Efficient mode jumping MCMC for Bayesian variable selection in GLMM

Aliaksandr Hubin *
Department of Mathematics, University of Oslo
and
Geir Storvik
Department of Mathematics, University of Oslo

2017-04-13

Abstract

Generalized linear mixed models (GLMM) are used for inference and prediction in a wide range of different applications providing a powerful scientific tool for the researchers and analysts coming from different fields. In most of these fields more and more sources of data are becoming available introducing a variety of hypothetical explanatory variables for these models to be considered. Selection of an optimal combination of these variables is thus becoming crucial. In a Bayesian setting, the posterior distribution of the models can be viewed as a relevant measure for the model evidence, based on the observed data. The number of models to select from is exponential in the number of candidate variables, moreover the search space in this context is often extremely non-concave and has numerous local extrema or statistically speaking modes. Hence efficient search algorithms have to be adopted for evaluating the posterior distribution within a reasonable amount of time. In this paper a novel MCMC algorithm for search through the model space via efficient mode jumping for GLMMs is introduced. The algorithm is based on that marginal likelihoods can be efficiently calculated within each model. We recommend that either exact expressions or precise approximations of marginal likelihoods are applied. We further apply the suggested algorithm to some simulated data, the famous U.S. crime data, protein activity data and epigenetic data and compare its performance to some of the existing approaches like MCMC algorithms with local moves between models and the Bayesian adaptive sampling algorithm by Clyde et al. (2011).

Keywords: Bayesian variable selection; Generalized linear mixed models; Auxiliary variables; Combinatorial optimization; High performance computations.

*The authors gratefully acknowledge CELS project at the University of Oslo for giving us the opportunity, inspiration and motivation to write this article.
1 INTRODUCTION

In this paper we study variable selection in generalized linear mixed models (GLMM) addressed in a Bayesian setting. These models allow to carry out detailed modeling in terms of both linking responses and reasonably chosen explanatory variables via a proper link function and incorporating the unexplained variability and dependence structure between the observations via random effects. Being one of the most powerful modeling tools in modern statistical science (Stroup 2013) these models have proven to be efficient in numerous applications including the simple banking scoring problems (Grossi & Bellini 2006) and insurance claims modeling (David 2015), studies on the course of illness in schizophrenia and linking diet with heart diseases (Skrondal & Rabe-Hesketh 2003), analyzing sophisticated astrophysical data (de Souza et al. 2015), and genomics data (Lobraux & Melodelima 2015).

In most of these applications modern technologies allow to generate more and more data both in terms of the number of observations and in terms of the number of candidate explanatory variables (covariates), bringing a so-called big data issue in terms of both of these aspects. This means that efficient methodology for both estimating a single model based on a fixed choice of covariates and algorithms for searching for the optimal combination of these covariates are becoming important. The first issue on calculation of the posterior distributions of parameters as well as marginal likelihoods within GLMM has been extensively considered in the literature. For linear models with conjugate priors analytic expressions are available (Clyde et al. 2011). In more general settings, MCMC algorithms combined with e.g. Chib’s method (Chib 1995) can be applied, although computational expensive. See also Friel & Wyse (2012) for alternative MCMC based methods. For Gaussian latent variables, the computational task can be efficiently solved through the integrated nested Laplace approximation (INLA) approach (Rue et al. 2009). In Hubin & Storvik (2016) comparisons of INLA and MCMC based methods are presented, showing that INLA based approximations are extremely accurate and require much less computational effort than the MCMC approaches.

The model selection issue, however, still remains an extremely important and difficult problem. This problem in turn can be divided into two sub problems, in particular: which model selection criterion to address based on how well it works in terms of capturing the best model with respect to similarity to the true model (if one exists), and which algorithm to address to optimize the
objective function induced by this criterion on the space of candidate models.

Thus, choice of model selection criterion is the first important issue to consider. In a fully Bayesian context traditional model selection criteria like AIC or BIC become pretty biased. That is why Bayesian alternatives like the Deviance Information Criterion (DIC) and the Watanabe-Akaike Information Criterion (WAIC), Marginal Likelihoods of The Data (MLIK) or Posterior Model Probabilities (PMP) should get addressed. An explicit applied overview of numerous Bayesian model selection criteria is given by Piironen & Vehtari (2016). A more theoretical study is given by Gelman et al. (2014). Different criteria are aimed at different objectives and thus may well disagree in terms of the selected models. Two of the most important goals of model selection according to Gelman et al. (2014) can be viewed as prediction and finding the best model in terms of inference on the true model. WAIC and PMP correspondingly can be viewed as natural model selection criteria in these contexts. In our research PMP is going to be addressed as the criterion for variable selection. Additionally PMPs allow to address the uncertainty in the process of model selection and serve as weights for Bayesian model averaging of any parameters of interest across the models of interest.

The second issue in model selection is purely algorithmic, in particular, we aim at answering how to search for sufficiently good models reasonably fast. Algorithms for stochastic variable selection in the Bayesian settings have been previously addressed, but primarily in the combined space of models and parameters. George & McCulloch (1997) describe and compare various hierarchical mixture prior formulations for Bayesian variable selection in normal linear regression models. Then they describe computational methods including Gray Code sequencing and standard MCMC for posterior evaluation and exploration of the space of models. They also describe infeasibility of exhaustive exploration of the space of models for moderately large problems as well as inability of standard MCMC techniques to escape from local optima efficiently. Ghosh (2015) also addresses MCMC algorithms to estimate the posterior distribution over models. However, she mentions that estimates of posterior probabilities of individual models based on MCMC output are often not reliable because the number of MCMC samples is typically by far smaller than the size of the model space. Bové & Held (2011) consider an MCMC algorithm within the model space in cases where marginal likelihoods are available, but only allow local moves. Clyde et al. (2011) suggest a Bayesian adaptive sampling (BAS) algorithm as an alternative to MCMC, which for small num-
ber of variables carries out full enumeration whilst for larger ones, when the enumeration is not feasible, allows to carry out perfect sampling without replacement. The authors show that their algorithm can, under some conditions, outperform standard MCMC. Song & Liang (2015) address the case when there is by far more explanatory variables than observations and suggest a split and merge Bayesian model selection algorithm that first splits the set of covariates into a number of subsets, then finds relevant variables from these subsets and in the second stage merges these relevant variables and performs a new selection from the merged set. This algorithm in general cannot guarantee convergence to a global optimum or find the true posterior distribution of the models, however under some strict regularity conditions it does so asymptotically. Al-Awadhi et al. (2004) considered using several MCMC steps within a new model to obtain good proposals within the combined parameter and model domain while Yeh et al. (2012) proposed local annealing approaches. Multiple try MCMC methods with local optimization have been described by Liu et al. (2000). These methods fall into the category of generating auxiliary states for proposals (Storvik 2011, Chopin et al. 2013). Yet another approach for Bayesian model selection is addressed by Bottolo et al. (2011), who propose the moves of MCMC between local optima through a permutation based genetic algorithm that has a pool of solutions in a current generation suggested by the parallel tempered chains, which allows to achieve a reasonably good mixing of the chains and escape from local modes at a reasonable rate. A similar idea is considered by Frommlet et al. (2012).

In this paper we introduce a novel MCMC algorithm, the mode jumping MCMC (MJMCMC), for search through the model space. This approach is based on the idea of mode jumping within MCMC - resulting in an MCMC algorithm which manages to efficiently explore the model space by means of introducing locally optimized proposals and thus being able to jump between modes and escape from local optima. Mode jumping MCMC methods within a continuous space setting were first suggested by Tjelmeland & Hegstad (1999). In order to escape from a local mode we first make a large jump, then perform a local optimization and finally a concentric randomization around the local mode. In this article we modify the algorithm to the discrete space of possible models, requiring both new ways of making large jumps and performing local optimization. A specific feature of our approach is the utilization of marginal likelihoods within each model where model parameters are marginalized out. We include mixtures of proposal distributions and parallelization
for further improving the performance of the algorithm. A valid acceptance probability within the Metropolis-Hastings setting is constructed based on the use of backward kernels.

The paper consists of the modeling part (section 2), where statement of the mathematical problem addressed is given; the algorithmic part (section 3), where the main contributions of the paper, namely the suggested MJMCMC algorithm is described; the experimental part (section 4), where further discussion on specification of the particular applied models and application of our algorithms is presented: in particular, we address three examples with a normal Bayesian regression, an example with a logistic regression and finally an example with a Poisson regression with a latent Gaussian field model applied to a real epigenetic data set; the experimental part is followed by the discussion (section 5) with some conclusions and suggestions for further research; a proof of the balance equation for the suggested algorithm and detailed pseudo-codes for local combinatorial optimizers are given in appendix A and B available in the supplementary materials.

2 THE GENERALIZED LINEAR MIXED MODEL

In our notation the data we model via the generalized linear mixed model consists of a response $Y_t$ coming from the exponential family distribution and $p$ covariates $X_{ti}, i \in \{1,...,p\}$ for observations $t \in \{1,...,T\}$. We introduce latent indicators $\gamma_i \in \{0,1\}, i \in \{1,...,p\}$ defining if covariate $X_{ti}$ is included into the model ($\gamma_i = 1$) or not ($\gamma_i = 0$). We are also addressing the unexplained variability of the responses and the correlation structure between them through random effects $\delta_t$ with a specified parametric and sparse covariance matrix structure. Conditioning on the random effect we model the dependence of the responses on the covariates via a proper link function $g(\cdot)$ as in the standard generalized linear regression model, namely:

$$Y_t|\mu_t \sim f(y|\mu_t), \quad t \in \{1,...,T\}$$

$$\mu_t = g^{-1}(\eta_t)$$

$$\eta_t = \beta_0 + \sum_{i=1}^{p} \gamma_i \beta_i X_{ti} + \delta_t$$

$$\delta = (\delta_1, ..., \delta_T) \sim N_T(0, \Sigma_b).$$

Here $\beta_i \in \mathbb{R}, i \in \{0,...,p\}$ are regression coefficients showing in which way the corresponding covariates influence the linear predictor and $\Sigma_b = \Sigma_b(\psi) \in \mathbb{R}^T \times \mathbb{R}^T$ is the covariance structure of
the random effects. We then put relevant priors for the parameters of the model in order to make a fully Bayesian inference:

\[\gamma_i \sim \text{Binom}(1, q)\]  
\[q \sim \text{Beta}(\alpha_q, \beta_q)\]  
\[\beta | \gamma \sim N_{p, p}(\mu_{\beta, \gamma}, \Sigma_{\beta, \gamma})\]

where \(q\) is the prior probability of including a covariate into the model.

Let \(\gamma = (\gamma_1, ..., \gamma_p)\), which uniquely defines a specific model. Then there are \(L = 2^p\) different fixed models. We would like to find a set of the best models of this sort with respect to a certain model selection criterion, namely posterior model probabilities (PMP) - \(p(\gamma | y)\), where \(y\) is the observed data. We are also interested in inference on the simultaneous posterior distribution of the vector of all parameters \(\theta = \{\beta, \psi\}\) and the model \(\gamma\), namely \(p(\theta, \gamma | y)\). We assume that marginal likelihoods \(p(y | \gamma)\) (MLIK) and posterior distributions \(p(\theta | \gamma, y)\) are available for a given \(\gamma\), and then use the proposed MJMCMC algorithm to explore \(p(\gamma | y)\). In order to infer on \(p(\theta, \gamma | y)\), we will address the fact that \(p(\theta, \gamma | y) = p(\theta | \gamma, y)p(\gamma | y)\). PMP's \(p(\gamma | y)\) are given according to Bayes formula as

\[p(\gamma | y) = \frac{p(y | \gamma)p(\gamma)}{\sum_{\gamma' \in \Omega_{\gamma}} p(y | \gamma')p(\gamma')},\]

which keeps the trade-off between \(p(y | \gamma)\) and the prior knowledge about the models incorporated through \(p(\gamma)\). Also notice that in case \(p(\gamma) = p(\gamma')\), \(\forall \gamma, \gamma' \in \Omega_{\gamma}\) maximization of PMP becomes equivalent to maximization of MLIK while in general PMP will be proportional to the product of MLIK and the model prior. In any case, in order to calculate \(p(\gamma | y)\) we have to iterate through the whole model space \(\Omega_{\gamma}\), which becomes computationally infeasible for sets of extremely large cardinality. Therefore we aim at approximating \(p(\gamma | y)\) by means of exploration of some subspace \(V\) of \(\Omega_{\gamma}\) (Clyde et al. 2011), namely:

\[\hat{p}(\gamma | y) = \frac{\mathbb{1}(\gamma \in V)p(y | \gamma)p(\gamma)}{\sum_{\gamma' \in V} p(y | \gamma')p(\gamma')}.\]

Clyde et al. (2011) name these the renormalized (RM) model estimates. Notice that in (10) low \(p(y | \gamma)\) induces both low values of the numerator and small contributions to the denominator in (10), hence low \(p(y | \gamma)\) will have negligible influence on posterior marginal probabilities of other
models. On the other hand, models with high values of \( p(y|\gamma) \) are important to be addressed. This means that modes and near modal values of marginal likelihoods are particularly important for construction of reasonable \( \mathbb{V} \subset \Omega_{\gamma} \) and missing them can dramatically influence our estimates. This builds our motivation to construct an algorithm that is efficiently exploring local modes and near modal values in the space of models and minimizes the amount of visits of models with low posterior mass. In this context the denominator of (10), which we would like to be as high as possible, becomes an extremely relevant measure for the quality of the search in terms of being able to capture whether the algorithm visits all of the modes, whilst the cardinality of \( \mathbb{V} \) is desired to be moderately low in order to save computational time. The problem seems to be pretty challenging, because of both the cardinality of the discrete space \( \Omega_{\gamma} \) growing exponentially fast with respect to the number of covariates and the fact that \( \Omega_{\gamma} \) typically is multimodal in terms of MLIK and PMP, furthermore the modes are often pretty sparsely located. Thus, we have a problem of NP-hard exploration in a sparse non-concave space of models.

