Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

EXACT POSTERIOR SIMULATION FROM THE LINEAR LASSO REGRESSION

Zdravko I. Botev

School of Mathematics and Statistics
The University of New South Wales
Sydney, NSW 2052, AUSTRALIA

Yi-Lung Chen

School of Mathematics and Statistics
The University of New South Wales
Sydney, NSW 2052, AUSTRALIA

Pierre L'Ecuyer

DIRO, GERAD, and CIRRELT
Pavillon Aisenstadt, Université de Montréal
Montréal (Québec), CANADA

Shev MacNamara

School of Mathematical and Physical Sciences
University of Technology Sydney
Sydney, NSW 2007, AUSTRALIA

Dirk P. Kroese

School of Mathematics and Physics
The University of Queensland
Brisbane, QLD 4072, AUSTRALIA

ABSTRACT

The current popular method for approximate simulation from the posterior distribution of the linear Bayesian LASSO is a Gibbs sampler. It is well-known that the output analysis of an MCMC sampler is difficult due to the complex dependence amongst the states of the underlying Markov chain. Practitioners can usually only assess the convergence of MCMC samplers using heuristics. In this paper we construct a method that yields an independent and identically distributed (iid) draws from the LASSO posterior. The advantage of such exact sampling over the MCMC sampling is that there are no difficulties with the output analysis of the exact sampler, because all the simulated states are independent. The proposed sampler works well when the dimension of the parameter space is not too large, and when it is too large to permit exact sampling, the proposed method can still be used to construct an approximate MCMC sampler.

1 INTRODUCTION

Suppose that we are given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ of p predictors and a response vector $\boldsymbol{y} \in \mathbb{R}^n$. Then, the Bayesian LASSO inference for a linear model with unknown parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\sigma > 0$ requires simulation from the posterior pdf:

$$\pi(\boldsymbol{\beta}, \sigma \,|\, \boldsymbol{y}, \lambda) \propto \underbrace{\frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\|\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2}{2\sigma^2}\right)}_{\text{likelihood of } \boldsymbol{\beta} \,|\, (\sigma,\boldsymbol{y})} \underbrace{\frac{\lambda^p}{(2\sigma)^p} \exp\left(-\frac{\lambda\|\boldsymbol{\beta}\|_1}{\sigma}\right)}_{\text{prior of } \boldsymbol{\beta} \,|\, (\sigma,\lambda)} \overbrace{\sigma^{-2}}^{\text{prior of } \sigma}, \tag{1}$$

where $\|\cdot\|_p$ is the L_p-norm, and $\lambda \geq 0$ is the LASSO parameter which controls the level of shrinkage in the estimator. With no shrinkage, that is $\lambda = 0$, the mode of the posterior occurs at the *ordinary least squares* estimator $\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}\|_2$ and $\hat{\sigma} = s/\sqrt{n}$, where $s^2 := \|\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\|_2^2$ is the sum of squared residuals.

Currently, the available methods for simulation from (1) are the Gibbs samplers from Park and Casella (2008) and from Polson et al. (2014). These methods provide only an approximate draw from the posterior, with an accuracy that is notoriously difficult to quantify rigorously. Park and Casella (2008) and Polson et al. (2014) address this issue with (heuristic) graphical convergence diagnostics.

In this paper, we show that when $p < 100$ (that is, not too large), it is sometimes possible to construct a rejection sampler for exact simulation from (1). A key advantage of exact simulation over the approximate Gibbs sampler is that the difficult issues of bias and dependence in the output analysis disappear. This is the *raison d'étre* for the field of *perfect sampling* (Propp and Wilson 1996). The lack of bias (due to exact simulation) is especially advantageous when all the simulations are run in a parallel computing environment using many processors (Glynn 2016).

Our rejection algorithm for simulation from the target pdf (1) uses as proposal a sequential importance sampling density, which relies on computing the QL decomposition of the matrix $\mathbf{X}$. This sequential importance density depends on a number of parameters, which have to be well-tuned to achieve a good acceptance rate. We show how this tuning can be accomplished by solving a convex optimization program with a unique global solution.

The numerical results suggest that exact sampling is possible for roughly $p < 100$ (it becomes less efficient as p increases). Ultimately, for very large parameter spaces (large p), the only available option for simulation from (1) remains MCMC sampling. We argue that even in this situation our sequential sampling proposal is valuable, because it can be used as a proposal in an independence random-walk MCMC sampler.

The rest of the paper is structured as follows. Section 2 describes our acceptance-rejection approach to simulating from the Bayesian posterior. We briefly explain how the sequential importance sampling density can be deployed as a proposal density in an MCMC sampler. Then, in section 3 we provide two numerical examples based on widely-used datasets (Efron et al. 2004), and finally we draw conclusions and suggest future directions for research.

2 EXACT SIMULATION FROM THE POSTERIOR DISTRIBUTION

Before presenting the sequential sampling algorithm in detail, we first explain how one may go about constructing a simple rejection sampler (Kroese et al. 2011, Chapter 3) for the target (1).

