | $\kappa\mu_i^2$ |
| Inverse-Gaussian | $-\frac{1}{2\xi}\sum_{i=1}^{n}\frac{e_i^2}{\mu_i^2 y_i}$ | $\frac{1}{\xi}\sum_{i=1}^{n}\left(\frac{e_i}{\mu_i^2 y_i}\right) X_{i,j}$ | $\xi\mu_i^3$ |

**Table 1.** Log-likelihoods (up to constants independent of $\boldsymbol{\beta}$) and gradients for commonly used target distributions. The quantity $\eta_i = \bar{\mathbf{x}}_i^{\mathrm{T}}\boldsymbol{\beta} + \beta_0$ denotes the linear predictor for sample $y_i$, and $e_i = y_i - \mu_i$. The normal distribution uses the identity link $\mu_i = \eta_i$; the binomial uses the logit link $\mu_i = (1+\exp(-\eta_i))^{-1}$; the remaining distributions use the log-link $\mu_i = \exp(\eta_i)$. All distributions are parameterised so that $\mathbb{E}[y_i] = \mu_i$. The final column identifies the dispersion parameter.

## 3 Two new samplers for the generalized horseshoe

In this section we discuss two new sampling schemes for the shrinkage hyperparameters in the generalized horseshoe hierarchy (1). More specifically, we develop two samplers to target the density

$$p(z \,|\, m, p, a, b) \propto z^{2a-p-1}(1+z^2)^{-a-b}e^{-m/z^2} \tag{5}$$

This density generalizes the conditional distributions for the shrinkage hyperparameters $\lambda_j$ and $\tau$ in the GHS hierarchy (1); for example, the conditional distribution for a local shrinkage hyperparameter $\lambda_j$ is

$$p(z = \lambda_j \,|\, \beta_j^2/(2\tau^2\sigma^2), 1, a, b)$$

and for the global shrinkage hyperparameter $\tau$ is

$$p\left(z = \tau \,|\, \left(\frac{1}{2\sigma^2}\right)\sum_{j=1}^{p}\frac{\beta_j^2}{\lambda_j^2}, p, a, b\right).$$

We develop two approaches to sample from (5). We provide an inverse gamma mixture of inverse gamma (IGIG) distributions as an alternative to the gamma-gamma (GG) sampler. We also detail a reasonable straightforward rejection sampler that exploits the log-concavity of the density (5) under the transformation $\xi = \log z$. In contrast to the the GG and IGIG samplers, the rejection sampler simulates uncorrelated random draws. It is also easily adapted to sample from a truncated form of (5), which is of potential interest in light of the results presented in [14].

### 3.1 Inverse Gamma-Inverse Gamma Sampler

The following proposition generalizes the inverse gamma-inverse gamma representation of the half-Cauchy density utilised in [10]. This allows us to build a Gibbs sampler for the generalized horseshoe estimator.

**Proposition 1.** *Let* $x^2 \,|\, \nu, b \sim \mathrm{IG}(b, 1/\nu)$ *and* $\nu \,|\, a \sim \mathrm{IG}(a, 1)$. *Then*

$$p(x) \propto x^{2a-1}(1+x^2)^{-a-b}.$$

The proof is a straightforward application of integration by substitution. Using Proposition 1, we can build a sampler for the density (5) in the case that $a > 0$, $b > 0$. Introduce the auxiliary variable $\nu$; the Gibbs sampler then iterates:

1. First sample
$$z^2 \sim \mathrm{IG}\left(\frac{p}{2} + b, m + \frac{1}{\nu}\right).$$
2. Then sample the auxilliary variable
$$\nu \sim \mathrm{IG}\left(a + b, 1 + \frac{1}{z^2}\right).$$

Marginally, the random variable $z$ will follow the distribution (5). In contrast to the gamma-gamma sampler discussed in Section 1.2, the inverse gamma-inverse gamma sampler only requires samples from inverse gamma distributions, rather than the substantially more complex generalised inverse Gaussian distribution needed by the gamma-gamma hierarchy. This makes implementation substantially more straightforward.