For any other important parameters \( \Delta \) the posterior marginal distribution within our notation becomes

\[
p(\Delta|y) = \sum_{\gamma \in \Omega_{\gamma}} p(\Delta|\gamma, y)p(\gamma|y). \tag{11}
\]

For instance posterior marginal inclusion probability \( p(\gamma_j = 1|y) \) can be expressed for \( j \in \{1, \ldots, p\} \) as

\[
p(\gamma_j = 1|y) = \sum_{\gamma' \in \Omega_{\gamma}} \mathbb{1}(\gamma'_j = 1)p(\gamma'|y), \tag{12}
\]

which can be approximated using (10) as

\[
\hat{p}(\gamma_j = 1|y) = \sum_{\gamma' \in \mathbb{V}} \mathbb{1}(\gamma'_j = 1)p(\gamma'|y), \tag{13}
\]

giving a measure for assessing importance of the covariates involved into the search procedure. Both (10) and (13) are consistent, but only asymptotically unbiased. In practice \( p(y|\gamma) \) (MLIK) and \( p(\theta|\gamma, y) \) are often not available analytically. We then rely on some precise approximations \( \hat{p}(y|\gamma) \) and \( \hat{p}(\theta|\gamma, y) \). Such approximations introduce additional errors in (10), but we assume them to be small enough to be ignored. More details on this are given in Hubin & Storvik (2016).
3 MONTE CARLO MARKOV CHAIN

MCMC algorithms (Robert & Casella 2005) and their extensions have been extremely popular for the exploration of model spaces for model selection, being capable (at least in theory) of providing the researchers with samples from the posterior distribution of the models. Typically, these algorithms work in the combined space of both models and parameters. In our research, however, we assume some methods for within model calculations, including marginal likelihoods \( p(y|\gamma) \), are available. Having obtained \( p(y|\gamma) \) we can use equation (10) to approximate posterior model probabilities (9). The most important thing for us then becomes building a method to explore the model space in a way to efficiently switch between potentially sparsely located modes, whilst avoiding visiting models with a low \( p(y|\gamma) \) too often. Moreover we would like to take advantage of the recent advances in technologies and efficiently use multiple cores in our exploration for the methods that can be parallelized. This section first presents theoretical background on MCMC and some theoretical extensions of it that can be addressed to reach our goals. Then we address our MJMCMCs. Finally we discuss tuning of control parameters within the algorithm and opportunities for parallel computing of the suggested algorithm.

3.1 Theoretical background of MCMC

Metropolis-Hastings algorithms (Robert & Casella 2005) are a class of MCMC methods for drawing from a complicated target distribution, which in our setting will be \( \pi(\gamma) = p(\gamma|y) \) living on some model space \( \Omega_\gamma \). Given some proposal distribution \( q(\gamma^*|\gamma) \), the Metropolis-Hastings algorithm accepts the proposed \( \gamma' = \gamma^* \) with probability

\[
rm(\gamma, \gamma^*) = \min \left\{ 1, \frac{\pi(\gamma^*)q(\gamma|\gamma^*)}{\pi(\gamma)q(\gamma^*|\gamma)} \right\}
\]

and otherwise remains in the old state \( \gamma' = \gamma \). This will generate a Markov chain which, given the chain is irreducible and aperiodic, will have \( \pi \) as stationary distribution. Storvik (2011) and Chopin et al. (2013) describe high potential flexibility in choices of proposals by means of generating additional auxiliary states allowing cases where \( q(\gamma^*|\gamma) \) is not directly available. The auxiliary states can be for example chains generated by large jumps combined with local optimizers allowing for jumps to alternative modes. There are different ways to address this flexibility, in particular Storvik (2011) shows that the detailed balance equation under some conditions is
satisfied for the general case addressed further. Assume the current state \( \gamma \sim \pi(\gamma) \). Generate \( (\chi^*, \gamma^*) \sim q(\chi^*, \gamma^*|\gamma) \) as well as \( \chi|\gamma, \chi^*, \gamma^* \sim h(\chi|\gamma, \chi^*, \gamma^*) \) for some arbitrary chosen \( h(\cdot|\cdot) \).

Then \( \chi \) and \( \chi^* \) are auxiliary states. Accept \( \gamma' = \gamma^* \) with the following acceptance probability

\[
    r_m(\chi, \gamma; \chi^*, \gamma^*) = \min\left\{ 1, \frac{\pi(\gamma^*) q(\chi, \gamma^*|\gamma) h(\chi^*|\gamma^*, \chi, \gamma)}{\pi(\gamma) q(\chi^*, \gamma^*|\gamma) h(\chi|\gamma, \chi^*, \gamma^*)} \right\}, \tag{15}
\]

or remain in the previous state otherwise. Then \( \gamma' \sim \pi(\gamma') \). In a typical setting \( \chi^* \) is generated first, followed by \( \gamma^* \). The extra \( \chi \) is needed in order to calculate a legal acceptance probability, relating to a backwards move.

Write now \( q(\chi^*, \gamma^*|\gamma) = q(\chi^*|\gamma) q(\gamma^*|\chi^*, \gamma) \). A convenient choice of \( h \) is \( h(\chi|\gamma, \gamma^*, \chi^*) = q(\chi|\gamma) \) in which case (15) reduces to

\[
    r_m(\gamma, \gamma^*) = \min\left\{ 1, \frac{\pi(\gamma^*) q_r(\gamma|\chi_k)}{\pi(\gamma) q_r(\gamma^*|\chi_k)} \right\}. \tag{16}
\]

Here we only need to consider transitions from the last auxiliary state with kernel \( q_r \). Thus we do not need to store data from the numerous intermediate steps. This seems to be a rather convenient choice in terms of memory and computations, moreover it is extremely similar to the standard Metropolis-Hastings acceptance probabilities and is similar to the acceptance probability used in Al-Awadhi et al. (2004). Acceptance ratios for such a procedure, however, might become relatively low, since in the backward move the probability \( q_r(\gamma|\chi_k) \) might be rather small for the concentric randomization kernels. Alternative choices are considered in appendix B available in the supplementary material.

Within the described procedure we are generating samples from the target distribution, i.e. the posterior model probability - \( p(\gamma|\mathbf{y}) \), which can then be approximated as

\[
    \tilde{p}(\gamma|\mathbf{y}) = \sum_{i=1}^{W} \mathbb{1}(\gamma^{(i)} = \gamma) \xrightarrow{d \ W \to \infty} p(\gamma|\mathbf{y}), \tag{17}
\]

whilst posterior marginal inclusion probabilities (12) for \( j \in \{1, ..., p\} \) can be estimated as

\[
    \tilde{p}(\gamma_j = 1|\mathbf{y}) = \sum_{i=1}^{W} \mathbb{1}(\gamma_j^{(i)} = 1) \xrightarrow{d \ W \to \infty} p(\gamma_j|\mathbf{y}), \tag{18}
\]

where \( W \) is the total number of MCMC samples. Although estimates (17) and (18) are asymptotically consistent, (10) and (12) will often be preferable estimators since convergences of the MCMC based approximations (17) and (18) are much slower.
3.2 Direct proposals

Consider first the standard Metropolis-Hastings algorithm (14). Given that the $\gamma_j$’s are binary, changes corresponds to swaps between the values 0 and 1. One can address various options for generating proposals. A simple proposal is to first select the number of components to change, e.g. $S \sim \text{Unif}\{\zeta,...,\eta\}$, followed by a sample of size $S$ without replacement from $\{1,...,p\}$. This implies that in (14) the proposal probability for switching from $\gamma$ to $\gamma^*$ becomes symmetric, which simplifies calculation of the acceptance probability. Other possibilities for proposals are summarized in Table 1, allowing, among others, different probabilities of swapping for the different components. Such probabilities can for instance be associated with marginal inclusion probabilities from a preliminary MCMC run.

| Type   | Proposal $q(\gamma^*|\gamma)$ probability                                                                 | Label                                           |
|--------|----------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Type 1 | $\prod_{i \in \{\zeta,...,S\}} \rho_i \frac{p \cdot (\eta-\zeta+1)}{(S)^{\zeta}(\eta-\zeta+1)}$          | Random change with random size of the neighborhood |
| Type 2 | $\prod_{i \in \{\zeta,...,S\}} \rho_i \frac{p \cdot (\eta-\zeta+1)}{(S)^{\eta}}$                          | Random change with fixed size of the neighborhood |
| Type 3 | $\frac{1}{(S)^{\zeta}(\eta-\zeta+1)}$                                                                  | Swap with random size of the neighborhood         |
| Type 4 | $\frac{1-1(\sum(y_i)=p)}{p-\sum(y_i=1)\sum(y_i=0)}$                                                     | Swap with fixed size of the neighborhood          |
| Type 5 | $\frac{1-1(\sum(y_i)=p)}{\sum(y_i=1)\sum(y_i=0)}$                                                     | Uniform addition of a covariate                  |
| Type 6 | $\frac{1-1(\sum(y_i)=p)}{\sum(y_i=1)\sum(y_i=0)}$                                                     | Uniform deletion of a covariate                  |

Table 1: Types of proposals suggested for moves between models during an MCMC procedure. Here $S$ is either a deterministic or random ($S \sim \text{Unif}\{\zeta,...,\eta\}$) size of the neighborhood; $\rho_i$ is the probability of inclusion of variable $\gamma_i$.

3.3 MJMCMC - the mode jumping MCMC

All of the proposals in Table 1 might be a good initial way to iterate between models, however when the search space is extremely large with sparsely located modes it might take quite a lot of time to generate a good alternative state and accept the move with these “blindly drawn” proposals. This may lead to low acceptance ratios in terms of jumps between modes and as a result both slow convergence of MCMC and poor exploration of $\Omega_\gamma$. In order to increase the quality of proposals and consequently both improve the acceptance ratio and increase the probability of escaping from local optima, a number of locally optimized proposals can be suggested. To be specific, the more general setup in (15) will be applied. Assume $\chi' = (\chi_0^*, \chi_k^*)$ where now $\chi_0^*$ is
generated according to one of the proposals in Table 1 with a large $S$. $\chi_0^*$ is then modified to $\chi_k^*$ through some optimization or simulation steps in order to move towards a local mode. Finally, $\gamma^*$ is a randomized version of $\chi_k^*$, perhaps again using one of the proposals in Table 1. The procedure is illustrated in Figure 1 where the backward sequence $\gamma^* \rightarrow \chi_0 \rightarrow \chi_k \rightarrow \gamma$, needed for calculating the acceptance probability, is included.

A pseudo-code for the algorithm is given in Algorithm 1. For this algorithm three proposals need to be specified, $q_l(\cdot|\cdot)$ specifying the first large jump (actually specifying the indexes to be swapped in the large jump), $q_o(\cdot|\cdot)$ specifying the local optimizer, and $q_r(\cdot|\cdot)$ specifying the last randomization. In order to avoid the MJMCMC procedure to get stuck in some local mode, we suggest to carry out locally optimized proposals to escape from it at a reasonable frequency $\varrho$ and otherwise make an ordinary MCMC step. This algorithm can be modified to include random choices of different proposal kernels $q_l, q_o, \text{and } q_r$ and parallelized using the multiple try MCMC idea. Technical details of the parallel MJMCMC with mixtures of proposals are given in appendix B. The proof of $\pi$-invariance of the suggested MJMCMC procedures is given in appendix A in the supplementary materials.

