Adaptive Monte Carlo for Bayesian Variable Selection in Regression Models

Demetris Lamnisos, Jim E. Griffin and Mark F.J. Steel *

April 10, 2012

Abstract

This article describes methods for efficient posterior simulation for Bayesian variable selection in Generalized Linear Models with many regressors but few observations. The algorithms use a proposal on model space which contains a tuneable parameter. An adaptive approach to choosing this tuning parameter is described which allows automatic, efficient computation in these models. The method is applied to examples from normal linear and probit regression. Relevant code and datasets are posted as an online supplement.

Keywords: Linear Regression; Probit Regression; Metropolis-within-Gibbs

1 Introduction

The availability of datasets with large numbers of variables has lead to interest in the use of variable selection methods for regression models with large numbers of potential regressors. In this paper, we will concentrate on Bayesian variable selection methods applied to datasets with hundreds of regressors where Markov chain Monte Carlo methods can be effectively

*Cyprus University of Technology and University of Kent and University of Warwick

used. We will work in the context of Generalized Linear Models (GLM) (McCullagh and Nelder, 1989) where it is assumed that $y_1, y_2, \ldots, y_n$ are observed dependent variables and the i-th observation is associated with a set of p regressors $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and that

$$f(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}, \tau) = \exp\left\{a_i^{-1}(\tau)\left[y_i g_i - b(g_i) + c(y_i, \tau)\right]\right\} \tag{1}$$

where $g_i = g(\eta_i)$ is the link function and $\eta_i = \alpha + \boldsymbol{x}_i\boldsymbol{\beta}$ is the linear predictor, with intercept $\alpha \in \mathbb{R}$ and regression coefficient $\boldsymbol{\beta} \in \mathbb{R}^p$. Different choices of a, b and c lead to different conditional models for y_i. Linear regression and probit regression have been the most thoroughly explored models with large datasets and this paper will concentrate on these examples (although several methods are applicable to all GLMs). For example, Lee et al. (2003) and Sha et al. (2003, 2004) consider using MCMC methods for Bayesian variable selection in a probit regression model to find gene expression levels related to a binary response (such as diseased or non-diseased). In linear regression, large numbers of variables are common in spectroscopy, chemometrics or proteomics and can also occur in economics. In both regression models, we would usually assume that only a subset of the regressors are needed to predict y_i and the vector of indicator variables $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_p)$ is introduced to represent inclusion ($\gamma_i = 1$) or exclusion ($\gamma_i = 0$) of the i-th regressor leading to model size $p_\gamma = \sum_{i=1}^{p} \gamma_i$ for $\boldsymbol{\gamma}$. The model $\boldsymbol{\gamma}$ uses the linear predictor $\eta_i = \alpha + \boldsymbol{x}_i^\gamma \boldsymbol{\beta}_\gamma$ in equation (1), where $\boldsymbol{x}_i^\gamma$ and $\boldsymbol{\beta}_\gamma$ group the p_γ elements for which $\gamma_i = 1$. Bayesian inference proceeds by placing a prior on $\boldsymbol{\gamma}$ as well as α, $\boldsymbol{\beta}_\gamma$ and τ.

Summaries of the posterior distribution are typically computed using either efficient methods to select a small subset of interesting models combined with an approximation for $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ (Yeung et al., 2005; Hans et al., 2007; Clyde et al., 2011) or Markov chain Monte Carlo methods that produce samples from the posterior distribution using a Gibbs sampler. This paper follows the second approach. The model $\boldsymbol{\gamma}$ is usually updated by a Metropolis-Hastings step which proposes new models by either including an excluded variable, excluding an included variable or swapping one included variable with one excluded variable. This is similar in spirit to a Random Walk Metropolis (RWM) sampler, which generates a Markov chain by proposing a new value as a perturbed version of the current value with the difference

that here the state space is a lattice. In the RWM sampler, the variance of the pertubations can be controlled to obtain an optimal RWM sampler. Lamnisos et al. (2009) find that acceptance rates for Metropolis-Hastings samplers that include, exclude or swap a single variable usually have high acceptance rates for variable selection in probit regression models when there are many regressors and few observations. They describe a tuneable proposal for variable selection problems with a single parameter defined on $[0, 1]$ controlling the difference between the current and proposed model. They find that in probit modelling with many regressors, the optimal sampler occurs for a proposal which leads to an average acceptance rate close to 0.234, which is the value for the optimal RWM sampler in many continuous problems and in simple discrete problems on a lattice (Roberts and Rosenthal, 2001; Sherlock and Roberts, 2009; Roberts, 1998).

Metropolis-Hastings samplers with tuneable proposal can lead to efficient algorithms but tuning can be time-consuming. Recently, there has been interest in adaptive Monte Carlo methods where the distribution of the proposal is adjusted during the MCMC run. These methods are difficult to implement in general since the Markov property is violated and standard theory for convergence of the chain to the target distribution does not apply. However, convergence to the target distribution can be verified for particular forms of adjustment. The first adaptive algorithm that could be shown to converge to the target distribution was introduced by Haario et al. (2001) who used methods from Stochastic Approximation. This important idea and other methods are reviewed by Andrieu and Thoms (2008).

Our goal is to automatically produce a Markov chain with good mixing properties which gives accurate answers with the smallest possible run length. The posterior distribution will be complicated and vast (there will be 2^p potential models if there are p potential regressors). Adaptive methods are important because MCMC methods often mix slowly (and so proposals that encourage good mixing are important) and the running of many pilot runs is unsatisfactory due to the large number of iterations needed to give good estimates of posterior summaries. Such methods have been previously applied to variable selection problems by Nott and Kohn (2005). They allow the probability that a particular variable is proposed

to be included in or removed from the model to adapt over the chain. This is rather different from the method developed here where the expected number of variables to be updated, rather than the variable-specific update probability, is adapted over the chain. The method uses a form of proposal for variable selection described by Lamnisos et al. (2009) which can be tuned in a similar way to a RWM sampler.

The paper is organised as follows: Section 2 describes the Bayesian approach to variable selection in linear regression, as well as a tuneable proposal on model space and an adaptive MCMC algorithm in linear regression. Section 3 describes MCMC algorithms for Bayesian variable selection in generalized linear models and their adaptive versions, as well as ergodicity results of the adaptive MCMC algorithms. Section 4 discusses some numerical examples that illustrate the utility of the approach and finally the Discussion contains some concluding comments. The code and data are freely available at http://www.amstat.org/publications/jcgs.

2 MCMC algorithms for Bayesian Variable Selection in Linear Regression

The normal **linear regression model** has greater analytical tractability than other GLMs and so we start by discussing that model here. It assumes that observations $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$ are generated by

$$\boldsymbol{y} \sim \mathrm{N}(\alpha \mathbf{1} + \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma, \tau \mathbf{I}_n).$$

Within a Bayesian analysis, $\alpha, \boldsymbol{\beta}_\gamma, \tau$ and $\boldsymbol{\gamma}$ are given prior distributions. The intercept α is given the commonly used noninformative improper prior for location parameters

$$\pi(\alpha) \propto 1, \tag{2}$$

the regression coefficients have the multivariate normal prior

$$\boldsymbol{\beta}_\gamma | \tau, \boldsymbol{\gamma} \sim \mathrm{N}_{p_\gamma}(\mathbf{0}, \tau \boldsymbol{V}_\gamma), \tag{3}$$

and the error variance τ is assigned the usual noninformative improper prior for scale parameters

$$\pi(\tau) \propto \frac{1}{\tau}. \tag{4}$$

Two quite common choices for the prior covariance $\boldsymbol{V}_\gamma$ are $c\mathbf{I}_{p_\gamma}$ and $c(\boldsymbol{X}'_\gamma \boldsymbol{X}_\gamma)^{-1}$ which involve a single hyperparameter. Finally, the prior on $\boldsymbol{\gamma}$ assumes that $\pi(\gamma_i = 1) = w$, independently for $i = 1, \ldots, p$.