### 3.2 Rejection Sampling

The GG and IGIG samplers all have a one-hundred percent acceptance rate, but suffer from autocorrelation due to their reliance on auxiliary variables. An alternative to this approach is rejection sampling, in which we trade a reduced acceptance rate for the removal of autocorrelation in the samples. As a quick refresher, a rejection sampler for a target density $p(x)$ works by first drawing a sample from a proposal distribution $q(x)$, and then accepting this sample if $p(x)/q(x) > u$, where $u \sim U(0,1)$. The proposal distribution must satisfy

$q(x) \geq p(x)$ for all $x$ (i.e., the proposal must upper-bound the target density), and ideally, must be straightforward to generate samples from. The closer $q(x)$ is to $p(x)$, the higher the rate of acceptance.

An efficient rejection sampler for $\lambda$ can be devised by noting that if $\lambda$ follows the conditional distribution (5), then the probability density for the transformed variable $\xi = \log \lambda$ (i.e., we are sampling the logarithm of the hyperparameters) is

$$p(\xi \,|\, m, p, a, b) \propto e^{-e^{-2\xi}m} e^{-\xi(p-2a)} (1 + e^{2\xi})^{-a-b}. \tag{6}$$

It is straightforward to verify that the density (6) is log-concave, and that $-\log p(\xi \,|\, m, p, a, b) \asymp \xi$ as $\xi \to \infty$. We therefore use a proposal density built by sandwiching a uniform density between two appropriately chosen exponential distributions, as this is guaranteed to bound the density (6) from above [7]. The mode of the density (6) is given by

$$\xi' = \frac{1}{2}\left[\log\left(2(a+m) - p + \sqrt{8m(2b+p) + (p-2a-2m)^2}\right) - \log(4b+2p)\right].$$

We place the uniform density on the interval $(L, R)$ which is chosen such that $L < \xi' < R$, and then place the two exponential distributions on either side of the mode; to find the break-points $L$ and $R$ for the three components, first define

$$\begin{aligned} l(\xi) &= -\log p(\xi \,|\, m, p, a, b) \\ &= e^{-2\xi}m + (p-2a)\xi + (a+b)\log\left(1 + e^{2\xi}\right) \end{aligned} \tag{7}$$

and

$$g(\xi) = -2a + \frac{2(a+b)e^{2\xi}}{1+e^{2\xi}} - 2e^{-2\xi}m + p \tag{8}$$

as the derivative of $l(\xi)$. We then set

$$\xi_L = \xi' - \frac{0.85}{\sqrt{p}}, \quad \xi_R = \xi' + \frac{1.3}{\sqrt{p}}.$$

These are the points that will be used to build the two exponential components of our proposal density; the break-points for our proposal density are then given by

$$L = \xi_L - \frac{l(\xi_L) - l(\xi')}{g(\xi_L)}, \quad R = \xi_R - \frac{l(\xi_R) - l(\xi')}{g(\xi_R)}$$

The proposal density is then given by

$$q(\xi) \propto \begin{cases} e^{-g(\xi_L)(\xi - L)} & \text{for } -\infty < \xi < L \\ 1 & \text{for } L < \xi < R \\ e^{-g(\xi_R)(\xi - R)} & \text{for } R < \xi < \infty \end{cases}.$$

Sampling from $q(\xi)$ is straightforward, as the normalizing constants for each of the components is straightforward: $K_L = -1/g(\xi_L)$, $K_C = R - L$, and $K_R = 1/g(\xi_R)$, where $K_L$, $K_C$ and $K_R$ denote the normalizing terms for the left, central and right hand pieces respectively, and set $K = K_L + K_C + K_R$. The algorithm is as follows.

1. First, sample
$$u_1 \sim U(0,1),\ u_2 \sim U(0,1),\ u_3 \sim U(0,1).$$
2. Next, check $u_1$:
   (a) If $u_1 \in (0,\ K_L/K)$ then
   $$x \leftarrow -\frac{\log(1-u_2)}{g(\xi_L)} + L, \quad q \leftarrow l(\xi_L) + g(\xi_L)(x - \xi_L)$$
   (b) If $u_1 \in (K_L/K,\ (K_L + K_C)/K)$ then
   $$x \leftarrow (R-L)u_2 + L, \quad q \leftarrow l(\xi')$$
   (c) If $u_1 \in ((K_L + K_C)/K,\ 1)$ then
   $$x \leftarrow -\frac{\log(1-u_2)}{g(\xi_R)} + R, \quad q \leftarrow l(\xi_R) + g(\xi_R)(x - \xi_R)$$
3. Determine if we accept $x$; check if
   $$\log u_3 < q - l(x).$$
   If so accept $x$; otherwise, reject $x$ and return to Step 1.