In order for the acceptance probability to be high, it is crucial that the auxiliary variables in the
Algorithm 1 Mode jumping MCMC

1: procedure MJMCMC(N)  
2: \( \gamma \leftarrow \gamma_0 \) \hfill \triangleright \text{define the initial state} 
3: for \( t = 1, \ldots, N \) do  
4: \quad if Unif[0; 1] \leq \varrho \text{ then} \hfill \triangleright \text{large jump with local optimization}  
5: \quad \quad \mathbf{J}^* \sim q_l(\cdot | \gamma) \hfill \triangleright \text{indices for large jump}  
6: \quad \quad \mathbf{\chi}_0^* \leftarrow \text{SWAP}(\gamma, \mathbf{J}^*) \hfill \triangleright \text{large jump}  
7: \quad \quad \mathbf{\chi}_k^* \sim q_o(\cdot | \mathbf{\chi}_0^*) \hfill \triangleright \text{local optimization}  
8: \quad \quad \gamma^* \sim q_r(\cdot | \mathbf{\chi}_k^*) \hfill \triangleright \text{randomization around the mode}  
9: \quad \quad \mathbf{\chi}_0 \leftarrow \text{SWAP}(\gamma^*, \mathbf{J}^*) \hfill \triangleright \text{reverse large jump}  
10: \quad \quad \mathbf{\chi}_k \sim q_o(\cdot | \mathbf{\chi}_0) \hfill \triangleright \text{local optimization}  
11: \quad r \leftarrow r_m(\mathbf{\chi}; \gamma; \mathbf{\chi}^*, \gamma^*) \hfill \triangleright \text{from (16)}  
12: \quad \text{else} \hfill \triangleright \text{ordinary proposal}  
13: \quad \quad \gamma^* \sim q(\cdot | \gamma)  
14: \quad \quad r \leftarrow r_m(\gamma, \gamma^*) \hfill \triangleright \text{from (14)}  
15: \quad \text{end if}  
16: \quad \text{if Unif[0; 1] \leq r \ then} \hfill \triangleright \text{accept the move}  
17: \quad \quad \gamma \leftarrow \gamma^*  
18: \quad \text{end if}  
19: \text{end for}  
20: \text{end procedure}  

reverse sequence \( \mathbf{\chi} \) make \( \gamma \) plausible (easiest seen in (16) where \( q(\gamma|\mathbf{\chi}, \gamma^*) \) should be large). One way to achieve this is to choose \( q_l(\mathbf{\chi}_0^*|\gamma) \) to be symmetric, increasing the probability of returning close to the initial mode in the reverse step. The symmetry is achieved by swapping the same set of \( \gamma_j \)'s in the large jumps in the forward simulation as in the backwards simulation. Namely, in large jumps we use \textit{swap with fixed or random size of the neighborhood} kernels from Table 1 with typically reasonably large size of the neighborhood \( S \). We record the components \( \mathbf{J}^* \) that have been swapped. The large jump is followed by local optimization of the target distribution \( \pi(\gamma^*) \) with the transition kernel \( q_o(\mathbf{\chi}_k^*|\mathbf{\chi}_0^*) \) describing the chosen optimizer. In our current implementation we require that only the components that do not correspond to \( \mathbf{J}^* \) can be changed in optimization.
transition kernels. Appendix A.2 shows that the acceptance probability (16) is valid also in this case.

Different local learning and optimization routines can be applied for the generation of $\chi_k^*$, both deterministic and stochastic ones. This allows to efficiently escape from local modes by generating a remote state and then taking a few greedy optimization steps so as to generate a good proposal. In principle, various kinds of heuristics (Pearl 1984) and meta-heuristics (Blum & Roli 2003) including accept the best neighbor, accept the first improving neighbor, forward and backward regression, simulated annealing, threshold acceptance, local MCMC, tabu search, ant colony optimization or genetic combinatorial optimization algorithms can be adopted for the local optimization procedures $q_o(\cdot)$ of MJMCMC, however most of them are pretty expensive computationally. We will consider several feasible computationally options: local simulated annealing (SA) optimization, local greedy optimization, and local MCMC methods. In the implementation of MJMCMC we currently allow parallel versions of the mentioned algorithms. In addition, mixtures of them are allowed. Pseudo-codes for these algorithms as well as the parallelization strategy can be found in appendix B.

Once the optimization is performed, we use the randomizing kernel $q_r$ to make randomization $\gamma^* \sim q_r(\gamma^*|\chi_k^*)$. We typically use randomizing kernels with a high mass on a small neighborhood around the mode but with a positive probability for any change. The two possible appropriate kernels from Table 1 are the random change of either random $S \sim \text{Unif}\{1,\ldots,p\}$ or deterministic $S = p$ number of components with reasonably small but positive probabilities $0 < \rho_i \ll 1$. This guarantees that the MJMCMC procedure is irreducible in $\Omega_\gamma$.

As a rule of thumb, based on suggestions of Tjelmeland & Hegstad (1999) and our own experience, we recommended that in not more than $\varrho = 5\%$ of the cases the jumps between the modes are performed. This is believed to provide the global Markov chain with both good mixing between the modes and accurate exploration of the regions around the modes. However, some tuning might well be required for the particular practical applications.

A small illustrative example with just 10 covariates $x_1,\ldots,x_{10}$ and thus 1024 models in $\Omega_\gamma$ is considered. We generated $Y \sim N(1 + 10x_1 + 0.89x_8 + 1.43x_5, 1)$ with correlated binary covariates (see supplementary material for details) and 1000 observations. We used a Gaussian linear regression with a Zellner’s g-prior (Zellner 1986) with $g = 1000$. This model has tractable MLIKs
Forward

| Model   | log(MLIK) | Model   | log(MLIK) |
|---------|-----------|---------|-----------|
| Initial mode | γ =1010110111 1606.21 | γ ∗=1101100001 1612.27 |
| Large jump | χ ∗=1001110001 1541.51 | χ0 =1110100111 1608.55 |
| Optimize | χ ∗=1101100000 1616.16 | χk =1010100110 1612.00 |
| Randomize | γ ∗=1101100001 1612.27 | γ =1101100001 1606.21 |

Acceptance probability: min \{1, 541.11\}, accept γ' = γ ∗=1101100001

Table 2: Illustration of a typical MJMCMC step with locally optimized proposals. The red components correspond to components swapped in the large jumps, the blue components - to the ones changed in the optimizer, the green components of γ - to the randomization step.

3.4 Calculation of marginal densities

Acceptance probabilities of Metropolis-Hastings algorithm (14), Metropolis-Hastings with auxiliary states algorithm (15), and MJMCMC (16) require calculation of \( \pi(\gamma) = p(\gamma|y) \propto p(\gamma)p(y|\gamma) \), where the complicated part is the marginal density \( p(y|\gamma) \). In practice exact calculation of the marginal density can only be performed in simple models such as linear Gaussian ones, so alternatives need to be considered. The two commonly addressed options are either to use approximative methods or unbiased estimates.

**Approximative methods** Use approximative methods that are accurate enough to neglect the approximation errors involved. Such approximative approaches have been used in various settings of Bayesian variable selection and Bayesian model averaging. Laplace’s method (Tierney & Kadane 1986) has been widely used, but is based on rather strong assumptions. The Harmonic mean estimator (Newton & Raftery 1994) is an easy to implement MCMC based method but can give high variability in the estimates. Chib’s method (Chib 1995), and its extension (Chib & Jeliazkov 2001), and also MCMC based approaches, have gained increasing popularity and
can be very accurate provided enough MCMC iterations are performed. Approximate Bayesian Computation (Marin et al. 2012) has also been considered in this context, being much faster than MCMC alternatives, but also giving cruder approximations. Variational methods (Jordan et al. 1999) provide lower bounds for the marginal likelihoods and have been used for model selection in e.g. mixture models (McGrory & Titterington 2007). Integrated nested Laplace approximation (INLA, Rue et al. 2009) provides accurate estimates of marginal likelihoods within the class of latent Gaussian models. In the context of generalized linear models, Clyde et al. (2011) suggest some approximations provided that informative priors for the coefficients are used.

Unbiased estimates  An alternative is to insert unbiased estimates of $p(y|\gamma)$ into (14). Andrieu & Roberts (2009) name this the pseudo-marginal approach and show that this leads to exact (in the sense of converging to the right distribution) algorithms. Importance sampling (Beaumont 2003) and particle filter (Andrieu et al. 2010) are two approaches that can be used within this setting. In general, the convergence rate will depend on the amount of Monte Carlo effort that is applied. Doucet et al. (2015) provide some guidelines.

Our implementation of the MJMCMC algorithm allows for all of the available possibilities for calculation of marginal likelihoods and assumes that the approximation error can be neglected. In the experiments from section 4 we have applied exact evaluations in the case of linear Gaussian models, approximations based on the assumed informative priors in case of generalized linear models (Clyde et al. 2011), and INLA (Rue et al. 2009) in the case of latent Gaussian models. Bivand et al. (2015) also apply INLA within an MCMC setting, but then concentrating on hyperparameters that (currently) can not be estimated within the INLA framework. Friel & Wyse (2012) performed comparison of some of the mentioned approaches for calculation of marginal likelihoods, including Laplace’s approximations, harmonic mean approximations, Chib’s method and others. Hubin & Storvik (2016) reported some comparisons of INLA and other methods for approximating marginal likelihood. There it is demonstrated that INLA provides extremely accurate approximations on marginal likelihoods in a fraction of time compared to Monte Carlo based methods. Hubin & Storvik (2016) also demonstrated that by means of adjusting tuning parameters within the algorithm (the grid size and threshold values within the numerical integration procedure, Rue et al. 2009) one can often make the difference between INLA and unbiased
methods of estimating of the marginal likelihood arbitrary small.

3.5 Parallelization and tuning parameters of the search

With large number of potential explanatory variables it is important to be able to utilize multiple cores and GPUs of either local machines or clusters in parallel to get the model selection results reasonably fast. General principles of utilizing multiple cores in local optimization are provided in Eksioglu et al. (2002). To put it short, at every step of the optimization one can simultaneously draw several proposals with respect to a certain transition kernel during the optimization procedure and then sequentially calculate the transition probabilities as the proposed models are evaluated by the corresponding CPUs, GPUs or clusters in the order they are returned. In those iterations of MJMCMC where no mode jumps are performed, we are utilizing multiple cores by means of addressing multiple try MCMC to explore the solutions around the current mode. The parallelization strategies are described in detail in appendix B of the paper.

In practice, tuning parameters of the local optimization routines such as the choice of the neighborhood, generation of proposals within it, the cooling schedule for simulated annealing (Michiels et al. 2010) or number of steps in greedy optimization also become crucially important and it yet remains unclear whether we can optimally tune them before or during the search. Tuning the probabilities of addressing different local optimizers and different proposals in the mixture can be beneficial. Such tuning is a sophisticated mathematical problem, which we are not trying to resolve optimally within this article, however we suggest a simple practical idea for obtaining reasonable solutions. Within the BAS algorithm, an important feature was to utilize the marginal inclusion probabilities of different covariates. We have introduced this in our algorithms as well by allowing insertion of estimates of the ρᵢ’s in proposals given in Table 1 based on some burn-in period. These proposals do however have to be combined with other proposals in order to obtain irreducibility. Additional literature review on search parameter tuning can be found in Luo (2016).
4 EXPERIMENTS

In this section we are going to apply the MJMCMC algorithm to different data sets and analyze the results in relation to other algorithms. Initially we address examples from Clyde et al. (2011) to compare the performance of our approach to some existing algorithms such as BAS (Clyde et al. 2011) and competing MCMC methods such as MCMC model composition (MC³, Madigan et al. 1995, Raftery et al. 1997) and the random-swap (RS) algorithm (Clyde et al. 2011). BAS carries out sampling without repetition from the space of models with respect to the adaptively updated marginal inclusion probabilities, whilst both MC³ and RS are simple MCMC procedures based on the standard Metropolis-Hastings algorithm with proposals chosen correspondingly as an inversion or a random change of one coordinate in \( \gamma \) at a time (Clyde et al. 2011). For the cases when full enumeration of the model space is possible we additionally compare all of the aforementioned approaches to the benchmark TOP method that consists of the best quantile of models in terms of the posterior probability for the corresponding number of addressed models \( ||V|| \) and can not by any chance be outperformed in terms of the posterior mass captured. The examples addressed include a simulated data set with 15 covariates and 100 observations, a famous U.S. Crime Data (Raftery et al. 1997) with 15 covariates and 47 observations, a simulated example based on a data set with multiple joint dependencies between 20 covariates with 2000 observations, and a protein activity data with 88 covariates and 96 observations (Clyde et al. 1998). Finally in the last example we address Arabadopsis epigenetic data. This example involves 13 covariates and 1502 observations.