The priors (2), (3) and (4) result in an analytical expression for the marginal likelihood $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ of model $\boldsymbol{\gamma}$ given by

$$\pi(\boldsymbol{y}|\boldsymbol{\gamma}) \propto |\boldsymbol{X}_\gamma^{\mathrm{T}} \boldsymbol{X}_\gamma + \boldsymbol{V}_\gamma^{-1}|^{-1/2} \, |\boldsymbol{V}_\gamma|^{-1/2} \, (\tilde{\boldsymbol{y}}^{\mathrm{T}} \tilde{\boldsymbol{y}} - \boldsymbol{y}^{\mathrm{T}} \boldsymbol{X}_\gamma (\boldsymbol{X}_\gamma^{\mathrm{T}} \boldsymbol{X}_\gamma + \boldsymbol{V}_\gamma^{-1})^{-1} \boldsymbol{X}_\gamma^{\mathrm{T}} \boldsymbol{y})^{-(n-1)/2},$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \bar{y}\mathbf{1}$ and $\bar{y}$ is the mean of the response $\boldsymbol{y}$.

This analytical expression for the marginal likelihood $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ facilitates the development of MCMC methods that are used to estimate posterior model characteristics. A Metropolis-Hastings algorithm with the posterior model distribution $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$ as stationary distribution can be implemented. This Metropolis-Hastings linear regression (MH-LR) algorithm proceeds as follows:

Algorithm 1 (MH-LR) *Let $\boldsymbol{\gamma}$ be the current state of the chain.*

1. *Select model $\boldsymbol{\gamma}'$ with probability $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$.*

2. *Jump to the model $\boldsymbol{\gamma}'$ with probability*

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min \left\{ 1, \frac{\pi(\boldsymbol{y}|\boldsymbol{\gamma}') \, \pi(\boldsymbol{\gamma}') \, q(\boldsymbol{\gamma}|\boldsymbol{\gamma}')}{\pi(\boldsymbol{y}|\boldsymbol{\gamma}) \, \pi(\boldsymbol{\gamma}) \, q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})} \right\}.$$

The availability of a closed form expression for $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ allows a Metropolis-Hastings algorithm to be defined directly on $\boldsymbol{\gamma}$. An important part of MH-LR is the model proposal $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ which controls the mixing of the algorithm.

2.1 A Tuneable Proposal On Model Space

Lamnisos et al. (2009) propose a new general model proposal $q_\zeta(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ which draws a new model in the following way:

1. A value $N^{(t)}$ is generated from a Binomial distribution with a fixed number, $N-1$, of trials and success probability ζ.

2. One of three possible moves: "Add", "Delete" and "Swap" is chosen uniformly at random. If Add is selected then $N^{(t)}+1$ regressors are chosen to be added to those included in $\boldsymbol{\gamma}$ to form $\boldsymbol{\gamma}'$, if Delete is selected then $N^{(t)}+1$ regressors are chosen to be removed from the model and if Swap is selected then $N^{(t)}+1$ included regressors are swapped with $N^{(t)}+1$ exluded regressors without changing the model size (provided $p_\gamma \geq N^{(t)}+1$; if not, the "Add" step is chosen for $p_\gamma < N^{(t)}+1$ and either the "Add" or "Swap" for $p_\gamma = N^{(t)}+1$. In those cases, the model proposal and reverse model proposal are slightly adjusted to consider those conditions).

This model proposal combines local moves with more global ones by simultaneously changing a block of variables. Two parameters determine this proposal: N is the maximum number of variables that can be changed from the current model $\boldsymbol{\gamma}$ and ζ determines the degree of "localness" since the mean number of variables proposed to be changed is $4/3 \times (1+(N-1)\times\zeta)$. If $\zeta = 0$, we have the standard proposal which adds or deletes a single variable or swaps one variable into the model and one variable out of the model. However, more ambitious moves which change more variables are increasingly likely to be proposed as ζ increases. The value of N will usually be fixed and the parameter ζ chosen to control the mixing of the chain. The application of this proposal to microarray data by Lamnisos et al. (2009) suggests that the optimum effective sample size is obtained when the average acceptance rate falls in the range 0.25 to 0.40. This is true for a wide-range of sampling schemes. Rather like RWM samplers, this optimal choice of acceptance rate can be achieved by carefully tuning the parameter ζ of the model proposal using a series of pilot runs. In each pilot run, the sampler is run for a chosen value of ζ and the average acceptance rate

calculated. If the acceptance rate is too high then ζ is increased in the next run and if the acceptance rate is too low then ζ is decreased in the next run. However, this tuning process will involve trial and error and so is typically a computationally expensive task.

2.2 Adaptive MCMC Algorithm in Linear Regression

As an alternative solution to a series of pilot runs, we consider adaptive MCMC algorithms which can automatically handle this parameter tuning. This problem is similar to adaptation in RWM samplers since there is a tuneable proposal and a target acceptance rate to be achieved. Therefore, we adopt ideas of the Adaptive Random Walk Metropolis (ARWM) algorithm proposed by Atchadé and Rosenthal (2005) to develop adaptive MCMC algorithms for variable selection that adapt sequentially the scale parameter ζ.

In our case the scale parameter $\zeta \in [0, 1]$ and similarly to Atchadé and Rosenthal (2005) we define the following function of ζ

$$\rho(\zeta) = \begin{cases} 0 & \text{if} \quad \zeta < 0 \\ \zeta & \text{if} \quad \zeta \in [0,1] \\ 1 & \text{if} \quad \zeta > 1. \end{cases}$$

The aim of this function is to contain the adaptive algorithm inside $[0, 1]$. Finally, a positive sequence of real numbers $s^{(t)} = \zeta_0/t$ is defined. The pseudocode representation of the adaptive MH-LR algorithm (denoted by ADMH-LR) adjusts the model proposal step and adds an extra step (step 3 below) to the corresponding non-adaptive algorithm (Algorithm 1).

Algorithm 2 (ADMH-LR) *Let $\boldsymbol{\gamma}$ be the current state of the chain and $\zeta^{(t)} \in [0, 1]$.*

1. *Select model $\boldsymbol{\gamma}'$ with probability $q_{\zeta^{(t)}}(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$.*

2. *Jump to the model $\boldsymbol{\gamma}'$ with probability*

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\pi(\boldsymbol{y}|\boldsymbol{\gamma}')\,\pi(\boldsymbol{\gamma}')\,q(\boldsymbol{\gamma}|\boldsymbol{\gamma}')}{\pi(\boldsymbol{y}|\boldsymbol{\gamma})\,\pi(\boldsymbol{\gamma})\,q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})}\right\}.$$

3. Compute

$$\zeta^{(t+1)} = \rho(\zeta^{(t)} + s^{(t)}(\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') - \bar{\tau})). \tag{5}$$

The acceptance rate is monitored by (5), where $\bar{\tau}$ is a value chosen from the range 0.25 to 0.4. The algorithm decreases the scale parameter $\zeta^{(t+1)}$ when the acceptance rate is small and increases $\zeta^{(t+1)}$ when the acceptance rate is high. The sequence of scale parameters $\zeta^{(t)}$ converges to a value that results in the target acceptance rate $\bar{\tau}$ (if it is achievable).

