Biometrika (2021), **108**, 1, *pp.* 53–69
Printed in Great Britain
doi: 10.1093/biomet/asaa055
Advance Access publication 5 October 2020

In search of lost mixing time: adaptive Markov chain Monte Carlo schemes for Bayesian variable selection with very large *p*

By J. E. GRIFFIN

Department of Statistical Science, University College London, Gower Street, London WC1E 6BT, U.K.
j.griffin@ucl.ac.uk

K. G. ŁATUSZYŃSKI and M. F. J. STEEL

Department of Statistics, University of Warwick, Coventry, CV4 7AL, U.K.
K.G.Latuszynski@warwick.ac.uk M.Steel@warwick.ac.uk

Summary

The availability of datasets with large numbers of variables is rapidly increasing. The effective application of Bayesian variable selection methods for regression with these datasets has proved difficult since available Markov chain Monte Carlo methods do not perform well in typical problem sizes of interest. We propose new adaptive Markov chain Monte Carlo algorithms to address this shortcoming. The adaptive design of these algorithms exploits the observation that in large-*p*, small-*n* settings, the majority of the *p* variables will be approximately uncorrelated a posteriori. The algorithms adaptively build suitable nonlocal proposals that result in moves with squared jumping distance significantly larger than standard methods. Their performance is studied empirically in high-dimensional problems and speed-ups of up to four orders of magnitude are observed.

Some key words: Expected squared jumping distance; High-dimensional data; Large-*p*, small-*n* problem; Linear regression; Optimal scaling; Spike-and-slab prior; Variable selection.

1. Introduction

The availability of large datasets has led to an increasing interest in variable selection methods applied to regression models with many potential variables, but few observations, so-called large-*p*, small-*n* problems. Frequentist approaches have mainly concentrated on point estimates under assumptions of sparsity using penalized maximum likelihood procedures (Hastie et al., 2015). Bayesian approaches to variable selection are an attractive and natural alternative, and lead to a posterior distribution on all possible models which can address model uncertainty for variable selection and prediction. A growing literature provides a theoretical basis for good posterior properties in large-*p* problems (see, e.g., Johnson & Rossell, 2012; Castillo et al., 2015).

The posterior probabilities of all possible models can usually only be calculated or approximated if *p* is smaller than 30. If *p* is larger, Markov chain Monte Carlo methods are typically used to sample from the posterior distribution (George & McCulloch, 1997; O'Hara & Sillanpää, 2009; Clyde et al., 2011). García-Donato & Martínez-Beneito (2013) discuss the benefits

of such methods. The most widely used Markov chain Monte Carlo algorithm in this context is the Metropolis–Hastings sampler, where new models are proposed using add-delete-swap samplers (Brown et al., 1998; Chipman et al., 2001). For example, this approach is used by Nikooienejad et al. (2016) in a binary regression model with a nonlocal prior for the regression coefficients on a dataset with 7129 genes. Some supporting theoretical understanding of convergence is available for the add-delete-swap samplers, e.g., conditions for rapid mixing in linear regression models have been derived by Yang et al. (2016). Others have considered more targeted moves in model space. For example, Titsias & Yau (2017) introduce the Hamming ball sampler which more carefully selects the proposed model in a Metropolis–Hastings sampler, in a similar way to shotgun variable selection (Hans et al., 2007), and Schäfer & Chopin (2013) develop a sequential Monte Carlo method that uses a sequence of annealed posteriors. Several authors use more general shrinkage priors and develop suitable Markov chain Monte Carlo algorithms for high-dimensional problems (see, e.g., Bhattacharya et al., 2016). Nonlocal priors (Johnson & Rossell, 2012) are adopted in Shin et al. (2018), who use screening for high dimensions. Zanella & Roberts (2019) combine Markov chain Monte Carlo and importance sampling ideas in their tempered Gibbs sampler.

The challenge of performing Markov chain Monte Carlo for Bayesian variable selection in high dimensions has led to several developments sacrificing exact posterior exploration. For example, Liang et al. (2013) use the stochastic approximation Monte Carlo algorithm (Liang et al., 2007) to efficiently explore model space. In another direction, variable selection can be performed as a post-processing step after fitting a model including all variables (see, e.g., Bondell & Reich, 2012; Hahn & Carvalho, 2015). Several authors develop algorithms that focus on high posterior probability models. In particular, Rockova & George (2014) propose a deterministic expectation-maximization-based algorithm for identifying posterior modes, while Papaspiliopoulos & Rossell (2017) develop an exact deterministic algorithm to find the most probable model of any given size in block-diagonal design models.

Alternatively, Markov chain Monte Carlo methods for variable selection can be tailored to the data to allow faster convergence and mixing using adaptive ideas (see, e.g., Green et al., 2015, § 2.4, and references therein). Several strategies have been developed in the literature for both the Metropolis-type algorithms (Ji & Schmidler, 2013; Lamnisos et al., 2013) and Gibbs samplers (Nott & Kohn, 2005; Richardson et al., 2010). Our proposal is a Metropolis–Hastings kernel that learns the relative importance of the variables, unlike previous work (see, e.g., Ji & Schmidler, 2013; Lamnisos et al., 2013). A similar strategy is used by Zanella & Roberts (2019) in a Gibbs sampling framework. This leads to substantially more efficient algorithms than commonly used methods in high-dimensional settings, and for which the computational cost of one step scales linearly with p. The algorithms adaptively build suitable nonlocal Metropolis–Hastings-type proposals that result in moves with expected squared jumping distance (Gelman et al., 1996) significantly larger than standard methods. In idealized examples the limiting versions of our adaptive algorithms converge in $\mathcal{O}(1)$ and result in super-efficient sampling. They outperform independent sampling in terms of the expected squared jump distance and also in the sense of the central limit theorem asymptotic variance. This is in contrast to the behaviour of optimal local random walk Metropolis algorithms that on analogous idealized targets need at least $\mathcal{O}(p)$ samples to converge (Roberts et al., 1997). The performance of our algorithms is studied empirically in realistic high-dimensional problems for both synthetic and real data. In particular, in § 4.1, for a well-studied synthetic data example, speed-ups of up to four orders of magnitude are observed compared to standard algorithms. Moreover, in § 4.2, we show the efficiency of the method in the presence of multicollinearity on a real-data example with $p = 100$ variables, and in § 4.3, we present real-data gene expression examples with $p = 22\,576$ and with $p = 79\,748$, and reliably

estimate the posterior inclusion probabilities for all variables. The Supplementary Material has results from three datasets with moderate p and high correlations used in Schäfer & Chopin (2013), indicating that our algorithms outperform most other methods in the literature. The algorithms have the potential to be parallelized across the multiple chains and to be applied to non-Gaussian models or more general prior structures.

2. Design of the adaptive samplers

2.1. *The setting*

Our approach is applicable to general regression settings, but we will focus on normal linear regression models. This will allow for clean efficiency comparisons independent of model-specific sampling details, e.g., of a reversible jump implementation. We define $\gamma = (\gamma_1, \ldots, \gamma_p) \in \Gamma = \{0, 1\}^p$ to be a vector of indicator variables with $\gamma_i = 1$ if the ith variable is included in the model and $p_\gamma = \sum_{j=1}^{p} \gamma_j$. We consider the model specification

$$y = \alpha 1_n + X_\gamma \beta_\gamma + e, \qquad e \sim N(0, \sigma^2 I_n),$$

where y is an $(n \times 1)$-dimensional vector of responses, a_q represents a q-dimensional column vector with entries a, and X_γ is an $(n \times p_\gamma)$-dimensional data matrix formed using the included variables. We consider Bayesian variable selection and, for clarity of exposition and validity of comparisons, we will assume the commonly used prior structure

$$p(\alpha, \sigma^2, \beta_\gamma, \gamma) \propto \sigma^{-2} p(\beta_\gamma \mid \sigma^2, \gamma) p(\gamma), \tag{1}$$

with $\beta_\gamma \mid \sigma^2, \gamma \sim N(0, \sigma^2 V_\gamma)$, and $p(\gamma) = h^{p_\gamma}(1-h)^{p-p_\gamma}$. The hyperparameter $0 < h < 1$ is the prior probability that a particular variable is included in the model, and V_γ is often chosen as proportional to $(X_\gamma^{\mathrm{T}} X_\gamma)^{-1}$, a g-prior, or to the identity matrix. In both cases, the marginal likelihood $p(y \mid \gamma)$ can be calculated analytically. The prior can be further extended with hyperpriors, for example adopting $h \sim \mathrm{Be}(a, b)$.