2.1 Naive Rejection Sampler

Using the Pythagorean identity $\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = s^2 + \|\mathbf{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\|_2^2$, the posterior (1) is proportional to:

$$\tilde{\pi}(\boldsymbol{\beta}, \sigma) = \frac{\sqrt{2}}{s} g_{(n+1)/2}\left(\frac{\sigma\sqrt{2}}{s}\right) \times \frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left(-\frac{\|\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\|_2^2}{2\sigma^2}\right) \times \frac{\lambda^p}{2^p} \exp\left(-\frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right),$$

where $g_\alpha(\sigma)$ is the density

$$g_\alpha(\sigma) := \frac{\sigma^{-2\alpha-1}}{\Gamma(\alpha)} \exp\left(-\frac{1}{\sigma^2}\right)$$

with $\alpha = (n+1)/2$ degrees of freedom. Note that if X is Gamma distributed with shape parameter α and scale unity, then $1/\sqrt{X}$ has density $g_\alpha(x)$. When $\mathbf{X}$ is full rank, a natural proposal is to simulate σ from the density $\frac{\sqrt{2}}{s} g_{(n+1)/2}\left(\frac{\sigma\sqrt{2}}{s}\right)$. Then, given σ, we simulate $\boldsymbol{\beta}$ from the multivariate Gaussian density $\phi_\Sigma(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})$ with mean $\hat{\boldsymbol{\beta}}$ and covariance matrix $\Sigma = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$. With such simulated $(\boldsymbol{\beta}, \sigma)$, the (likelihood) ratio of $\tilde{\pi}(\boldsymbol{\beta}, \sigma)$ to the joint density $\frac{\sqrt{2}}{s} g_{(n+1)/2}\left(\frac{\sigma\sqrt{2}}{s}\right) \times \phi_\Sigma(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \times \frac{\lambda^p}{2^p} \exp\left(-\frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right)$ yields

$$\frac{\lambda^p}{2^p} \exp\left(-\frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right),$$

which is bounded from above by $\frac{\lambda^p}{2^p} =: c$. Hence, we immediately have the following rejection algorithm.

Algorithm 1 : Naive rejection sampler for $\pi(\boldsymbol{\beta}, \sigma \,\big|\, \boldsymbol{y}, \lambda)$.

Require: λ; least squares estimate $\hat{\boldsymbol{\beta}}$; sum of squared residuals s^2
  repeat
    Simulate $U \sim \mathsf{U}(0,1)$ and $\sigma \sim \mathsf{Gamma}(\alpha, 1)$, independently.
    $\sigma \leftarrow s/\sqrt{2\sigma}$
    $\boldsymbol{\beta} \sim \mathsf{Normal}(\hat{\boldsymbol{\beta}}; \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$
  until $U \leq \exp\left(-\frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right)$
  return $(\boldsymbol{\beta}, \sigma)$, a draw from the posterior.

This rejection algorithm will only be useful if the probability of acceptance

$$\mathbb{P}\Big[U \leq \exp\left(-\tfrac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right)\Big] = \tfrac{1}{c}\iint \tilde{\pi}(\boldsymbol{\beta}, \sigma)\mathrm{d}\boldsymbol{\beta}\mathrm{d}\sigma \tag{2}$$

is high. Unfortunately, in practice the marginal likelihood $\iint \tilde{\pi}(\boldsymbol{\beta}, \sigma)\mathrm{d}\boldsymbol{\beta}\mathrm{d}\sigma$ is much smaller than the enveloping constant c and as a result the probability of acceptance is too low to make the naive rejection Algorithm 1 practicable. For example, for the diabetes example in the numerical Section 3, the acceptance probability is less than 10^{-7}. Since Algorithm 1 is too simplistic to be efficient, we next propose an alternative with higher acceptance probability. The proposal has the additional advantage that it does not require a full rank data matrix $\mathbf{X}$.

2.2 Sequential Monte Carlo Sampler

Before giving the details of the much more elaborate sequential sampling, we need to introduce the following notation. Let

$$\mathrm{Lap}_\mu(z) = \frac{1}{2}\exp\left(-|z| + \mu z + \ln(1-\mu^2)\right), \qquad |\mu| < 1$$

be the exponentially tilted version of the Laplace density on $\mathbb{R}$ with tilting parameter μ. Appendix 5.1 gives a simple algorithm for simulating from Lap_μ. Further, let chi_ν be the density of the χ_ν distribution:

$$\mathrm{chi}_\nu(r) = \frac{\exp(-\frac{r^2}{2} + (\nu-1)\ln r)}{2^{\nu/2-1}\Gamma(\nu/2)}, \quad r > 0$$

Note that if $X \sim \chi_\nu$, then $\sqrt{X} \sim \chi^2_\nu$ (the chi-squared distribution). Lastly, if ϕ is the pdf of the standard normal and $\overline{\Phi}$ is its complementary cdf, we define the normal-Laplace density as

$$\mathrm{nl}(z; \lambda, \alpha) = \phi(z)\exp\Big(-\lambda|z-\alpha| - \xi(\lambda, \alpha)\Big),$$

where we have the normalizing constant:

$$\xi(\lambda, \alpha) = -\frac{\alpha^2 + \ln(2\pi)}{2} + \ln\Big[\frac{\overline{\Phi}(\lambda+\alpha)}{\phi(\lambda+\alpha)} + \frac{\overline{\Phi}(\lambda-\alpha)}{\phi(\lambda-\alpha)}\Big]$$

Fast simulation from this density is feasible and described in Appendix 5.2.

Recall that the QL decomposition of a matrix $\mathbf{X} \in \mathbb{R}^n \times \mathbb{R}^p$ is $\mathbf{X} = \mathbf{QL}$, where $\mathbf{Q} \in \mathbb{R}^n \times \mathbb{R}^p$ is a matrix of n orthonormal (column) vectors and $\mathbf{L} \in \mathbb{R}^p \times \mathbb{R}^p$ is a lower triangular matrix with elements $\{l_{ij}\}$, whose rank equals the column rank of $\mathbf{X}$. The QL factorization is just another version of the more common QR matrix factorization (Golub and Van Loan 2012).