Following Clyde et al. (2011), approximations for model probabilities (10) and marginal inclusion probabilities (13) based on a subspace of models are further referred to as RM (renormalized) approximations, whilst the corresponding MCMC based approximations (17) and (18) are referred to as MC approximations. The validation criteria addressed include biases and squared root of the mean squared errors of parameters of interest based on multiple replications of each algorithm as described by Clyde et al. (2011). In addition to marginal inclusion probabilities, we also include
two global measures

\[ I(\gamma) = \sum_{\gamma' \in \Omega} p(y|\gamma')p(\gamma') - \sum_{\gamma' \in \mathcal{V}} p(y|\gamma')p(\gamma'), \tag{19} \]

\[ C(\gamma) = \frac{\sum_{\gamma' \in \mathcal{V}} p(y|\gamma')p(\gamma')}{\sum_{\gamma' \in \Omega} p(y|\gamma')p(\gamma')} \tag{20} \]

describing the errors made by using only a subset of the models. Mixtures of different proposals from Table 1 and local optimizers mentioned in section 3.3 were used in the studied examples in MJMCMC algorithm. A validation of the gain in using such mixtures is given in example 4.2. The details on the choices and frequencies of different proposals are given in the appendix sections C.1-C.5 in Tables C.2, C.4, C.6, C.8, and C.9. Settings of the combinatorial optimizers for the addressed examples are reported in appendix C, Table C.1.

4.1 Example 1

In this experiment we compare MJMCMC to BAS and competing MCMC methods (MC$^3$, RS) using simulated data following the same linear Gaussian regression model as Clyde et al. (2011) with $p = 15$ and $T = 100$. All columns of the design matrix except for the ninth were generated from independent standard normal random variables and then centered. The ninth column was constructed so that its correlation with the second column was approximately 0.99. The regression parameters were chosen as $\beta_0 = 2$, $\beta = (0.48, 8.72, 1.76, 1.87, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ while the precision $\phi = 1$. For the parameters in each model Zellner’s g-prior with $g = T$ is used. This leads to the marginal likelihood of the model to be proportional to

\[ p(y|\gamma) \propto (1 + g)^{(T-p-1)/2}(1 + g[1 - R_\gamma^2])^{-(T-1)/2}, \tag{21} \]

where $R_\gamma^2$ is the usual coefficient of determination of a linear regression model; with this scaling, the marginal likelihood of the null model (the model containing no covariates) is 1.0. To complete the prior specification, we use (5) with $q = 0.5$. This leads to a rather simple example with two main modes in the model space. Simple approaches are expected to work pretty well in this case. The exact posterior model probabilities may be obtained by enumeration of the model space in this case, making comparison with the truth possible.

In the BAS algorithm 3276 models unique were visited (about 10% of the total number of models). When running the MCMC algorithms approximately the same number of iterations
were used. For the MJMCMC algorithm calculation of marginal likelihoods of models are stored making it unnecessary to recompute these when a model is revisited. Therefore, for MJMCMC also a number of iterations giving the number of unique models visited comparable with BAS was included. For each algorithm 100 replications were performed.

Table 3, showing the root mean squared errors for different quantities, demonstrate that MJMCMC is outperforming simpler MCMC methods in terms of RM approximations of marginal poste-

| Par | True | TOP | MJMCMC | BAS | MC | RS |
|-----|------|-----|--------|-----|----|----|
| $\Delta$ | $\pi_j$ | - | RM | MC | RM | MC | RM | MC | RM |
| $\gamma_{12}$ | 0.09 | 0.29 | 2.11 | 5.31 | 1.19 | 5.73 | 1.23 | 2.77 | 4.27 |
| $\gamma_{14}$ | 0.10 | 0.28 | 2.13 | 6.99 | 1.13 | 6.25 | 1.14 | 2.92 | 4.31 |
| $\gamma_{10}$ | 0.11 | 0.28 | 2.31 | 7.41 | 1.21 | 7.84 | 1.15 | 3.06 | 4.31 |
| $\gamma_{8}$ | 0.12 | 0.27 | 1.97 | 6.44 | 1.09 | 7.80 | 0.97 | 2.77 | 4.01 |
| $\gamma_{6}$ | 0.13 | 0.25 | 2.25 | 8.87 | 1.27 | 8.46 | 1.05 | 3.12 | 4.74 |
| $\gamma_{7}$ | 0.14 | 0.25 | 2.06 | 7.75 | 1.29 | 8.51 | 1.05 | 3.45 | 4.52 |
| $\gamma_{13}$ | 0.15 | 0.24 | 2.42 | 9.98 | 1.36 | 8.79 | 1.15 | 3.50 | 4.87 |
| $\gamma_{11}$ | 0.16 | 0.24 | 2.36 | 9.38 | 1.22 | 8.31 | 1.13 | 3.64 | 4.71 |
| $\gamma_{15}$ | 0.17 | 0.23 | 1.96 | 9.38 | 1.08 | 9.73 | 0.78 | 3.92 | 4.27 |
| $\gamma_{5}$ | 0.48 | 0.00 | 1.22 | 15.66 | 0.50 | 12.90 | 0.27 | 3.69 | 1.41 |
| $\gamma_{9}$ | 0.51 | 0.10 | 1.15 | 16.35 | 0.38 | 12.92 | 0.37 | 16.70 | 5.62 |
| $\gamma_{2}$ | 0.54 | 0.07 | 1.46 | 20.69 | 0.58 | 15.38 | 0.39 | 16.56 | 5.25 |
| $\gamma_{1}$ | 0.74 | 0.18 | 2.15 | 6.43 | 1.06 | 5.97 | 1.20 | 4.10 | 3.55 |
| $\gamma_{3}$ | 0.91 | 0.25 | 1.61 | 3.03 | 0.92 | 3.33 | 1.57 | 2.96 | 3.66 |
| $\gamma_{4}$ | 1.00 | 0.01 | 0.00 | 6.08 | 0.00 | 2.66 | 0.00 | 0.01 | 0.01 |
| $I(\gamma)$ | 0.00 | 1.52 | 7.94 | 39.79 | 3.55 | 36.15 | 3.16 | 33.61 | 25.35 |
| $C(\gamma)$ | 1.00 | 0.99 | 0.89 | 0.89 | 0.95 | 0.95 | 0.95 | 0.72 | 0.72 |
| Eff | $2^{15}$ | 3276 | 1906 | 1906 | 3212 | 3212 | 3276 | 400 | 400 |
| Tot | $2^{15}$ | 3276 | 3276 | 3276 | 6046 | 6046 | 3276 | 3276 | 3276 |

Table 3: Average root mean squared error (RMSE) over the 100 repeated runs of every algorithm on the simulated linear regression data; the values reported in the table are RMSE $\times 10^2$ for $\Delta = \gamma_j$ and RMSE $\times 10^5$ for $I(\gamma)$. Tot is the total number of generated proposals, while Eff is the number of unique models visited during the iterations of the algorithms. RM corresponds to using the renormalization procedure (13) while MC corresponds to using the MC procedure (18). The two runs of MJMCMC are based on different Eff. The corresponding biases are reported in the appendix C.1 in Table C.3.
rior inclusion probabilities, individual model probabilities and the total captured mass. However, the MC approximations seem to be slightly poorer for this example. Whenever both MC and RM approximations are available one should address the latter since they always have less noise. Comparing MJMCMC results to RM approximations provided by BAS (MC are not available for this method), MJMCMC is performing slightly worse when we have 3276 proposals (but 1906 unique models visited). However MJMCMC becomes equivalent to BAS when we consider 6046 proposals with 3212 unique models visited in MJMCMC (corresponding to similar computational time as BAS). In this example we are not facing a really multiple mode issue having just two modes. All MCMC based methods tend to revisit the same states from time to time and for such a simple example one can hardly ever beat BAS, which never revisits the same solutions and simultaneously draws the models to be estimated in a smart and adaptive way with respect to the current marginal posterior inclusion probabilities of individual covariates.

4.2 Example 2

In this example we are going to address a real U.S. Crime data set, first introduced by Vandaele (1978) and much later stated to be a test bed for evaluation of methods for model selection (Raftery et al. 1997). The data set consists of 47 observations on 15 covariates and the responses, which are the corresponding crime rates. We will compare performance of the algorithms based on a linear Bayesian regression model with a Zellner’s g-prior, $g = 47$, making the marginal likelihood to be of form (21).

This is a more sophisticated example with much more local modes, which results in that all simple MCMC methods easily get stuck and have extremely poor performances in terms of the captured mass and precision of both the marginal posterior inclusion probabilities and the posterior model probabilities. For this example MJMCMC gives a much better performance than other MCMC methods in terms of both MC and RM based estimations as well as the posterior mass captured (Table 4). Using 3276 iterations, BAS slightly outperforms MJMCMC. However, when running MJMCMC so that the number of unique models visited ($\|V\|$) are comparable with BAS, MJMCMC gives better results compared to BAS in terms of posterior mass captured, biases and root mean squared errors for both posterior model probabilities and marginal inclusion probabilities (Table 4).
BAS has the property of never revisiting the same solutions, whilst all MCMC based procedures tend to do that with respect to the corresponding posterior probabilities. In the situation when generating a proposal is much cheaper than estimation of the model (which is the case in this example) and we are storing the results for the already estimated models, having generated a bit more models by MJMCMC does not seem to be a serious issue. Those unique models that are visited have a higher posterior mass than those suggested by BAS (for the same number of models visited). Furthermore MJMCMC (like BAS) is guaranteed to escape from local modes and never gets stuck there for an unreasonably long number of steps and thus avoids retardation potentially connected to this issue, common for simpler MCMC procedures. For this example we also report the results of MJMCMC with no mixtures of proposals allowed (labeled MJMCMC* in Table 4) for the case with 3264 unique models visited on average. Here only swaps of 2 components at a time were allowed as the proposals. One can see that MJMCMC* gives much better results than standard MCMC methods, however the results obtained by the MJMCMC algorithm with a mixture of proposals were even better. We have tested this on some other examples too and the use of mixtures was always beneficial and thus recommended. For this reason in other experiments only the cases with mixtures of proposals are addressed.

4.3 Example 3

In the third example we are considering a new simulated data set based on logistic regression. We generated $p = 20$ covariates as a mixture of binary and continuous variables. The correlation structure is shown in Figure 2 while the full details of how the data was generated is given in Appendix C.4 in the supplementary material.

