Individual adaptation: an adaptive MCMC scheme for variable selection problems

J. E. Griffin, K. Łatuszyński and M. F. J. Steel*

Abstract

The increasing size of data sets has lead to variable selection in regression becoming increasingly important. Bayesian approaches are attractive since they allow uncertainty about the choice of variables to be formally included in the analysis. The application of fully Bayesian variable selection methods to large data sets is computationally challenging. We describe an adaptive Markov chain Monte Carlo approach called Individual Adaptation which adjusts a general proposal to the data. We show that the algorithm is ergodic and discuss its use within parallel tempering and sequential Monte Carlo approaches. We illustrate the use of the method on two data sets including a gene expression analysis with 22 577 variables.

Keywords: Bayesian variable selection; spike-and-slab priors; high-dimensional data; large $p$, small $n$ problems; linear regression

1 Introduction

The problem of choosing a subset of potential variables to include in a linear model is an important, and well-studied, problem in statistics. Let $y$ be an $(n \times 1)$-dimensional vector of responses and $X$ be an $(n \times p)$-dimensional data matrix. The indicator variable $\gamma_i$ denotes whether the $i$-th variable is included in the model (when $\gamma_i = 1$)
and we define \( p_\gamma = \sum_{j=1}^{p} \gamma_j \). The linear regression model is

\[
y = \alpha \mathbf{1} + X_\gamma \beta_\gamma + \epsilon
\]

where \( \mathbf{1} \) is an \((n \times 1)\)-dimensional vector of 1’s, \( X_\gamma \) is the sub-matrix of \( X \) where the \( i \)-th column is included if \( \gamma_i = 1 \), \( \beta_\gamma \) is a \((p_\gamma \times 1)\)-dimensional vector and \( \epsilon \sim N(0, \sigma^2 I_n) \).

It will be useful to define the notation \( \theta_\gamma = (\alpha, \beta_\gamma) \).

Bayesian methods are attractive for the variable selection problem since they can formally incorporate uncertainty about the form of the model and provide Bayesian model averaged (BMA) estimates of common parameters and predictions. These can be substantially more accurate than those from a single model. A prior distribution is placed on the parameters \( \theta_\gamma \) and \( \sigma^2 \) jointly with the model \( \gamma \). The most commonly used prior structure is

\[
p(\alpha, \sigma^2, \beta_\gamma, \gamma) \propto \sigma^{-2} p(\beta_\gamma | \sigma^2, \gamma) p(\gamma)
\]

with \( \beta_\gamma | \sigma^2, \gamma \sim N(0, \sigma^2 V_\gamma) \) 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 \( V_\gamma \) is often chosen as proportional to \((X_\gamma^T X_\gamma)^{-1}\) (a g-prior) or to an identity matrix (implying conditional prior independence between the regression coefficients). The use of these methods extends beyond regression problems and underlies Bayesian approaches to many problems, such as flexible curve and surface estimation.

Posterior inference is challenging since the number of models \((2^p)\) is very large if \( p \) is not small and the posterior distribution may be highly multi-modal. Interest normally centres around low-dimensional summaries such as posterior inclusion probabilities (PIP’s) or predictive distributions for future observations. There is a large literature on computational strategies for model uncertainty problems and, particularly, regression models, see e.g. George and McCulloch (1997); Dellaportas et al. (2002); O’Hara and Sillanpää (2009); Bottolo and Richardson (2010); Clyde et al. (2011) and references therein. There are two main computational approaches: Markov chain Monte Carlo (MCMC) sampling and heuristic search methods aiming to find the highest posterior probability models. García-Donato and Martínez-Beneito (2013) provide an interesting comparison of these two methods which they term empirical and renormalization respectively. They show that the renormalization method is prone to biased estimates of posterior probabilities whereas the MCMC method can provide consistent estimates. Successful estimation using the empirical method depends on having a representative sample from the posterior
distribution. This is challenging since the model space is large and the posterior distribution is potentially multi-modal. Many MCMC schemes have been proposed for this model (see e.g. García-Donato and Martinez-Beneito, 2013) but these increasingly struggle to provide representative samples as $p$ becomes larger. The difficulty of sampling from the posterior distribution is a particular problem with large numbers of covariates which is becoming increasingly common in many applications (with $p$ in the tens of thousands).

The complexity of the posterior distribution has lead to interest in methods where the computational algorithm adapts to the data. For example, Kwon et al. (2011) consider building transition probabilities using the correlation matrix of the regressors. Alternatively, the algorithm can be adapted during the run. Nött and Kohn (2005) developed a Gibbs sampling algorithm which allows the algorithm to adapt to the marginal inclusion probabilities (the posterior probability that a variable is included in the model). Richardson et al. (2010) focus on high dimensional sparse multi-response regression models, that are central to genomics, and develop an adaptive Gibbs sampler for identifying hot spots in this context. Lamnisos et al. (2013) construct a tuneable proposal distribution in a Metropolis-Hastings algorithm and describe an adaptive algorithm which tunes this parameter to achieve a pre-specified average acceptance rate. Ji and Schmidler (2013) use a mixture distribution for the proposal kernel and adapt its parameters to minimize the Kullback-Leibler divergence from the target distribution. The problem of multi-modality can be addressed using standard computational techniques such as parallel tempering or sequential Monte Carlo samplers (Schäfer and Chopin, 2013 with application to variable selection) which use powered versions of the posterior distribution.

This paper describes a flexible adaptive Metropolis-Hastings algorithm that is cheap to implement per iteration and is able to efficiently traverse the model space. This leads to substantially more efficient algorithms than commonly-used methods. The adaptation step relies on the optimal acceptance rate criterion (Roberts et al. 1997, Roberts 1998, Roberts and Rosenthal 2001). The adaptation parameter is a vector of length $2p$ which allows the deletion and addition of each variable conditional on the current model to be optimised individually. This flexibility allows the variables included in the model to change quickly and leads to substantial improvements in mixing. Each individual adaptation step is cheap as the marginal likelihood is calculated using a fraction of the variables which has the same order as the typical $a posteriori$ model size. We also show how this adaptive kernel can be used as a building block for interchain adaptation, parallel tempering and sequential Monte Carlo schemes in more challenging multi-modal problems. We also verify its ergodicity.
under some typical regularity assumptions.

The paper is organised as follows: Section 2 introduces a new adaptive kernel for variable selection which we term “individual adaptation”, Section 3 discusses some methods for accelerating the convergence of the algorithm to the target acceptance probability. Section 4 considers their use as a building block in more complex algorithms for exploring posteriors with well-separated modes. Ergodicity of the algorithms is discussed in Section 5. Section 6 presents the application of the methods to datasets with \( p = 100 \) and \( p = 22,576 \) possible covariates, and Section 7 concludes. Supplementary material includes proofs of the ergodicity of the algorithms and a further example using sequential Monte Carlo and parallel tempering methods. Matlab code is available from http://www.kent.ac.uk/smsas/personal/jeg28/index.htm.

2 The individual adaptation algorithm

We will consider inference in Bayesian variable selection with a linear regression model and conjugate prior as in (1) using a Metropolis-Hastings sampler. In this case, the marginal likelihood \( p(y|\gamma) \) can be calculated analytically and a sampler can be directly run on \( \gamma \).