Since the naive rejection sampler in the previous section is not practicable, we now describe a more complex rejection sampler to simulate from π, which can be written as:

$$\pi(\boldsymbol{\beta}, \sigma \,\big|\, \boldsymbol{y}, \lambda) \propto \phi_{\sigma^2\mathbf{I}}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})\lambda^d \exp\left(-(2+d)\ln\sigma - \lambda\|\boldsymbol{\beta}\|_1/\sigma\right) \tag{3}$$

Given the QL decomposition of the data matrix $\mathbf{X}$, we make the change of variables from $(\boldsymbol{\beta}, \sigma)$ to $(\boldsymbol{z}, r)$:

$$r = s/\sigma, \quad \boldsymbol{z} = \boldsymbol{\beta}/\sigma$$

Setting $\boldsymbol{\gamma} := \mathbf{L}\hat{\boldsymbol{\beta}}/s$, this change of variables in (3) yields the joint density in $(\boldsymbol{z}, r)$:

$$\pi(\boldsymbol{z}, r \,|\, \boldsymbol{y}, \lambda) \propto f(\boldsymbol{z}, r) := \lambda^d \mathrm{chi}_{n+1}(r) \exp\left(-\tfrac{1}{2} \textstyle\sum_i \left(l_{ii} z_i - r\gamma_i + \sum_{j<i} l_{ij} z_j \right)^2 - \lambda \sum_j |z_j| \right) \quad (4)$$

The idea now is to exploit the lower triangular structure of $\mathbf{L}$ in order to sample the vector $(\boldsymbol{Z}, R)$ sequentially as follows:

$$R \longrightarrow (Z_1 \,|\, R) \longrightarrow (Z_2 \,|\, R, Z_1) \longrightarrow (Z_3 \,|\, R, Z_1, Z_2) \longrightarrow \cdots$$

To proceed with this idea, set

$$\alpha_j(r, z_1, \ldots, z_{j-1}) := -r\gamma_j + \sum_{k<j} l_{jk} z_k,$$

and $\mathscr{J} := \{j : l_{jj} \neq 0\}$, so that simplification of the right hand side of (4) yields:

$$f(\boldsymbol{z}, r) = \lambda^d \mathrm{chi}_{n+1}(r) \exp\left(- \textstyle\sum_{j \notin \mathscr{J}} \frac{\alpha_j^2}{2} - \sum_{j \in \mathscr{J}} \frac{(l_{jj} z_j + \alpha_j)^2}{2} - \lambda \sum_j |z_j| \right)$$

Observe that, given fixed $z_1, \ldots, z_{j-1}$, the function

$$\exp\big(- \tfrac{\alpha_j^2}{2} - \tfrac{(l_{jj} z_j + \alpha_j)^2}{2} \big)$$

depends only on the variable z_j. This key observation suggests simulating $(\boldsymbol{Z}, R)$ sequentially via a proposal density $g(\boldsymbol{z}, r)$, defined as follows.

Algorithm 2 : Defining the proposal density $g(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$.

Require: tuning parameters η and $\boldsymbol{\mu}$ (to be specified later in Theorem 1)
Compute QL decomposition $\mathbf{X} = \mathbf{QL}$; $\hat{\boldsymbol{\beta}} \leftarrow \mathrm{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|_2$; $s^2 \leftarrow \|\mathbf{X}\hat{\boldsymbol{\beta}} - \boldsymbol{y}\|_2^2$; $\boldsymbol{\gamma} \leftarrow \mathbf{L}\hat{\boldsymbol{\beta}}/s$
$R \sim \mathrm{chi}_{n+1}(r)$
for $j = 1, \ldots, p$ **do**
  if $j \notin \mathscr{J}$ **then**
    $Z_j \sim \mathrm{Lap}_{\mu_j}(\lambda z_j)$
  else
    $\alpha_j \leftarrow -R\gamma_j + \sum_{k<j} l_{jk} Z_k$
    $Z_j \sim \mathrm{nl}(l_{jj} z_j + \alpha_j - \mu_j; \lambda/|l_{jj}|, \alpha_j - \mu_j)$
return A draw $(\boldsymbol{Z}, R)$ from the proposal density.

Note that the algorithm requires as input the tuning parameters $\boldsymbol{\mu}$ and η, which will be specified below. If we use g as a proposal density to simulate from a density proportional to f, then the likelihood ratio is $f(\boldsymbol{z}, r)/g(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$. Therefore, the log-likelihood, $\ln(f(\boldsymbol{z}, r)/g(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta))$, is given by:

$$\begin{aligned} \psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta) = & \sum_{j \in \mathscr{J}} \left(\frac{\mu_j^2}{2} - \mu_j (l_{jj} z_j + \alpha_j) \right) - \lambda \sum_{i \notin \mathscr{J}} \mu_j z_j - \sum_{j \neq \mathscr{J}} \frac{\alpha_j^2}{2} \\ & + \frac{\eta^2}{2} - r\eta + n \ln r + \ln \overline{\Phi}(-\eta) + \mathrm{const.} + d \ln \lambda \\ & + \sum_{j \in \mathscr{J}} \xi(\lambda/l_{jj}, \alpha_j - \mu_j) - \sum_{j \notin \mathscr{J}} \ln(1 - \mu_j^2) \end{aligned}$$

To apply the rejection method we need to compute the supremum of the (log-)likelihood ratio (we did the same for the naive rejection Algorithm 1 and obtained $\sup_{\boldsymbol{\beta}} \frac{\lambda^p}{2^p} \exp\left(-\frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right) = \frac{\lambda^p}{2^p} = c$). In other words, we need to compute

$$\sup_{\boldsymbol{z},r} \psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$$

Then, similar to the acceptance rate (2) for the naive rejection Algorithm 1, the acceptance rate of a rejection algorithm with proposal $g(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$ is:

$$c(\boldsymbol{\mu}, \eta) := \frac{\iint f(\boldsymbol{z}, r) \mathrm{d}\boldsymbol{z}\mathrm{d}r}{\sup_{\boldsymbol{z},r} \exp(\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta))} \tag{5}$$

Notice that this acceptance probability depends on the tuning parameters $\boldsymbol{\mu}$ and η. In particular, we can maximize the acceptance probability by minimizing the (log-)likelihood ratio with respect to the tuning parameters. In other words, we can find the best possible tuning parameters by solving the saddle-point (or minimax) program:

$$(\boldsymbol{\mu}^*, \eta^*) := \operatorname*{argmin}_{\boldsymbol{\mu},\eta} \sup_{\boldsymbol{z},r} \psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$$

Fortunately, this saddle-point problem is tractable due to the fact that ψ is a convex-concave function of its arguments.