The accepted sample $x$ can be transformed back to the original space using $z = e^x$.

## 4 Experimental results

We undertook several simulation experiments to assess the comparitive performance of the new sampler algorithms: mGrad-1, mGrad-2 and the new hyperparameter samplers. In all experiments we used the effective sample size per second (ESS/$s$) as a measure of performance of the samples. The ESS measures how much correlation is present in a chain of MCMC samples; the higher the correlation, the less information is contributed by each sample.

In all simulated examples we used the following experimental procedure. For a given sample size $n$ and number of predictors $p$, we generated a design matrix from a multivariate normal distribution with covariance between predictors given by $\mathrm{Cov}(X_i, X_j) = \rho^{|i-j|}$. Then, we randomly selected 15 predictors to be associated, and generated their coefficients from a Student-$t$ distribution with a degrees-of-freedom equal to ten. We then rescaled the coefficients so that the signal-to-noise ratio of the regression model was equal to three for the Poisson models and 1.5 for the binomial models; the intercept was fixed at $\beta_0 = 1$ for Poisson models and $\beta_0 = 0$ for binomial models. Finally, we generated $n = 200$ data points from this model. These choices produced models with a realistic, sparse mix of stronger and weaker effects, and which were not (near) linearly seperable in the case of binomial regression. All tests were conducted on a Microsoft Surface Pro 2016 laptop. Additional experiments were performed but are not included in this article due to space constraints.[3]

[3] Available at `https://dschmidt.org`

| Prior | Sampler | $p = 50$ | $p = 250$ | $p = 500$ |
|---|---|---|---|---|
| $(a = 1/2, b = 1/2)$ | Rejection | $(\mathbf{2147}, \mathbf{7329}, 11933)$ | $(48, 661, 1302)$ | $(5.2, 286, 540)$ |
| | IGIG | $(2044, 8028, \mathbf{14111})$ | $(\mathbf{49}, \mathbf{664}, \mathbf{1397})$ | $(\mathbf{5.3}, \mathbf{299}, \mathbf{586})$ |
| | GG | $(1558, 6098, 11016)$ | $(29, 500, 1207)$ | $(3.1, 221, 538)$ |
| $(a = 1/4, b = 1/2)$ | Rejection | $(\mathbf{1766}, \mathbf{6709}, \mathbf{11700})$ | $(\mathbf{45}, \mathbf{617}, \mathbf{1314})$ | $(\mathbf{5.1}, \mathbf{283}, 554)$ |
| | IGIG | $(966, 4399, 10526)$ | $(31, 525, 1277)$ | $(4.7, 270, \mathbf{572})$ |
| | GG | $(1245, 4995, 9063)$ | $(29, 483, 1121)$ | $(3.4, 224, 509)$ |

**Table 2.** (minimum, median, maximum) effective samples per second for three generalized horseshoe local shrinkage hyperparameter samplers: a rejection sampler, the inverse-gamma inverse-gamma (IGIG) sampler and the gamma-gamma (GG) sampler. The quantities $a$ and $b$ are the concentration and tail hyperparameters for the generalized horseshoe prior.

### 4.1 Comparison of GHS hyperparameter samplers

We tested the performance of the three GHS local hyperparameter samplers: the gamma-gamma (GG) sampler (Section 1.2, [1]), the inverse gamma-inverse gamma (IGIG) sampler (Section 3.1) and the rejection sampler (Section 3.2). We tested their performance on a Gaussian linear model with $p = 50$, $p = 250$ and $p = 500$ predictors generated as per the procedure in Section 4, using a correlation of $\rho = 0.9$. The samplers for the coefficients was the usual conditionally conjugate multivariate Gaussian. We tested two prior settings: $(a = 1/2, b = 1/2)$, i.e., the regular horseshoe prior, and $(a = 1/4, b = 1/2)$, which concentrates more prior probability mass around the origin. For a fair comparison, we implemented the generalized inverse Gaussian sampler and the rejection sampler in C. The IGIG sampler was implemented in pure MATLAB.