We define a very general proposal on model space with parameters \( A = (A_1, \ldots, A_p) \), \( D = (D_1, \ldots, D_p) \) with \( 0 < A_j, D_j < 1 \) and \( \eta = (A, D) \). A new model, \( \gamma' \), is proposed independently, conditional on \( \gamma \), according to the transition density

\[
q_\eta(\gamma, \gamma') = p(\gamma'|\gamma) = \prod_{j=1}^{p} p(\gamma'_j|\gamma_j) = \prod_{j=1}^{p} q_{\eta,j}(\gamma_j, \gamma'_j)
\]

where \( q_{\eta,j}(\gamma_j = 0, \gamma'_j = 1) = A_j \), \( q_{\eta,j}(\gamma_j = 0, \gamma'_j = 0) = 1 - A_j \), \( q_{\eta,j}(\gamma_j = 1, \gamma'_j = 0) = D_j \), and \( q_{\eta,j}(\gamma_j = 1, \gamma'_j = 1) = 1 - D_j \). The values of \( \gamma'_1, \ldots, \gamma'_p \) are conditionally independent and so can be quickly sampled. The tuning parameter \( A_j \) is the probability that the \( j \)-th variable is added to the model (if it is currently excluded) and \( D_j \) is the probability that the \( j \)-th variable is deleted from the model (if it is currently included). The proposed model is accepted using the standard Metropolis-Hastings acceptance probability

\[
a_\eta(\gamma, \gamma') = \min \left\{ 1, \frac{p(y|\gamma')p(\gamma')q_\eta(\gamma', \gamma)}{p(y|\gamma)p(\gamma)q_\eta(\gamma, \gamma')} \right\}.
\]

The proposal allows multiple variables to be added or deleted from the model and, consequently, we do not need separate add, remove or swap moves as in the stan-
standard multi-move proposal (Brown et al., 1998). If the number of additions and deletions is different, the model size will be proposed to change. The expected proposed change in the model size, given $\gamma$, is 

$$\sum_{i=1}^{p} I(\gamma_j = 0) A_j - \sum_{i=1}^{p} I(\gamma_j = 1) D_j$$

and the total number of variables proposed to be changed is 

$$\sum_{i=1}^{p} I(\gamma_j = 0) A_j + \sum_{i=1}^{p} I(\gamma_j = 1) D_j$$

Unconditionally, these equal 

$$\sum_{i=1}^{p} p(\gamma_j = 0) A_j - \sum_{i=1}^{p} p(\gamma_j = 1) D_j$$

and 

$$\sum_{i=1}^{p} p(\gamma_j = 0) A_j + \sum_{i=1}^{p} p(\gamma_j = 1) D_j$$

respectively. Therefore, smaller values of $A_j$ and $D_j$ will tend to lead to smaller changes in the model. However, the effect on proposed model size of changing an individual $A_j$ or $D_j$ depends on the posterior inclusion probability (PIP) for the $j$-th variable. The value of $A_j$ will only have a large effect on the average size of change if $p(\gamma_j = 0 | y)$ is large. The proposal is more general than the one proposed by Lamnisos et al. (2009) and is easily extended to allow the probabilities of adding or deleting each variable from the model to change over the run of the sampler.

Working with this proposal seems, at first, problematic since there are $2p$ tuning parameters $A$ and $D$ which must be specified at the start of the algorithm and we have little guidance on their choice. Our solution is to follow the idea of Lamnisos et al. (2013) and choose values of these tuning parameters which give a prespecified acceptance rate by adapting these tuning parameters during the MCMC run. Schäfer and Chopin (2013) note that the usual form of average acceptance rate for Metropolis-Hastings samplers is not appropriate for the variable selection problem (or other problems on discrete spaces) where moves which do not change the model (i.e. $\gamma$ and $\gamma'$ are the same) have positive probability. These have an acceptance probability of 1 but do not help mixing since the model does not change. They suggest using instead the mutation rate which is defined to be

$$\bar{a}_M = \int C(\gamma, \gamma') a_\eta(\gamma, \gamma') q_\eta(\gamma, \gamma') p(\gamma | y) d\gamma' d\gamma,$$

where $C(\gamma, \gamma') = 0$ if $\gamma_j' = \gamma_j$ for all $j$ and 1 otherwise.

The individual adaptation (IA) algorithm targets a particular value, $\tau$, of the mutation rate. Let $\gamma^{(i)}$ be the value of $\gamma$ at the start of the $i$-th iteration, $\gamma'$ be the subsequently proposed value and $\eta^{(i)} = (A^{(i)}, D^{(i)})$ be the value of the tuning parameters used at the $i$-th iteration. We define for $j = 1, \ldots, p$

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

$$\gamma_j^{D(i)} = \begin{cases} 
1 & \text{if } \gamma_j' \neq \gamma_j^{(i)} \text{ and } \gamma_j^{(i)} = 1 \\
0 & \text{otherwise}
\end{cases}$$
The values of $A^{(i)}$ and $D^{(i)}$ are adapted using for $j = 1 \ldots, p$

\[
\log \left( \frac{A_j^{(i+1)} - \epsilon}{1 - A_j^{(i+1)} - \epsilon} \right) = \log \left( \frac{A_j^{(i)} - \epsilon}{1 - A_j^{(i)} - \epsilon} \right) + \phi_i \gamma_j^{A^{(i)}} \left( a_{\eta^{(i)}} \left( \gamma^{(i)}, \gamma^{'} \right) - \tau \right) \tag{3}
\]

and

\[
\log \left( \frac{D_j^{(i+1)} - \epsilon}{1 - D_j^{(i+1)} - \epsilon} \right) = \log \left( \frac{D_j^{(i)} - \epsilon}{1 - D_j^{(i)} - \epsilon} \right) + \phi_i \gamma_j^{D^{(i)}} \left( a_{\eta^{(i)}} \left( \gamma^{(i)}, \gamma^{'} \right) - \tau \right) \tag{4}
\]

where $0 < \epsilon < 1/2$ and $\epsilon$ is small, $\phi_i = O(i^{-\lambda})$ for some constant $1/2 < \lambda \leq 1$ and $a_{\eta^{(i)}} \left( \gamma^{(i)}, \gamma^{'} \right)$ represents the acceptance probability at the $i$-th iteration. The transformation implies that $\epsilon < A_j^{(i)} < 1 - \epsilon$ and $\epsilon < D_j^{(i)} < 1 - \epsilon$ and the algorithm targets an average mutation rate of $\tau$ if that is attainable. Clearly, if the current acceptance probability exceeds $\tau$, $A_j$ for the currently excluded variables will be increased, as well as $D_j$ for the variables that are in the current model. This implies larger proposed model changes, so will tend to decrease the mutation rate.

The starting values of $A$ and $D$ can have a considerable effect on the convergence of the tuning parameters towards values which have an average mutation rate of $\tau$. We have found that the following starting values work well in practice: $A_j^{(1)} = \nu/(1-h)p$ and $D_j^{(1)} = \nu/(hp)$, where $hp$ is the prior mean model size (see (1)). The range of values taken by $A_j^{(1)}$ and $D_j^{(1)}$ imply that $\epsilon(1-h)p < \nu < (1-\epsilon)hp$ if $h < 1/2$ (which will be true in large $p$ settings). If the initial value of $\gamma$ is generated from the prior, this choice of $A^{(1)}$ and $D^{(1)}$ implies that the expected number of proposed changes from $\gamma^{(1)}$ is $2\nu$. We have used the value $\nu = 1$ in our examples and found that the performance of the algorithm is robust to choices in the range 0.25 to 4 in Example 1.

The efficiency of the algorithm with respect to the choice of $\tau$ has been empirically studied in Example 1 and appears not to be very sensitive as long as $\tau$ is not too close to 0 or 1. This confirms other empirical and theoretical studies on scaling and in particular we note that for discrete state spaces the optimal $\tau$ will depend on the problem, see e.g. Figure 3 on page 282 of Roberts (1998). The accelerated versions of the algorithm described below (RAPA and MCA) appear even more robust to the choice of $\tau$. 

6
3 Accelerated individual adaptation algorithms

The convergence of $A$ and $D$ can be slow if $p$ is large. This does not affect the ergodicity of the adaptive chain but it can affect the mixing of the chain in MCMC runs of practically sensible length. Therefore, we consider two possible methods for accelerating the algorithm. The first method uses $r$ independent MCMC chains but shares the proposal parameters across the chains which are updated after the iteration of each independent chain. Craiu et al. (2009) empirically show that a related approach 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). This will be referred to as multiple chain acceleration (MCA) (which differs from the parallel tempering methods described in Section 4).