We will consider sampling from the target distribution $\pi_p(\gamma) = p(\gamma \mid y)$ using a nonsymmetric Metropolis–Hastings kernel. Let the probability of proposing to move from model γ to γ' be

$$q_\eta(\gamma, \gamma') = \prod_{j=1}^{p} q_{\eta,j}(\gamma_j, \gamma_j'), \tag{2}$$

where $\eta = (A, D) = (A_1, \ldots, A_p, D_1, \ldots, D_p)$, $q_{\eta,j}(\gamma_j = 0, \gamma_j' = 1) = A_j$ and $q_{\eta,j}(\gamma_j = 1, \gamma_j' = 0) = D_j$. The proposal can be quickly sampled, the parameterization allows optimization of the expected squared jumping distance, and multiple variables can be added to or deleted from the model in one iteration. The proposed model is accepted using the standard Metropolis–Hastings acceptance probability

$$a_\eta(\gamma, \gamma') = \min\left\{1, \frac{\pi_p(\gamma') q_\eta(\gamma', \gamma)}{\pi_p(\gamma) q_\eta(\gamma, \gamma')}\right\}.$$

2.2. *In search of lost mixing time: optimizing the sampler*

The transition kernel in (2) is highly parameterized, with $2p$ parameters, and these will be tuned using adaptive Markov chain Monte Carlo methods (see, e.g., Andrieu & Thoms, 2008;

Roberts & Rosenthal, 2009; Green et al., 2015). These methods allow the tuning of parameters on the fly to improve mixing using some computationally accessible performance criterion whilst maintaining the ergodicity of the chain. Suppose that μ_p is a p-dimensional probability density function which has the form $\mu_p = \prod_{j=1}^{p} f$. A commonly used result is that the optimal scale of a random walk proposal for μ_p leads to a mean acceptance rate of 0.234 as $p \to \infty$ for some smooth enough f. The underlying analysis also implies that the optimized random walk Metropolis will converge to stationarity in $\mathcal{O}(p)$ steps. This is a useful guide even in moderate dimensions, and well beyond the restrictive independent, identically distributed product form assumption of Roberts et al. (1997). Lamnisos et al. (2013) show that this rule can be effectively used to tune a Metropolis–Hastings sampler for Bayesian variable selection. However, other results suggest that other optimal scaling rules could work well in Bayesian variable selection problems. Firstly, Neal et al. (2012) establish, under additional regularity conditions, that if f is discontinuous, the optimal mean acceptance rate for a Metropolis–Hastings random walk is $e^{-2} \approx 0.1353$ and the chain mixes in $\mathcal{O}(p^2)$ steps, an order of magnitude slower than with smooth target densities f. Rather surprisingly, Lee & Neal (2018) show that the optimally tuned independence sampler in this setting recovers the $O(p)$ mixing and acceptance rate of 0.234 without any additional smoothness conditions. Secondly, Roberts (1998) considers optimal scaling of the random walk Metropolis–Hastings algorithm on $\Gamma = \{0,1\}^p$ for the product measures

$$\mu_p(\gamma_1, \ldots, \gamma_p) = s^{p_\gamma}(1-s)^{p-p_\gamma}, \qquad \gamma = (\gamma_1, \ldots, \gamma_p) \in \Gamma, \quad 0 < s < 1.$$

If s is close to $1/2$, the optimal $\mathcal{O}(p)$ mixing rate occurs as p tends to infinity if the mean acceptance rate is 0.234. If $s \to 0$ as $p \to \infty$, the numerical results of Roberts (1998, § 3) indicate that the optimally tuned random walk Metropolis proposes to change two γ_js at a time, but that the acceptance rate deteriorates to zero resulting in the chain not moving. This suggests that the actual mixing in this regime is slower than the $\mathcal{O}(p)$ observed for smooth continuous densities.

In Bayesian variable selection, it is natural to assume that the variables differ in posterior inclusion probabilities and so we consider target densities that have the form

$$\pi_p(\gamma) = \prod_{j=1}^{p} \pi_j^{\gamma_j}(1-\pi_j)^{1-\gamma_j}, \qquad \gamma \in \Gamma, \tag{3}$$

where $0 < \pi_j < 1$ for $j = 1, \ldots, p$. Consider the nonsymmetric Metropolis–Hastings algorithm with the product form proposal $q_\eta(\gamma, \gamma')$ given by (2) targeting the posterior distribution given by (3). Note that $\alpha_\eta(\cdot, \cdot) \equiv 1$ for any choice of $\eta = (A, D)$, satisfying

$$\frac{A_j}{D_j} = \frac{\pi_j}{1-\pi_j} \qquad \text{for every } j. \tag{4}$$

To discuss optimal choices of η, we consider several commonly used criteria for Markov chains with stationary distribution π and transition kernel P on a finite discrete state space Γ. The mixing time of a Markov chain (Roberts & Rosenthal, 2004) is $\rho := \min\{t : \max_{\gamma \in \Gamma} \|P^t(\gamma, \cdot) - \pi(\cdot)\|_{\mathrm{TV}} < 1/2\}$, where $\|\cdot\|_{\mathrm{TV}}$ is the total variational norm. If $\Gamma = \{0,1\}^p$, it is natural to define the expected squared jumping distance (Gelman et al., 1996) as $E_\pi\left(\sum_{j=1}^{p} |\gamma_j^{(0)} - \gamma_j^{(1)}|^2\right)$, where $\gamma^{(0)}$ and $\gamma^{(1)}$ are two consecutive values in a Markov chain trajectory, which is the average number of variables changed in one iteration. Suppose that the Markov chain is ergodic; then, for

any function $f : \Gamma \to \mathbb{R}$, $\frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} f(\gamma^{(k)}) \xrightarrow{D} N(E_\pi f, \sigma^2_{P,f})$, where the constant $\sigma^2_{P,f}$ depends on the transition kernel P and function f. Consider transition kernels P_1 and P_2. If $\sigma^2_{P_1,f} \leqslant \sigma^2_{P_2,f}$ for every f, then P_1 dominates P_2 in Peskun ordering (Peskun, 1973). If P_1 dominates all other kernels from a given class, then P_1 is optimal in this class with respect to Peskun ordering. Apart from toy examples, Peskun ordering can rarely be established without further restrictions. Hence, for the variable selection problem, where posterior inclusion probabilities are often of interest, we consider Peskun ordering for the class $\mathbb{L}(\Gamma)$ of linear combinations of univariate functions,

$$\mathbb{L}(\Gamma) := \left\{ f : \Gamma \to \mathbb{R} : f(\gamma) = a_0 + \sum_{j=1}^{p} a_j f_j(\gamma_j) \right\}. \tag{5}$$

We consider two proposals which satisfy (4): the independent proposal for which $A_j = 1 - D_j = \pi_j$, and the informed proposal for which $A_j = \min\left(1, \frac{\pi_j}{1-\pi_j}\right)$ and $D_j = \min\left(1, \frac{1-\pi_j}{\pi_j}\right)$. The following proposition shows that the informed proposal has more desirable properties.

PROPOSITION 1. *Consider the class of Metropolis–Hastings algorithms with target distribution given by* (3)*, and proposal* $q_\eta(\gamma, \gamma')$ *given by* (2) *with the independent or informed proposal. Let* $\mathrm{var}_\pi f$ *be the stationary variance of* f *under* $\pi_p(\gamma)$ *and* $\pi^{(j)} := \{1 - \pi_j, \pi_j\}$*. Then:*

(i) *The independent proposal leads to*

  (a) *independent sampling and optimal mixing time* $\rho = 1$*;*
  (b) *the expected squared jumping distance* $E_\pi(\Delta^2) = 2\sum_{j=1}^{p} \pi_j(1 - \pi_j)$*;*
  (c) *the asymptotic variances* $\sigma^2_{P,f} = \mathrm{var}_\pi f$ *for arbitrary* f *and* $\sigma^2_{P,f} = \mathrm{var}_\pi f = \sum_{j=1}^{p} a_j^2 \mathrm{var}_{\pi^{(j)}} f_j$ *for* $f \in \mathbb{L}(\Gamma)$*.*

(ii) *The informed proposal leads to*

  (a) *the expected squared jumping distance* $E_\pi(\Delta^2) = 2\sum_{j=1}^{p} \min(1 - \pi_j, \pi_j)$*, which is maximal;*
  (b) *the asymptotic variance* $\sigma^2_{P,f} = \sum_{j=1}^{p} \left\{2\max(1 - \pi_j, \pi_j) - 1\right\} a_j^2 \mathrm{Var}_{\pi^{(j)}} f_j$ *for* $f \in \mathbb{L}(\Gamma)$*, which is optimal with respect to the Peskun ordering for the class of linear functions* $\mathbb{L}(\Gamma)$ *defined in* (5)*.*

Remark 1. The differences of the expected squared jumping distance and asymptotic variance for the two proposals is largest when π_j is close to $1/2$.