For each experiment we ran the chains for $10^4$ burnin iterations, and then collected $2 \times 10^4$ samples. The results are shown in Table 1. Overall, the rejection sampler performed the best, but the IGIG sampler was competitive with, or superior to, the rejection sampler for all but the case of $p = 50$ and $a = 1/4$, with both being largely superior to the GG sampler. The performance of the IGIG sampler, coupled with its simple implementation, recommend it as an excellent choice of sampler for generalized horseshoe hierarchies.

### 4.2 Comparison of samplers for coefficients

We tested the performance of our samplers for two distributions: the Poisson, for which a scale mixture of normals (SMN) sampler is not known, and logistic (binomial) regression for which an SMN sampler exists [17]. For both models we compared the mGrad-1 and mGrad-2 sampling algorithms presented in Section 2 against the NUTS sampler (using the RStan `stan_glm()` function). For Poisson regression we also tested against the generalized elliptical slice sampler

| Distribution | Correlation | Sampler | $p = 50$ | $p = 100$ | $p = 250$ |
|---|---|---|---|---|---|
| Poisson | $\rho = 0.5$ | mGrad-1 | $(108, 270, 615)$ | $(\mathbf{42}, \mathbf{398}, \mathbf{776})$ | $(\mathbf{55}, \mathbf{358}, \mathbf{777})$ |
| | | mGrad-2 | $(\mathbf{127}, \mathbf{351}, \mathbf{673})$ | $(18, 136, 235)$ | $(4.8, 27, 49)$ |
| | | pCNL | $(14, 36, 108)$ | $(3.8, 18, 69)$ | $(3.7, 17, 75)$ |
| | | NUTS | $(26, 35, 36)$ | $(16, 23, 24)$ | $(9.5, 14, 14)$ |
| | | GESS | $(4.3, 13, 25)$ | $(0.6, 3.3, 7.8)$ | $(0.1, 0.7, 1.9)$ |
| | $\rho = 0.9$ | mGrad-1 | $(12, 56, 165)$ | $(7.1, 69, 223)$ | $(2.8, \mathbf{103}, \mathbf{339})$ |
| | | mGrad-2 | $(\mathbf{157}, \mathbf{489}, \mathbf{806})$ | $(\mathbf{25}, \mathbf{157}, \mathbf{305})$ | $(1.6, 30, 59)$ |
| | | pCNL | $(6.0, 18, 56)$ | $(3.3, 14, 39)$ | $(1.9, 12, 39)$ |
| | | NUTS | $(29, 40, 41)$ | $(20, 28, 28)$ | $(\mathbf{7.8}, 12, 12)$ |
| | | GESS | $(4.0, 11, 21)$ | $(0.6, 3.3, 7.9)$ | $(0.1, 0.8, 2.2)$ |
| Binomial | $\rho = 0.5$ | mGrad-1 | $(46, 315, 694)$ | $(12, 165, 425)$ | $(8.0, \mathbf{269}, \mathbf{656})$ |
| | | mGrad-2 | $(52, 352, 747)$ | $(4.6, 91, 208)$ | $(0.6, 24, 54)$ |
| | | SMN | $(\mathbf{179}, \mathbf{772}, \mathbf{1936})$ | $(\mathbf{15}, \mathbf{260}, \mathbf{894})$ | $(3.1, 201, 486)$ |
| | | NUTS | $(56, 73, 76)$ | $(39, 54, 56)$ | $(\mathbf{17}, 26, 27)$ |
| | $\rho = 0.9$ | mGrad-1 | $(4.6, 33, 99)$ | $(3.3, 49, 161)$ | $(4.1, 76, 312)$ |
| | | mGrad-2 | $(25, 317, 705)$ | $(11, 135, 298)$ | $(1.2, 25, 61)$ |
| | | SMN | $(\mathbf{96}, \mathbf{721}, \mathbf{1775})$ | $(\mathbf{49}, \mathbf{483}, \mathbf{1159})$ | $(5.1, \mathbf{161}, \mathbf{478})$ |
| | | NUTS | $(32, 42, 44)$ | $(34, 47, 48)$ | $(\mathbf{17}, 26, 27)$ |

**Table 3.** (minimum, median, maximum) effective samples per second for various sampling algorithms. mGrad-1 and mGrad-2 refer to the two gradient-based sampling algorithms developed in this article, pCNL is the pre-conditioned Crank Nicholson sampler, NUTS is the no U-turn sampler and GESS is the generalized elliptical slice sampler.