A second approach uses the reverse acceptance probability acceleration (RAPA) method. The individual adaptation algorithm updates $A_j$ only if $\gamma_j^A(i) = 1$ or $D_j$ only if $\gamma_j^D(i) = 1$. This potentially wastes information since the Metropolis-Hastings algorithm considers a pair of models (the current and the proposed) and we only use the acceptance probability for moving from the current to the proposed. The Metropolis-Hastings acceptance ratio for the reverse move from proposed to current, $a_\eta(\gamma', \gamma)$, can also be calculated using the values needed to compute $a_\eta(\gamma, \gamma')$. To include the acceptance probability $a_\eta(\gamma', \gamma)$ in the update of $A$ and $D$, we need to keep the mutation rate targeting $\tau$ in the stochastic approximation algorithm. This is $\bar{a}_M$ in (2), which is just an expectation with respect to the posterior and the transition. A second chain $(\delta, \delta')$ can be constructed from $\gamma$ and $\gamma'$ in the following way

$$(\delta, \delta') = \begin{cases} (\gamma', \gamma) & \text{with probability } a_\eta(\gamma, \gamma') \\ (\gamma, \gamma') & \text{with probability } 1 - a_\eta(\gamma, \gamma') \end{cases}.$$

The stationary distribution of $\delta$ is the posterior distribution due to properties of the Metropolis-Hastings algorithm which implies that $p(\delta, \delta') = p(\delta | y) q_\eta(\delta, \delta')$. 

7
It follows that we can write (2) as
\[
\int C(\delta, \delta') a_\eta(\delta, \delta') q_\eta(\delta, \delta') p(\delta | y) \, d\delta' \, d\delta \\
= E[C(\delta, \delta') a_\eta(\delta, \delta')]
\]
\[
= a_\eta(\gamma', \gamma') E[C(\gamma', \gamma) a_\eta(\gamma', \gamma)] + (1 - a_\eta(\gamma, \gamma')) E[C(\gamma, \gamma') a_\eta(\gamma, \gamma')].
\]

Taking a weighted average of this expression and \(E[C(\gamma, \gamma') a_\eta(\gamma, \gamma')]\) gives
\[
w a_\eta(\gamma, \gamma') E[C(\gamma', \gamma) a_\eta(\gamma', \gamma)] + (1 - w a_\eta(\gamma, \gamma')) E[C(\gamma, \gamma') a_\eta(\gamma, \gamma')].
\]

Therefore, and noticing that \(C(\gamma, \gamma') = C(\gamma', \gamma)\), an accelerated version of the adaptive algorithm (the \textbf{IA-RAPA} algorithm) uses the following updates:

1. \(\log \left( \frac{A_j^{(i+1)} - \epsilon}{1 - A_j^{(i+1)} - \epsilon} \right) = \log \left( \frac{A_j^{(i)} - \epsilon}{1 - A_j^{(i)} - \epsilon} \right) + \phi_i \gamma_j^{(i)} (a_\eta(y_i) (\gamma(i), \gamma(i)) - \tau) (1 - w a(y_i) (\gamma(i), \gamma(i)))
\]
2. \(\log \left( \frac{D_j^{(i+1)} - \epsilon}{1 - D_j^{(i+1)} - \epsilon} \right) = \log \left( \frac{D_j^{(i)} - \epsilon}{1 - D_j^{(i)} - \epsilon} \right) + \phi_i \gamma_j^{(i)} (a_\eta(y_i) (\gamma(i), \gamma(i)) - \tau) w a(y_i) (\gamma(i), \gamma(i))
\]
3. \(\log \left( \frac{A_j^{(i+1)} - \epsilon}{1 - A_j^{(i+1)} - \epsilon} \right) = \log \left( \frac{A_j^{(i)} - \epsilon}{1 - A_j^{(i)} - \epsilon} \right) + \phi_i \gamma_j^{(i)} (a_\eta(y_i) (\gamma(i), \gamma(i)) - \tau) (1 - w a(y_i) (\gamma(i), \gamma(i)))
\]
4. \(\log \left( \frac{D_j^{(i+1)} - \epsilon}{1 - D_j^{(i+1)} - \epsilon} \right) = \log \left( \frac{D_j^{(i)} - \epsilon}{1 - D_j^{(i)} - \epsilon} \right) + \phi_i \gamma_j^{(i)} (a_\eta(y_i) (\gamma(i), \gamma(i)) - \tau) w a(y_i) (\gamma(i), \gamma(i))
\]

Whereas \(w = 0\) corresponds to the standard IA algorithm, we will use \(w = 0.5\) for IA-RAPA in the applications below.

4 \textbf{Multi-modal posterior distributions}

The individual adaptation algorithm behaves like a Metropolis-Hastings random walk (albeit running on a very high-dimensional space). However, in common with all random walk samplers, the algorithm can become stuck in local modes if the modes are sufficiently well-separated. We will consider methods which use a sequence of annealed versions of the posterior distribution
\[
\pi_k(\gamma | y) \propto p(y | \gamma)^{t_k} \pi(\gamma), \quad k = 1, \ldots, m
\]
where the parameters \(0 < t_1 < t_2 < \cdots < t_m = 1\) are referred to as temperatures (with smaller \(t_j\) referring to higher temperatures). The density \(\pi_m(\gamma | y)\) is the posterior density \(p(\gamma | y)\) of interest. The density at other temperatures will be flatter than the posterior distribution and is more likely to allow for
moves between the local modes. Our adaptive algorithm is potentially well-suited to this approach since it can quickly explore the model space at high temperatures (the posterior raised to a power close to 0) and so rapidly move between local modes. We consider two implementations: a parallel tempering and a sequential Monte Carlo algorithm (Schäfer and Chopin 2013) which use a sequence of annealed versions of the posterior distribution.

The parallel tempering (PT) algorithm has long been used to improve convergence of MCMC algorithms for multi-modal posterior distributions. Its use in MCMC for Bayesian variable selection was first proposed by Jasra et al. (2007). The algorithm runs a chain at each temperature and proposes to swap the current value in two chains in such a way that the chains are drawn from the correct distribution. The idea is formalized by defining a joint target for $\gamma^* = (\gamma^*_1, \ldots, \gamma^*_m)$,

$$
\pi(\gamma^* | y) = \prod_{k=1}^{m} \pi_k(\gamma^*_k | y)
$$

where $\pi_k(\gamma^*_k | y) \propto p(y | \gamma^*_k)^{t_k} \pi(\gamma^*_k)$. An MCMC algorithm is run on the target $\pi(\gamma^* | y)$ with two types of moves. Firstly, an MCMC algorithm updates $\gamma^*_k$ for all values of $k$. Secondly, a Metropolis-Hastings algorithm is introduced which proposes to swap $\gamma^*_k$ with $\gamma^*_l$ where $k$ and $l$ are drawn from some distribution. In practice, the proposed value is often chosen by first drawing a value $k$ uniformly from $\{1, \ldots, m - 1\}$ and then choosing $l = k + 1$. This restricts the algorithm to swaps between chains at consecutive temperatures.