Theorem 1 (Log-likelihood properties) The saddle-point optimization problem with optimal value

$$\psi^* = \inf_{\boldsymbol{\mu},\eta} \sup_{\boldsymbol{z},r} \psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$$

has a unique solution, $(\boldsymbol{x}^*, r^*, \boldsymbol{\mu}^*, \eta^*)$, which coincides with that of the nonlinear system $\nabla\psi = \mathbf{0}$, where the gradient is with respect to the vector $(\boldsymbol{z}, r, \boldsymbol{\mu}, \eta)$.

The details of the proof are given in Appendix 5.3. A consequence of this theorem is that we can compute the optimal tilting parameters by solving the nonlinear system $\nabla\psi = \mathbf{0}$ using, for example, any built-in routine for solving a system of nonlinear equations (in our implementation we use Matlab's `fsolve.m` to compute the solution). The closed-form formulas for the components of the gradient $\nabla\psi$ are quite messy, see Appendix 5.4, but pose no difficulties with their computer implementation. Note that $\nabla\psi = \mathbf{0}$ needs to be solved only once, regardless of how many draws one simulates from the posterior. In summary, the rejection scheme for simulating one draw from the posterior (3) reads as follows.

Algorithm 3 : Simulating from posterior $\pi(\boldsymbol{\beta}, \sigma \,|\, \boldsymbol{y}, \lambda)$.

Require: shrinkage parameter $\lambda > 0$
  Solve the nonlinear system $\nabla\psi = \mathbf{0}$ to obtain the solution $(\boldsymbol{z}^*, r^*; \boldsymbol{\mu}^*, \eta^*)$
  $\psi^* \leftarrow \psi(\boldsymbol{z}^*, r^*; \boldsymbol{\mu}^*, \eta^*)$
  repeat
    $E \sim \mathsf{Exp}(1)$, that is, E is an exponential r.v. with rate unity
    $(\boldsymbol{Z}, R) \sim g(\boldsymbol{z}, r; \boldsymbol{\mu}^*, \eta^*)$ using Algorithm 2
  until $E > \psi^* - \psi(\boldsymbol{Z}, R; \boldsymbol{\mu}^*, \eta^*)$
  $\sigma \leftarrow s/R$
  $\boldsymbol{\beta} \leftarrow \sigma \boldsymbol{Z}$
  return A draw $(\boldsymbol{\beta}, \sigma)$ from the posterior density (3).

2.3 Independence Metropolis Sampler

When the dimension p is too large, the curse of dimensionality makes it very difficult to construct a proposal whose acceptance probability is not too small. In such cases, MCMC sampling remains the only

viable option that we know for (approximate) simulation. Even in this situation, the sequential sampling proposal in Algorithm 2 is useful, because it can serve as a proposal in an independence Metropolis simulation framework (Kroese et al. 2011, Chapter 6). In particular, given the Markov chain is in the state $(\boldsymbol{Z}', R') = (\boldsymbol{z}', r')$, we accept the proposal move $(\boldsymbol{Z}, R) \sim g(\boldsymbol{z}, r; \boldsymbol{\mu}^*, \eta^*)$ with probability

$$\alpha(\boldsymbol{Z}, R \,\big|\, \boldsymbol{z}', r') := 1 \wedge \exp(\psi(\boldsymbol{Z}, R; \boldsymbol{\mu}^*, \eta^*) - \psi(\boldsymbol{z}', r'; \boldsymbol{\mu}^*, \eta^*))$$

This yields the following MCMC algorithm for approximate simulation from (3).

Algorithm 4 : Approximate simulation from posterior (3).

Require: an initial state $(\boldsymbol{Z}_0, R_0)$; length of chain m
  Solve the nonlinear system $\nabla\psi = \mathbf{0}$ to obtain the solution $(\boldsymbol{z}^*, r^*; \boldsymbol{\mu}^*, \eta^*)$.
  for $t = 1, \ldots, m$ **do**
    Simulate $(\boldsymbol{Z}, R) \sim g(\boldsymbol{z}, r; \boldsymbol{\mu}^*, \eta^*)$ using Algorithm 2
    Simulate $U \sim \mathsf{U}(0, 1)$, independently
    if $U < \alpha(\boldsymbol{Z}, R \,\big|\, \boldsymbol{Z}_{t-1}, R_{t-1})$ **then**
      $(\boldsymbol{Z}_t, R_t) \leftarrow (\boldsymbol{Z}, R)$
    else
      $(\boldsymbol{Z}_t, R_t) \leftarrow (\boldsymbol{Z}_{t-1}, R_{t-1})$
    $\sigma_t \leftarrow s/R_t$
    $\boldsymbol{\beta}_t \leftarrow \sigma_t \boldsymbol{Z}_t$
  return $(\boldsymbol{\beta}_1, \sigma_1), \ldots, (\boldsymbol{\beta}_m, \sigma_m)$

Interestingly, maximization (with respect to the tuning parameters $\boldsymbol{\mu}, \eta$) of the acceptance probability (5) for Algorithm 3 also maximizes an upper bound on the mixing rate of the Markov chain in Algorithm 4. To see this, let $\kappa_t(\cdot \,\big|\, \boldsymbol{\beta}, \sigma)$ be the t-step transition kernel of the Markov chain in Algorithm 4, and define the total variation distance:

$$D_t := \sup_{\boldsymbol{\beta}, \sigma} \sup_{A} \left| \kappa_t(A \,\big|\, \boldsymbol{\beta}, \sigma) - \pi(A \,\big|\, \boldsymbol{y}, \lambda) \right|.$$

Then, the MCMC sampler in Algorithm 4 satisfies (Mengersen and Tweedie 1996, Theorem 2.1) :

$$D_t \leq 2 \left(1 - \frac{1}{\sup_{\boldsymbol{\mu}, \eta} c(\boldsymbol{\mu}, \eta)}\right)^t.$$

That is, a good approximation to the posterior (3) is not only useful for exact simulation via rejection sampling, but also for approximate MCMC sampling via the independence Metropolis sampling.