3 MCMC algorithms for Bayesian Variable Selection in Generalized Linear Models

The marginal likelihood $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ is not analytically available for other GLMs and a sampler must be defined on the joint posterior of $\boldsymbol{\gamma}$, α, $\boldsymbol{\beta}_\gamma$ and τ. Let us focus on the particular case of the **probit model**. If the GLM is a probit regression model then Albert and Chib (1993) show that the model can be written in the following way

$$z_i \sim \mathrm{N}(\alpha + \boldsymbol{x}_i^\gamma \boldsymbol{\beta}_\gamma, 1) \tag{6}$$

where $y_i = 0$ if $z_i < 0$ and $y_i = 1$ if $z_i > 1$ and $\boldsymbol{x}_i^\gamma$ denotes the ith row of $\boldsymbol{X}_\gamma$. Latent variables $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^{\mathrm{T}}$ are introduced which allow the model to be expressed as a linear regression with known error variance in z_i. The prior on $\boldsymbol{\beta}_\gamma$ will be similar to (3), namely

$$\boldsymbol{\beta}_\gamma|\boldsymbol{\gamma} \sim \mathrm{N}_{p_\gamma}(\mathbf{0}, \boldsymbol{V}_\gamma),$$

and for α we now take a vague Normal prior

$$\alpha \sim \mathrm{N}(0, h),$$

with some large value for h (we adopt $h = 100$ in the sequel). In probit regression, unlike linear regression, the use of the improper prior in (2) can yield improper posteriors for α and $\boldsymbol{\beta}_\gamma$ if the design matrix $\tilde{\boldsymbol{X}}_\gamma = (\mathbf{1} : \boldsymbol{X}_\gamma)$ does not satisfy two conditions (Lamnisos,

2010). Furthermore, identification issues between α and the scale of $\boldsymbol{V}_\gamma$ are raised when a prior distribution is specified on the scale parameter, as documented in Lamnisos (2010). We shall define $\boldsymbol{\theta}_\gamma = (\alpha, \boldsymbol{\beta}_\gamma^{\mathrm{T}})^{\mathrm{T}}$ in what follows. Various MCMC methods for dealing with these models are described by Lamnisos et al. (2009). One of these methods is the Holmes and Held (2006) (H-H) algorithm that uses the data augmentation approach of probit regression to develop an efficient between-model move. This algorithm proceeds as follows:

Algorithm 3 (H-H) *Suppose that $(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)$ are the current values of the chain*

1. *Generate $z_1, z_2, \ldots, z_n$ from the truncated normal distribution $z_i \sim \mathbf{N}(\alpha + \boldsymbol{x}_i^\gamma \boldsymbol{\beta}_\gamma, 1)$ and $z_i > 0$ if $y_i = 1$ or $z_i < 0$ if $y_i = 0$.*

2. *Select model $\boldsymbol{\gamma}'$ with probability $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$.*

3. *Jump to the model $\boldsymbol{\gamma}'$ with probability*

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}')q(\boldsymbol{\gamma}|\boldsymbol{\gamma}')\pi(\boldsymbol{z}|\boldsymbol{\gamma}')}{\pi(\boldsymbol{\gamma})q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})\pi(\boldsymbol{z}|\boldsymbol{\gamma})}\right\}.$$

4. *If $\boldsymbol{\gamma}'$ is accepted, draw $\boldsymbol{\theta}_{\gamma'} \sim \mathbf{N}((\tilde{\boldsymbol{X}}_{\gamma'}^{\mathrm{T}}\tilde{\boldsymbol{X}}_{\gamma'} + \tilde{\boldsymbol{V}}_{\gamma'}^{-1})^{-1}\tilde{\boldsymbol{X}}_{\gamma'}^{\mathrm{T}}\boldsymbol{z}, (\tilde{\boldsymbol{X}}_{\gamma'}^{\mathrm{T}}\tilde{\boldsymbol{X}}_{\gamma'} + \tilde{\boldsymbol{V}}_{\gamma'}^{-1})^{-1})$.*

The between-model acceptance probability of the H-H algorithm is independent of parameter states and it is similar to the acceptance probability of MH-LR algorithm with target distribution $\pi(\boldsymbol{\gamma}|\boldsymbol{z})$. The H-H algorithm can be made adaptive by updating ζ at each iteration using the recursion in (5). The pseudocode representation for the adaptive H-H algorithm (denoted by ADH-H) has the form

Algorithm 4 (ADH-H) *Let $(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)$ be the current state and $\zeta^{(t)} \in [0, 1]$, then*

1. *Generate $z_1, z_2, \ldots, z_n$ from the truncated normal distribution $z_i \sim \mathbf{N}(\alpha + \boldsymbol{x}_i^\gamma \boldsymbol{\beta}_\gamma, 1)$ and $z_i > 0$ if $y_i = 1$ or $z_i < 0$ if $y_i = 0$.*

2. *Select model $\boldsymbol{\gamma}'$ with probability $q_{\zeta^{(t)}}(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$.*

3. *Jump to the model $\boldsymbol{\gamma}'$ with probability*

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}')q_{\zeta^{(t)}}(\boldsymbol{\gamma}|\boldsymbol{\gamma}')\pi(\boldsymbol{z}|\boldsymbol{\gamma}')}{\pi(\boldsymbol{\gamma})q_{\zeta^{(t)}}(\boldsymbol{\gamma}'|\boldsymbol{\gamma})\pi(\boldsymbol{z}|\boldsymbol{\gamma})}\right\}.$$

4. *Compute*

$$\zeta^{(t+1)} = \rho(\zeta^{(t)} + s^{(t)}(\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') - \bar{\tau})).$$

5. *If $\boldsymbol{\gamma}'$ is accepted, draw $\boldsymbol{\theta}_{\gamma'} \sim \mathrm{N}((\tilde{\boldsymbol{X}}_{\gamma'}^{\mathrm{T}}\tilde{\boldsymbol{X}}_{\gamma'} + \tilde{\boldsymbol{V}}_{\gamma'}^{-1})^{-1}\tilde{\boldsymbol{X}}_{\gamma'}^{\mathrm{T}}\boldsymbol{z}, (\tilde{\boldsymbol{X}}_{\gamma'}^{\mathrm{T}}\tilde{\boldsymbol{X}}_{\gamma'} + \tilde{\boldsymbol{V}}_{\gamma'}^{-1})^{-1})$.*

These algorithms could only be applied to a restricted class of GLMs that has a data augmentation representation leading to a latent model which is linear in the regression coefficients and involves errors that are either a mixture of normals (Holmes and Held, 2006) or can be approximated by a mixture of normals (Frühwirth-Schnatter and Wagner, 2006; Frühwirth-Schnatter and Frühwirth, 2007). Furthermore, these algorithms are likely to mix slowly because the auxiliary variable $\boldsymbol{z}$ is correlated with $(\boldsymbol{\theta}_\gamma, \boldsymbol{\gamma})$, as is seen from (6), and $\boldsymbol{z}$ is updated from its full conditional distribution.