There are a number of drawbacks with this algorithm which can be addressed using adaptive ideas. Firstly, the temperature schedule $t_1, \ldots, t_{m-1}$ must be chosen. Recent work has suggested that the optimal choice of temperature schedule should maintain an acceptance rate of 0.234 for swaps between chains (Atchadé et al. 2011). An adaptive algorithm that exploits this idea is suggested by Miasojedow et al. (2013) and adopted in our algorithm. Secondly, the distribution for higher temperatures (smaller values of $t_j$) should be relatively flat to allow easier exploration. However, standard variable selection algorithms may move slowly across these targets since only one variable is changed in the model at each iteration. We use different tuning parameters for each chain and define $\eta_k$ to be the value of the tuning parameters for the $k$-th chain. The individual adaptation algorithm allows more than one variable to be changed at each iteration in any chain and so
should avoid the problem with standard variable selection algorithms. In summary, one iteration of the full individual adaptation-parallel tempering (IA-PT) algorithm is

- For $k = 1, \ldots, m$ do individual adaptation updating with $\pi_k$ as the target distribution and tuning parameters $\eta_k$.
- Choose $k$ uniformly from $\{1, \ldots, m-1\}$ and set $l = k+1$. Propose to swap $\gamma^{(k)}$ with $\gamma^{(l)}$ and accept the move with acceptance probability

$$\min \left\{ 1, \frac{p(y|\gamma^{(k)}_{t_k}) p(y|\gamma^{(l)}_{t_l})}{p(y|\gamma^{(k)}_{t_l}) p(y|\gamma^{(l)}_{t_k})} \right\}.$$  

- Let $\rho_{j-1}^{(h)} = t_j^{(h)} - t_{j-1}^{(h)}$, $j = 1, 2, \ldots, m-1$. These values are updated to

$$\rho_j^{(h+1)} = \begin{cases} 
\rho_j^{(h)} & \text{if } j = 1, \ldots, l-1, l+1, \ldots, m-1, \\
\rho_j^{(h)} + \zeta_h (a - \hat{a}) & \text{if } j = l
\end{cases}$$

where $\zeta_h$ is $O(h^{-\lambda})$ for some constant $1/2 < \lambda \leq 1$, $a$ is the Metropolis-Hastings acceptance probability and $\hat{a}$ is the target average acceptance probability for the parallel tempering moves. Finally, the temperatures are updated to $t_j^{(h+1)} = t_{j-1}^{(h+1)} + \rho_{j-1}^{(h+1)}$, $j = 1, 2, \ldots, m-1$.

As we discussed in Section 3, multiple chains can lead to faster convergence of the proposal parameters. A multiple chain acceleration version of the IA-PT algorithm can be defined where all chains share the same proposal parameters and temperature schedule and which will be referred to as the MCA-IA-PT algorithm.

Schäfer and Chopin (2013) propose a related sequential Monte Carlo (SMC) algorithm using the sequence of distributions $\pi_1(\gamma|y), \ldots, \pi_m(\gamma|y)$. They suggest sampling from this sequence of distribution using an SMC algorithm and choosing the sequence of powers $t_j$ adaptively. The IA-SMC algorithm proceeds by alternating selection steps with MCMC steps as follows. Let $t_0 = 0$ and $N$ particles $\gamma_1^\dagger, \ldots, \gamma_N^\dagger$ are chosen from $\pi_0(\gamma^\dagger) = \pi(\gamma^\dagger)$.

1. At the $k$-th selection step - calculate the weight of the $j$-th particle which is distributed according to $\pi_{k-1}$ as

$$w_j \propto p\left(y|\gamma_j^\dagger\right)^{t_k-t_{k-1}}, \quad j = 1, \ldots, N.$$
A sample which reweights according to $w_1, \ldots, w_N$ is selected. Any reweighting scheme can be used but we have used systematic resampling in our examples. The new sample is distributed according to $\pi_k$. The value of $t_k$ is chosen so that the Effective Sample Size is approximately $cN$ for some $0 < c < 1$.

2. MCMC step - $K$ iterations of the individual adaptation algorithm are run for each particle using a common set of $A$ and $D$.

The algorithm proceeds until $t_k = 1$. We have chosen the value $c = 0.9$. This is a conservative choice and often leads to small changes from $t_{k-1}$ to $t_k$ but smaller values of $c$ typically lead to substantially increased problems with particle degeneracy. This leads to a value of $m$ which is chosen adaptively and so is random. The individual adaptation algorithm for each $k$ starts from the values of $A$ and $D$ at the end of the $(k-1)$-th step but the iteration counter is re-set. This allows the algorithm to use information about these tuning parameters from updating the chains for $\pi_1, \ldots, \pi_{k-1}$ but also allows these values to be quickly adapted at each step. The tuning parameters are assumed common for all particles and so changes in the shape from $\pi_{k-1}(\gamma|y)$ to $\pi_k(\gamma|y)$ can be quickly learnt in the algorithm. An alternative scheme for adaptation in SMC is discussed by Fearnhead and Taylor (2013).

5 Ergodicity of the Algorithms

Since adaptive MCMC algorithms violate the Markov condition, the standard and well developed Markov chain theory can not be used to establish ergodicity and we need to derive appropriate results for our algorithms. In particular, it is well known that even simple and seemingly reasonable adaptive algorithms may fail to converge (Roberts and Rosenthal, 2007; Bai et al., 2011; Łatuszyński et al., 2013).

Here we provide some fairly general ergodicity results in the case when the model parameters can be integrated out and the marginal likelihood $p(y|\gamma)$ is available analytically.

Recall that $\pi(\gamma|y) \propto p(y|\gamma)p(\gamma)$, the target posterior on the model space $M$ and the vector of adaptive parameters

$$
\eta^{(i)} = (A^{(i)}, D^{(i)}) \in [\varepsilon, 1 - \varepsilon]^{2p} \equiv \Delta_\varepsilon
$$
at time $i$. By $P_{\eta}(\gamma, \cdot)$ denote the non-adaptive Markov chain kernel corresponding to the fixed choice of $\eta$. Thus under dynamics of the individual adaptation algorithm

$$\mathbb{P}\left[\gamma^{(i+1)} \in S \, \middle| \, \gamma^{(i)} = \gamma, \eta^{(i)} = \eta\right] = P_{\eta}(\gamma, S), \quad S \subseteq M.$$ 

In the case of multiple chain acceleration, where $r$ copies of the chain are run, the respective model state space is the product space and thus the current state of the algorithm at time $i$ is $\gamma^{(i)} \otimes r \in M^r$ and the stationary distribution is the product density $\pi^{\otimes r}$ on $M^r$. Clearly, when $r = 1$ then the multiple chain becomes a single chain and thus all the notions and results in the sequel stated for multiple chains acceleration are valid for the single chain algorithm.

To assess ergodicity, we need to define the distribution of the adaptive algorithm at time $i$, and the associated total variation distance: for $S \subseteq M^r$

$$L^{(i)}[(\gamma^{\otimes r}, \eta), S] := \mathbb{P}\left[\gamma^{(i)} \in S \, \middle| \, \Gamma_0 = \gamma^{\otimes r}, \eta^{(0)} = \eta\right],$$

$$T(\gamma^{\otimes r}, \eta, i) := \|L^{(i)}[(\gamma^{\otimes r}, \eta), \cdot] - \pi^{\otimes r}(\cdot)\|_{TV}$$

$$= \sup_{S \in M^r} |L^{(i)}[(\gamma^{\otimes r}, \eta), S] - \pi^{\otimes r}(S)|.$$ 

Defining $\pi(f) = \sum f(\gamma)\pi(\gamma|y)$, we show that all algorithms are ergodic, i.e.

$$\lim_{i \to \infty} T(\gamma^{\otimes r}, \eta, i) = 0 \quad \text{for every} \quad \gamma^{\otimes r} \in M^r, \quad (9)$$

and satisfy a Weak Law of Large Numbers, i.e.

$$\frac{1}{i} \sum_{k=1}^{i} f(\gamma_k) \xrightarrow{i \to \infty} \pi(f) \quad \text{in probability, for every} \quad f : M^r \to \mathbb{R} \quad (10)$$

and every $\gamma^{\otimes r, (0)} \in M^r, \eta^{(0)} \in \Delta_{\varepsilon}$.

We first establish the following result.

**Lemma 1.** The kernel $P_{\eta}(\gamma, S)$ leads to a simultaneously uniform ergodic chain. For all $\delta > 0$ there exists $N = N(\delta) \in \mathbb{N}$ such that

$$\|P_{\eta}^N(\gamma^{\otimes r}, \cdot) - \pi^{\otimes r}(\cdot)\|_{TV} \leq \delta \quad \text{for all} \quad \gamma^{\otimes r} \in M^r \text{ and } \eta \in \Delta_{\varepsilon},$$

Our first result considers non-tempered versions of the algorithm.
Theorem 1. Assume that \( p(y|\gamma) \pi(\gamma) \) is available analytically for all \( \gamma \in M \) and \( \varepsilon > 0 \) in (3), (4), or in (5)-(8), respectively. Then each of the algorithms: IA, RAPA-IA, MCA-IA and MCA-RAPA-IA is ergodic and satisfies a Weak Law of Large Numbers.