3 NUMERICAL EXAMPLES

In this section we consider two numerical examples with datasets that have been widely used in the literature as benchmarks for small-scale (as opposed to "big data") statistical inference. Thus, these are examples for which exact simulation is feasible via Algorithm 3.

3.1 Diabetes Dataset

We take the "diabetes dataset" on $n = 442$ patients from Efron et al. (2004). For each patient, we have a record of $p = 10$ predictor variables (age, sex, body mass index, blood pressure, and six blood serum measurements), and a response variable, which measures the severity of nascent diabetes. We then wish to determine which of the predictors are most relevant to the response variable using the LASSO model (1), where $\boldsymbol{y} \in \mathbb{R}^n$ is the vector of centralized response variables, and $\mathbf{X}$ is the standardized matrix of

predictors. The LASSO parameter λ is chosen to be $\lambda = 0.24$, which is the value that maximizes the marginal likelihood; see Park and Casella (2008).

The results of simulating from the posterior are given in the last two columns of Table 1, which also shows the ordinary least squares estimate. Figure 1 shows the estimated marginal distributions of each of the ten predictors with the ordinary least squares point-estimate superimposed as a (blue) dot on the boxplots.

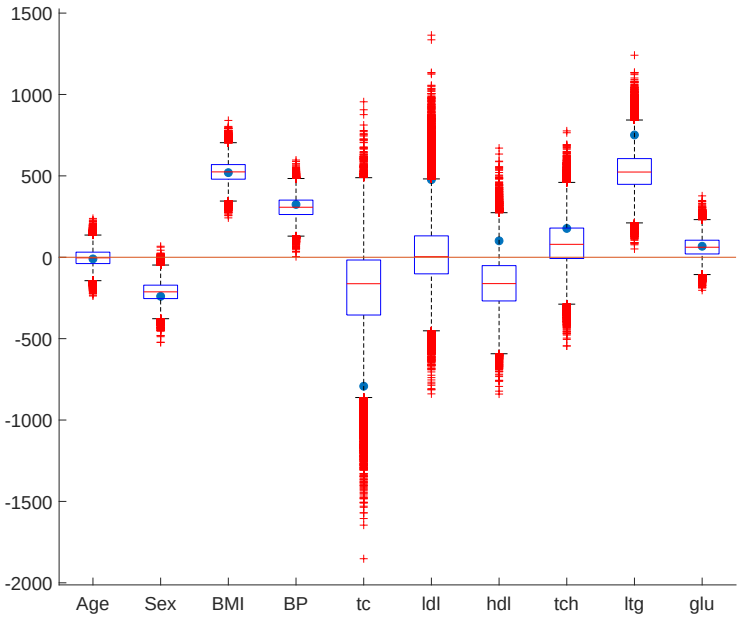

Figure 1: Marginal distributions of each predictor coefficient (as boxplots). The statistically significant predictors appear to be sex, body mass index (BMI), blood pressure (BP), and lgt.

Table 1: Simulation results from the posterior density (diabetes dataset) using 10^5 iid draws.

	Ord. least sq.	Posterior Median	Estimated 95% credible interval
age of patient (age)	-10	-3.1	(-110,102)
Gender of patient (sex)	-239	-212	(-332,-91)
body mass index (BMI)	519	524	(394,655)
blood pressure (BP)	324	307	(179,435)
tc	-792	-161	(-807,232)
ldl	476	2.7	(-327,491)
hdl	101	-160	(-454,153)
tch	177	82	(-186, 374)
ltg	751	523	(309,791)
glu	67	61	(-53,188)

The acceptance rate of the sequential sampling Algorithm 3 was estimated to be approximately 0.39. In contrast, the naive rejection Algorithm 1 has an acceptance probability smaller than 10^{-7} (making the event of drawing from the posterior a rare event, and the probability of acceptance a rare-event probability).

In addition, we compared the output of Algorithm 3 (taking about 7 seconds) with the output of the popular Park&Casella Gibbs sampler (taking about 16 seconds). We observed that the boxplots computed

from the output of the Gibbs sampler (not shown here) do not extend as much as the boxplots on Figure 1. This suggests that the exact sampler is better at exploring the tails of the posterior density.

3.2 Boston Housing Dataset

As another example, we consider the Boston housing dataset from Harrison Jr and Rubinfeld (1978), which attempts to explain housing prices in the Boston area from the $p = 13$ predictors given in Table 2 and $n = 506$ observations. The table confirms that Algorithm 3 simulates from the posterior accurately, because the MCMC simulations in Polson et al. (2014) suggest the same list of statistically relevant predictors (crime levels, proximity to waterfront, number of rooms, distance to employment centers, etc) using the shrinkage value of $\lambda = 5.71$. For this example, the acceptance rate of the sequential rejection sampler was estimated to be approximately 0.67.

Table 2: Posterior estimates for the Boston Housing dataset using 10^5 iid draws.