A total of $2^{20} = 1 048 576$ potential models need to be explored. Additionally in this example $T = 2000$, which makes estimation of a single model significantly slower than in the previous examples. We are using the AIC-prior (Clyde et al. 2011) for the regression coefficients of the linear predictor leading to the following approximation of the log marginal likelihood:

$$\hat{p}(y|\gamma) \propto -\frac{1}{2} \left( D(y) + 2 \sum_{i=1}^{p} \gamma_i \right), \quad (22)$$

where $D(y)$ is the deviance for the logistic regression model. This choice was made in order to compare the results with implementations of BAS, RS and MC$^3$ available in the supplementary
| Par | True | TOP | MJMCMC | BAS | MC³ | RS | MJMCMC* |
|-----|------|-----|--------|-----|-----|----|---------|
| Δ   | $\pi_j$ | -   | RM    | MC  | RM  | MC | RM     |
| γ₈  | 0.16 | 3.51 | 6.57  | 10.68 | 5.11 | 10.29 | 5.21 | 6.49 | 3.49 | 5.87 | 3.31 | 6.23 | 9.06 |
| γ₁₃ | 0.16 | 3.34 | 7.46  | 10.54 | 5.60 | 10.19 | 6.26 | 8.62 | 3.39 | 8.83 | 3.05 | 6.38 | 10.54 |
| γ₁₄ | 0.19 | 3.24 | 8.30  | 12.43 | 6.30 | 12.33 | 6.20 | 6.58 | 2.55 | 6.22 | 2.46 | 7.15 | 10.91 |
| γ₁₂ | 0.22 | 3.27 | 6.87  | 13.61 | 5.57 | 13.64 | 3.10 | 5.81 | 6.23 | 4.93 | 6.49 | 3.49 | 5.87 |
| γ₅  | 0.23 | 2.56 | 6.30  | 13.45 | 4.59 | 13.65 | 1.84 | 6.07 | 13.05 | 5.13 | 12.77 | 5.39 | 10.90 |
| γ₉  | 0.23 | 3.27 | 9.49  | 16.21 | 7.40 | 16.21 | 9.27 | 5.99 | 2.99 | 5.70 | 2.60 | 7.68 | 11.06 |
| γ₇  | 0.29 | 2.31 | 4.37  | 13.63 | 3.45 | 12.73 | 2.28 | 4.74 | 9.61 | 3.46 | 9.70 | 3.91 | 10.10 |
| γ₄  | 0.30 | 1.57 | 6.18  | 19.22 | 3.79 | 17.31 | 0.99 | 13.24 | 21.84 | 13.53 | 21.48 | 4.63 | 13.22 |
| γ₆  | 0.33 | 1.92 | 8.61  | 19.71 | 6.14 | 19.49 | 3.11 | 10.19 | 7.47 | 10.99 | 7.12 | 5.87 | 15.43 |
| γ₁  | 0.34 | 2.51 | 11.32 | 22.68 | 7.29 | 20.50 | 8.43 | 22.89 | 25.19 | 23.63 | 24.71 | 7.58 | 12.97 |
| γ₃  | 0.39 | 0.43 | 3.95  | 11.13 | 2.38 | 6.99 | 5.02 | 21.48 | 30.24 | 21.39 | 29.94 | 2.99 | 12.66 |
| γ₂  | 0.57 | 1.58 | 5.92  | 13.21 | 3.82 | 9.03 | 13.78 | 30.81 | 37.57 | 29.27 | 37.15 | 5.11 | 14.04 |
| γ₁₁ | 0.59 | 0.58 | 3.57  | 13.49 | 2.37 | 15.94 | 4.04 | 11.88 | 21.79 | 11.16 | 21.31 | 2.77 | 12.77 |
| γ₁₀ | 0.77 | 3.25 | 7.62  | 7.28  | 5.97 | 4.78 | 15.45 | 21.83 | 19.18 | 20.53 | 19.65 | 6.41 | 14.27 |
| γ₁₅ | 0.82 | 3.48 | 9.23  | 4.45  | 6.89 | 5.85 | 14.50 | 69.68 | 76.81 | 69.19 | 76.30 | 6.75 | 14.76 |
| I(γ) | 0.00 | 11.44 | 16.83 | 24.92 | 10.00 | 22.22 | 12.47 | 34.39 | 45.68 | 34.03 | 44.18 | 16.31 | 24.51 |
| C(γ) | 1.00 | 0.86 | 0.58  | 0.58  | 0.71 | 0.71 | 0.66 | 0.10 | 0.10 | 0.10 | 0.10 | 0.60 | 0.60 |
| Eff | 2¹⁵ | 3276 | 1909  | 1909  | 3237 | 3237 | 3276 | 829  | 829  | 1071 | 1071 | 3264 | 3264 |
| Tot | 2¹⁵ | 3276 | 3276  | 3276  | 5936 | 3276 | 3276 | 3276 | 3276 | 4295 | 4295 |

Table 4: Average root mean squared error (RMSE) over the 100 repeated runs of every algorithm on the Crime data; the values reported in the table are RMSE $\times 10^2$ for $\Delta = \gamma_j$ and RMSE $\times 10^5$ for I(γ). See the caption of Table 3 for further details. The corresponding biases are reported in the appendix C.2 in Table C.5.

to Clyde et al. (2011), where this approximation is considered. In that way, the model search procedures are compared based on the same selection criterion.

Even though the correlation structure between the covariates in this example is generally speaking sparse, one can find quite some significant correlations between the covariates involved. This induces both multimodality of the space of models and sparsity of the locations of the modes and creates an interesting example for comparison of different search strategies. As one can see in Table 5, MJMCMC for the same number of estimated models outperforms both pure
BAS and RS by far in terms of posterior mass captured and root mean square errors of marginal inclusion probabilities and model probabilities. Notice that even for almost two times less originally visited models in $\mathcal{V}$, comparing to BAS, MJMCMC gives almost the same results in terms of the posterior mass captured and errors. Also it is worth mentioning that MJMCMC for the given number of unique models visited did not outperform the combination of MCMC and BAS, that is recommended by Clyde et al. (2011) for larger model spaces; both of them gave approximately identical results.

4.4 Example 4

This experiment is based on a much larger model space in comparison to all of the other examples. We address the protein activity data (Clyde et al. 1998) and consider all main effects together with the two-way interactions and quadratic terms of the continuous covariates resulting in 88 covariates in total. This corresponds to the model space of cardinality $2^{88}$. This model space is additionally multimodal, which is the result of having high correlations between numerous of the addressed covariates (17 pairs of covariates have correlations above 0.95). We analyze the data set using the Bayesian linear regression with a Zellner’s g-prior, $g = 96$ (the data has 96 observations). We then compare the performance of MTMCMC, BAS and RST. The reported RST results are based on the RS algorithm run for $88 \times 2^{20}$ iterations and a thinning rate $\frac{1}{88}$. BAS was run with several choices of initial sampling probabilities such as uniformly distributed within the model space one, eplogp (Clyde et al. 2011) adjusted, and those based on RM and MC approximations obtained by
| Par | True | TOP | MJMCMC | BAS | MCBAS | RS |
|-----|------|-----|--------|-----|-------|----|
| Δ   | π_j  | -   | RM  | MC | RM  | MC | RM  | RM |
| γ_6 | 0.29 | 0.00 | 7.38 | 15.54 | 4.54 | 16.62 | 6.47 | 3.67 | 6.01 | 2.11 |
| γ_8 | 0.31 | 0.00 | 6.23 | 15.50 | 3.96 | 16.94 | 5.58 | 3.02 | 5.37 | 2.55 |
| γ_{12} | 0.35 | 0.00 | 4.86 | 14.62 | 2.78 | 13.66 | 4.22 | 2.12 | 3.91 | 2.37 |
| γ_{15} | 0.35 | 0.00 | 4.55 | 15.24 | 2.56 | 15.45 | 4.66 | 1.64 | 3.40 | 2.56 |
| γ_2  | 0.36 | 0.00 | 4.90 | 16.52 | 2.92 | 17.39 | 5.42 | 2.45 | 3.65 | 2.61 |
| γ_{20} | 0.37 | 0.00 | 4.82 | 14.35 | 2.66 | 14.08 | 3.32 | 1.80 | 4.15 | 2.18 |
| γ_3  | 0.40 | 0.00 | 9.25 | 20.93 | 5.65 | 22.18 | 9.75 | 4.82 | 6.76 | 2.83 |
| γ_{14} | 0.44 | 0.00 | 3.14 | 17.54 | 1.58 | 16.24 | 3.73 | 1.30 | 1.33 | 2.93 |
| γ_{10} | 0.44 | 0.00 | 4.60 | 18.73 | 2.29 | 17.90 | 4.87 | 1.30 | 1.51 | 2.42 |
| γ_5  | 0.46 | 0.00 | 3.10 | 17.17 | 1.53 | 16.97 | 4.06 | 1.51 | 1.09 | 2.85 |
| γ_9  | 0.61 | 0.00 | 3.68 | 16.29 | 1.63 | 13.66 | 3.89 | 1.39 | 2.19 | 2.35 |
| γ_4  | 0.88 | 0.00 | 5.66 | 6.70 | 3.74 | 6.26 | 6.60 | 5.57 | 7.61 | 2.15 |
| γ_{11} | 0.91 | 0.00 | 5.46 | 6.81 | 3.95 | 6.90 | 4.66 | 3.14 | 4.32 | 1.57 |
| γ_1  | 0.97 | 0.00 | 1.90 | 1.74 | 1.35 | 1.34 | 2.43 | 1.96 | 2.30 | 1.1 |
| γ_{13} | 1.00 | 0.00 | 0.00 | 0.43 | 0.00 | 0.32 | 0.00 | 0.00 | 0.00 | 0.37 |
| γ_7  | 1.00 | 0.00 | 0.00 | 0.57 | 0.00 | 0.41 | 0.00 | 0.00 | 0.00 | 0.33 |
| γ_{16} | 1.00 | 0.00 | 0.00 | 0.41 | 0.00 | 0.33 | 0.00 | 0.00 | 0.00 | 0.23 |
| γ_{17} | 1.00 | 0.00 | 0.00 | 0.43 | 0.00 | 0.39 | 0.00 | 0.00 | 0.00 | 0.23 |
| γ_{18} | 1.00 | 0.00 | 0.00 | 0.47 | 0.00 | 0.35 | 0.00 | 0.00 | 0.00 | 0.24 |
| γ_{19} | 1.00 | 0.00 | 0.00 | 0.52 | 0.00 | 0.36 | 0.00 | 0.00 | 0.00 | 0.41 |
| I(γ) | 0.00 | 0.00 | 1.36 | 2.95 | 0.69 | 2.72 | 1.21 | 0.63 | 1.54 | 2.42 |
| Cap  | 1.00 | 1.00 | 0.72 | 0.72 | 0.85 | 0.85 | 0.74 | 0.85 | 0.68 | 0.68 |
| Eff  | 2^{20} | 10000 | 5148 | 5148 | 9988 | 9988 | 10000 | 10000 | 1889 | 1889 |
| Tot  | 2^{20} | 10000 | 9998 | 9998 | 19849 | 19849 | 10000 | 10000 | 10000 | 10000 |

Table 5: Average root mean squared error (RMSE) from the 100 repeated runs of every algorithm on the simulated logistic regression data; the values reported in the table are RMSE $\times 10^2$ for $\Delta = \gamma_j$ and RMSE $\times 10^5$ for $I(\gamma)$. See the caption of Table 3 for further details. The corresponding biases are reported in the appendix C.3 in Table C.7.

the RST algorithm. For the first two initial sampling probabilities BAS was run for $2^{20}$ iterations, whilst for the latter two first RST was run for $88 \times 2^{19}$ iterations providing $2^{19}$ models for estimating
initial sampling probabilities and then BAS was run for the other $2^{19}$ iterations. MJMCMC was run until $2^{20}$ unique models were obtained. All of the algorithms were run on 10 replications. In Figure 3 box-plots of the best 100,000 models captured by the corresponding replications of the algorithms as well as posterior masses captured by them are displayed. BAS with both uniform and eplogp initial sampling probabilities perform rather poorly in comparison to other methods, whilst BAS combined with RM approximations from RST as well as MJMCMC show the most promising results. BAS with RM initial sampling probabilities usually manages to find models with the highest MLIK, however MJMCMC in general captures by far higher posterior mass within the same amount of unique models addressed. Marginal inclusion probabilities obtained by the best run of MJMCMC with a mass of $8.56 \times 10^{20}$ are reported in Figure 3, whilst those obtained by other methods can be found in Clyde et al. (2011). Since MJMCMC obtains the highest posterior mass, we expect that the corresponding RM estimates of the marginal inclusion probabilities are the least biased, moreover they perfectly agree with MC approximations obtained by MJMCMC ensuring that the procedure has reached the stationary mode. Although MJMCMC in all of the obtained replications outperformed the competitors in terms of the posterior mass captured, it itself exhibits significant variation between the runs (right panel of Figure 3). The latter issue can be explained by that we are only allowing visiting $3.39 \times 10^{-19}\%$ of the total model space in
the addressed replications, which might be not enough to always converge to the same posterior mass captured depending on the starting point. Note however that the variability in the results obtained from different runs of MJMCMC clearly indicates that more iterations are needed, while the other methods may indicate (wrongly) that sufficient iterations are performed.

4.5 Example 5

In this example we illustrate how MJMCMC works for GLMM models. As illustration, we address genomic and epigenomic data on Arabadopsis. Arabadopsis is a plant model organism with a lot of genomic/epigenomic data easily available (Becker et al. 2011). At each position on the genome, a number of reads are allocated. At locations with a nucleotide of type cytosine nucleobase (C), reads are either methylated or not. Our focus will be on modeling the amount of methylated reads through different covariates including (local) genomic structures, gene classes and expression levels. The studied data was obtained from the NCBI GEO archive (Barrett et al. 2013).