A comprehensive analysis of the individual adaptation algorithm with other generalised linear models or with linear models whose parameters are given a non-conjugate prior distributions requires an involved case by case treatment, and is beyond the scope of this paper. However, we note that if the prior distributions are supported on a compact set and all involved densities are continuous and everywhere positive, establishing ergodicity for a specific model will, with some technical care, typically be possible. The following theorem establish the ergodicity of the parallel tempered MCMC algorithm.

Theorem 2. Assume that \( p(y|\gamma)^t \pi(\gamma) \) is available analytically and is finite for all \( 0 < t \leq 1 \) and \( \gamma \in M \) and \( \varepsilon > 0 \) in (3), (4), or in (5)-(8), respectively. Then each of the algorithms: IA-PT, RAPA-IA-PT, MCA-IA-PT and MCA-RAPA-IA-PT is ergodic and satisfies a Weak Law of Large Numbers.

Finally Theorem 1 combined with standard results for SMC algorithms can be used to show that the IA-SMC algorithm is ergodic as well as its variations with MCA and RAPA.

6 Applications

6.1 Tecator Data

The tecator data contains 172 observations and 100 variables. They have been previously analysed using Bayesian linear regression techniques by Griffin and Brown (2010), who give a description of the data, and Lamnisos et al. (2013). The prior used was (1) with \( V_{\gamma} = 100I \) and \( h = 5/100 \). We generated 10 independent runs of the algorithms with different tuning parameters and without thinning. If multiple chain acceleration was used, the number of iterations in each chain was divided by the number of chains. This fixes the total number of iterations so that run times are the same for all algorithms.

Figure 1 shows the average mutation rate as a function of \( \tau \) for the IA algorithm with MCA only and IA-MCA with IA-RAPA. Both algorithm were
Figure 1: Tecator data: average mutation rate over 10 independent runs as a function of $\tau$ with MCA only and MCA with RAPA 100 000 iterations and 1 000 000 iterations after a burn-in of 100 000 iterations. The number of chains were: 1 (solid line), 5 (dashed line) and 25 (dot-dashed line). The thin solid line is $y = x$

able to effectively target the chosen average mutation rate for most values of $\tau$ with both 100 000 and 1 000 000 iterations after a burn-in of 100 000 iterations. Unsurprisingly, the targeting improves as the number of iterations or the number of chains is increased. All algorithms struggle with targeting larger values of $\tau$ but these are not in a range that we would consider to be optimal.

Figure 2 shows the effect of $\tau$ on the average effective sample size (ESS) with different number of multiple chains and with or without the RAPA step (using $w = 0.5$). In all case, the ESS was maximized by $\tau$ between 0.35 and 0.55 but was relatively constant over this range. This is largely in keeping with previous work on optimal acceptance rates for Metropolis-Hastings ran-
Figure 2: Tecator data: average ESS over 10 independent runs as a function of $\tau$ using MCA only and MCA-RAPA. The number of chains in MCA were: 1 (solid line), 5 (dashed line) and 25 (dot-dashed line).

dom walk samplers on discrete spaces and implies that the performance of the algorithm is not overly sensitive to choice of $\tau$. Both acceleration steps tended to lead to larger effective sample sizes at all values of $\tau$. The effect of MCA was much less pronounced when RAPA was used. The improvement of MCA-RAPA over MCA in targeting the correct rate (particularly, for a single chain) leads to the slightly larger ESS with the addition of a RAPA step.

As a comparison, 10 independent runs of a multi-move Metropolis-Hastings

IA-RAPA, $\tau = 0.45$  
Lamnisos et al, $\tau = 0.3$  
Multi-move MH

Figure 3: Tecator data: model size for the last 5000 iterations from a single run of the IA-RAPA algorithm with $\tau = 0.45$ and two competitors
algorithm with add, remove and swap moves and the adaptive algorithm of Lamnisos et al. (2013) were run. The multi-move sampler had an average ESS of 30 332 and the adaptive algorithm had an average ESS of 40 000 (pretty much unaffected by the value of \( \tau \) in the range \((0.25,0.7)\)). The best individual adaptation algorithm had an ESS around 200 000 which represents roughly a six-fold increase over the multi-move sampler and roughly a five-fold increase over the adaptive algorithm. The mixing of different algorithms with the tecator data is further illustrated in Figure 3 which shows trace plots of the model size for a randomly chosen run. It is clear that the IA-RAPA algorithm leads to much better mixing than the two competitors.

\[ \tau = 0.35 \quad \tau = 0.45 \quad \tau = 0.55 \]

Figure 4: Tecator data: PIP’s estimated from a single run of the IA-RAPA algorithm with \( \tau = 0.35, \tau = 0.45 \) and \( \tau = 0.55 \)

Figure 5: Tecator data: scatter plots of pairs of PIP’s estimated from a single run of the IA-RAPA algorithm with \( \tau = 0.35, \tau = 0.45 \) and \( \tau = 0.55 \). The thin solid line is \( y = x \)

Insight into the behaviour of the algorithm is provided by looking at the results of single runs of the IA-RAPA algorithm with different values of \( \tau \)
with a burn-in period of 100,000 iterations, a subsequent sample of 1 million iterations taken and no thinning. Figure 4 shows the PIP’s and Figure 5 shows scatter-plots of pairs of the estimated posterior inclusion probabilities with the different values of \( \tau \). These indicate very strong agreement across the runs of the individual adaptation algorithms with different \( \tau \).

\[
\begin{align*}
\tau &= 0.35 & \tau &= 0.45 & \tau &= 0.55
\end{align*}
\]

Figure 6: Tecator data: empirical probability mass function of the number of variables proposed to be changed at each iteration during a single run of the IA-RAPA algorithm with \( \tau = 0.35 \), \( \tau = 0.45 \) and \( \tau = 0.55 \).

The empirical probability mass function of the number of variables proposed to be changed at each step of the algorithm is shown in Figure 6. The modal value is 14 for \( \tau = 0.35 \) with a sizeable spread of values from 5 to 22. This illustrates that relatively large changes in the model are possible in this example. The location and spread of the distribution becomes smaller as \( \tau \) increases and less ambitious moves are proposed.

The values of \( A \) and \( D \) for a single run of the algorithm with different values of \( \tau \) are shown in Figures 7. Overall, the values of \( A \) tend to decrease as \( \tau \) increases. Therefore, the algorithm proposes less ambitious moves which leads to a larger average mutation rate. The values of \( D_j \) tend to be close to 0 or 1 when \( \tau = 0.35 \). The value is close to zero for variables which have a higher inclusion probability whereas \( D_j \) is close to one for variables which have a lower PIP. Therefore, the algorithm will usually propose to remove variables with low PIP’s if they are currently included in the model and tend to not propose removing variables with high PIP’s. This type of behaviour is critical for rapid mixing in this type of problem. If a variable has a low PIP, say 0.05 or 0.1, the best mixing would occur if this variable was removed.
from the model as quickly as possible after being added (whilst maintaining the correct PIP). The values of $D$ become less extreme as $\tau$ increases.