The Automatic Generic sampler described by Green (2003) and extended by Lamnisos et al. (2009) avoids using the auxiliary variable $\boldsymbol{z}$ in the between-model move and is applicable to any GLM. The MLE of $\boldsymbol{\theta}_\gamma$ is asymptotically normally distributed and so the full conditional of $\boldsymbol{\theta}_\gamma$ can be approximated by a normal distribution . The automatic generic method exploits the fact that a normally distributed random variable, $\boldsymbol{x}$, with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$ can be written as $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}$ where $\boldsymbol{\Sigma}^{1/2}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}$ and $\boldsymbol{\epsilon}$ is standard normal with the same dimension as $\boldsymbol{x}$. In what follows we shall use the notation $\tilde{\boldsymbol{X}}_\gamma = (\mathbf{1} : \boldsymbol{X}_\gamma)$ and $\tilde{\boldsymbol{V}}_\gamma = \begin{pmatrix} h & \mathbf{0}' \\ \mathbf{0} & \boldsymbol{V}_\gamma \end{pmatrix}$. The Automatic Generic (AG) algorithm is then:

Algorithm 5 (AG) *Let $(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)$ be the current state of the chain.*

1. *Select model $\boldsymbol{\gamma}'$ with probability $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$.*

2. *Propose $\boldsymbol{\theta}_{\gamma'}$ in the following way. Let $\boldsymbol{\mu}_\gamma$ and $\boldsymbol{\Sigma}_\gamma$ be an approximation of the mean and variance of the posterior distribution of $\boldsymbol{\theta}_\gamma$ and let $\boldsymbol{B}_\gamma$ be the Cholesky decomposition*

of $\boldsymbol{\Sigma}_\gamma$ *and* $\boldsymbol{v} = \boldsymbol{B}_\gamma^{-1}(\boldsymbol{\theta}_\gamma - \boldsymbol{\mu}_\gamma)$*. Then we propose* $\boldsymbol{\theta}_{\gamma'} = \boldsymbol{\mu}_{\gamma'} + \boldsymbol{B}_{\gamma'}\boldsymbol{v}'$ *where*

$$\boldsymbol{v}' = \begin{cases} (v_1, \ldots, v_{p_{\gamma'}})^{\mathrm{T}} & \text{if } p_{\gamma'} < p_\gamma \\ \boldsymbol{v} & \text{if } p_{\gamma'} = p_\gamma \\ (\boldsymbol{v}^{\mathrm{T}}, \boldsymbol{\epsilon}^{\mathrm{T}})^{\mathrm{T}} & \text{if } p_{\gamma'} > p_\gamma \end{cases}$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_{p_{\gamma'} - p_\gamma})^{\mathrm{T}}$ *has i.i.d.* $\mathbf{N}(0, 1)$ *elements.*

3. *Jump to the model* $\boldsymbol{\gamma}'$ *and parameter* $\boldsymbol{\theta}_{\gamma'}$ *with probability* $\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}', \boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_{\gamma'}) =$

$$\min\left\{1, \frac{\pi(\boldsymbol{\gamma}', \boldsymbol{\theta}_{\gamma'}) q(\boldsymbol{\gamma}|\boldsymbol{\gamma}') \pi(\boldsymbol{y}|\boldsymbol{\theta}_{\gamma'}) |\boldsymbol{B}_{\gamma'}|}{\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma) q(\boldsymbol{\gamma}'|\boldsymbol{\gamma}) \pi(\boldsymbol{y}|\boldsymbol{\theta}_\gamma) |\boldsymbol{B}_\gamma|} \times K\right\}$$

where

$$K = \begin{cases} (2\pi)^{-\frac{1}{2}(p_\gamma - p_{\gamma'})} \exp\left\{-\frac{1}{2}(\boldsymbol{\epsilon}')^{\mathrm{T}}(\boldsymbol{\epsilon}')\right\} & \text{if } p_{\gamma'} < p_\gamma \\ 1 & \text{if } p_{\gamma'} = p_\gamma \\ (2\pi)^{\frac{1}{2}(p_{\gamma'} - p_\gamma)} \exp\left\{\frac{1}{2}\boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}\right\} & \text{if } p_{\gamma'} > p_\gamma, \end{cases}$$

and $\boldsymbol{\epsilon}'$ *is the obvious counterpart of* $\boldsymbol{\epsilon}$*.*

There are several methods of finding $\boldsymbol{\mu}_\gamma$ and $\boldsymbol{\Sigma}_\gamma$. Two are considered in this paper: the Laplace approximation and the Iterated Weighted Least Squares (IWLS) approximation for one iteration (see Lamnisos et al. (2009) and its supplemental material for more details). The Laplace method involves an optimization algorithm in each iteration to find accurate estimates of $\boldsymbol{\mu}_\gamma$ and $\boldsymbol{\Sigma}_\gamma$ while the IWLS method use a single cycle of the Bayesian IWLS algorithm to find rough estimates of $\boldsymbol{\mu}_\gamma$ and $\boldsymbol{\Sigma}_\gamma$. However, the IWLS approximation has much lower computational cost.

Alternative automatic methods for moving between models are described by Brooks et al. (2003) which are applied to probit regression by Lamnisos et al. (2009). In the case of proposals that increase the model size, the coefficient vector is completed with $\boldsymbol{u}_\gamma(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}$ has a standard normal distribution. The Conditional Maximization method chooses $\boldsymbol{\mu}$ to maximize the posterior distribution $\pi(\boldsymbol{\theta}_\gamma, \boldsymbol{u}_\gamma|\boldsymbol{y})$ with respect to $\boldsymbol{u}_\gamma$. The variance, σ^2, is chosen to ensure that the acceptance probability of the move with $\boldsymbol{u}_\gamma(\boldsymbol{\epsilon}) = \boldsymbol{\mu}$ is 1 and is given by

$$\sigma = \left(\frac{\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)\ q(\boldsymbol{\gamma'}|\boldsymbol{\gamma})\ \pi(\boldsymbol{y}|\boldsymbol{\theta}_\gamma)}{(2\pi)^{(p_{\gamma'} - p_\gamma)/2}\ \pi(\boldsymbol{\gamma}', \boldsymbol{\theta}_{\gamma'})\ q(\boldsymbol{\gamma}|\boldsymbol{\gamma'})\ \pi(\boldsymbol{y}|(\boldsymbol{\theta}_\gamma, \boldsymbol{\mu}))}\right)^{\frac{1}{p_{\gamma'} - p_\gamma}}.$$

Here

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}', \boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_{\gamma'}) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}', \boldsymbol{\theta}_{\gamma'})\ q(\boldsymbol{\gamma}|\boldsymbol{\gamma}')\ \pi(\boldsymbol{y}|\boldsymbol{\theta}_{\gamma'})\ (\sigma^2)^{\frac{1}{2}(p_{\gamma'}-p_\gamma)}}{\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)\ q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})\ \pi(\boldsymbol{y}|\boldsymbol{\theta}_\gamma)} K\right\} \tag{7}$$

where

$$K = (2\pi)^{\frac{1}{2}(p_{\gamma'}-p_\gamma)} \exp\left\{\frac{1}{2}\boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}\right\}.$$

The pseudocode representation of the Conditional Maximization (CM) method is as follows:

Algorithm 6 (CM) *If current state is* $(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)$*, then*

1. *Select model* $\boldsymbol{\gamma}'$ *with probability* $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$.
2. *Determine the location* $\boldsymbol{\mu}$ *and the scale* σ *of the proposal random variable* $\boldsymbol{u}_\gamma$ *as described above.*
3. *Generate* $\boldsymbol{\epsilon} \sim \mathbf{N}_{p_{\gamma'}-p_\gamma}(\mathbf{0}, \mathbf{I}_{p_{\gamma'}-p_\gamma})$.
4. *Set* $\boldsymbol{u}_\gamma(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$ *and* $\boldsymbol{\theta}_{\gamma'} = (\boldsymbol{\theta}_\gamma, \boldsymbol{u}_\gamma)$.
5. *Jump to the model* $\boldsymbol{\gamma}'$ *and* $\boldsymbol{\theta}_{\gamma'}$ *with probability given by (7). Otherwise, the proposal is rejected.*

Algorithms 5 and 6 can be made adaptive in the same way as Algorithm 3 by changing $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ to $q_{\zeta^{(t)}}(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ and updating ζ after calculating the acceptance probability of the Metropolis-Hastings step. The proposed adaptive MCMC algorithms adapt the model proposal $q_\zeta(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ while the adaptive MCMC algorithm of Ji and Schmidler (2009) adapt the weight of the point mass component of a point mass mixture proposal on each β_i to approximate the posterior inclusion probability of its associated variable.