We model the number of methylated reads $Y_t, \in \{1, \ldots, R_t\}$ per loci $t, \in \{1, \ldots, T\}$ to be Poisson distributed with a mean $\mu_t, \in \mathbb{R}^+$ modeled via the log link to the chosen covariates $X_t = \{X_{t,1}, \ldots, X_{t,p}\}, t, \in \{1, \ldots, T\}$, an offset defined by the total number of reads per location $R_t, \in \mathbb{N}$, and a spatially correlated random effect, which is modeled via an AR(1) process with parameter $\rho, \in \mathbb{R}$, namely $\delta_t = \rho \delta_{t-1} + \epsilon_t, \in \mathbb{R}$ with $\epsilon_t \sim N(0, \sigma^2_\epsilon), t, \in \{1, \ldots, T\}$. Thus, we take into
account spatial dependence structures of methylation rates along the genome as well as the variance of the observations not explained by the covariates. Since in general the ratio of methylated bases is low, we have preferred the Poisson distribution of the responses to the binomial. We then put relevant priors for the parameters of the model in order to make a fully Bayesian inference. We use the binomial priors (5) for $\gamma$ and the Gaussian priors (7) for the regression coefficients. For the parameters within the random effects, we first reparametrize to $\psi_1 = \log \frac{1}{\sigma^2_{\epsilon,t}}(1 - \rho^2)$, $\psi_2 = \log \frac{1+\rho}{1-\rho}$ and assume

$$\psi_1 \sim \text{Gamma}(1, 5 \times 10^{-5}),$$

$$\psi_2 \sim \text{N}(0, 0.15^{-1}).$$

We have addressed $p = 13$ different covariates in addition to the intercept. Among these covariates we address a factor with 3 levels corresponding to whether a location belongs to a CGH, CHH or CHG genetic region, where H is either A, C or T and thus generating two covariates $X_1$ and $X_2$ corresponding to whether a location is CGH or CHH. The second group of factors indicates whether a distance to the previous cytosine nucleobase (C) in DNA is 1, 2, 3, 4, 5, from 6 to 20 or greater than 20 inducing the binary covariates $X_3 - X_8$. The third factor corresponds to whether a location belongs to a gene from a particular group of genes of biological interest, these groups are indicated as $M_\alpha$, $M_\gamma$, $M_\delta$ or $M_0$ inducing 3 additional covariates $X_9 - X_{11}$. Finally, we have
Table 6: Average root mean squared error (RMSE) from the 100 simulated runs of MJMCMC on the epigenetic data; the values reported in the table are RMSE $\times 10^2$ for $\Delta = \gamma_j$; RMSE $\times 10^5$ for $I(\gamma)$.

As one can see from Table 6 within just the 385 best unique models (2.35% of the total model space) we are able to capture almost full posterior mass for this problem. The model space, as shown in Figure 6, has very few sparsely located modes in a pretty large model space. In this example we compare MJMCMC and a simple RS MCMC algorithm, the latter is allowed to only swap one component per iteration. This example does contain most of the mass in just two closely located models as can be seen in Figure 6. That is why a simple MCMC can capture essentially
most of the mass after 10 000 iterations. At the same time there are a few small modes that lie a bit further from the region of the high concentration of mass, which the simple MCMC algorithm could not capture. Essentially, MCMC was staying within a few modes for most of the time, never being able to travel to the more remote parts of the model space and generating very few (155 on average) unique models. The number of unique models (155) is here very low compared to the total number of models visited (10 000). If there were more sparsely located remote modes the simple MCMC algorithm would run into the problems similar to those discussed in the previous examples and miss a significant amount of mass. For MJMCMC, we run the algorithm until 3160 models where visited, resulting in 1758 unique models. As a result, MJMCMC captures the mass from the remote small modes, adding a bit to the captured mass, which allows it to slightly outperform the simple MCMC algorithm. As can be seen in Table 6, MJMCMC outperforms the simple MCMC algorithm in terms of the errors of marginal model probabilities for both RM approach and the MC approach. Marginal inclusion probabilities in terms of RM are also more precise when MJMCMC is used, but the MC based approximations of them are slightly better for the MCMC case.
According to marginal inclusion probabilities, factors of whether the location is CGH or CHH are both extremely significant, as well as the higher cut off for the level of expression. Additionally factors for $M_\alpha$ and $M_\delta$ groups of genes have non-zero marginal inclusion probabilities and reasonably high significance. In future it would be of an interest to obtain additional covariates such as whether a nucleobase belongs to a particular part of the gene like promoter or a coding region. Furthermore, it is definitely of interest to address factors whether a base is located within a CpG island or whether it belongs to a transposone. Moreover interactions of these covariates may be interesting. Alternative choices of the response distributions (e.g. binomial or negative binomial) and/or type of random effects ($AR(k)$, $ARMA(l,k)$) might also be of an interest.

5 SUMMARY AND DISCUSSION

In this article we have introduced the mode jumping MCMC (MJMCMC) approach for estimating posterior model probabilities and performing Bayesian model averaging and selection. The algorithm incorporates the ideas of MCMC with the possibility of large jumps combined with local optimizers to generate proposals in the discrete space of models. Unlike standard MCMC methods applied to variable selection, the developed procedure avoids getting stuck in local modes and manages to iterate through all of the important models much faster. It also in many cases outperforms Bayesian Adaptive Sampling (BAS), having the tendency to capture a higher posterior mass within the same amount of unique models visited. This can be explained by that for problems with numerous covariates BAS requires good initial marginal inclusion probabilities to perform well. Clyde et al. (2011) demonstrated that estimates of marginal inclusion probabilities obtained from preliminary MCMC runs could largely improve BAS. A combination of MJMCMC with BAS could possibly improve both algorithms even further.

The EMJMCMC R-package is developed and currently available from the Git Hub repository: http://aliaksah.github.io/EMJMCMC2016/. The methodology depends on the possibility of calculating marginal likelihoods within models accurately. The developed package gives a user high flexibility in the choice of methods to obtain marginal likelihoods. Whilst the default choice for marginal likelihood calculations is based on INLA (Rue et al. 2009), we also have adopted efficient C based implementations for exact calculations in Bayesian linear regression and Bayesian logistic and Poisson regressions with g-priors as well as other priors. Several model selection criteria for
the class of methods are also addressed. Extensive parallel computing for both MCMC moves and local optimizers is available within the developed package; in particular, with a default option a user specifies how many threads are addressed within the in-build `mclapply` function or `snow` based parallelization, however an advanced user can specify his own function to parallelize computations on both the MCMC and local optimization levels taping, for instance, modern graphical processing units - GPUs, which in turn allows additional efficiency and flexibility.

Whilst estimators (10) for marginal inclusion and posterior model probabilities based on Bayes formula and obtained by MJMCMC, as noticed by Clyde et al. (2011), are Fisher consistent, they remain generally speaking biased; although their bias reduces to zero asymptotically. MCMC based estimators such as (17) or (18), which are both consistent and unbiased, are also available through our procedure; these estimators however tend to have a much higher variance than the aforementioned ones. As one of the further developments it would be of an interest to combine knowledge available from both groups of estimators to adjust for bias and variance, which is vital for higher dimensional problems.

Another aspect that requires being discussed is the model selection criteria. As stated in the introduction, WAIC, DIC and PMP can sometimes disagree about the results of model selection. In order to avoid confusion, the researcher should be clear about the stated goals. If the goal is prediction rather than inference one should adjust for that and use AIC, WAIC or DIC rather than BIC pr PMP as selection criterion in MJMCMC. These choices are possible within the `EMJMCMC` package as well.

Based on several experiments, we can claim MJMCMC to be a rather competitive algorithm that is addressing the wide class of Generalized Linear Mixed Models (GLMM). In particular for this class of models one can incorporate a random effect, which both models the variability unexplained by the covariates and introduces dependence between observations, creating additional modeling flexibility. Estimations of parameters of such models and Bayesian inference within them becomes significantly harder in comparison to simple GLM. This creates the necessity to address parallel computing extensively. We have enabled the latter within our package by means of combining methods for calculating marginal likelihoods, such as the INLA methodology, and parallel MJMCMC algorithm.

Currently we use decision variables only on the level of choice of covariates, however the
mode jumping procedure can be easily extended to more general cases. In future it would be of interest to extend the procedure to model selection and model averaging jointly across covariates, link functions, random effect structures and response distributions. Such extensions will require even more accurate tuning of control parameters of the algorithm introducing another important direction for further research.

SUPPLEMENTARY MATERIAL

R package: R package EMJMCMC to perform the efficient mode jumping MCMC described in the article. (EMJMCMC_1.2.tar.gz; GNU zipped tar file). The package is also maintained at [http://aliaksah.github.io/EMJMCMC2016/](http://aliaksah.github.io/EMJMCMC2016/)

Data and code: Data (simulated and real) and R code for MJMCMC algorithm, post-processing and creating figures wrapped together into a reference based EMJMCMC class. (code-and-data.zip; zip file containing the data, code and a read-me file (readme.pdf))

Proofs and Pseudo code: Proofs of the ergodicity of MJMCMC procedure, pseudo codes for MJMCMC and local combinatorial optimizers, parallelization strategies and some supplementary tables for the experiments. (appendix.pdf)

ACKNOWLEDGMENTS

The authors gratefully acknowledge the CELS project at the University of Oslo, [http://www.mn.uio.no/math/english/research/groups/cels/index.html](http://www.mn.uio.no/math/english/research/groups/cels/index.html), for giving us the opportunity, inspiration and motivation to write this article.

REFERENCES

Al-Awadhi, F., Hurn, M. & Jennison, C. (2004), ‘Improving the acceptance rate of reversible jump MCMC proposals’, *Statistics and Probability Letters* **69**(2), 189 – 198.

Andrieu, C., Doucet, A. & Holenstein, R. (2010), ‘Particle Markov chain Monte Carlo methods’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**(3), 269–342.

Andrieu, C. & Roberts, G. O. (2009), ‘The pseudo-marginal approach for efficient Monte Carlo computations’, *The Annals of Statistics* **2**(2), 697–725.
Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M. et al. (2013), ‘NCBI GEO: archive for functional genomics data sets update’, *Nucleic acids research* **41**(D1), D991–D995.

Beaumont, M. A. (2003), ‘Estimation of population growth or decline in genetically monitored populations’, *Genetics* **164**(3), 1139–1160.

Becker, C., Hagmann, J., Müller, J., Koenig, D., Stegle, O., Borgwardt, K. & Weigel, D. (2011), ‘Spontaneous epigenetic variation in the arabidopsis thaliana methylome’, *Nature* **480**(7376), 245–249.

Bivand, R., Gómez-Rubio, V. & Rue, H. (2015), Spatial data analysis with R-INLA with some extensions, American Statistical Association.

Blum, C. & Roli, A. (2003), ‘Metaheuristics in combinatorial optimization: Overview and conceptual comparison’, *ACM Computing Surveys (CSUR)* **35**(3), 268–308.

Bottolo, L., Chadeau-Hyam, M., Hastie, D. I., Langley, S. R., Petretto, E., Tiret, L., Tregouet, D. & Richardson, S. (2011), ‘ESS++: a C++ object-oriented algorithm for Bayesian stochastic search model exploration’, *Bioinformatics* **27**(4), 587–588.

Bové, D. S. & Held, L. (2011), ‘Bayesian fractional polynomials’, *Statistics and Computing* **21**(3), 309–324.

Chib, S. (1995), ‘Marginal likelihood from the Gibbs output’, *Journal of the American Statistical Association* **90**(432), 1313–1321.

Chib, S. & Jeliazkov, I. (2001), ‘Marginal likelihood from the Metropolis–Hastings output’, *Journal of the American Statistical Association* **96**(453), 270–281.

Chopin, N., Jacob, P. E. & Papaspiliopoulos, O. (2013), ‘SMC2: an efficient algorithm for sequential analysis of state space models’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75**(3), 397–426.

Clyde, M. A., Ghosh, J. & Littman, M. L. (2011), ‘Bayesian adaptive sampling for variable selection and model averaging’, *Journal of Computational and Graphical Statistics* **20**(1), 80–101.

Clyde, M., Parmigiani, G. & Vidakovic, B. (1998), ‘Multiple shrinkage and subset selection in wavelets’, *Biometrika* **85**(2), 391–401.

David, M. (2015), ‘Auto insurance premium calculation using generalized linear models’, *Procedia Economics and Finance* **20**, 147 – 156.

de Souza, R., Cameron, E., Killeidar, M., Hilbe, J., Vilalta, R., Maio, U., Biffi, V., Ciardi, B. & Riggs, J. (2015), ‘The overlooked potential of generalized linear models in astronomy, i: Binomial regression’, *Astronomy and Computing* **12**, 21 – 32.
Doucet, A., Pitt, M. K., Deligiannidis, G. & Kohn, R. (2015), ‘Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator’, *Biometrika* **102**(2), 295.