Remark 2. In discrete spaces, Schäfer & Chopin (2013) argue that the mutation rate

$$\bar{a}_{\mathrm{M}} = \int \mathbb{I}(\gamma \neq \gamma')\, a_\eta(\gamma, \gamma')\, q_\eta(\gamma, \gamma')\, \pi(\gamma)\, \mathrm{d}\gamma'\, \mathrm{d}\gamma,$$

which excludes moves which do not change the model, is more appropriate than the average acceptance rate. The mutation rate is $\bar{a}_{\mathrm{M}} = 1 - \prod_{j=1}^{p} \left\{(1 - \pi_j)^2 + \pi_j^2\right\}$ with independent sampling and $\bar{a}_{\mathrm{M}} = 1 - \prod_{j=1}^{p} |2\pi_j - 1|$ with the informed proposal. Therefore, the informed proposal always leads to a higher mutation rate.

Remark 3. Zanella (2020) discusses a framework for designing informed proposals in discrete spaces, and discusses optimal choices under Peskun ordering.

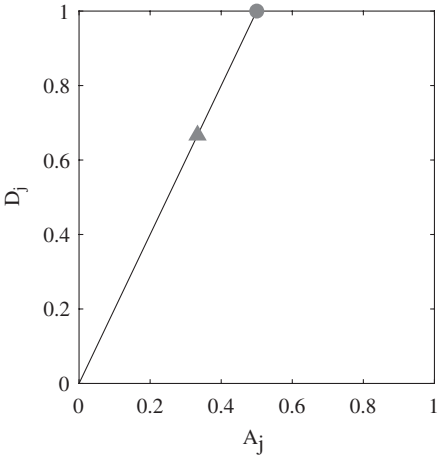

Fig. 1. The solid black line shows pairs (A_j, D_j) corresponding to $\pi = 1/3$ and $\zeta_j \in [0, 1]$ in (6). The independent proposal (i), marked with a triangle, is a shrunk version of the informed proposal (ii), marked with a bullet.

These results suggest that the informed proposal should be preferred to the independent proposal when designing a Metropolis–Hastings sampler for idealized posteriors of the form in (3). In practice, the posterior distribution will not have a product form, but can anything be said about its form when p is large? The following result sheds some light on this issue. We define $\mathrm{BF}_j(\gamma_{-j})$ to be the Bayes factor of including the jth variable given the values of $\gamma_{-j} = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_p)$ and denote by γ_0 the vector γ without γ_j and γ_k.

PROPOSITION 2. *Let* $a = \frac{\mathrm{BF}_j(\gamma_k=1,\gamma_0)}{\mathrm{BF}_j(\gamma_k=0,\gamma_0)}$. *If (i)* $a \to 1$ *or (ii)* $a \to A < \infty$ *and* $\mathrm{BF}_j(\gamma_k = 0, \gamma_0)h \to 0$ *then* $p(\gamma_j = 1 \mid \gamma_k = 1, \gamma_0) \to p(\gamma_j = 1 \mid \gamma_k = 0, \gamma_0)$.

This result gives conditions under which γ_j and γ_k are approximately independent. Condition (ii) is interesting in large p settings: γ_j and γ_k are approximately independent if p is large, and so h is small, and $\mathrm{BF}_j(\gamma_k = 0, \gamma_0)$ is not large, i.e., the evidence in favour of including γ_j is not large. This will be the case for all variables apart from the most important. Although this result provides some reassurance, there will be some posterior correlation in many problems, and the informed proposal may propose changing too many variables, leading to low acceptance rates. This can be addressed by using a scaled proposal of the form

$$A_j = \zeta_j \min\left(1, \frac{\pi_j}{1-\pi_j}\right), \qquad D_j = \zeta_j \min\left(1, \frac{1-\pi_j}{\pi_j}\right). \tag{6}$$

The family of these proposals for $\zeta_j \in [0, 1]$ form a line segment for (A_j, D_j) between $(0, 0)$ and $\left\{\min\left(1, \frac{\pi_j}{1-\pi_j}\right), \min\left(1, \frac{1-\pi_j}{\pi_j}\right)\right\}$, which is illustrated in Fig. 1. The independent proposal corresponds to the point on this line where $\zeta_j = \max(\pi_j, 1 - \pi_j)$.

In the next section we devise adaptive Markov chain Monte Carlo algorithms to tune proposals of the form (2) so that the A_j and D_j lie approximately on this line. Larger values of ζ_j tend to lead to larger jumps, whereas smaller values of ζ_j tend to increase acceptance. These algorithms aim to find a point which balances this trade-off. We define two strategies for adapting η: exploratory individual adaptation and adaptively scaled individual adaptation.

With both forms of adaptation, we run independent parallel chains which share the same proposal parameters and refer to this as multiple-chain acceleration. Craiu et al. (2009) show empirically that running multiple independent Markov chains with the same adaptive parameters improves the rate of convergence of adaptive algorithms towards their target acceptance rate in the context of the classical adaptive Metropolis algorithm of Haario et al. (2001); see also Bornn et al. (2013). At this point, it is helpful to define some notation. Let $\eta^{(i)} = (A^{(i)}, D^{(i)})$ and $\gamma^{(i)}$ be the values of η and γ at the start of the ith iteration, and γ' be the subsequently proposed value. Let $a_i = a_{\eta^{(i)}}(\gamma^{(i)}, \gamma')$ be the acceptance probability at the ith iteration. We define, for $j = 1, \ldots, p$,

$$\gamma_j^{A(i)} = \begin{cases} 1 \text{ if } \gamma_j' \neq \gamma_j^{(i)} \text{ and } \gamma_j^{(i)} = 0, \\ 0 \text{ otherwise;} \end{cases} \qquad \gamma_j^{D(i)} = \begin{cases} 1 \text{ if } \gamma_j' \neq \gamma_j^{(i)} \text{ and } \gamma_j^{(i)} = 1, \\ 0 \text{ otherwise,} \end{cases}$$

and the map $\text{logit}_\epsilon : (\epsilon, 1-\epsilon) \to \mathbb{R}$ by $\text{logit}_\epsilon(x) = \log(x-\epsilon) - \log(1-x-\epsilon)$, where $0 \leqslant \epsilon \leqslant 1/2$. This reduces to the usual logit transform if $\epsilon = 0$.

2.3. *Remembrance of things past: exploratory individual adaptation*

The first adaptive strategy is a general purpose method that we term exploratory individual adaptation. It aims to find pairs (A_j, D_j) on the line segment defined by (4) which lead to good mixing. Proposals with larger values of A_j and D_j will tend to propose more changes to the included variables, but will also tend to reduce the average acceptance probability or mutation rate. The method introduces two tuning parameters, τ_{L} and τ_{U}. There are three types of updates for $A^{(i)}$ and $D^{(i)}$ which move towards the correct ratio A_j/D_j and then along the segment, where the slope of the segment depends on π_j. Unless otherwise stated, $A_j^{(i+1)} = A_j^{(i)}$ and $D_j^{(i+1)} = D_j^{(i)}$.

Both the expansion step and the shrinkage step change $A_j^{(i+1)}$, and $D_j^{(i+1)}$ for j in $\gamma^{A(i)}$ and $\gamma^{D(i)}$ to adjust the average squared jumping distance whilst maintaining that $A_j^{(i+1)}/D_j^{(i+1)} \approx A_j^{(i)}/D_j^{(i)}$. The expansion step is used if a promising move is proposed, when $a_i > \tau_{\text{U}}$, and sets $A_j^{(i+1)}$ and $D_j^{(i+1)}$ larger than $A_j^{(i)}$ and $D_j^{(i)}$, respectively. Similarly, the shrinkage step is used if an unpromising move has been proposed, when $a_i < \tau_{\text{L}}$, and $A_j^{(i+1)}$ and $D_j^{(i+1)}$ are set smaller than $A_j^{(i)}$ and $D_j^{(i)}$.