(GESS) and pCNL algorithm; however, as both of these were dominated by mGrad-1 we did not test them for binomial regression. For logistic regression we also compared against the optimised scale mixture of normals (SMN) sampler implemented in the `bayesreg` package for MATLAB. We used the IGIG sampler for the horseshoe hierarchy for mGrad-1, mGrad-2, GESS, SMN and pCNL.

We tested the samplers for two settings of correlation $\rho = \{0.5, 0.9\}$, and generated a different model for each combination of $p = \{50, 100, 250\}$ and $\rho$. To make the comparisons as favourable for NUTS as possible we compute ESS/$s$ based only on the sampling times, and ignore warmup. We note that the mGrad algorithms require substantially less warmup time for tuning than NUTS. For NUTS we ran the chains for $10^3$ warmup samples and then collected the following $10^3$ samples. There were no convergence issues. For the other samplers we ran the chains for $10^4$ burnin iterations and the collected $2 \times 10^4$ samples. For each

test and each sampler we produced 10 chains and averaged the ESS/$s$ scores across the chains. The results are shown in Table 3.

In all cases the NUTS sampler exhibited an interesting property: the spread of ESS/$s$ values was small, with the minimum ESS/s being close to the maximum ESS/$s$. For Poisson regression the NUTS sampler had higher minimum ESS/$s$ than mGrad-1 when $\rho = 0.9$. In the case of Poisson regression, the mGrad-1 algorithm is highly competitive with the NUTS sampler, even for smaller $p$, and is uniformly superior for $\rho = 0.5$. The mGrad-2 algorithm exhibits superior performance to mGrad-1 for smaller $p$ and higher correlation $\rho$, but has poorer performance for $p = 250$ as the expensive matrix inversions outweigh the improvement in mixing. The pCNL and GESS algorithms performed uniformly worse than mGrad-1.

For logistic regression, the NUTS algorithm performed substantially better than for Poisson regression. The SMN sampler generally achieved the highest median and maximum ESS/$s$ scores, while the NUTS sampler uniformly achieved the higher minimum ESS/$s$ than mGrad-1. The mGrad-1 algorithm is largely inferior to the SMN sampler, but generally achieved higher median and maximum ESS/$s$ than the NUTS sampler. The mGrad-2 algorithm is uniformly worse than SMN in the setting of logistic regression, which is unsurprising as its base time complexity is similar to the SMN approach. We note that due to a different model being used for each combination of $p$ and $\rho$, the ESS/$s$ scores do not necessarily decrease as $p$ increases as the performance of all the samplers can vary depending on the structure of the underlying model.

**Additional Test for $p = 1,000$.** We performed an additional experiment for a much larger design matrix of $p = 1,000$ predictors with $\rho = 0.9$ and 50 non-zero coefficients for Poisson regression. We considered only the mGrad-1 and NUTS sampler; the NUTS sampler achieved a maximum ESS/$s$ of 7.8, while the mGrad-1 algorithm achieved a minimum/median/maximum of $\approx (0.8, 30, 143)$, which suggests that the simplicity of the algorithm potentially allows it to remain competitive with NUTS even for large $p$.

**Sensitivity to Model Structure**. We also performed an additional experiment to examine the sensitivity of mGrad-1 and NUTS to model structure. We generated the same design matrix and coefficients as used in the experiments for Poisson regression with $\rho = 0.5$, $p = 100$ but rescaled the coefficients to have a signal-to-noise ratio (SNR) of 9. The NUTS sampler achieved a maximum ESS of $\approx 8$ while the mGrad-1 sampler achieved a minimum/median/maximum of $\approx (20, 120, 245)$. In both cases this is roughly a three-fold reduction in comparison to the results obtained when the SNR was 3 (from Table 3). The sensitivity of NUTS is primarily driven by increased sampling time rather than changes in raw ESS, while for mGrad-1 the sampling time is unaffected but the increased correlation in the chains reduces the overall ESS/$s$.