The values of $A_j$ and $D_j$ at the end of each run tend to be different (although, many final values of $A_j$ and $D_j$ will be similar across different runs). As we have already mentioned, the convergence of the sampler does not depend on the convergence of the $A_j$’s or $D_j$’s. However, the ratio $A_j/D_j$ tends to have a consistent value across different runs and different values of $\tau$. Figure 8 shows that $A_j/D_j$ is typically very close to $\psi_j/(1-\psi_j)$ where $\psi_j$ is the PIP of the $j$-th variable. As a simple explanation of this effect, consider a posterior for $\gamma$ which is independent: then the Metropolis-Hastings acceptance rate of both adding and removing a variable will be 1 if $A_j/D_j = \psi_j/(1 - \psi_j)$ and so this maximizes the overall acceptance rate. Of course, the posterior distribution will typically be far from independent and this chain will not lead to optimal performance in general.
Figure 8: Tecator data: scatterplot of $A_j/D_j$ at the end of 10 different runs of the IA-RAPA algorithm against $\psi_j/(1-\psi_j)$ where $\psi_j$ is the PIP of the $j$-th regressor calculated using all runs with $\tau = 0.35$, $\tau = 0.45$ and $\tau = 0.55$

6.2 PCR Data

Bondell and 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. Using quantitative real-time polymerase chain reaction (PCR) several physiological phenotypes are recorded. We consider one of these phenotypes, phosphoenolpyruvate carboxykinase (PEPCK) as the response variable. Bondell and Reich (2012) apply their method to both a subset of 2,000 variables (selected on the basis of marginal correlations with the response) and the full data set. We use our adaptive algorithm on the full data set of 22,576 variables. In prior (1) we adopt $V_\gamma = 100I$ and a hierarchical prior was used for $\gamma$ by assuming that $h \sim \text{Be}(1, (p - 5)/5)$ which implies that the prior mean number of included variables is 5. An MCA-PT-IA algorithm was run with $\tau = 0.35$, $m = 6$ temperatures, $r = 5$ or 25 multiple chains, and 24,000,000 iterations (the number of iterations for each chain was divided by the number of multiple chains leading to comparable computational times). Three independent runs of the algorithms were done for each combination of tuning parameters.

Figure 9 shows the PIP’s with 5 and 25 multiple chains. The results indicate that two genes are particularly predictive of the response with PIP’s over 0.5. There are also many other variables with smaller but non-negligible PIP’s. Results from the different runs are in good agreement. Figure 10 shows pairwise comparisons of the PIP’s for each algorithmic parameter set-
Figure 9: PCR Data Example: PIP’s for three runs of the MCA-IA-PT algorithm with 6 temperatures.

Estimated PIP’s are close, particularly for the variables with high PIP’s. Figure 11 shows the posterior distribution of model size from the three runs. The posterior mean model sizes calculated using output from the three runs were 20.2, 20.7 and 20.2 with 5 chains and 20.2, 19.6 and 19.5 with 25 chains. This results is quite sensitive to the choice of the prior on model space. For example, setting $h = 5/22,576$ (rather than using the hierarchical prior, while keeping the same prior mean model size) leads to much smaller model sizes. The posterior mean model sizes in the three runs were 8.8, 8.9 and 9.0 with 5 chains and 8.4, 8.0 and 8.7 with 25 chains. This is in line with the fact that the prior with a fixed $h$ is much more informative than the hierarchical prior (see Ley and Steel, 2009). However, the ranking of the variables in terms of PIP is largely unchanged. The posterior mean model size with the hierarchical prior is much larger than the ones reported by Bondell and Reich (2012) using their marginal sets method.

Figure 12 shows a trace plot of the model size averaged over the multiple chains. This indicate that the average model size for all runs stabilizes around...
Figure 10: PCR Data: scatter plots of the PIP’s for three runs of the MCA-IA-PT algorithm with 6 temperatures

The results with 25 chains have a smaller variability since the average at every iteration involves a larger number of draws.

Bondell and Reich (2012) applied their method to a subset of 2,000 variables chosen to have the largest correlation with the response. Figure 14 shows a scatter plot of the PIP’s for these 2,000 genes with both the full data set and the subset. Six of the eight genes with PIP’s using the full data over 0.1 are included in the reduced data set (with the third and fourth most important genes being excluded). In addition, 10 of the 17 genes with PIP’s over 0.05 are included and 43 of the 164 genes with PIP’s over 0.01 are included. The diminishing proportion of genes included in the reduced data set as we lower the PIP threshold is not surprising since the reduced set is chosen using the marginal relationship between the response and the genes. Figure 13 shows the PIP’s using only the reduced data set from three runs of the MCA-PT-IA algorithm with 6 temperatures. The top two genes from the full data set are the most important but the discrimination between impor-
Figure 11: PCR Data: posterior distribution of model size for three runs of the MCA-IA-PT algorithm with 6 temperatures

tant and unimportant genes is less clear with several variables whose PIP’s are around 0.5 using the subset but are much smaller using all the data (see Figure 11). The posterior mean model size with the reduced data set was 27.6 (averaged across the three runs) compared with 20.4 for the full data set. This suggests that the reduction method removes some simpler models which are well-supported by the data from the set of possible models. These results illustrate the potential problems that can arise by screening variables based on the marginal relationship with the response, such as the popular SIS (sure independence screening) and iterative SIS procedures of Fan and Lv (2008) and the Bayesian subset regression method of Liang et al. (2013). Bondell and Reich (2012) also use SIS on the full data set in combination with SCAD (smoothly clipped absolute deviation; Fan and Li, 2001) which results in very small models (mean model size is 2.3) and relatively poor prediction.
Figure 12: PCR Data: trace plots of the model size averaged across the multiple chains using the MCA-IA-PT algorithm

7 Discussion

Markov chain Monte Carlo methods for Bayesian variable selection has traditionally been considered a difficult problem associated with slow mixing. The individual adaptation algorithm is a method which can substantially improve mixing and lead to much more accurate estimates of posterior quantities, such as posterior inclusion probabilities. It leads to six- and seven-fold improvements in effective sample size in our examples and effectively opens the door for formal Bayesian model selection and model averaging analyses involving very large numbers of covariates, such as over 22 thousand in one of our examples.

These results illustrate the potential of carefully constructed adaptive Monte Carlo schemes in difficult problems for Bayesian inference. Much work on adaptive Monte Carlo has concentrated on problems where hand-tuning of algorithms is feasible but tiresome. The proposal in this paper has $2p$ pa-
Figure 13: PCR Data Example: PIP’s for the reduced data using three runs of the MCA-PT-IA algorithm with 6 temperatures and 5 chains.

Figure 14: PCR Data: A scatter plot of PIP’s calculated using the full data set and the subset

parameters and tuning is only possible using adaptive Monte Carlo ideas. The development of similar algorithms where hand-tuning would be infeasible represents an interesting, and as yet virtually unexplored, area for future research.
A Supplementary material for “Individual adaptation: an adaptive MCMC scheme for variable selection problems”

A.1 Proofs of Ergodicity Results

Proof of Lemma 1. To verify the result it is enough to check that the whole state space $M^r$ is $1−$small with the same constant $\beta > 0$, (c.f. ?), that is check for example that there exists $\beta > 0$ s.t. for every $\eta \in \Delta_\varepsilon$ and every $\gamma^{\otimes r}, \gamma'^{\otimes r} \in M^r$ we have

$$P_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r}) \geq \beta.$$  \hfill (11)