	Ord. least sq.	Posterior Median	Estimated 95% credible interval
per capita crime rate by town (crim)	-0.10	-0.098	(-0.16, -0.032)
proportion of zoned land (zn)	0.047	0.048	(0.021,0.076)
proportion of non-retail business (indus)	0.011	-0.034	(-0.15, 0.084)
Charles River exposure (chas)	2.75	1.75	(0.14, 3.47)
nitric oxide pollution (nox)	-12.0	-1.49	(-6.41 , 0.70)
number of rooms (rm)	4.61	4.079	(3.54, 4.62)
proportion built before 1940 (age)	-0.0023	-0.010	(-0.035 , 0.015)
distance to CBD (dis)	-1.28	-1.17	(-1.54, -0.80)
highway access (rad)	0.25	0.25	(0.13, 0.38)
property tax rate (tax)	-0.011	-0.013	(-0.020, -0.0059)
quality of schools (ptratio)	-0.73	-0.72	(-0.93, -0.51)
proportion of ethnic diversity (b)	0.011	0.010	(0.0056, 0.015)
economic status of residents (lstat)	-0.48	-0.53	(-0.63,-0.44)

4 CONCLUSIONS

Bayesian posterior simulation requires approximate MCMC simulation of random vectors in a high-dimensional space. When the Bayesian model is simple enough and the dimension of the space is not too large, one can often do better than approximate MCMC simulation. In this paper we showed that the Bayesian LASSO linear model is simple enough to permit exact simulation from the posterior density. This was achieved by exploiting the lower triangular structure of the $\mathbf{L}$ matrix in the QL factorization of the regression matrix $\mathbf{X}$. The lower triangular structure suggested a natural sequential simulation algorithm that helps draw random vectors from the posterior.

Whenever it is feasible, exact simulation has the advantage that all draws from the posterior are independent and identically distributed, thus simplifying all output analysis. In contrast, the analysis of the MCMC output is much more difficult and subjective — it requires that we estimate a Markov chain burn-in period and examine auto-correlation plots to decide whether the Markov chain converges fast enough to the target density.

The proposed sampler assumes that the shrinkage parameter λ is given to us. As for future work, one may assign a prior distribution on λ and then determine its value by making it part of the posterior simulation output. Similar to our treatment of σ and $\boldsymbol{\beta}$, the major difficulty will be in selecting a prior for λ that will yield a posterior pdf, which is concave in λ and convex in its tuning parameters.

ACKNOWLEDGMENTS

Yi-Lung Chen is supported by the Australian Government Research Training Program Scholarship. Zdravko Botev has been supported by the Australian Research Council grant DE140100993. P. L'Ecuyer has been supported by a discovery grant from NSERC-Canada, a Canada Research Chair, and an Inria International Chair. D. P. Kroese has been supported by ACEMS: https://acems.org.au/.

5 APPENDIX

5.1 Simulating from an Exponentially Tilted Laplace Distribution

Algorithm 5 : Simulating an Exponentially twisted Laplace variable

Require: twisting parameter μ such that $|\mu| < 1$
$U, V \sim \mathsf{U}(0,1)$, independently
if $U < (1+\mu)/2$ **then**
$X \leftarrow -\frac{\ln(V)}{1-\mu}$
else
$X \leftarrow \frac{\ln(V)}{1+\mu}$
return X distributed from the pdf $\mathrm{Lap}_\mu(x)$

Note that when $\mu = 0$, we revert to simulating from the untilted Laplace density: $\exp(-|x|)/2$.

5.2 Normal-Laplace density

The normal-Laplace density, $\mathrm{nl}(z;\lambda,\alpha) \propto \phi(z)\exp(-\lambda|z-\alpha|)$, has a mixture form:

$$w_1 \frac{\phi(z+\lambda)\mathbb{I}\{z>\alpha\}}{\overline{\Phi}(\lambda+\alpha)} + w_2 \frac{\phi(z-\lambda)\mathbb{I}\{z<\alpha\}}{\overline{\Phi}(\lambda-\alpha)}$$

with weights $w_{1,2} \propto \overline{\Phi}(\lambda \pm \alpha)/\phi(\lambda \pm \alpha)$. Hence, we have the following simulation algorithm.

Algorithm 6 : Simulating an normal-Laplace variable

Require: parameters $\lambda > 0$ and $\alpha \in \mathbb{R}$
$U \sim \mathsf{U}(0,1)$
$w_1 \leftarrow \overline{\Phi}(\lambda+\alpha)/\phi(\lambda+\alpha)$
$w_2 \leftarrow \overline{\Phi}(\lambda-\alpha)/\phi(\lambda-\alpha)$
if $U < w_2/(w_1+w_2)$ **then**
Simulate $Z \sim \mathsf{Normal}(\lambda,1)$, conditional on $Z < \alpha$
else
Simulate $Z \sim \mathsf{Normal}(-\lambda,1)$, conditional on $Z > \alpha$
return Z as distributed from the pdf $\mathrm{nl}(z;\lambda,\alpha)$

Simulating a truncated normal random variable is best accomplished using the algorithms described by Botev and L'Ecuyer (2017). These algorithm are robust even deep into the tail of the normal distribution, where naive approaches exhibit numerical instability and underflow errors.

5.3 Proof of Theorem

We note that the idea of solving a minimax program was used in previous works (Botev 2017; Botev and L'Ecuyer 2015) to design efficient algorithms for simulation from the truncated student and normal

distributions. The proof of this theorem is thus similar in spirit to the proofs given in those previous works. In particular, the proof relies on the following facts (Prékopa 1973):

1. A log-concave measure $\mathbb{P}$ satisfies

$$\mathbb{P}[\boldsymbol{X} \in \sum_i w_i A_i] \geq \prod_i (\mathbb{P}[\boldsymbol{X} \in A_i])^{w_i}$$

   for some probability vector $\boldsymbol{w}$. Here $\sum_i w_i A_i$ is to be understood as a *Minkowski sum*, namely,

$$w_1 A_1 + w_2 A_2 = \{w_1 \boldsymbol{x}_1 + w_2 \boldsymbol{x}_2 : \boldsymbol{x}_1 \in A_1, \boldsymbol{x}_2 \in A_2\}.$$

2. If $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ is a log-concave function, then the marginal

$$g(\boldsymbol{x}) = \int_{\mathbb{R}^{d_2}} f(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y}$$

   is a log-concave function as well.
3. The indicator function $\mathbb{I}[\boldsymbol{x} \in C]$ of a convex set C (for example, the sub-level set of convex function) is a log-concave function;
4. The product of log-concave functions is again log-concave.