## 5 Summary

In comparison to NUTS and SMN, the mGrad-1 algorithm is substantially easier to implement, requiring only knowledge of likelihood and gradient information. The entire algorithm, including the tuning can be implemented in around 50 lines of MATLAB code. This simplicity, coupled with the competitive performance of the mGrad-1 algorithm, demonstrates that it is a very useful addition to the suite of sampling procedures avaiable for Bayesian regression. A similar conclusion can be drawn regarding the new inverse gamma-inverse gamma sampler for the generalized horseshoe hyperparameters: in terms of performance it is roughly equivalent to the rejection sampler, and largely superior to the standard gamma-gamma sampler, while being substantially simple to implement than both. We therefore recommend this sampler to researchers looking to implement horseshoe and generalized horseshoe hierarchies for new models. The mGrad-1 sampler and SMN sampler for generalized linear generalized horseshoe regression models are both implemented in the `bayesreg`[4] Bayesian regression package.

## References

1. Armagan, A., Dunson, D.B., Clyde, M.: Generalized beta mixtures of Gaussians. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 24. 523–531 (2011)
2. Bhadra, A., Datta, J., Polson, N.G., Willard, B.: The horseshoe+ estimator of ultra-sparse signals (2016), arXiv:1502.00560
3. Bhattacharya, A., Chakraborty, A., Mallick, B.K.: Fast sampling with Gaussian scale-mixture priors in high-dimensional regression. Biometrika 103(4), 985–991 (2016), arXiv:1506.04778
4. Bhattacharya, A., Pati, D., Pillai, N.S., Dunson, D.B.: Dirichlet—Laplace priors for optimal shrinkage. Journal of the American Statistical Association 110, 1479–1490 (2015)
5. Carvalho, C.M., Polson, N.G., Scott, J.G.: The horseshoe estimator for sparse signals. Biometrika 97(2), 465–480 (2010)
6. Cotter, S., Roberts, G., Stuart, A., White, D.: MCMC methods for functions: modifying old algorithms to make them faster. Statistical Science 28, 424–446 (2014)
7. Gilks, W.R., Wild, P.: Adaptive rejection sampling for Gibbs sampling. Journal of the Royal Statistical Society. Series C (Applied Statistics) 41(2), 337–348 (1992)
8. Hoffman, M.D., Gelman, A.: The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. Journal of Machine Learning Research 15, 1351–1381 (2014)
9. Makalic, E., Schmidt, D.F.: High-dimensional Bayesian regularised regression with the bayesreg package (2016), arXiv:1611.06649
10. Makalic, E., Schmidt, D.F.: A simple sampler for the horseshoe estimator. IEEE Signal Processing Letters 23(1), 179–182 (2016)
11. Nelder, J.A., Wedderburn, R.W.M.: Generalized linear models. Journal of the Royal Statistical Society. Series A (General) 135(3), 370–384 (1972)

[4] Available at `https://au.mathworks.com/matlabcentral/fileexchange/60823`

12. Nishihara, R., Murray, I., Adams, R.P.: Parallel MCMC with generalized elliptical slice sampling. Journal of Machine Learning Research 15, 2087–2112 (2014)
13. Park, T., Casella, G.: The Bayesian lasso. Journal of the American Statistical Association 103(482), 681–686 (June 2008)
14. van der Pas, S., Szabó, B., van der Vaart, A.: Adaptive posterior contraction rates for the horseshoe (2017), arXiv:1702.03698v1
15. Polson, N.G., Scott, J.G., Windle, J.: The Bayesian bridge. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76(4) (2014)
16. Polson, N.G., Scott, J.G.: Shrink globally, act locally: Sparse Bayesian regularization and prediction. In: Bayesian Statistics. vol. 9 (2010)
17. Polson, N.G., Scott, J.G., Windle, J.: Bayesian inference for logistic models using Pólya-gamma latent variables 108(504), 1339–1349 (2013)
18. Rue, H.: Fast sampling of Gaussian markov random fields. Journal of the Royal Statistical Society (Series B) 63(2), 325–338 (2001)
19. Schmidt, D.F., Makalic, E.: Adaptive Bayesian shrinkage estimation using log-scale shrinkage priors (2017), https://arxiv.org/abs/1801.02321
20. Titsias, M.K., Papaspiliopoulos, O.: Auxiliary gradient-based sampling algorithms. Journal of the Royal Statistical Society (Series B) 80(4), 749–767 (2018)