First decompose the move into proposal and acceptance

$$P_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r}) = q_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r}) \times a_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r}),$$

and notice that by the proposal construction $q_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r}) \geq \varepsilon^r p$ since $|M^r| = rp$. Similarly

$$a_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r}) = \min \left\{ 1, \frac{\pi^{\otimes r}(\gamma'^{\otimes r}) q_\eta(\gamma'^{\otimes r}, \gamma^{\otimes r})}{\pi^{\otimes r}(\gamma^{\otimes r}) q_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r})} \right\} \geq \frac{\pi^{\otimes r}(\gamma'^{\otimes r}) q_\eta(\gamma'^{\otimes r}, \gamma^{\otimes r})}{\pi^{\otimes r}(\gamma^{\otimes r}) q_\eta(\gamma^{\otimes r}, \gamma'^{\otimes r})} \geq \frac{\pi^r_m \times \varepsilon^r p}{\varepsilon^r p},$$

where $\pi_m := \min_{\gamma \in M^r} \pi(\gamma)$. Consequently in (11) we can take

$$\beta = \pi^r_m \times \varepsilon^{2rp},$$

and we have established simultaneous uniform ergodicity. \hfill \Box

Proof of Theorem 1. Theorem 1 follows from Theorem 1 (ergodicity) and Theorem 5 (WLLN) of Roberts and Rosenthal (2007). Precisely, simultaneous uniform ergodicity for nonadaptive kernels holds via Lemma 1. Moreover, it is routine to check that the proposal satisfies diminishing adaptation, and consequently by Lemma 4.21 (ii) of Łatuszyński et al. (2013) applied with discrete topology of the variable selection context, also the transition kernels satisfy diminishing adaptation i.e. the random variable

$$D_i := \sup_{\gamma^{\otimes r} \in M^r} \|P_{\eta(i+1)}(\gamma^{\otimes r}, \cdot) - P_{\eta(i)}(\gamma^{\otimes r}, \cdot)\|$$

converges to 0 in probability as $i \to \infty$. \hfill \Box
Proof of Theorem 2. We conclude Theorem 2 from Theorem 1 (ergodicity) and Theorem 3 (WLLN) of ?. To this end we need an analogue of Lemma 1 for the parallel tempering version of the kernel to verify simultaneous uniform ergodicity. This can be established along the same lines as Lemma 1, necessarily with additional notational complication that we omit here for brevity. Similarly, it is routine to verify that the parallel tempering adaptive kernel proposals satisfy diminishing adaptation and again by Lemma 4.21 (ii) of Łatuszyński et al. (2013) applied with discrete topology of the variable selection context, also the transition kernels satisfy diminishing adaptation. □

A.2 Example: Boston Housing data

We considered the Boston housing data previously analyzed by Schäfer and Chopin (2013) in the context of mixing of MCMC algorithms for variable selection. Here we have \( n = 506 \) observations on the log of the median values of owner-occupied housing which are modelled through a linear regression model using \( p = 104 \) potential covariates. We use the prior in equation (1) of the paper with \( V_\gamma = 100I \) and \( h = 5/104 \). The problem differs from the previous one in that \( n > p \), but there is reported evidence of multimodality in the posterior on model space. Thus, we use the methods described in Section 4 and consider the ability of our adaptive algorithm in combination with both the sequential Monte Carlo (SMC) and parallel tempering (PT) algorithms. The complicated nature of the posterior distribution is illustrated by the results in Table 1. The two models with the largest posterior probability differ by only one variable. However, the difference between those models and the model with the third largest posterior probability is much greater. Therefore, it will be difficult for many MCMC algorithms to traverse this posterior distribution. The IA-SMC algorithm was run with 92 500 particles and \( K = 1 \), 18 500 particles and \( K = 5 \), 9 250 particles and \( K = 10 \), 3 700 particles and \( K = 25 \) and finally 1 850 particles and \( K = 50 \) and the IA-PT algorithm \((m = 8)\) was run with a burn-in period of 12 500 and, subsequently, for 525 000 iterations with no thinning. This was found to lead to similar run-times for the different algorithms.

The ESS may not be well-estimated from a single run if the run leans to biased estimates. An alternative is the mean squared error of the estimate
| Model | Post. Prob. |
|-------|------------|
| 5 6 8 | 0.243      |
| 5 6 8 | 0.140      |
| 5 6 8 | 0.031      |
| 5 6 8 | 0.022      |
| 5 6 8 | 0.021      |
| 5 6 8 | 0.018      |
| 5 6 8 | 0.016      |
| 5 6 8 | 0.015      |
| 5 6 8 | 0.013      |
| 5 6 8 | 0.013      |
| 5 6 8 | 0.012      |
| 5 6 8 | 0.011      |

Table 1: The 10 models with the highest posterior probability for the Boston housing data (the variable names are given in the Appendix).
across multiple runs. This is an estimate of the variance of the Monte Carlo estimate when the Monte Carlo estimates are unbiased. However, it naturally includes a penalty for the sampler producing biased estimates. Rather than use Mean Squared Error, the accuracy of the algorithms was evaluated using a Weighted Mean Squared Error (WMSE)

\[ \text{WMSE} = \sum_{i=1}^{M} \sum_{j=1}^{p} w_j (\hat{\theta}_{ij} - \theta_{ij}^*)^2 \]

where \( M \) is the number of replicate MCMC or SMC runs, \( \hat{\theta}_{ij} \) is the estimated posterior inclusion probability for the \( j \)-th variables in the \( i \)-th run and \( \theta_{ij}^* \) is a “gold-standard” estimate of the posterior inclusion probability for the \( j \)-th variable. The weights \( w_j \) are assumed to be such that \( \sum_{j=1}^{p} w_j = 1 \) and \( w_j \) represents the importance of the \( j \)-th variables. We chose \( M = 60 \) and \( w_j \propto \theta_{ij}^* \) in our comparisons. The gold standard value of \( \theta_{ij}^* \) was calculated using output from the PT chains and SMC with \( K = 25 \) and \( K = 50 \) which had the highest levels of accuracy.

The WMSE is shown in Figure 15. The WMSE for the PT algorithm with \( m = 8 \) is shown as a dashed line and decreases with \( \tau \). The graph also shows the WMSE’s for the SMC algorithm with \( K \) MCMC steps in the re-weighting
step. These range from $K = 1$ to $K = 50$. The WMSE decreases with the number of steps for each value of $\tau$ with the WMSE for $K = 50$ having a similar WMSE to the PT algorithm for small $\tau$ but for $\tau \geq 0.3$ the PT algorithm does a lot better. The effect of $\tau$ on the WMSE differs according to the number of Metropolis-Hastings steps. The WMSE tends to increase with $\tau$ for $K = 1$ and $K = 5$ whereas WMSE is not that much affected by $\tau$ for $K = 10$, $K = 25$ and $K = 50$. The simple Metropolis-Hastings algorithm was run with a burn-in period of 12500 with 9 750 000 subsequent iterations with no thinning. This took the same computational times as the other algorithms and so represents a comparison to the more complicated algorithms for multi-modal distributions. The WMSE for the simple MH algorithm was 0.0071 which is smaller than all algorithm apart from the IA-SMC algorithm with $K = 50$ and $K = 25$ with smaller values of $\tau$ and the IA-PT algorithm. The improvement of the IA-PT over the simple MH algorithm is still substantial. The acceptance rate is roughly 2% for the simple MH algorithm and so the adaptive algorithm of Lamnisos et al. (2013) would reduce to the simple MH algorithm for this data set.

References

Atchadé, Y. F., G. O. Roberts, and J. S. Rosenthal (2011). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing* 21, 555–568.

Bai, Y., G. Roberts, and J. Rosenthal (2011). On the containment condition for adaptive Markov chain Monte Carlo algorithms. *Advances and Applications in Statistics* 21, 1–54.

Bondell, H. D. and B. J. Reich (2012). Consistent high-dimensional variable selection via penalized credible regions. *Journal of the American Statistical Association* 107, 1610–1624.

Bornn, L., P. E. Jacob, P. Del Moral, and A. Doucet (2013). An adaptive interacting Wang-Landau algorithm for automatic density exploration. *Journal of Computational and Graphical Statistics* 22, 749–773.
Bottolo, L. and S. Richardson (2010). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis* 5, 583–618.

Brown, P. J., M. Vannucci, and T. Fearn (1998). Multivariate Bayesian variable selection and prediction. *Journal of the Royal Statistical Society, B* 60, 627–641.

Clyde, M. A., J. Ghosh, and M. L. Littman (2011). Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics* 20, 80–101.

Craiu, R. V., J. Rosenthal, and C. Yang (2009). Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association* 104, 1454–1466.

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

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96, 1348–1360.

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society, B* 70, 849–911 (with discussion).

Fearnhead, P. and B. M. Taylor (2013). An adaptive sequential Monte Carlo sampler. *Bayesian Analysis* 8, 411–438.