The correction step aims to increase the average acceptance rate by correcting the ratio between A and D. If $\tau_{\text{L}} < a_i < \tau_{\text{U}}$, we set $A_j^{(i+1)} > A_j^{(i)}$ and $D_j^{(i+1)} < D_j^{(i}$ if $\gamma_j^{D(i)} = 1$, and $A_j^{(i+1)} < A_j^{(i)}$ and $D_j^{(i+1)} > D_j^{(i)}$ if $\gamma_j^{A(i)} = 1$.
The following updates achieve these properties:

$$\text{logit}_\epsilon A_j^{(i+1)} = \text{logit}_\epsilon A_j^{(i)} + \phi_i \times \left[\gamma_j^{A(i)} d_i(\tau_{\text{U}}) + \gamma_j^{D(i)} d_i(\tau_{\text{L}}) - \gamma_j^{A(i)}\{1 - d_i(\tau_{\text{U}})\}\right], \quad (7)$$

$$\text{logit}_\epsilon D_j^{(i+1)} = \text{logit}_\epsilon D_j^{(i)} + \phi_i \times \left[\gamma_j^{D(i)} d_i(\tau_{\text{U}}) + \gamma_j^{A(i)} d_i(\tau_{\text{L}}) - \gamma_j^{D(i)}\{1 - d_i(\tau_{\text{U}})\}\right], \quad (8)$$

for $j = 1 \ldots, p$ where $d_i(\tau) = \mathbb{I}(a_i \geqslant \tau)$ and $\phi_i = O(i^{-\lambda})$ for some constant $1/2 < \lambda \leqslant 1$. The gradient fields of these updates are shown in the Supplementary Material. The transformation implies that $\epsilon < A_j^{(i)} < 1 - \epsilon$ and $\epsilon < D_j^{(i)} < 1 - \epsilon$, and we assume that $0 < \epsilon < 1/2$. It also implies diminishing adaptation, since the derivative of the inverse logit is bounded; see Lemma 2. Based on several simulation studies, we suggest taking $\tau_{\text{L}} = 0.01$ and $\tau_{\text{U}} = 0.1$. As discussed in § 2.2, targeting a low acceptance rate is often beneficial in irregular cases, so we expect this choice to be robust in real data applications. In all our simulations with this

parameter setting, the resulting mean acceptance rate was between 0.15 and 0.35, i.e., in the high efficiency region identified in Roberts et al. (1997). We also suggest an initial choice of parameters such that $A_j^{(1)}/D_j^{(1)} \approx h/(1-h)$ as this summarizes the prior information on $\pi_j/(1-\pi_j)$; in particular, $D_j^{(1)} \equiv 1$ and $A_j^{(1)} \equiv h$ often works well. The parameter ϵ controls the minimum and maximum values of A_i and D_i. In the large-p setting, $A_i \approx \epsilon$ for unimportant variables and the expected number of those unimportant variables proposed to be included at each iteration will be approximately $p\epsilon$, since the number of excluded, unimportant variables will be close to p. This expected value can be controlled by choosing $\epsilon = 0.1/p$. The exploratory individual adaptation algorithm is described in Algorithm 1, and we indicate its transition kernel at time i as $P^{\mathrm{EIA}}_{\eta^{(i)}}$.

Algorithm 1. Exploratory individual adaptation.

for $i = 1$ to $i = M$
  sample $\gamma' \sim q_{\eta^{(i)}}(\gamma^{(i)}, \cdot)$ and $U \sim U(0,1)$;
  if $U < a_{\eta^{(i)}}(\gamma^{(i)}, \gamma')$ then $\gamma^{(i+1)} := \gamma'$, else $\gamma^{(i+1)} := \gamma^{(i)}$;
  update $A^{(i+1)}$ using (7) and $D^{(i+1)}$ using (8);
endfor

2.4. *Remembrance of things past: adaptively scaled individual adaptation*

Algorithm 1 learns two parameters $A_j^{(i)}$ and $D_j^{(i)}$ for each variable and, we shall see, can slowly converge to optimal values if p is large. Alternatively, we could learn $\pi_1, \ldots, \pi_p$ from the chain to approximate the slope of the line defined by (4) and use the proposal (6) with the same scale parameter for all variables. We term this approach the adaptively scaled individual adaptation proposal. In particular, we use

$$A_j^{(i)} = \zeta^{(i)} \min\left\{1, \hat{\pi}_j^{(i)}/\left(1-\hat{\pi}_j^{(i)}\right)\right\} \quad \text{and} \quad D_j^{(i)} = \zeta^{(i)} \min\left\{1, \left(1-\hat{\pi}_j^{(i)}\right)/\hat{\pi}_j^{(i)}\right\}, \tag{9}$$

for $j = 1, \ldots, p$, where $0 < \zeta^{(i)} < 1$ is a tuning parameter and $\hat{\pi}_j^{(i)}$ is a Rao–Blackwellized estimate of the posterior inclusion probability of variable j at the ith iteration. Like Ghosh & Clyde (2011), we work with the Rao–Blackwellized estimate conditional on the model, marginalizing over α, β_γ and σ^2, in contrast to Guan & Stephens (2011) who condition on the model parameters. We assume that $V_\gamma = gI_{p_\gamma}$, where I_q is the $q \times q$ identity matrix. After N posterior samples, $\gamma^{(1)}, \ldots, \gamma^{(N)}$, the Rao–Blackwellised estimate of $\pi_j = p(\gamma_j = 1 \mid y)$ is

$$\hat{\pi}_j = \frac{1}{N}\sum_{k=1}^{N} \frac{\tilde{h}_j^{(k)}\,\mathrm{BF}_j\left(\gamma_{-j}^{(k)}\right)}{1-\tilde{h}_j^{(k)}+\tilde{h}_j^{(k)}\,\mathrm{BF}_j\left(\gamma_{-j}^{(k)}\right)}, \tag{10}$$

where $\tilde{h}_j^{(k)} = h$ if h is fixed or $\tilde{h}_j^{(k)} = \frac{\#\gamma_{-j}^{(k)}+1+a}{p+a+b}$ if $h \sim \mathrm{Be}(a,b)$. Let $Z_\gamma = [1_n\ X_\gamma]$, $\Lambda_\gamma = \begin{pmatrix} 0 & 0_{p_\gamma}^{\mathrm{T}} \\ 0_{p_\gamma} & V_\gamma^{-1} \end{pmatrix}$, $F = (Z_\gamma^{\mathrm{T}} Z_\gamma + \Lambda_\gamma)^{-1}$ and $A = y^{\mathrm{T}}y - y^{\mathrm{T}}Z_\gamma F Z_\gamma^{\mathrm{T}} y$. If $\gamma_j = 0$,

$$\mathrm{BF}_j(\gamma_{-j}) = d_j^{\uparrow\,-1/2} g^{-1/2} \left\{\frac{A - \frac{1}{d_j^{\uparrow}}(y^{\mathrm{T}}x_j - y^{\mathrm{T}}Z_\gamma F Z_\gamma^{\mathrm{T}} x_j)^2}{A}\right\}^{-n/2},$$

with $d_j^{\uparrow} = x_j^{\mathrm{T}} x_j + g^{-1} - (x_j^{\mathrm{T}} Z_\gamma) F (Z_\gamma^{\mathrm{T}} x_j)$. If $\gamma_j = 1$, we define z_j to be the ordered position of the included variables, $z_j = 1$ if j is the first included variable, etc.; then,

$$\mathrm{BF}_j(\gamma_{-j}) = d_j^{\downarrow -1/2} g^{-1/2} \left\{ \frac{A}{A + d_j^{\downarrow} (y^{\mathrm{T}} Z_\gamma F_{\cdot, z_j+1})^2} \right\}^{-n/2},$$

where $d_j^{\downarrow} = 1/F_{z_j+1, z_j+1}$. These results allow the contribution to the Rao–Blackwellized estimates for all values of j to be calculated in $O(p)$ operations at each iteration if the values of F and A, which are needed for calculating the marginal likelihood, are stored. Derivations are provided in the Supplementary Material. The value of $\zeta^{(i)}$ is updated using

$$\mathrm{logit}_\epsilon \zeta^{(i+1)} = \mathrm{logit}_\epsilon \zeta^{(i)} + \phi_i (a_i - \tau), \tag{11}$$

where τ is a targeted acceptance rate. We use $\epsilon = 0.1/p$ as in Algorithm 1. We shall see in Lemma 2 that adaptively scaled individual adaptation also satisfies diminishing adaptation by verifying that the Rao–Blackwellized estimate in (10) evolves at the rate $1/i$, and reiterating the argument about inverse logit derivatives. To avoid proposing to change no variable with high probability, we set $\zeta^{(i+1)} = 1/\Delta^{(i+1)}$ if $\zeta^{(i+1)} \Delta^{(i+1)} < 1$, where $\Delta^{(i+1)} = 2 \sum_{j=1}^{p} \min\left(\pi_j^{(i+1)}, 1 - \pi_j^{(i+1)}\right)$. This ensures that the algorithm will propose to change at least one variable with high probability. The adaptively scaled individual adaptation algorithm is described in Algorithm 2, and we indicate its transition kernel at time i as $P_{\eta^{(i)}}^{\mathrm{ASI}}$. We use $\kappa = 0.001$ to avoid the estimated probabilities becoming very small.