Eksioglu, S. D., Pardalos, P. M. & Resende, M. G. (2002), Parallel metaheuristics for combinatorial optimization, in R. Corra, I. Dutra, M. Fiallos & F. Gomes, eds, ‘Models for Parallel and Distributed Computation’, Vol. 67 of *Applied Optimization*, Springer US, pp. 179–206.

Friel, N. & Wyse, J. (2012), ‘Estimating the evidence – a review’, *Statistica Neerlandica* **66**(3), 288–308.

Frommlet, F., Ljubic, I., Arnardttir Helga, B. & Bogdan, M. (2012), ‘QTL Mapping Using a Memetic Algorithm with Modifications of BIC as Fitness Function’, *Statistical Applications in Genetics and Molecular Biology* **11**(4), 1–26.

Gelman, A., Hwang, J. & Vehtari, A. (2014), ‘Understanding predictive information criteria for Bayesian models’, *Statistics and Computing* **24**(6), 997–1016.

George, E. I. & McCulloch, R. E. (1997), ‘Approaches for Bayesian variable selection’, *Statistica Sinica* **7**(2), 339–373.

Ghosh, J. (2015), ‘Bayesian model selection using the median probability model’, *Wiley Interdisciplinary Reviews: Computational Statistics* **7**(3), 185–193.

Grossi, L. & Bellini, T. (2006), Credit risk management through robust generalized linear models, in S. Zani, A. Cerioli, M. Riani & M. Vichi, eds, ‘Data Analysis, Classification and the Forward Search’, Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, pp. 377–386.

Hubin, A. & Storvik, G. (2016), ‘Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA)’. arXiv:1611.01450v1.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. & Saul, L. K. (1999), ‘An introduction to variational methods for graphical models’, *Machine learning* **37**(2), 183–233.

Liu, J. S., Liang, F. & Wong, W. H. (2000), ‘The multiple-try method and local optimization in Metropolis sampling’, *Journal of the American Statistical Association* **95**(449), 121–134.

Lobraux, S. & Melodelima, C. (2015), ‘Detection of genomic loci associated with environmental variables using generalized linear mixed models’, *Genomics* **105**(2), 69 – 75.

Luo, G. (2016), ‘A review of automatic selection methods for machine learning algorithms and hyper-parameter values’, *Network Modeling Analysis in Health Informatics and Bioinformatics* **5**(1), 1–16.

Madigan, D., York, J. & Allard, D. (1995), ‘Bayesian graphical models for discrete data’, *International Statistical Review/Revue Internationale de Statistique* pp. 215–232.
Marin, J.-M., Pudlo, P., Robert, C. P. & Ryder, R. J. (2012), ‘Approximate Bayesian computational methods’, *Statistics and Computing* **22**(6), 1167–1180.

McGrory, C. A. & Titterington, D. (2007), ‘Variational approximations in Bayesian model selection for finite mixture distributions’, *Computational Statistics & Data Analysis* **51**(11), 5352–5367.

Michiels, W., Aarts, E. & Korst, J. (2010), *Theoretical Aspects of Local Search*, 1st edn, Springer Publishing Company, Incorporated.

Newton, M. A. & Raftery, A. E. (1994), ‘Approximate Bayesian inference with the weighted likelihood bootstrap’, *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 3–48.

Pearl, J. (1984), *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Piironen, J. & Vehtari, A. (2016), ‘Comparison of bayesian predictive methods for model selection’, *Statistics and Computing* pp. 1–25.

Raftery, A. E., Madigan, D. & Hoeting, J. A. (1997), ‘Bayesian model averaging for linear regression models’, *Journal of the American Statistical Association* **92**(437), 179–191.

Robert, C. P. & Casella, G. (2005), *Monte Carlo Statistical Methods*, Springer Texts in Statistics, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Rohde, D. L. T. (2002), ‘Methods for binary multidimensional scaling’, *Neural Comput.* **14**(5), 1195–1232.

Rue, H., Martino, S. & Chopin, N. (2009), ‘Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations’, *Journal of the Royal Statistical Society* **71**(2), 319–392.

Skrodnal, A. & Rabe-Hesketh, S. (2003), ‘Some applications of generalized linear latent and mixed models in epidemiology: repeated measures, measurement error and multilevel modeling’, *Norwegian Journal of Epidemiology* **13**(2), 265–278.

Song, Q. & Liang, F. (2015), ‘A split-and-merge Bayesian variable selection approach for ultrahigh dimensional regression’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **77**(5), 947–972.

Storvik, G. (2011), ‘On the flexibility of Metropolis-Hastings acceptance probabilities in auxiliary variable proposal generation’, *Scandinavian Journal of Statistics* **38**, 342–358.

Stroup, W. W. (2013), *Generalized linear mixed models : modern concepts, methods and applications*, CRC Press, Taylor & Francis, Boca Raton.
Tierney, L. & Kadane, J. B. (1986), ‘Accurate approximations for posterior moments and marginal densities’, *Journal of the american statistical association* **81**(393), 82–86.

Tjelmeland, H. & Hegstad, B. K. (1999), ‘Mode jumping proposals in MCMC’, *Scandinavian journal of statistics* **28**, 205–223.

Vandaele, W. (1978), ‘Participation in illegitimate activities: Ehrlich revisited’, *Deterrence and Incapacitation* **1**, 270–335.

Yeh, Y. T., Yang, L., Watson, M., Goodman, N. & Hanrahan, P. (2012), ‘Synthesizing open worlds with constraints using locally annealed reversible jump MCMC’, *ACM Transactions on Graphics* **31**(4), 56–58.

Zellner, A. (1986), ‘On assessing prior distributions and bayesian regression analysis with g-prior distributions’, *Bayesian inference and decision techniques: Essays in Honor of Bruno De Finetti* **6**, 233–243.

**APPENDIX**

**A  MCMC WITH AUXILIARY VARIABLES**

**A.1  $\pi$ invariance**

Assume the current state to be $\gamma \sim \pi(\gamma)$. We generate $(\gamma^*, \chi^*) \sim q(\gamma^*, \chi^*|\gamma)$ and $\chi|\gamma, \chi^*, \gamma^* \sim h(\chi|\gamma, \chi^*, \gamma^*)$. The new state $\gamma^*$ is accepted with probability

$$r_m(\chi, \gamma; \chi^*, \gamma^*) = \min \left\{ 1, \frac{\pi(\gamma^*)q(\chi, \gamma|\gamma^*)h(\chi^*|\gamma^*, \chi)\pi(\gamma)q(\chi^*, \gamma^*|\gamma)h(\chi|\gamma, \chi^*, \gamma^*)}{\pi(\gamma)q(\chi^*, \gamma^*|\gamma)h(\chi|\gamma, \chi^*, \gamma^*)} \right\}$$

This approach, with $\chi$ and $\chi^*$ are auxiliary variables as auxiliary variables, was considered by Storvik (2011). We will here derive an alternative proof of $\pi$-invariance in $\Omega_\gamma$. In the next subsection we will show that the MJMCMC algorithm is a special case of this general MCMC algorithm.

- First, we will show that the constructed Markov chain is $\pi$-invariant in space $\Omega_\gamma$. Let $A(\gamma; \gamma^*)$ be the transition kernel for the Markov chain. We will show that the detailed balance is satisfied by means integrating out the auxiliary states $\chi$ and $\chi^*$ with respect to the chosen $q(\cdot|\cdot)$ and $h(\cdot|\cdot)$ functions in the case of $\gamma^* \neq \gamma$:...
Consider now the MJMCMC algorithm where \( A.2 \) Derivation of the acceptance probability for MJMCMC

\[ \pi(\gamma)A(\gamma; \gamma^*) = \pi(\gamma) \int_{\Omega} \int_{\Omega^*} q(\gamma^*, \gamma^* | \gamma) h(\gamma | \gamma^*, \gamma^*) r_m(\gamma, \gamma; \gamma^*) d\gamma d\gamma^* \]
\[ = \int_{\Omega} \int_{\Omega^*} \pi(\gamma) q(\gamma^*, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*) \times \]
\[ \min \left\{ 1, \frac{\pi(\gamma) q(\gamma^*, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*)}{\pi(\gamma) q(\gamma^*, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*)} \right\} d\gamma^* d\gamma \]
\[ = \int_{\Omega} \int_{\Omega^*} \pi(\gamma) q(\gamma, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*) \times \]
\[ \min \left\{ \frac{\pi(\gamma) q(\gamma, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*)}{\pi(\gamma) q(\gamma, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*)} \times 1 \right\} d\gamma^* d\gamma^* \]
\[ = \pi(\gamma^*) \int_{\Omega} \int_{\Omega^*} q(\gamma, \gamma | \gamma) h(\gamma^* | \gamma, \gamma^*) r_m(\gamma, \gamma^*; \gamma, \gamma) \]
\[ = \pi(\gamma^*) A(\gamma^*; \gamma). \]

Note that in the derivation above change of the order of integration is always possible according to Fubini’s theorem, since all of the addressed probability measures are measurable and bounded by definition. Thus, we have shown the Markov chain to be \( \pi \)-invariant in \( \Omega_\gamma \). ▷

A.2 Derivation of the acceptance probability for MJMCMC

Consider now the MJMCMC algorithm where \( \chi^* = (J^*, \chi_0^*, \chi_k^*) \), with \( J^* \) being the set of components of \( \gamma \) that are swapped, \( \chi_0^* \) being a deterministic function of \( \gamma \) and \( J^* \) while \( \chi_k^* \) is the result after local optimization. We further define \( \chi = \chi_k \). Then, \( q(\gamma^*, \chi^* | \gamma) \) = \( q(J^*) q_0(\chi_k^* | \gamma) q_r(\gamma^* | \chi_k^*) \) while \( h(\gamma | \gamma^*, \gamma^*) = q_0(\chi_k | \chi_0) \) where \( \chi_0 \) is a deterministic function of \( \gamma^* \) and \( J^* \). Then

\[ r_m(\chi, \gamma; \chi^*, \gamma^*) = \min \left\{ 1, \frac{\pi(\gamma^*) q(\chi, \gamma | \gamma^*) h(\gamma^* | \gamma^*; \gamma, \gamma)}{\pi(\gamma) q(\chi^*, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*)} \right\} \]
\[ = \min \left\{ 1, \frac{\pi(\gamma^*) q(J^*) q_0(\chi_k^* | \chi_0) q_r(\gamma^* | \chi_k^*) q_0(\chi_k | \chi_0)}{\pi(\gamma) q(J^*) q_0(\chi_k^* | \chi_0) q_r(\gamma^* | \chi_k^*) q_0(\chi_k | \chi_0)} \right\} \]
\[ = \min \left\{ 1, \frac{\pi(\gamma^*) q_r(\gamma | \chi_k)}{\pi(\gamma) q_r(\gamma^* | \chi_k^*)} \right\}. \]

showing equation (16).

An alternative algorithm is to use \( \chi_k \sim q_r(\cdot | \gamma^*) \) with the symmetric constraint \( q_r(\gamma | \gamma') = q_r(\gamma' | \gamma) \). In that case we obtain

\[ r_m(\chi, \gamma; \chi^*, \gamma^*) = \min \left\{ 1, \frac{\pi(\gamma^*) q(\chi, \gamma | \gamma^*) h(\gamma^* | \gamma^*; \gamma, \gamma)}{\pi(\gamma) q(\chi^*, \gamma^* | \gamma) h(\gamma^* | \gamma, \gamma^*)} \right\} \]
\[ = \min \left\{ 1, \frac{\pi(\gamma^*) q(J^*) q_0(\chi_k^* | \chi_0) q_r(\gamma | \chi_k) q_0(\chi_k | \chi_0)}{\pi(\gamma) q(J^*) q_0(\chi_k^* | \chi_0) q_r(\gamma^* | \chi_k^*) q_0(\chi_k | \chi_0)} \right\} \]
\[ = \min \left\{ 1, \frac{\pi(\gamma^*) q_0(\chi_k | \chi_0)}{\pi(\gamma) q_0(\chi_k | \chi_0)} \right\}. \quad (A-1) \]
which focuses on the performance of the proposed mode in comparison to the current state and the last step of the optimization procedure. Here however \( q_\theta (\chi_k | \chi_0) \) may exhibit similar issues as \( q_r (\gamma | \chi_k) \) in (16). If further we apply a deterministic local optimizer, this reduces to

\[
\begin{align*}
    r_m (\chi, \gamma; \chi^*, \gamma^*) = \min \left\{ 1, \frac{\pi(\gamma^*)}{\pi(\gamma)} \right\}
\end{align*}
\] (A-2)

which results in a pure comparison of the current and proposed states. We do not restrict the reader to either use (16) or (A-1)-(A-2) and suggest that (15) in general are adopted in order to obtain the optimal acceptance ratio for the given model and data within the suggested MJMCMC algorithm.