Now consider the normalizing constant, $\xi(\lambda, \alpha)$, of the normal-Laplace density. We will show that it is log-concave in the second argument. If $E \sim \mathsf{Exp}(1)$ is an exponential r.v., then ξ can be written as

$$\xi(\lambda, \alpha) = \ln \mathbb{P}[E > \lambda |Z - \alpha|] = \ln \mathbb{P}[\{\lambda Z - \lambda\alpha - E < 0\} \cap \{-\lambda Z + \lambda\alpha + E < 0\}]$$

Now the sets $\mathscr{C}_1 = \{z - \alpha - e/\lambda < 0\}$ and $\mathscr{C}_2 = \{-z + \alpha + e/\lambda < 0\}$ are convex in (z, e, α). Hence, the intersection is also convex and the indicator function of $\mathscr{C}_1 \cap \mathscr{C}_2$ is a log-concave function of (z, e, α). Now the joint density, $\phi(z)\exp(-e)$, of the Gaussian and exponential variables, (Z, E), is also log-concave in (z, e). Therefore, by the marginalization property above, the integral with respect to (z, e) will yield a log-concave function in the third argument of (z, e, α). In other words, $\xi(\cdot, \alpha)$ is log-concave in the second argument. We can now address each of the following properties of the log-likelihood ψ.

Concavity in $\boldsymbol{z}$. Inspection of ψ shows that, ignoring linear terms in $\boldsymbol{z}$ and all other variables kept as constants, ψ is a sum of $\xi(\cdot, \alpha_j(\boldsymbol{z}) - \mu_j)$ terms. Using the facts: 1) α_j is a linear function of $\boldsymbol{z}$; 2) ξ is log-concave in its second argument; 3) the (product) sum of (log-) concave functions is (log-) concave; we conclude that ψ is concave in $\boldsymbol{z}$.

Concavity in r. The dependence of ψ on r is given by $-r\eta + n \ln r$, which is clearly concave in r.

Convexity in η. The dependence of ψ on η is given by the expression $\eta^2/2 - r\eta + \ln \overline{\Phi}(-\eta)$. Since the linear term does not affect the convexity, we need to show that $\ln \overline{\Phi}(\eta) + \eta^2/2$ is convex. That this is the case follows from the fact that

$$\ln \overline{\Phi}(\eta) + \frac{\eta^2}{2} = \ln \int_0^\infty \phi(x) \exp(\eta x) \mathrm{d}x + \text{const.}$$

is the cumulant generating function of the truncated normal distribution (truncated to $(0, \infty)$).

Convexity in $\boldsymbol{\mu}$. ψ depends on $\boldsymbol{\mu}$ through a weighted linear combination of the terms $\mu_j, -\ln(1-\mu_j^2)$, and $\mu_j^2/2 + \xi(\cdot, \alpha_j - \mu_j)$. Thus, if these terms are convex, then the convexity in $\boldsymbol{\mu}$ follows from the fact that the sum of convex functions is convex.

It is clear that $-\ln(1-\mu^2)$ is convex, because its second derivative is $2(\mu^2+1)/(1-\mu^2)^2 > 0$. What is less obvious is that $\mu^2 2 + \xi(\lambda, \mu)$ is convex in μ, because we showed above that ξ is concave in its

second argument. To see the convexity, consider

$$\begin{aligned}\frac{\mu^2}{2} + \xi(\lambda,\mu) &= \frac{\mu^2}{2} + \ln \int_{-\infty}^{\infty} \phi(z)\exp(-\lambda|z-\mu|)\mathrm{d}z \\ &= \ln \int_{-\infty}^{\infty} \phi(y)\exp(-\lambda|y|)\exp(y\mu)\mathrm{d}y \\ &= \xi(\lambda,0) + \ln \int_{-\infty}^{\infty} \frac{\phi(y)\exp(-\lambda|y|)}{\int_{\mathbb{R}}\phi(z)\exp(-\lambda|z|)\mathrm{d}z}\exp(y\mu)\mathrm{d}y\end{aligned}$$

The last expression is the constant $\xi(\lambda,0)$ plus the cumulant generating function (a function of μ) of the density proportional to $\phi(y)\exp(-\lambda|y|)$. Since the Laplace transform (or moment generating) function of a density is log-convex, the cumulant generating function is convex.

Putting all results together. Finally, from all of the above we can conclude that $\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$ is a concave-convex function that satisfies the saddle-point condition

$$\inf_{\boldsymbol{\mu},\eta}\sup_{\boldsymbol{z},r}\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta) = \sup_{\boldsymbol{z},r}\inf_{\boldsymbol{\mu},\eta}\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$$

Using the fact that, if $f(\boldsymbol{x}, \boldsymbol{y})$ is a convex function for each fixed $\boldsymbol{y}$, then the pointwise supremum with respect to $\boldsymbol{y}$ is also convex, we conclude that $\sup_{\boldsymbol{z},r}\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$ is convex in $(\boldsymbol{\mu}, \eta)$. Hence, to find the minimum we set the gradient of ψ (with respect to $(\boldsymbol{\mu}, \eta)$) to zero. In other words, $\inf_{\boldsymbol{\mu},\eta}\sup_{\boldsymbol{z},r}\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$ has the same value as the concave optimization $\sup_{\boldsymbol{z},r}\psi(\boldsymbol{z}, r; \boldsymbol{\mu}, \eta)$, subject to $\nabla_{\boldsymbol{\mu},\eta}\psi = \mathbf{0}$. Therefore, this last constrained concave optimization can be solved by setting $\nabla\psi = \mathbf{0}$.