Algorithm 2. Adaptively scaled individual adaptation.

for $i = 1$ to $i = M$
  sample $\gamma' \sim q_{\eta^{(i)}}(\gamma^{(i)}, \cdot)$ and $U \sim U(0,1)$;
  if $U < a_{\eta^{(i)}}(\gamma^{(i)}, \gamma')$ then $\gamma^{(i+1)} := \gamma'$, else $\gamma^{(i+1)} := \gamma^{(i)}$;
  Update $\hat{\pi}_1^{(i+1)}, \ldots, \hat{\pi}_p^{(i+1)}$ as in (10) and set $\tilde{\pi}_j^{(i+1)} = \kappa + (1 - 2\kappa)\, \hat{\pi}_j^{(i+1)}$;
  Update $\zeta^{(i+1)}$ as in (11);
  Calculate $A_j^{(i+1)} = \zeta^{(i+1)} \min\left\{1, \tilde{\pi}_j^{(i+1)} / \left(1 - \tilde{\pi}_j^{(i+1)}\right)\right\}$ for $j = 1, \ldots, p$;
  Calculate $D_j^{(i+1)} = \zeta^{(i+1)} \min\left\{1, \left(1 - \tilde{\pi}_j^{(i+1)}\right) / \tilde{\pi}_j^{(i+1)}\right\}$ for $j = 1, \ldots, p$;
endfor

3. Ergodicity of the algorithms

Since adaptive Markov chain Monte Carlo algorithms violate the Markov condition, the standard and well-developed Markov chain theory cannot be used to establish ergodicity and we need to derive appropriate results for our algorithms. We verify the validity of our algorithms by establishing the conditions introduced in Roberts & Rosenthal (2007), namely simultaneous uniform ergodicity and diminishing adaptation.

The target posterior specified in § 2.1 on the model space Γ is

$$\pi_p(\gamma) = \pi_p(\gamma \mid y) \propto p(y \mid \gamma)\, p(\gamma), \tag{12}$$

with $p(y \mid \gamma)$ available analytically, and the vector of adaptive parameters at time i is

$$\eta^{(i)} = (A^{(i)}, D^{(i)}) \quad \in \quad [\epsilon, 1-\epsilon]^{2p} =: \Delta_\epsilon, \qquad \text{with } 0 < \epsilon < 1/2, \tag{13}$$

with the update strategies in Algorithm 1 or 2. The nonadaptive Markov kernel corresponding to a fixed choice of η is denoted as $P_\eta(\gamma, \cdot)$. Under the dynamics of either algorithm, for $S \subseteq \Gamma$ we have

$$\begin{aligned} P_\eta(\gamma, S) &= \mathbb{P}\big(\gamma^{(i+1)} \in S \mid \gamma^{(i)} = \gamma, \eta^{(i)} = \eta\big) \\ &= \sum_{\gamma' \in S} q_\eta(\gamma, \gamma') a_\eta(\gamma, \gamma') + \mathbb{I}(\gamma \in S) \sum_{\gamma' \in \Gamma} q_\eta(\gamma, \gamma') \big\{1 - a_\eta(\gamma, \gamma')\big\}. \end{aligned} \tag{14}$$

In the case of multiple-chain acceleration, where L copies of the chain are run, the model state space becomes the product space and the current state of the algorithm at time i is $\gamma^{\otimes L,(i)} = (\gamma^{1,(i)}, \ldots, \gamma^{L,(i)}) \in \Gamma^L$. The single-chain version corresponds to $L = 1$ and all results apply.

To assess ergodicity, we need to define the distribution of the adaptive algorithm at time i, and the associated total variation distance: for the lth copy of the chain $\{\gamma^{l,(i)}\}_{i=0}^{\infty}$ and $S \subseteq \Gamma$ define

$$\begin{aligned} \mathcal{L}^{l,(i)}\big\{(\gamma^l, \eta), S\big\} &:= \mathbb{P}\big(\gamma^{l,(i)} \in S \mid \gamma^{l,(0)} = \gamma^l, \eta^{(0)} = \eta\big); \\ T^l(\gamma^l, \eta, i) &:= \big\|\mathcal{L}^{l,(i)}\big\{(\gamma^l, \eta), \cdot\big\} - \pi_p(\cdot)\big\|_{\mathrm{TV}} = \sup_{S \in \Gamma} \big|\mathcal{L}^{l,(i)}\big\{(\gamma^l, \eta), S\big\} - \pi_p(S)\big|. \end{aligned}$$

We show that all the algorithms considered are ergodic, see (15), and satisfy a strong law of large numbers, see (16), i.e., for any starting point $\gamma^{\otimes L} \in \Gamma^L$ and any initial parameter value $\eta \in \Delta_\epsilon$, we have:

$$\lim_{i \to \infty} T^l(\gamma^l, \eta, i) = 0, \qquad \text{for any } l = 1, \ldots, L; \tag{15}$$

$$\frac{1}{L} \sum_{l=1}^{L} \frac{1}{k} \sum_{i=1}^{k} f(\gamma^{l,(i)}) \overset{k \to \infty}{\longrightarrow} \pi_p(f) \quad \text{almost surely, for any } f : \Gamma \to \mathbb{R}. \tag{16}$$

To this end we establish the following lemmas.

LEMMA 1 (Simultaneous uniform ergodicity). *The family of Markov chains defined by transition kernels P_η in (14), targeting $\pi_p(\gamma)$ in (12), is simultaneously uniformly ergodic for any $\epsilon > 0$ in (13), and so is the multiple chain acceleration version. That is, for any $\delta > 0$ there exists $N = N(\delta, \epsilon) \in \mathbb{N}$ such that, for any starting point $\gamma^{\otimes L} \in \Gamma^L$ and any parameter value $\eta \in \Delta_\epsilon$,*

$$\big\|P_\eta^N(\gamma^{\otimes L}, \cdot) - \pi_p^{\otimes L}(\cdot)\big\|_{\mathrm{TV}} \leqslant \delta.$$

LEMMA 2 (Diminishing adaptation). *Recall the constant $1/2 \leqslant \lambda \leqslant 1$ defining the adaptation rate $\phi_i = O(i^{-\lambda})$ in (7), (8) or (11), and the parameter $\kappa > 0$ in Algorithm 2. Then both algorithms, exploratory individual adaptation and adaptively scaled individual adaptation, satisfy*

diminishing adaptation. More precisely, their transition kernels satisfy

$$\sup_{\gamma\in\Gamma} \left\| P^{\bullet}_{\eta^{(i+1)}}(\gamma,\cdot) - P^{\bullet}_{\eta^{(i)}}(\gamma,\cdot) \right\| \leqslant Ci^{-\lambda}, \qquad \textit{for some } C < \infty, \tag{17}$$

where $\bullet$ *stands for exploratory or adaptively scaled individual adaptation.*

Simultaneous uniform ergodicity together with diminishing adaptation leads to the following theorem.

THEOREM 1 (Ergodicity and the strong law of large numbers). *Consider the target* $\pi_p(\gamma)$ *of* (12)*, the constants* $1/2 \leqslant \lambda \leqslant 1$ *and* $\epsilon > 0$ *defining respectively the adaptation rate* $\phi_i = O(i^{-\lambda})$ *and region in* (7)*,* (8) *or* (11)*, and the parameter* $\kappa > 0$ *in Algorithm 2. Then, ergodicity* (15) *and the strong law of large numbers* (16) *hold for each of the algorithms, exploratory individual adaptation and adaptively scaled individual adaptation, and their multiple-chain acceleration versions.*

Remark 4. Lemma 2 and Theorem 1 remain true with any $\lambda > 0$; however, $\lambda > 1$ results in finite adaptation (see, e.g., Roberts & Rosenthal, 2007), and $\lambda < 1/2$ is rarely used in practice for finite-sample stability concerns.