The MCMC algorithms of this section could be optionally supported with a within-model move that enables a better exploration of the parameter space. In the case of probit regression, the data augmentation representation could be used to implement the within-model move which involves Steps 1 and 4 of the H-H algorithm.

3.1 Ergodicity of the Adaptive MCMC algorithms

General conditions for ergodicity of the adaptive algorithms are discussed by Roberts and Rosenthal (2007) who establish two sufficient conditions: diminishing adaptation and simultaneous uniform ergodicity. The diminishing adaptation requires that the amount of adaptation diminishes at each iteration, which can be achieved by modifying the scale parameter ζ by smaller and smaller amounts. Roberts and Rosenthal (2007) give some sufficient conditions for simulteaneous uniform ergodicity.

The adaptive algorithms introduced in this paper clearly satisfy the diminishing adaptation condition. The proposal density $q_\zeta(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ of the adaptive algorithms is for $k = 1, \ldots, N$

$$q_\zeta(\boldsymbol{\gamma}'|\boldsymbol{\gamma}) = \begin{cases} \dfrac{1}{3\,|\boldsymbol{\gamma}^+|} \begin{pmatrix} N-1 \\ k-1 \end{pmatrix} \zeta^{k-1}(1-\zeta)^{N-k} & \text{if} \quad \displaystyle\sum_{i=1}^{p} |\gamma_i' - \gamma_i| = k, \text{addition} \\ \dfrac{1}{3\,|\boldsymbol{\gamma}^0|} \begin{pmatrix} N-1 \\ k-1 \end{pmatrix} \zeta^{k-1}(1-\zeta)^{N-k} & \text{if} \quad \displaystyle\sum_{i=1}^{p} |\gamma_i' - \gamma_i| = 2k, \text{swap} \\ \dfrac{1}{3\,|\boldsymbol{\gamma}^-|} \begin{pmatrix} N-1 \\ k-1 \end{pmatrix} \zeta^{k-1}(1-\zeta)^{N-k} & \text{if} \quad \displaystyle\sum_{i=1}^{p} |\gamma_i' - \gamma_i| = k, \text{deletion} \\ 0, \text{ otherwise,} \end{cases}$$

where $|\boldsymbol{\gamma}^+| = \#$ of neighboring models of $\boldsymbol{\gamma}$ with dimension $p_\gamma + k$, $|\boldsymbol{\gamma}^0| = \#$ of neighboring models of $\boldsymbol{\gamma}$ with dimension p_γ and $|\boldsymbol{\gamma}^-| = \#$ of neighboring models of $\boldsymbol{\gamma}$ with dimension $p_\gamma - k$. The ADMH-LR algorithms will satisfy the simultaneous uniform ergodicity condition since the state space $\mathcal{X} = \{0,1\}^p$ is finite and the proposal density $q_\zeta(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ is continuous with respect to ζ in the close interval $[0,1]$. We conjecture that these properties will also lead to the ergodicity of the other algorithms. Certain sufficient conditions ensuring that simultaneous uniform ergodicity condition holds for a specific adaptive MCMC algorithm are discussed in Roberts and Rosenthal (2007) and Bai et al. (2011). Finally, the diminishing adaptation condition and the continuity of $q_\zeta(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ in $[0,1]$ do not depend on the magnitude of the acceptance rate for each ζ and they are not violated if the target acceptance rate $\bar{\tau}$ is not achievable.

4 Illustrations

The performance of the adaptive MCMC algorithms is evaluated using examples from linear and probit regressions. All the adaptive MCMC samplers start with initial parameter value $\zeta_0 = 0.5$ and the algorithms were run for 2,000,000 iterations with a burn-in period of 100,000 iterations and thinned every 10th iteration resulting in an MCMC sample size T of 190,000. We specify the value 0.3 as a target acceptance rate $\bar{\tau}$ because the optimum effective sample size of the MCMC algorithms that explore the model and parameter space of our problem is obtained when acceptance rates are between 0.25 and 0.4. Adopting $\bar{\tau} = 0.234$ instead makes very little difference to our results. We compare the adaptive version to its non-adaptive counterpart using the parameter settings $\zeta = 0,\ 0.25,\ 0.5,\ 0.75,\ 0.95$ and $N = 4$. All the MCMC samplers have been replicated five times with random starting values. In both applications, we assume that $\boldsymbol{V}_\gamma$ is a diagonal matrix $c\mathbf{I}_{p_\gamma}$. This implies that the coefficients are independent *a priori* and we choose $c = 5$ which is the value chosen by Sha et al. (2004). We also use mean prior model size pw equal to five.

The efficiency of an MCMC sampler can be measured using the Effective Sample Size (ESS) which is $T/(1 + 2\sum_{j=1}^{\infty} \rho_j)$ for an MCMC run of length T with lag j autocorrelation ρ_j (*e.g.*, Liu, 2001). The interpretation is that the MCMC sampler leads to the same accuracy of estimates as a Monte Carlo sampler (where all the draws are independent) run for ESS iterations. In this paper, the MCMC output monitored consists of the components γ_i of $\boldsymbol{\gamma}$ since the posterior inclusion probabilities are the main quantities of interest in variable selection. An estimate of the integrated autocorrelation time $\tau_i = 1 + 2\sum_{j=1}^{\infty} \rho_j$ for each γ_i was computed using the Lag Window Estimator (Geyer, 1992) with Parzen window kernel. We calculate the median m of τ_i's for each algorithm and estimate the Effective Sample Size by ESS $= T/m$. The algorithms have different running times and so we define the efficiency ratio for a sampler to be

$$\text{ER(Sampler)} = \frac{\text{ESS(Sampler)}}{\text{CPU(Sampler)}},$$

which standardizes the effective sample size by CPU run time and so penalizes computation-

ally inefficient algorithms. We are interested in the performance of each adaptive algorithm to the non-adaptive algorithm with $\zeta = 0$ (which is the standard MCMC proposal for these types of models and represents a baseline) and with the optimal value of ζ among five candidates ($\zeta = 0, 0.25, 0.5, 0.75, 0.95$), which is the value resulting in the highest ESS. The relative efficiency of the non-adaptive over the adaptive algorithm is defined by

$$\text{R.E} = \frac{\text{ER(Non-Adaptive)}}{\text{ER(Adaptive)}}.$$

4.1 Linear Regression

An adaptive version of the MH-LR algorithm was applied to the Tecator dataset ($n = 172$, $p = 100$) which is discussed in Griffin and Brown (2010). Figure 1 displays the es-

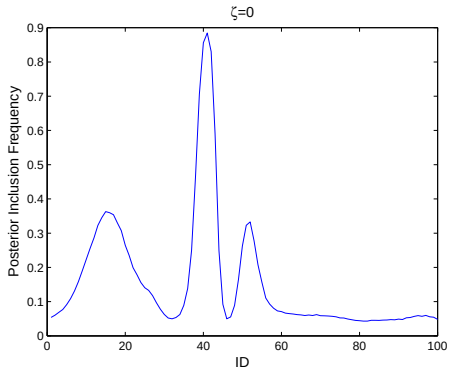

Figure 1: Estimated posterior inclusion probabilities of Tecator data with MH-LR algorithm and $\zeta = 0$

timated posterior inclusion probabilities using the MH-LR algorithm with $\zeta = 0$. The posterior inclusion probabilities are very similar for different values of ζ. Figure 2 shows how the new general model proposal improves the ESS, even though it decreases the between-model acceptance rate. The MH-LR algorithm has maximum ESS for $\zeta = 0.25$ which gives an acceptance rate of 0.33. This acceptance rate falls in the range 0.25 to 0.4 and this result is consistent with that found in probit regression.