5.4 Computing the gradient $\nabla\psi$ in closed form.

Let $w_{1,2} \propto \overline{\Phi}(\lambda \pm \alpha)/\phi(\lambda \pm \alpha)$ be normalized weights. Then, we can write

$$\begin{aligned}\partial\xi/\partial\alpha &= \lambda(2w_1 - 1) =: \xi_2(\lambda,\alpha) \\ \partial\xi/\partial\lambda &= \lambda + \alpha(2w_1 - 1) - 2\exp(-\xi(\lambda,\alpha) - \alpha^2/2) =: \xi_1(\lambda,\alpha)\end{aligned}$$

Setting $\tilde{\mathbf{L}} = \mathbf{L} - \mathrm{diag}(\mathbf{L})$ (that is, same as $\mathbf{L}$, but with zeros down the main diagonal) and $\xi_{2j} := \xi_2(\lambda/l_{jj}, \alpha_j - \mu_j)$, we can write the gradient $\nabla\psi$ as:

$$\begin{aligned}\partial\psi/\partial z_i &= -\mu_i(l_{ii}\mathbb{I}_{\{i\in\mathscr{J}\}} + \mathbb{I}_{\{i\notin\mathscr{J}\}}) + \textstyle\sum_{j\in\mathscr{J}}\tilde{l}_{ji}\xi_{2j} - \sum_{j\notin\mathscr{J}}\mu_j\tilde{l}_{ji} \\ \partial\psi/\partial r &= -\eta + (\nu-1)/r - \textstyle\sum_{j\in\mathscr{J}}\gamma_j\xi_{2j} + \sum_{j\notin\mathscr{J}}\mu_j\gamma_j \\ \partial\psi/\partial\mu_i &= (\mu_i - (l_{ii}z_i + \alpha_i) - \xi_{2j})\mathbb{I}_{\{i\in\mathscr{J}\}} - (z_i - 2\mu_i/(\lambda^2 - \mu_i^2))\mathbb{I}_{\{i\notin\mathscr{J}\}} \\ \partial\psi/\partial\eta &= \eta - r + \phi(\eta)/\Phi(\eta)\end{aligned}$$

REFERENCES

Botev, Z. I. 2017. "The normal law under linear restrictions: simulation and estimation via minimax tilting". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79(1):125–148.

Botev, Z. I., and P. L'Ecuyer. 2015. "Efficient probability estimation and simulation of the truncated multivariate Student-t distribution". *Proceedings of the 2015 Winter Simulation Conference*, 380–391. IEEE Press.

Botev, Z. I., and P. L'Ecuyer. 2017. "Simulation from the Normal Distribution Truncated to an Interval in the Tail". *Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2016)*, 23–29. ICST Publications, Brussels.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani. 2004. "Least angle regression". *The Annals of Statistics* 32(2):407–499.

Glynn, P. W. 2016. "Exact simulation vs exact estimation". *Proceedings of the 2016 Winter Simulation Conference*, 193–205. IEEE Press.

Golub, G. H., and C. F. Van Loan. 2013. *Matrix computations*, Fourth Edition. John Hopkins University Press.

Harrison Jr, D., and D. L. Rubinfeld. 1978. "Hedonic housing prices and the demand for clean air". *Journal of Environmental Economics and Management* 5(1):81–102.

Kroese, D. P., T. Taimre, and Z. I. Botev. 2011. *Handbook of Monte Carlo methods*. John Wiley & Sons.

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

Park, T., and G. Casella. 2008. "The Bayesian Lasso". *Journal of American Statistical Association* 103(482):681–686.

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

Prékopa, A. 1973. "On logarithmic concave measures and functions". *Acta Scientiarum Mathematicarum* 34:335–343.

Propp, J. G., and D. B. Wilson. 1996. "Exact sampling with coupled Markov chains and applications to statistical mechanics". *Random Structures and Algorithms* 9(1-2):223–252.

AUTHOR BIOGRAPHIES

ZDRAVKO I. BOTEV is Senior Lecturer at the School of Mathematics and Statistics at the University of New South Wales in Sydney, Australia (UNSW Sydney). His research interests include splitting and adaptive importance sampling methods for rare-event simulation. His software offers one of the most accurate nonparametric density estimation methods, and it has been widely used in the applied sciences. For more information, visit his webpage: http://web.maths.unsw.edu.au/~zdravkobotev/.

YI-LUNG CHEN is a PhD candidate at the School of Mathematics and Statistics and a tutor at the University of New South Wales in Sydney, Australia (UNSW Sydney).

PIERRE L'ECUYER is Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada. He also holds an International Chair at Inria, in Rennes, France. He is a member of the CIRRELT and GERAD research centers. His main research interests are random number generation, quasi-Monte Carlo methods, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He has published over 270 scientific articles, book chapters, and books, and has developed software libraries and systems for random number generation and stochastic simulation. He has been a referee for 145 different scientific journals. More information can be found on his web page: http://www.iro.umontreal.ca/~lecuyer.

SHEV MACNAMARA is a Senior Lecturer in the School of Mathematical and Physical Sciences at the University of Technology Sydney, Australia. Previously, he has held appointments at MIT and the University of Oxford.

DIRK P. KROESE is a Professor of Mathematics and Statistics at The University of Queensland. He is the co-author of several influential monographs on simulation and Monte Carlo methods, including Handbook of Monte Carlo Methods and Simulation and the Monte Carlo Method, (3rd Edition). Dirk is a pioneer of the well-known Cross-Entropy method-an adaptive Monte Carlo technique, invented by Reuven Rubinstein, which is being used around the world to help solve difficult estimation and optimization problems in science, engineering, and finance. His personal website can be found under https://people.smp.uq.edu.au/DirkKroese/.