4. RESULTS

4.1. *Simulated data*

We consider the simulated data example of Yang et al. (2016). They assume that there are n observations and p regressors, and the data is generated from the model $y = X\beta^{\star} + e$ where $e \sim N(0, \sigma^2 I)$ for $\sigma^2 = 1$. The first 10 regression coefficients are nonzero, and we use

$$\beta^{\star} = \mathrm{SNR} \times \left(\frac{\sigma^2 \log p}{n}\right)^{1/2} (2, -3, 2, 2, -3, 3, -2, 3, -2, 3, 0, \ldots, 0)^{\mathrm{T}} \in \mathbb{R}^p.$$

The ith vector of regressors is generated as $x_i \sim N(0, \Sigma)$, where $\Sigma_{jk} = \rho^{|j-k|}$. In our examples we use the value $\rho = 0.6$, which represents a relatively large correlation between the regressors.

We are interested in the performance of the two adaptive algorithms relative to an add-delete-swap algorithm. The adaptive algorithms do not lead to Markov chains, and so the traditional estimator of the effective sample size based on autocorrelation is not applicable. We define the ratio of the relative time-standardized effective sample size of algorithm A versus algorithm B to be $r_{A,B} = (\mathrm{ESS}_A/t_A)/(\mathrm{ESS}_B/t_B)$, where ESS_A is the effective sample size for algorithm A. This is estimated by making 200 runs of each algorithm and calculating $\hat{r}_{A,B} = (s_B^2 t_B)/(s_A^2 t_A)$, where t_A and t_B are the median runtimes, and s_A^2 and s_B^2 are the sample variances of the posterior inclusion probabilities for algorithms A and B.

We use the prior in (1) with $V_\gamma = 9I$ and $h = 10/p$, implying a prior mean model size of 10. The posterior distribution changes substantially with the SNR and the size of the dataset. All 10 true nonzero coefficients are given posterior inclusion probabilities greater than 0.9 in the two high-SNR scenarios, SNR = 2 and SNR = 3, for each value of n and p, and no true nonzero coefficients are given posterior inclusion probabilities greater than 0.2 in the low-SNR scenario, SNR = 0.5, for each value of n and p. In the intermediate SNR scenario, SNR = 1, the numbers of true nonzero coefficients given posterior inclusion probabilities

Table 1. *Simulated data: median values of $\hat{r}_{A,B}$ for the posterior inclusion probabilities over all variables where B is the add-delete-swap Metropolis–Hastings algorithm and A is the exploratory or adaptively scaled individual adaptation algorithm*

		5 chains SNR				25 chains SNR			
(n,p)		0.5	1	2	3	0.5	1	2	3
(500, 500)	EIA	4.9	1.8	5.5	5.1	1.2	1.5	2.4	2.3
	ASI	1.7	21.3	31.8	7.5	2.0	36.0	42.7	12.6
(500, 5000)	EIA	8.7	2.2	718.0	81.5	7.1	2.9	2267.2	147.2
	ASI	29.9	126.9	2053.1	2271.3	53.5	353.3	12319.5	7612.3
(1000, 500)	EIA	5.9	16.3	7.7	4.2	1.6	80.7	4.4	1.8
	ASI	41.9	2.1	16.9	12.0	32.8	34.0	27.9	14.4
(1000, 5000)	EIA	2.2	2.2	9167.2	11.3	5.6	2.5	15960.7	199.8
	ASI	15.4	37.0	4423.1	30.8	54.9	53.4	11558.2	736.4

SNR, signal-to-noice ratio; EIA, exploratory individual adaptation algorithm; ASI, adaptively scaled individual adaptation algorithm.

greater than 0.9 are four and eight for $p = 500$, and three and zero for $p = 5000$. Generally, the results are consistent with our intuition that true nonzero regression coefficients should be detected with greater posterior probability for larger SNR, larger values of n and smaller values of p.

Table 1 shows the median relative time-standardized effective sample sizes for the exploratory individual adaptation and adaptively scaled individual adaptation algorithms with 5 or 25 multiple chains for different combinations of n, p and SNR. The median is taken across the estimated relative time-standardized effective sample sizes for all posterior inclusion probabilities.

The results show a wide variation in the relative performance of the adaptive algorithms and the add-delete-swap algorithm. As is common in work on Bayesian variable selection, see, e.g., Zanella & Roberts (2019), each result uses a single dataset and so the results have to be interpreted in a holistic way. Clearly, the adaptively scaled individual adaptation algorithm outperforms the exploratory individual adaptation algorithm for most settings with either 5 or 25 multiple chains. The performance of the exploratory individual adaptation and, especially, the adaptive scaled individual adaptation algorithm with 25 chains, is better than the corresponding performance with 5 chains for most cases. Concentrating on the results with the adaptively scaled individual adaptation algorithm, the largest increase in performance compared to the Metropolis–Hastings algorithm occurs with SNR = 2. In this case, there is three or four orders of magnitude improvement when $p = 5000$, and several orders of magnitude improvement for other SNRs with $p = 5000$. In smaller problems with $p = 500$ there are still substantial improvements in efficiency over the add-delete-swap Metropolis–Hastings sampler.

The superior performance of the adaptively scaled individual adaptation algorithm over the exploratory individual adaptation algorithm is due to the substantially faster convergence of the tuning parameters of the former algorithm to optimal values. Plotting posterior inclusion probabilities against A and D at the end of a run shows that, in most cases, the values of A_j are close to the corresponding posterior inclusion probabilities for both algorithms. However, the values of D_j are mostly close to 1 for adaptively scaled individual adaptation, but not for exploratory individual adaptation. If D_j is close to 1, then variable j is highly likely to be proposed to be removed if included in the model. This is consistent with the idealized super-efficient setting (ii) in Proposition 1 for $\pi_j < 0.5$, and leads to improved mixing rates for small π_j, since it allows

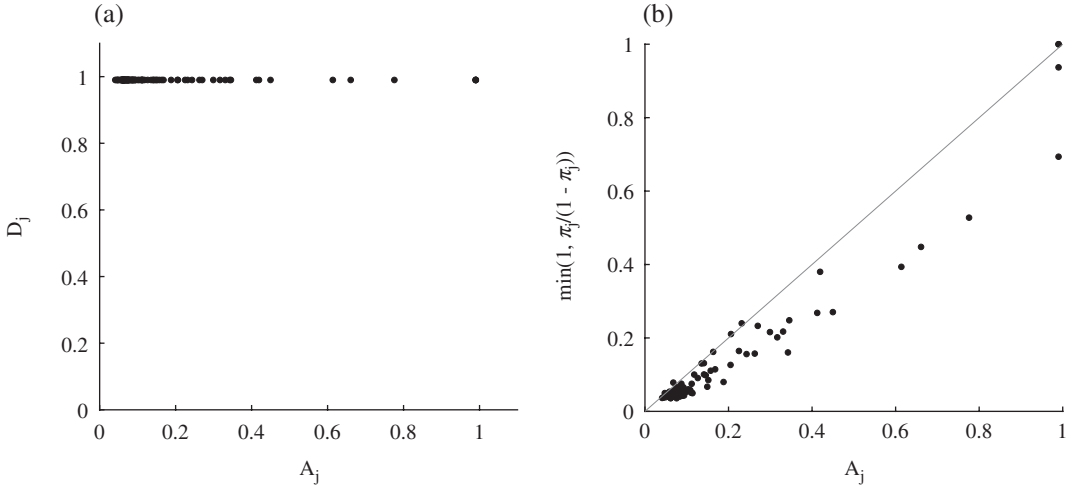

Fig. 2. Tecator data: the adaptive parameter $\eta = (A, D)$ for the exploratory individual adaptation algorithm. (a) Limiting values of the (A_j, D_j) pairs align at the top ends of the segments of Fig. 1, with the D_j close to 1, corresponding to the superefficient setting (ii) of Proposition 1; (b) The attained values of the A_j overestimate the idealized values $\min\{1, \pi_j/1 - \pi_j\}$ of setting (ii) in Proposition 1, indicating low dependence in the posterior.

for the variable to be included more often in a fixed run length. This can be hard to learn through exploratory individual adaptation since variables with low posterior inclusion probabilities will rarely be included and so the algorithm learns D_j slowly for those variables.