Table 1 presents results of the MC^3 algorithm (Madigan and York, 1995), the adaptive Gibbs (ADG) of Nott and Kohn (2005) and the adaptive and non-adaptive MH-LR with fixed

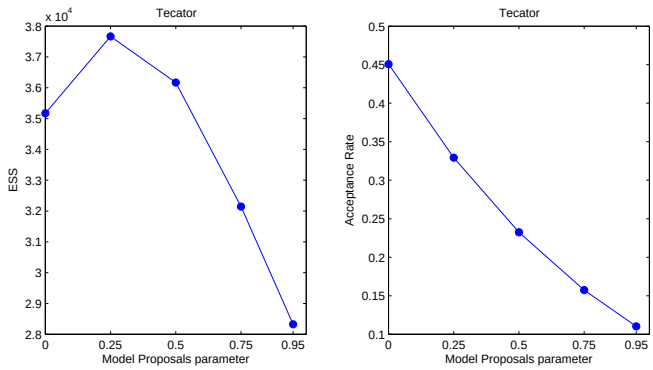

Figure 2: Effective sample size and acceptance rate of the MH-LR algorithm for five different model proposal parameters using the Tecator dataset

Method	ESS(s.e)	CPU(s.e)	ER(s.e)	R.E(s.e)
MC^3	10520 (28.1)	25836 (9.7)	0.41 (0.01)	
ADG	1462 (7.1)	22507 (43.0)	0.07 (0.01)	
MH-LR $(\zeta = 0)$	35644 (46.9)	26250 (7.4)	1.36 (0.01)	0.96(0.01)
MH-LR $(\zeta = 0.25)$	37581 (107.5)	26415 (3.9)	1.42 (0.01)	1.00 (0.01)
ADMH-LR	37437 (62.6)	26448 (5.9)	1.42 (0.01)	

Table 1: The effective sample size ESS, the CPU time in seconds, the efficiency ratio E.R of the MC^3, adaptive Gibbs and the adaptive and non-adaptive MH-LR with relative efficiencies of the non-adaptive algorithm over the adaptive algorithm for the Tecator dataset

values of ζ (the standard choice of $\zeta = 0$ and the optimal value among the values mentioned above). Standard errors for the estimates over the five replications are also provided. The MH-LR is almost 3.5 times more efficient than the MC^3 and almost 20 times more efficient than ADG. The relative efficiency of the sampling method with standard proposal $(\zeta = 0)$ is less than 1 indicating that the adaptive method is superior. The relative efficiency of the optimal non-adaptive algorithm is around 1 and therefore the adaptive algorithm achieves essentially the same efficiency as the optimal non-adaptive algorithms.

Figure 3 shows the trace plots of both the model proposal parameter ζ (left panel) and the empirical acceptance rate (right panel) of the adaptive algorithm for the Tecator dataset. The

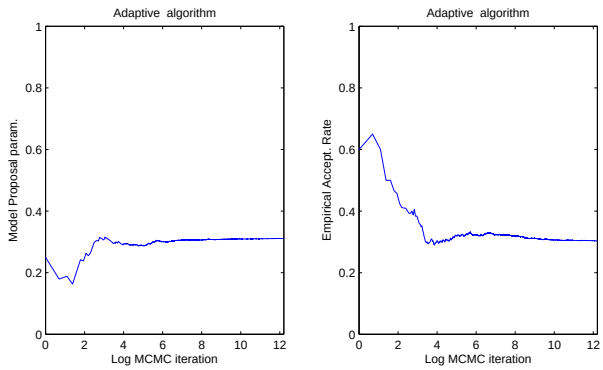

Figure 3: Trace plots of the model proposal parameter ζ and the empirical acceptance rate of the adaptive MH-LR algorithm for the Tecator dataset; MCMC iterations are represented on a log scale

parameter ζ of the adaptive algorithm converges to a value close to the optimal one obtained by manual tuning and the empirical acceptance rate converges to a value quite close to the target acceptance rate 0.3. This result illustrates that the adaptive MH-LR algorithm automatically finds model proposal parameters ζ that give asymptotically the target acceptance rate $\bar{\tau} = 0.3$.

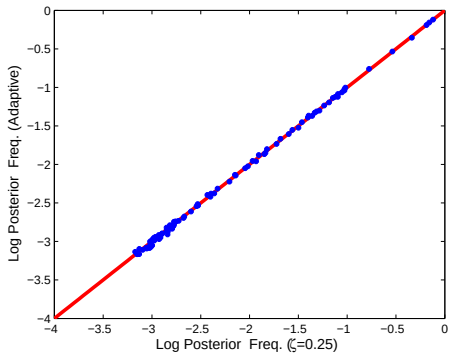

Figure 4: Scatter-plot of the log estimated posterior inclusion probabilities of the adaptive and optimal non-adaptive algorithms for the Tecator dataset

Figure 4 displays the scatter-plot of the log estimated posterior inclusion probabilities of the adaptive and optimal non-adaptive algorithms for the Tecator dataset. These log posterior inclusion probabilities are very similar for both algorithms.

4.2 Probit Regression

The performance of the adaptive MCMC algorithms in probit regression is evaluated using two microarray datasets. These are the Arthritis data (Sha et al., 2003) for which we have $n = 31$ observations and $p = 755$ possible regressors and the Colon Tumour data (Alon et al., 1999) with $n = 62$ and $p = 1224$. Adaptive versions of all algorithms in Section 3 were tested and are denoted as follows:

1. ADH-H : Adaptive Holmes and Held algorithm
2. ADAG-LA : Adaptive automatic generic sampler with Laplace approximation
3. ADAG-IWLS : Adaptive automatic generic sampler with Iterated Weighted Least Squares approximation
4. ADCM : Adaptive Conditional Maximization.

Non-adaptive versions of the algorithms are indicated by dropping the first two letters "AD".

Table 2 presents results of the MC^3 algorithm, the adaptive Gibbs (ADG) and the adaptive and non-adaptive algorithms discussed in Section 3 with fixed values of ζ (the standard choice of $\zeta = 0$ and the optimal value among the five values mentioned above) for the Arthritis and Colon Tumour datasets. The Automatic Generic samplers tend to have the highest ESS, followed by CM and H-H and finally the MC^3 and ADG which have considerably lower ESS. If we take computing time into account, the AG-IWLS sampler is the most efficient, following by H-H, AG-LA and CM. The AG-IWLS sampler is almost 30 times more efficient than MC^3 and ADG.

The relative efficiency for the adaptive algorithms against the standard proposal ($\zeta = 0$) is less than 1 only for the Automatic Generic algorithms indicating that those methods gain the most benefit from adaptation. Furthermore, the increase in performance depends on the form of posterior. However, the effect can be large in some cases. For example, the standard method only obtains 80% of the efficiency of the adaptive method with the Arthritis data and AG-LA and AG-IWLS algorithms. The Automatic Generic methods gain the most benefit

Table 2: The effective sample size ESS, the CPU time in seconds, the efficiency ratio E.R of the MC^3, adaptive Gibbs and the adaptive and non-adaptive algorithms discussed in Section 3 with relative efficiencies of the non-adaptive algorithm over the adaptive algorithm for the Arthritis and Colon Tumour datasets