García-Donato, G. and M. A. Martínez-Beneito (2013). On sampling strategies for Bayesian variable selection problems with large model spaces. *Journal of the American Statistical Association* 108, 340–352.

George, E. I. and R. E. McCulloch (1997). Approaches for Bayesian variable selection. *Statistica sinica* 7, 339–373.

Griffin, J. E. and P. J. Brown (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis* 5, 171–188.

Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli* 7, 223–242.
Jasra, A., D. A. Stephens, and C. C. Holmes (2007). Population-based reversible jump Markov chain Monte Carlo. *Biometrika* 94, 787–807.

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

Kwon, D., M. T. Landi, M. Vannucci, H. J. Issaq, D. Prieto, and R. M. Pfeiffer (2011). An efficient stochastic search for Bayesian variable selection with high-dimensional correlated predictors. *Computational Statistics and Data Analysis* 55, 2807–2818.

Lamnisos, D. S., J. E. Griffin, and M. F. J. Steel (2009). Transdimensional sampling algorithms for Bayesian variable selection in classification problems with many more variable than observations. *Journal of Computational and Graphical Statistics* 18, 592–612.

Lamnisos, D. S., J. E. Griffin, and M. F. J. Steel (2013). Adaptive Monte Carlo for Bayesian variable selection in regression models. *Journal of Computational and Graphical Statistics* 22, 729–748.

Łatuszyński, K., G. O. Roberts, and J. S. Rosenthal (2013). Adaptive Gibbs samplers and related MCMC methods. *The Annals of Applied Probability* 23, 66–98.

Ley, E. and M. F. J. Steel (2009). On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics* 24, 651–674.

Liang, F., Q. Song, and K. Yu (2013). Bayesian subset modelling for high-dimensional generalized linear models. *Journal of the American Statistical Association* 108, 589–606.

Miasojedow, B., E. Moulines, and M. Vihola (2013). An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics* 22, 649–664.

Nott, D. J. and R. Kohn (2005). Adaptive sampling for Bayesian variable selection. *Biometrika* 92, 747–763.
O’Hara, R. B. and M. J. Sillanpää (2009). A review of Bayesian variable selection methods: what, how and which. *Bayesian analysis* 4, 85–117.

Richardson, S., L. Bottolo, and J. S. Rosenthal (2010). Bayesian models for sparse regression analysis of high dimensional data. *Bayesian Statistics 9*, 539–568.

Roberts, G. O. (1998). Optimal metropolis algorithms for product measures on the vertices of a hypercube. *Stochastics and Stochastic Reports* 62, 275–283.

Roberts, G. O., A. Gelman, and W. R. Gilks (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability* 7, 110–120.

Roberts, G. O. and J. S. Rosenthal (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science* 16, 351–367.

Roberts, G. O. and J. S. Rosenthal (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability* 44, 458–475.

Schäfer, C. and N. Chopin (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing* 23, 163–184.

**Appendix: Variables for the Boston housing data**

This is a list of the variables that appear in Table T using the names given in the R package *spdep*.

| 5 | NOX | 24 | NOX × CRIM | 67 | TAX × RAD |
| 6 | RM  | 29 | RM × CRIM  | 78 | PTRATIO × TAX |
| 8 | DIS | 49 | DIS²       | 86 | B × DIS |
| 9 | RAD | 50 | RAD × CRIM | 91 | B²  |
| 10| TAX | 54 | RAD × NOX  | 97 | LSTAT × RM |
| 11| PTRATIO | 55 | RAD × RM  | 101 | LSTAT × TAX |
| 12| B   | 58 | RAD²       | 103 | LSTAT × B  |
| 13| LSTAT| 59 | TAX × CRIM | 104 | LSTAT² |
| 14| CRIM²| 61 | TAX × CHAS |   |
References

Atchadé, Y. F., G. O. Roberts, and J. S. Rosenthal (2011). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing* 21, 555–568.

Bai, Y., G. Roberts, and J. Rosenthal (2011). On the containment condition for adaptive Markov chain Monte Carlo algorithms. *Advances and Applications in Statistics* 21, 1–54.

Bondell, H. D. and B. J. Reich (2012). Consistent high-dimensional variable selection via penalized credible regions. *Journal of the American Statistical Association* 107, 1610–1624.

Bornn, L., P. E. Jacob, P. Del Moral, and A. Doucet (2013). An adaptive interacting Wang-Landau algorithm for automatic density exploration. *Journal of Computational and Graphical Statistics* 22, 749–773.

Bottolo, L. and S. Richardson (2010). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis* 5, 583–618.

Brown, P. J., M. Vannucci, and T. Fearn (1998). Multivariate Bayesian variable selection and prediction. *Journal of the Royal Statistical Society, B* 60, 627–641.

Clyde, M. A., J. Ghosh, and M. L. Littman (2011). Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics* 20, 80–101.

Craiu, R. V., J. Rosenthal, and C. Yang (2009). Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association* 104, 1454–1466.

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

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96, 1348–1360.
Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society, B* 70, 849–911 (with discussion).

Fearnhead, P. and B. M. Taylor (2013). An adaptive sequential Monte Carlo sampler. *Bayesian Analysis* 8, 411–438.

García-Donato, G. and M. A. Martínez-Beneito (2013). On sampling strategies for Bayesian variable selection problems with large model spaces. *Journal of the American Statistical Association* 108, 340–352.

George, E. I. and R. E. McCulloch (1997). Approaches for Bayesian variable selection. *Statistica sinica* 7, 339–373.

Griffin, J. E. and P. J. Brown (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis* 5, 171–188.

Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli* 7, 223–242.

Jasra, A., D. A. Stephens, and C. C. Holmes (2007). Population-based reversible jump Markov chain Monte Carlo. *Biometrika* 94, 787–807.

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

Kwon, D., M. T. Landi, M. Vannucci, H. J. Issaq, D. Prieto, and R. M. Pfeiffer (2011). An efficient stochastic search for Bayesian variable selection with high-dimensional correlated predictors. *Computational Statistics and Data Analysis* 55, 2807–2818.

Lamnisos, D. S., J. E. Griffin, and M. F. J. Steel (2009). Transdimensional sampling algorithms for Bayesian variable selection in classification problems with many more variable than observations. *Journal of Computational and Graphical Statistics* 18, 592–612.

Lamnisos, D. S., J. E. Griffin, and M. F. J. Steel (2013). Adaptive Monte Carlo for Bayesian variable selection in regression models. *Journal of Computational and Graphical Statistics* 22, 729–748.
Łatuszyński, K., G. O. Roberts, and J. S. Rosenthal (2013). Adaptive Gibbs samplers and related MCMC methods. *The Annals of Applied Probability* 23, 66–98.

Ley, E. and M. F. J. Steel (2009). On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics* 24, 651–674.

Liang, F., Q. Song, and K. Yu (2013). Bayesian subset modelling for high-dimensional generalized linear models. *Journal of the American Statistical Association* 108, 589–606.

Miasojedow, B., E. Moulines, and M. Vihola (2013). An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics* 22, 649–664.

Nott, D. J. and R. Kohn (2005). Adaptive sampling for Bayesian variable selection. *Biometrika* 92, 747–763.

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

Richardson, S., L. Bottolo, and J. S. Rosenthal (2010). Bayesian models for sparse regression analysis of high dimensional data. *Bayesian Statistics* 9, 539–568.

Roberts, G. O. (1998). Optimal metropolis algorithms for product measures on the vertices of a hypercube. *Stochastics and Stochastic Reports* 62, 275–283.

Roberts, G. O., A. Gelman, and W. R. Gilks (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability* 7, 110–120.

Roberts, G. O. and J. S. Rosenthal (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science* 16, 351–367.

Roberts, G. O. and J. S. Rosenthal (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability* 44, 458–475.
Schäfer, C. and N. Chopin (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing* 23, 163–184.