4.2. *Behaviour of the exploratory individual adaptation algorithm on the Tecator data*

The Tecator data contains 172 observations and 100 variables. They have previously been analysed using Bayesian linear regression techniques by Griffin & Brown (2010), who give a description of the data, and Lamnisos et al. (2013). The regressors show a high degree of multicollinearity, so this is a challenging example for Bayesian variable selection algorithms. The prior used was (1) with $V_\gamma = 100I$ and $h = 5/100$. Even short runs of the exploratory individual adaptation algorithm for this data, such as 5 multiple chains with 3000 burn in and 3000 recorded iterations, taking about 5 seconds on a laptop, show consistent convergence across runs.

Our purpose was to study the adaptive behaviour of the exploratory individual adaptation algorithm on this real data example, in particular to compare the idealized values of the A_j and D_j with the values attained by the algorithm.

We use multiple chain acceleration with 50 multiple chains over the total of 6000 iterations. The algorithm parameters were set to $\tau_L = 0.01$ and $\tau_U = 0.1$. The resulting mean acceptance rate was approximately 0.2, indicating close to optimal efficiency. The average number of variables proposed to be changed in a single accepted proposal was 23, approximately twice the average model size, meaning that in a typical move all of the current variables were deleted from the model, and a set of completely fresh variables was proposed.

Figure 2(a) shows how the exploratory individual adaptation algorithm approximates setting (ii) of Proposition 1, namely the super-efficient sampling from the idealized posterior (3). Figure 2(b) illustrates how the attained values of A_j somewhat overestimate the idealized values $\min\{1, \pi_j/(1 - \pi_j)\}$ of setting (ii) in Proposition 1. This indicates that the chosen parameter values

$\tau_L = 0.01$ and $\tau_U = 0.1$ of the algorithm overcompensates for dependence in the posterior, which is not very pronounced for this dataset. To quantify the performance, we ran both algorithms with adaptation in the burn-in only and calculated the effective sample size. With a burn-in of 10 000 iterations and 30 000 draws, the effective sample per multiple chain was 4015 with exploratory individual adaptation and 6673 with adaptively scaled individual adaptation. This is an impressive performance for both algorithms given the multicollinearity in the regressors. The difference in performance can be explained by the speed of convergence to optimal values for the proposal. To illustrate this, we reran the algorithms with the burn-in extended to 30 000 iterations: the effective sample per multiple chain was now 4503 with exploratory individual adaptation, but 6533 with adaptively scaled individual adaptation, indicating that the first algorithm had caught up somewhat. As a comparison, the effective sample size was 1555 for add-delete-swap and 15 039 for the Hamming ball sampler with a burn-in of 10 000 iterations. However, the Hamming ball sampler required 34 times the run time of the exploratory individual adaptation sampler, rendering the latter nine times more efficient in terms of time-standardized effective sample size.

This example and the previous one show that the simplified posterior (3) is a good fit with many datasets and can indeed be used to guide and design algorithms.

4.3. *Performance on problems with very large p*

Bondell & Reich (2012) described a variable selection problem with 22 576 variables and 60 observations on two inbred mouse populations. The covariates are gender and gene expression measurements for 22 575 genes. Three physiological phenotypes are recorded, and used as the response variable in the three datasets: PCRi, $i = 1, \ldots, 3$. We use prior (1) with $V_\gamma = gI$, where g is given a half-Cauchy hyperprior distribution, and a hierarchical prior was used for γ by assuming that $h \sim \text{Be}\left(1, \frac{p-5}{5}\right)$, which implies that the prior mean number of included variables is 5. We summarize the results using PCR1, while a more complete analysis of all PCR data is given in the Supplementary Material.

Another dataset, denoted SNP data, relates to genome-wide mapping of a complex trait. The data are weight measurements for 993 outbred mice and 79 748 single nucleotide polymorphisms, SNPs, recorded for each mouse. The testis weight is the response, the body weight is a regressor which is always included in the model and variable selection is performed on the 79 748 SNPs. The high dimensionality makes this a difficult problem and Carbonetto et al. (2017) use a variational inference algorithm, varbvs, for these data. We have used various prior specifications in (1), and present results for a half-Cauchy hyperprior on g and $h = 5/p$. Complete results for these data are also provided in the Supplementary Material.

For all datasets, the individual adaptation algorithms were run with $\tau_L = 0.05$ and $\tau_U = 0.23$, and $\tau = 0.234$. The exploratory individual adaptation algorithm had a burn-in of 2150 iterations and 10 750 subsequent iterations and no thinning, and the adaptively scaled individual adaptation had 500 burn-in and 2500 recorded iterations and no thinning, which gave very similar run times. Rao–Blackwellised updates of $\pi^{(i)}$ were only calculated during the burn-in, and posterior inclusion probability for the jth variable was estimated by the posterior mean of γ_j. In addition, we show results for the add-delete-swap algorithm and the weighted tempered Gibbs sampler of Zanella & Roberts (2019), which were the most promising alternatives. Three independent runs of all algorithms were executed to gauge the degree of agreement across runs. Using MATLAB and an Intel i7 3.60 GHz processor, each algorithm took about 25 minutes for the PCR data and around 2.5 hours for the SNP data.

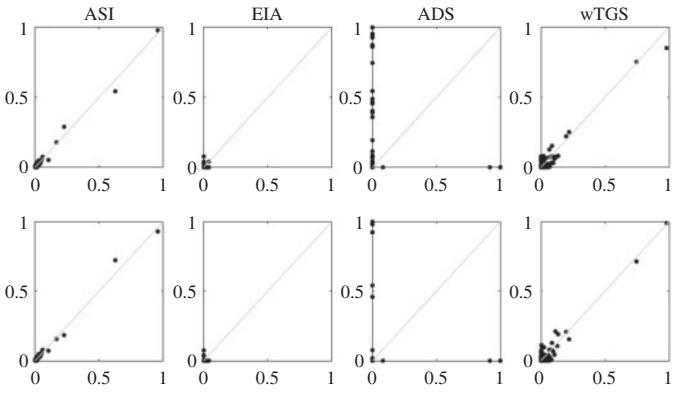

Fig. 3. PCR1 data: comparisons of posterior inclusion probabilities from pairs of runs with random g and h using adaptively scaled individual adaptation, ASI, exploratory individual adaptation, EIA, add-delete-swap, ADS, and weighted tempered Gibbs sampling, wTGS.

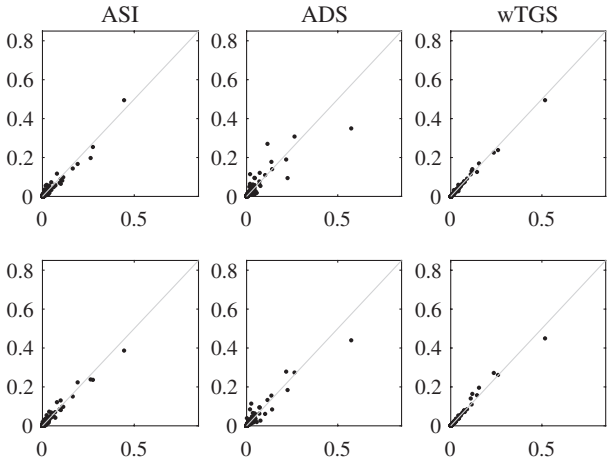

Fig. 4. SNP data: comparisons of posterior inclusion probabilities from pairs of runs with random g and fixed h using adaptively scaled individual adaptation, ASI, add-delete-swap, ADS, and weighted tempered Gibbs sampling, wTGS.

Figures 3 and 4 show pairwise comparisons between the different runs for each dataset. The estimates from each independent chain for the adaptively scaled individual adaptation algorithm are very similar, and indicate that the sampler is able to accurately represent the posterior distribution. The weighted tempered Gibbs algorithm performs equally well for the SNP data, but shows worse performance on the PCR1 dataset. The exploratory individual adaptation algorithm does not seem to converge rapidly enough to effectively deal with these very high-dimensional model spaces in the relatively modest running time allocated. Clearly, the add-delete–wap sampler is not able to adequately characterize the posterior model distribution for the PCR data, with dramatically different results across runs, but performs much better for the SNP data.

Acknowledgement

K.Ł. acknowledges support from the Royal Society and the Engineering and Physical Sciences Research Council. The authors thank two referees and an associate editor for their insightful comments that helped improve the paper.