Arthritis data

Method	ESS(s.e)	CPU(s.e)	ER(s.e)	R.E(s.e)
MC^3	1495 (2.2)	3955 (24.4)	0.38 (0.01)	
ADG	1241 (2.6)	3889 (21.8)	0.32 (0.01)	
H-H ($\zeta = 0$)	51498 (92.4)	5146 (3.2)	10.01 (0.02)	1.07 (0.02)
ADH-H	49908 (55.4)	5321 (2.4)	9.38 (0.01)	
AG-LA ($\zeta = 0$)	72584 (76.4)	19958 (7.2)	3.64 (0.01)	0.83 (0.02)
AG-LA ($\zeta = 0.5$)	95664 (83.8)	21096 (3.1)	4.53 (0.01)	1.03 (0.02)
ADAG-LA	94453 (251.3)	21424 (22.1)	4.41 (0.02)	
AG-IWLS ($\zeta = 0$)	64564 (81.8)	6155 (2.4)	10.49 (0.02)	0.81 (0.01)
AG-IWLS ($\zeta = 0.5$)	82822 (104.9)	6435 (4.5)	12.87 (0.02)	0.99 (0.01)
ADAG-IWLS	82385 (70.3)	6328 (7.0)	13.02 (0.02)	
CM ($\zeta = 0$)	73216 (40.7)	18564 (1.1)	3.94 (0.01)	1.52 (0.04)
CM ($\zeta = 0.5$)	85406 (58.4)	27774 (3.3)	3.08 (0.01)	1.18 (0.03)
ADCM	80246 (189.0)	30827 (80.3)	2.60 (0.01)	

Colon Tumour data

Method	ESS(s.e)	CPU(s.e)	ER(s.e)	R.E(s.e)
MC^3	1199 (64.4)	5498 (19.5)	0.22 (0.01)	
ADG	1080 (4.9)	4947 (23.9)	0.22 (0.01)	
H-H ($\zeta = 0$)	48330 (46.0)	7314 (8.5)	6.61 (0.01)	1.07 (0.02)
ADH-H	46202 (182.9)	7450 (5.1)	6.20 (0.02)	
AG-LA ($\zeta = 0$)	67643 (94.0)	21478 (7.3)	3.15 (0.01)	0.96 (0.02)
AG-LA ($\zeta = 0.5$)	73702 (49.1)	22049 (7.3)	3.34 (0.01)	1.02 (0.02)
ADAG-LA	73601 (123.0)	22250 (6.5)	3.28 (0.01)	
AG-IWLS ($\zeta = 0$)	63939 (53.0)	7889 (8.8)	8.11 (0.01)	0.96 (0.01)
AG-IWLS ($\zeta = 0.5$)	69845 (87.3)	8063 (7.4)	8.66 (0.02)	1.02 (0.01)
ADAG-IWLS	68815 (129.5)	8117 (5.9)	8.48 (0.02)	
CM ($\zeta = 0$)	58232 (106.2)	20189 (3.9)	2.88 (0.01)	1.46 (0.03)
CM ($\zeta = 0.5$)	59311 (97.0)	25410 (5.2)	2.33 (0.01)	1.18 (0.03)
ADCM	56786 (165.9)	28643 (182.0)	1.98 (0.02)	

from adaptation because their acceptance rates with standard proposal ($\zeta = 0$) are around 0.50 for Arthritis and Colon Tumour while the H-H algorithm results in acceptance rates of 0.41 and 0.36, respectively. Implementing an adaptive H-H algorithm in those cases will not improve efficiency as we are already close to optimum efficiency. The H-H algorithm almost always has the lowest acceptance rates compared to other algorithms in real applications (Lamnisos et al., 2009). The CM algorithm is also not benefitting from adaptation because the computational complexity of this algorithm increases considerably with ζ (each iteration of CM algorithm involves a maximization problem whose execution time increases with the number of variables proposed to be changed from the current model).

The effective sample size of the adaptive Automatic Generic samplers are very similar to the optimal non-adaptive algorithms in terms of mixing. Furthermore, the increase in CPU time of the adaptive algorithms is small. This leads to relative efficiencies quite close to 1 and therefore the adaptive AG algorithms achieve essentially the same efficiency as their optimal non-adaptive counterparts. Crucially, however, the adaptive algorithms avoid the pilot runs needed to tune the model proposal parameter ζ. Overall, the adaptive Automatic Generic algorithm with IWLS approximation seems to be the most efficient algorithm in probit regression with $p \gg n$ and it is the one recommended.

Figure 5 and Figure 6 show the trace plots of both the model proposal parameter ζ (left panels) and the empirical acceptance rate (right panels) of the adaptive algorithms for the Arthritis and Colon Tumour datasets, respectively. The parameter ζ of each adaptive algorithm converges to a value close to the optimal one obtained by manual tuning. Furthermore, the empirical acceptance rates converge to values quite close to the target acceptance rate of 0.3. These results illustrate that the adaptive MCMC algorithms automatically find model proposal parameters ζ that asymptotically lead to the desired acceptance rate $\bar{\tau} = 0.3$.

Figure 7 displays the scatter-plots of the log estimated posterior gene inclusion probabilities of the adaptive and optimal non-adaptive algorithms for the Arthritis and Colon Tumour datasets. The log posterior gene inclusion probabilities are very similar indicating empirically that the stationary distribution of the stochastic process generated by the adaptive

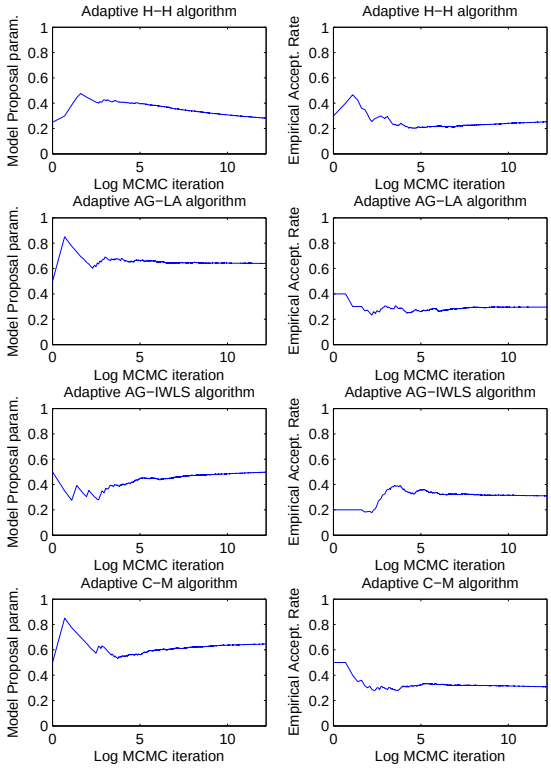

Figure 5: Trace plots of the model proposal parameter ζ and the empirical acceptance rate of the adaptive algorithms for the Arthritis dataset; MCMC iterations are represented on a log scale

MCMC algorithms is the target posterior distribution $\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma | \boldsymbol{y})$.

In all the applications, we choose $N = 4$ because the very large number of predictors and the high correlations among them result in a high acceptance probability for proposals which only change a single variable. Therefore, we wish to try some ambitious global moves to improve model space exploration. When the posterior model distribution is concentrated on few models because of either very large sample size n or low correlation among predictors then it is more reasonable to choose smaller N $(N = 2)$ because more global moves will only be accepted very infrequently.

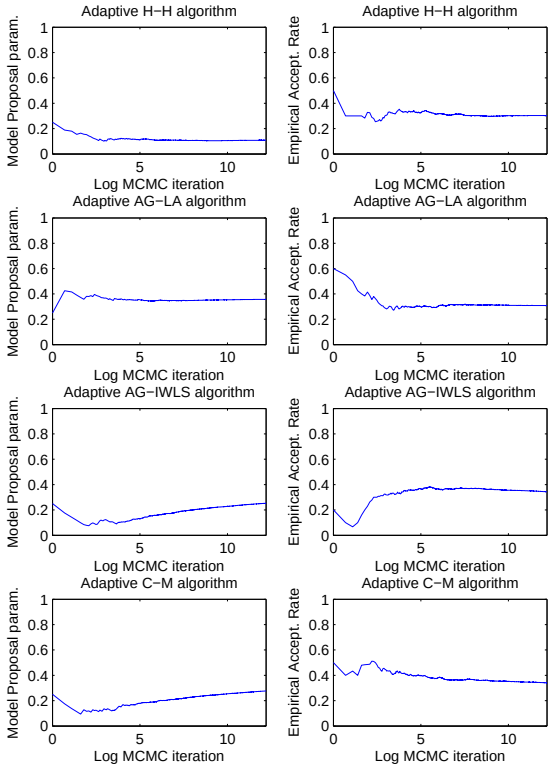

Figure 6: Trace plots of the model proposal parameter ζ and the empirical acceptance rate of the adaptive algorithms for the Colon Tumour dataset; MCMC iterations are represented on a log scale

5 Discussion

This paper describes an adaptive Monte Carlo algorithm for posterior simulation for variable selection in generalized linear models with many regressors, where acceptance rates for standard MCMC algorithms tend to be fairly large. The algorithm leads to Markov chains with good mixing properties without the need for pilot runs and outperforms previously proposed adaptive methods for variable selection. In fact, the effective sample sizes for the adaptive algorithms are almost identical to those for the algorithms run at an optimized value of the proposal parameter ζ (found using trial-and-error which requires separate tuning runs). The methods are useful when there are a large number of variables that could potentially be included in the model, which leads to high acceptance rates for standard algorithms. If the number of regressors is not large, then acceptance rates may not be high and a target ac-

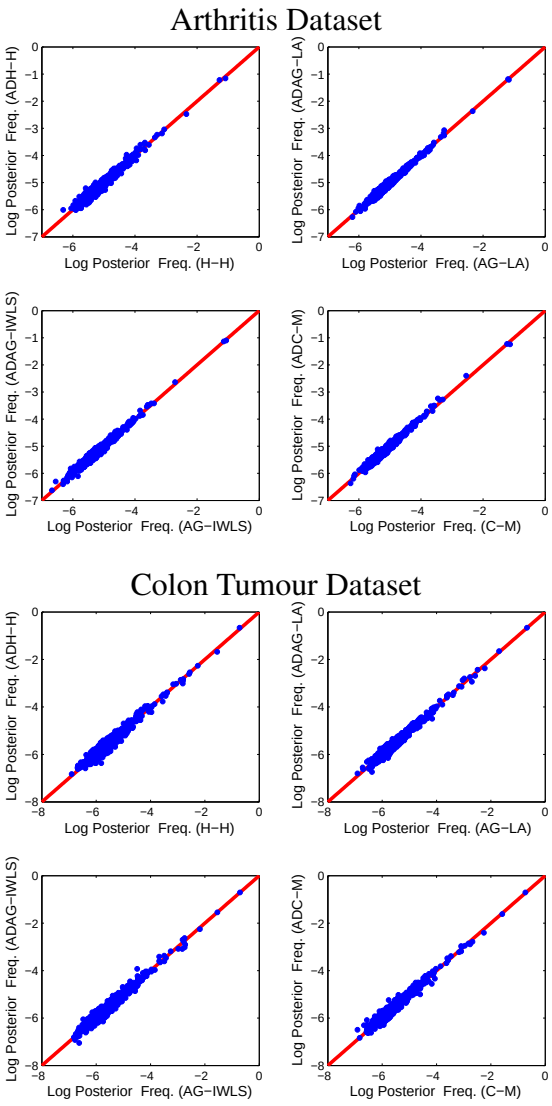

Figure 7: Scatter-plots of the log estimated posterior gene inclusion probabilities of the adaptive and optimal non-adaptive algorithms for the Arthritis and Colon Tumour datasets

ceptance rate of 0.3 may not be achievable. In this case, ζ should be zero and the value of $\zeta^{(t)}$ will converge to zero showing the robustness of the algorithm. Therefore, we suggest the use of adaptive MCMC algorithms to efficiently explore the model space using Bayesian variable selection in linear and probit regression problems with many covariates. Extensions to variable selection problems for other types of GLMs can also be accommodated. For

example, Frühwirth-Schnatter and Wagner (2006) and Frühwirth-Schnatter and Frühwirth (2007) propose an auxiliary mixture sampling approach which uses approximations through mixtures of normal distributions to develop easy Gibbs sampling schemes for Poisson and logistic regression models, respectively. The discussed adaptive model proposal can be easily implemented within these sampling approaches to provide efficient algorithms for Bayesian variable selection in such models.

Supplemental materials

Computer Code and Data: Supplemental materials for this article are contained in a zip archive and can be obtained in a single download. The archive contains all three datasets used in the paper (MATLAB MAT-files) and MATLAB files implementing the adaptive algorithms in the paper (MATLAB M-files). A detailed description of the supplemental materials is contained in the pdf file entitled "!Read Me.pdf" which is enclosed in the zip archive.

Acknowledgements

Demetris Lamnisos acknowledges support from the Centre for Research in Statistical Methodology (CRiSM) at the University of Warwick.

References

Albert, J. H. and S. Chib (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association 88*, 669–679.

Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of American 96*, 6745–6750.

Andrieu, C. and J. Thoms (2008). A tutorial on adaptive MCMC. *Statistics and Computing 18*, 343–373.

Atchadé, Y. F. and J. S. Rosenthal (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli 11*(5), 815–828.

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

Brooks, S. P., P. Giudici, and G. O. Roberts (2003). Efficient constuction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society. Series B 65*, 3–55.

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

Frühwirth-Schnatter, S. and R. Frühwirth (2007). Auxiliary mixture sampling with applications to logistic models. *Computational Statistics and Data Analysis 51*, 3509–3528.

Frühwirth-Schnatter, S. and H. Wagner (2006). Auxiliary mixture sampling for parameter-driven models of time series of small counts with applications to state space modelling. *Biometrika 93*, 827–841.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science 7*, 473–511.

Green, P. J. (2003). Trans-dimensional Markov chain Monte Carlo. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Stuctured Stochastic Systems*, pp. 179–206. Oxford University Press.

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

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

Hans, C., A. Dobra, and M. West (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association 102*, 507–516.

Holmes, C. C. and L. Held (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis 1*, 145–168.

Ji, C. and S. C. Schmidler (2009). Adaptive markov chain Monte Carlo for Bayesian variable selection. Duke University.

Lamnisos, D. (2010). Bayesian variable selection for binary regression with many more variables than observations. Ph.D. dissertation, University of Warwick.

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

Lee, K. E., N. Sha, E. R. Dougherty, M. Vannucci, and B. Mallick (2003). Gene selection: A Bayesian variable selection approach. *Bioinformatics 19*, 90–97.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York.

Madigan, D. and J. York (1995). Bayesian graphical models for discrete data. *International Statistical Review 63*, 215–232.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models* (2 ed.). Boca Raton: Chapman and Hall / CRC.

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

Roberts, G. O. (1998). Optimal metropolis algorithms for product measures on the vertices of a hypercube. *Stochastics and Stochastic Reports 62*(3-4), 275–283.

Roberts, G. O. and J. S. Rosenthal (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science 16*(4), 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.

Sha, N., M. Vannucci, P. J. Brown, M. K. Trower, G. Amphlett, and F. Falciani (2003). Gene selection in arthritis classification with large-scale microarray expression profiles. *Comparative and Functional Genomics 4*, 171–181.

Sha, N., M. Vannucci, M. G. Tadesse, P. J. Brown, I. Dragoni, N. Davies, T. C. Roberts, A. Contestabile, M. Salmon, C. Buckley, and F. Falciani (2004). Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics 60*, 812–819.

Sherlock, C. and G. O. Roberts (2009). Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli 15*, 774–798.

Yeung, K. Y., R. E. Bumgarner, and A. E. Raftery (2005). Bayesian model averaging: Development of an improved multi-class gene selection and classification tool for microarray data. *Bioinformatics 21*, 2394–2402.