SUPPLEMENTARY MATERIAL

Supplementary material available at *Biometrika* online includes proofs, derivations, details of adaptive parallel tempering versions of the algorithms, using the approach of Miasojedow et al. (2013), and further results. Code to run both algorithms is available from `https://jimegriffin.github.io/website/`.

REFERENCES

ANDRIEU, C. & THOMS, J. (2008). A tutorial on adaptive MCMC. *Statist. Comp.* **18**, 343–73.

BHATTACHARYA, A., CHAKRABORTY, A. & MALLICK, B. K. (2016). Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika* **4**, 985–91.

BONDELL, H. D. & REICH, B. J. (2012). Consistent high-dimensional variable selection via penalized credible regions. *J. Am. Statist. Assoc.* **107**, 1610–24.

BORNN, L., JACOB, P. E., DEL MORAL, P. & DOUCET, A. (2013). An adaptive interacting Wang–Landau algorithm for automatic density exploration. *J. Comp. Graph. Statist.* **22**, 749–73.

BROWN, P. J., VANNUCCI, M. & FEARN, T. (1998). Multivariate Bayesian variable selection and prediction. *J. R. Statist. Soc.* B **60**, 627–41.

CARBONETTO, P., ZHOU, X. & STEPHENS, M. (2017). varbvs: Fast variable selection for large-scale regression. *arXiv:* 1709.06597.

CASTILLO, I., SCHMIDT-HIEBER, J. & VAN DER VAART, A. (2015). Bayesian linear regression with sparse priors. *Ann. Statist.* **43**, 1986–2018.

CHIPMAN, H., GEORGE, E. I. & MCCULLOCH, R. E. (2001). The practical implementation of Bayesian model selection. In *Model Selection*, P. Lahiri, ed. pp. 65–116. Beachwood, OH: Institute of Mathematical Statistics.

CLYDE, M. A., GHOSH, J. & LITTMAN, M. L. (2011). Bayesian adaptive sampling for variable selection and model averaging. *J. Comp. Graph. Statist.* **20**, 80–101.

CRAIU, R. V., ROSENTHAL, J. & YANG, C. (2009). Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *J. Am. Statist. Assoc.* **104**, 1454–66.

GARCÍA-DONATO, G. & MARTÍNEZ-BENEITO, M. A. (2013). On sampling strategies for Bayesian variable selection problems with large model spaces. *J. Am. Statist. Assoc.* **108**, 340–52.

GELMAN, A., ROBERTS, G. O. & GILKS, W. R. (1996). Efficient Metropolis jumping rules. In *Bayesian Statistics 5*, J. M. Bernardo, J. O. Berger, A. P. David & F. M. Smith, eds. pp. 599–607. Oxford University Press.

GEORGE, E. I. & MCCULLOCH, R. E. (1997). Approaches for Bayesian variable selection. *Statist. Sinica* **7**, 339–73.

GHOSH, J. & CLYDE, M. A. (2011). Rao–Blackwellisation for Bayesian variable selection and model averaging in linear and binary regression: A novel data augmentation approach. *J. Am. Statist. Assoc.* **106**, 1041–52.

GREEN, P. J., ŁATUSZYŃSKI, K., PEREYRA, M. & ROBERT, C. P. (2015). Bayesian computation: A summary of the current state, and samples backwards and forwards. *Statist. Comp.* **25**, 835–62.

GRIFFIN, J. E. & BROWN, P. J. (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Anal.* **5**, 171–88.

GUAN, Y. & STEPHENS, M. (2011). Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *Ann. Appl. Statist.* **5**, 1780–815.

HAARIO, H., SAKSMAN, E. & TAMMINEN, J. (2001). An adaptive Metropolis algorithm. *Bernoulli* **7**, 223–42.

HAHN, P. R. & CARVALHO, C. M. (2015). Decoupling shrinkage and selection in Bayesian linear models: A posterior summary perspective. *J. Am. Statist. Assoc.* **110**, 435–48.

HANS, C., DOBRA, A. & WEST, M. (2007). Shotgun stochastic search for 'large p' regression. *J. Am. Statist. Assoc.* **102**, 507–16.

HASTIE, T., TIBSHIRANI, R. & WAINWRIGHT, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. London: Chapman & Hall / CRC.

JI, C. & SCHMIDLER, S. C. (2013). Adaptive Markov chain Monte Carlo for Bayesian variable selection. *J. Comp. Graph. Statist.* **22**, 708–28.

JOHNSON, V. E. & ROSSELL, D. (2012). Bayesian model selection in high-dimensional settings. *J. Am. Statist. Assoc.* **107**, 649–60.

LAMNISOS, D. S., GRIFFIN, J. E. & STEEL, M. F. J. (2013). Adaptive Monte Carlo for Bayesian variable selection in regression models. *J. Comp. Graph. Statist.* **22**, 729–48.

LEE, C. & NEAL, P. J. (2018). Optimal scaling of the independence sampler: Theory and practice. *Bernoulli* **24**, 1636–52.

LIANG, F., LIU, C. & CARROLL, R. J. (2007). Stochastic approximation in Monte Carlo computation. *J. Am. Statist. Assoc.* **102**, 305–20.

LIANG, F., SONG, Q. & YU, K. (2013). Bayesian subset modeling for high-dimensional generalized linear models. *J. Am. Statist. Assoc.* **108**, 589–606.

Miasojedow, B., Moulines, E. & Vihola, M. (2013). An adaptive parallel tempering algorithm. *J. Comp. Graph. Statist.* **22**, 649–64.

Neal, P., Roberts, G. & Yuen, W. K. (2012). Optimal scaling of random walk Metropolis algorithms with discontinuous target densities. *Ann. Appl. Prob.* **22**, 1880–927.

Nikooienejad, A., Wang, W. & Johnson, V. E. (2016). Bayesian variable selection for binary outcomes in high-dimensional genomic studies using non-local priors. *Bioinformatics* **32**, 1338–45.

Nott, D. J. & Kohn, R. (2005). Adaptive sampling for Bayesian variable selection. *Biometrika* **92**, 747–63.

O'Hara, R. B. & Sillanpää, M. J. (2009). A review of Bayesian variable selection methods: What, how and which. *Bayesian Anal.* **4**, 85–117.

Papaspiliopoulos, O. & Rossell, D. (2017). Bayesian block-diagonal variable selection and model averaging. *Biometrika* **104**, 343–59.

Peskun, P. H. (1973). Optimum Monte Carlo sampling using Markov chains. *Biometrika* **60**, 607–12.

Richardson, S., Bottolo, L. & Rosenthal, J. S. (2010). Bayesian models for sparse regression analysis of high-dimensional data. *Bayesian Statist.* **9**, 539–68.

Roberts, G. O. (1998). Optimal Metropolis algorithms for product measures on the vertices of a hypercube. *Stochast. Stochast. Rep.* **62**, 275–83.

Roberts, G. O., Gelman, A. & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Prob.* **7**, 110–20.

Roberts, G. O. & Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Prob. Surv.* **1**, 20–71.

Roberts, G. O. & Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Prob.* **44**, 458–75.

Roberts, G. O. & Rosenthal, J. S. (2009). Examples of adaptive MCMC. *J. Comp. Graph. Statist.* **18**, 349–67.

Rockova, V. & George, E. I. (2014). EMVS: The EM approach to Bayesian variable selection. *J. Am. Statist. Assoc.* **109**, 828–46.

Schäfer, C. & Chopin, N. (2013). Sequential Monte Carlo on large binary sampling spaces. *Statist. Comp.* **23**, 163–84.

Shin, M., Bhattacharya, A. & Johnson, V. E. (2018). Scalable Bayesian variable selection using nonlocal prior densities in ultrahigh-dimensional settings. *Statist. Sinica* **28**, 1053–78.

Titsias, M. K. & Yau, C. (2017). The Hamming ball sampler. *J. Am. Statist. Assoc.* **112**, 1598–611.

Yang, Y., Wainwright, M. & Jordan, M. I. (2016). On the computational complexity of high-dimensional Bayesian variable selection. *Ann. Statist.* **44**, 2497–532.

Zanella, G. (2020). Informed proposals for local MCMC in discrete spaces. *J. Am. Statist. Assoc.* **115**, 852–65.

Zanella, G. & Roberts, G. O. (2019). Scalable importance tempering and Bayesian variable selection. *J. R. Statist. Soc.* B **81**, 489–517.

[*Received on* 3 *May* 2019. *Editorial decision on* 20 *April* 2020]