### B DETAILS OF THE MJMCMC ALGORITHM

#### B.1 Choice of proposal distributions

The implementation of MJMCMC allows for great flexibility in the choices of proposal distributions for the large jumps, the local optimization and the last randomization.

- Table 1 lists the current possibilities for drawing indexes to swap in the first large jump. In this setting, one would choose distributions such that there is a large number of components that are swapped.

- An important ingredient of the MJMCMC algorithm is the choice of local optimizer. In the current implementation of the algorithm, several choices are possible; simulated annealing, greedy optimizers based on best neighbor optimization or first improving neighbor (Blum & Roli 2003) which is another variant of greedy local search accepting the first randomly selected solution better than the current. For each alternative the neighbors are defined through swapping a few of the \( \gamma_j \)'s in the current model.

- For the last randomization, again Table 1 lists the possibilities, but in this case a small number of swaps will be preferable.

Different possibilities to combine the optimizers and proposals in a hybrid setting are also possible. Then, at each iteration, which proposal distributions and which optimizer to use are randomly drawn from the set of possibilities, see Robert & Casella (2005, sec 10.3) for the validity of such procedures.

#### B.2 Multiple try MCMC algorithm

In addition to ordinary MCMC steps and mode jump MCMC, also multiple-try Metropolis (Liu et al. 2000) is considered in an extension of Algorithm 1. Multiple-try Metropolis (MTMCMC) is a sampling method that is a modified form of the Metropolis-Hastings method, designed to be able to properly parallelize the original Metropolis-Hastings algorithm. The idea of the method is to allow generating \( S \) trial proposals \( \chi^*_1, ..., \chi^*_S \) in parallel from a proposal distribution \( q_\theta (\cdot | \gamma) \). Then within the trial set \( \gamma^* \in \{ \chi^*_1, ..., \chi^*_S \} \) is selected with probability proportional to some importance weights \( w(\gamma, \chi^*_i) = \pi(\gamma) q(\chi^*_i | \gamma) \lambda(\chi^*_i, \gamma) \) where \( \lambda(\chi^*_i, \gamma) = \lambda(\gamma, \chi^*_i) \). In the reversed move
\(x_1, \ldots, x_{S-1}\) are generated from the proposal \(q(x|\gamma^*)\) and we put \(x_S = \gamma\). Finally, the move is accepted with probability

\[
r_m(\gamma, \gamma^*) = \min \left\{ 1, \frac{w(x_1, \gamma) + \cdots + w(x_S, \gamma)}{w(x_1, \gamma^*) + \cdots + w(x_S, \gamma^*)} \right\}.
\]

In the implementation of the algorithm, ordinary MCMC is considered as a special case of MTMCMC with \(S = 1\). We recommend (MT)MCMC is used in at least 95% of the iterations of MJMCMC. Simultaneously several sequential steps of MTMCMC can be seen as an alternative local combinatorial optimizer.

### B.3 Parallel computing in local optimizers

General principles of utilizing multiple cores in local optimization are provided in Eksioglu et al. (2002). One can simultaneously draw several proposals with respect to a certain transition kernel \(s_o(|\cdot|)\) during every step of the optimization procedure and then sequentially calculate the transition probabilities as the proposed models are evaluated by the corresponding CPUs, GPUs or clusters. Consider an optimizer with the acceptance probability function \(r^i_o(x; \gamma^*), i \in 1, \ldots, N\), which either changes over the time (iterations) or remains unchanged. For the greedy local search \(r^i_o(x; \gamma^*) = 1 \{ p(y|x) p(y) \geq p(y|x^*) p(y) \}, i \in 1, \ldots, N\). While for the implemented version of the simulated annealing algorithm we consider \(r^i_o(x; \gamma^*) = \min \left\{ 1, \exp \left( \frac{p(y|x)p(y) - p(y|x^*)p(y)}{t_i} \right) \right\}, i \in 1, \ldots, N\), where \(t_i\) is the SA temperature (Blum & Roli 2003) parameter at iteration \(i\). The proposed parallelization strategy is given in detail in Algorithm B.1.

**Algorithm B.1** Parallel optimization

1: procedure OPTIMIZE(N)
2: \(X^* \leftarrow X_0^*\)
3: for \(i = 1, \ldots, N\) do
4: \(X_{i,1}, \ldots, X_{i,K} \sim s_o(|\cdot|X^*)\) \(\triangleright\) make \(K\) proposals in parallel
5: for \(j = 1, \ldots, K\) do
6: \(r \leftarrow r^i_o(X_{i,j}; X^*)\) \(\triangleright\) calculate acceptance probability
7: if \(\text{Unif}[0; 1] \leq r\) then
8: \(X^* \leftarrow X_{i,j}\) \(\triangleright\) accept the transition
9: end if
10: end for
11: \(X_i^* \leftarrow X^*\)
12: end for
13: return \(X_N^*\)
14: end procedure
B.4 Parallel MJMCMC with a mixture of proposals

We suggest to use the auxiliary states to expand the MCMC with local optimization ideas to a given problem in a rather flexible way, based on (15). Here we also described the fully parallel version of Algorithm 1. The suggested MJMCMC approach allows to both mix between the local modes efficiently and explore the solutions around the modes simultaneously utilizing multiple CPUs and GPUs of a single machine or a cluster of nodes, whilst keeping the desired ergodicity of the MJMCMC procedure. Moreover this modification of Algorithm 1 allows for the mixtures of both local optimizers and proposals addressed within MJMCMC. The pseudo-code of the algorithm is given in Algorithm B.2 of the appendix. In this pseudo-code we consider the following notation:

- $\rho$ - the probability deciding if a large jump with local optimization is used; $P_g(\cdot)$ - the distribution for the choice of the MTMCMC proposal kernels;
- $P_o(\cdot)$ - the distribution for the choice of the local optimizers’ kernels;
- $P_l(\cdot)$ - the distribution for the choice of large jump transition kernel;
- $P_r(\cdot)$ - the distribution for the choice of the randomizing kernel;

The essential differences of the parallel version of the MJMCMC with a mixture of proposals (Algorithm B.2) from Algorithm 1, thus, are as follows:

- MTMCMC steps instead of MCMC steps are performed for the steps with no mode jumps;
- At the iterations with no mode jumps the proposal is chosen randomly as $q_g \sim P_g(\zeta)$;
- At the iterations with mode jumps the large jump proposals $q_l \sim P_l(\zeta)$, the optimization proposals $q_o \sim P_o(\zeta)$, and the randomizing kernels $q_r \sim P_r(\zeta)$ are chosen randomly;
- The optimization steps are parallelized as in section B.3.
### C SUPPLEMENTARY MATERIALS FOR THE EXPERIMENTS

| Example No | CPU Num | SA $S_t$ | $\Delta t$ | $t_0$ | $t_f$ | Greedy S | LS | FI | MTMCMC Size | Steps |
|------------|---------|---------|---------|-------|-------|--------|----|----|-----------|-------|
| 1          | 4       | 4       | 3       | 10    | $14 \times 10^{-5}$ | 15     | F  | T  | 4         | 15    |
| 2          | 4       | 4       | 3       | 10    | $14 \times 10^{-5}$ | 15     | F  | T  | 4         | 15    |
| 3          | 2       | 5       | 3       | 10    | $14 \times 10^{-5}$ | 20     | F  | T  | 2         | 20    |
| 4          | 10      | 18      | 3       | 10    | $14 \times 10^{-5}$ | 88     | F  | T  | 10        | 88    |
| 5          | 1       | 3       | 3       | 10    | $14 \times 10^{-5}$ | 13     | F  | T  | 2         | 13    |

Table C.1: Tuning parameters of the blocks of MJMCMC in the examples (Example No); CPU (Num) - the number of CPUs utilized within the examples; $S_t$ - number of iterations per temperature in SA algorithm; $\Delta t$ - cooling factor of the cooling schedule of SA algorithm; $t_0$ - initial temperature of SA algorithm; $t_f$ - final temperature of SA algorithm; S - number of iterations in Greedy algorithm (per run); LS - if local stop is allowed in Greedy algorithm; FI - if the first improving neighbor strategy is applied in Greedy algorithm; Size - number of proposals per step in MTMCMC algorithm; Steps - number of MTMCMC steps (only makes sense when MTMCMC is used as an optimizer).
C.1 Example 1

| Proposal | Optimizer | Frequency | Type 1  | Type 4  | Type 3  | Type 5  | Type 6  | Type 2  |
|----------|-----------|-----------|---------|---------|---------|---------|---------|---------|
| $q_g$    | -         | $\rho = 0.9836$ | 0.1176  | 0.3348  | 0.2772  | 0.0199  | 0.2453  | 0.0042  |
| $S$      | -         | -         | $\{2, 2\}$ | 2       | $\{2, 2\}$ | 1       | 1       | 15      |
| $\rho_i$ | -         | -         | $\tilde{p}(\gamma_i|y)$ | -       | -       | -       | -       | $\tilde{p}(\gamma_i|y)$ |
| $q_l$    | -         | 0.0164    | 0       | 1       | 0       | 0       | 0       | 0       |
| $S$      | -         | -         | -       | 4       | -       | -       | -       | -       |
| $\rho_i$ | -         | -         | -       | -       | -       | -       | -       | -       |
| $q_o$    | SA        | 0.5553    | 0.0788  | 0.3942  | 0.1908  | 0.1928  | 0.1385  | 0.0040  |
| $q_o$    | GREEDY    | 0.2404    | 0.0190  | 0.3661  | 0.2111  | 0.2935  | 0.1046  | 0.0044  |
| $q_o$    | MTMCMC    | 0.2043    | 0.2866  | 0.1305  | 0.2329  | 0.1369  | 0.2087  | 0.0040  |
| $S$      | -         | -         | $\{2, 2\}$ | 2       | $\{2, 2\}$ | 1       | 1       | 15      |
| $\rho_i$ | -         | -         | $\tilde{p}(\gamma_i|y)$ | -       | -       | -       | -       | $\tilde{p}(\gamma_i|y)$ |
| $q_r$    | -         | -         | 0       | 0       | 0       | 0       | 0       | 1       |
| $S$      | -         | -         | -       | -       | -       | -       | -       | -       |
| $\rho_i$ | -         | -         | -       | -       | -       | -       | -       | 0.0010  |

Table C.2: Other tuning parameters of MTMCMC for all proposal types ($q_g, q_l, q_o,$ and $q_r$) in example 1; Optimizer - to which optimizer the proposal belongs (if not relevant "-"); Frequency - the frequency at which the proposal is addressed ($\rho$ for $q_g$ and $1 - \rho$ for $q_l$) and the frequency within the set of local optimizers ($P_o$ for local optimizers); Type X - the frequency of proposal of type X Table 1; $S$ - maximal allowed size of the neighborhood for the corresponding proposal; $\rho_i$ - probability of change of component $i$ of the current solution (if applicable to the proposal).
| Par | True | TOP | MJMCMC | BAS | MC³ | RS |
|-----|------|-----|--------|-----|-----|----|
| Δ   | π_j  | -   | RM     | MC  | RM  | MC |
| γ₁₂ | 0.09 | -0.29 | -2.11  | -4.95 | -1.19  | -5.47 | -1.23 | -0.14 | -4.21 | 0.35 | -3.80 |
| γ₁₄ | 0.10 | -0.28 | -2.12  | -6.58 | -1.12  | -6.07 | -1.14 | -0.23 | -4.23 | 0.05 | -3.89 |
| γ₁₀ | 0.11 | -0.28 | -2.30  | -6.89 | -1.30  | -7.64 | -1.14 | -0.10 | -4.23 | 0.11 | -4.02 |
| γ₈  | 0.12 | -0.27 | -1.96  | -6.16 | -1.08  | -7.69 | -0.97 | 0.36  | -3.94 | -0.51 | -3.81 |
| γ₆  | 0.13 | -0.25 | -2.24  | -8.03 | -1.26  | -8.33 | -1.05 | -0.65 | -4.64 | 0.06 | -4.24 |
| γ₇  | 0.14 | -0.25 | -2.05  | -7.45 | -1.28  | -8.37 | -1.04 | -0.13 | -4.41 | 0.08 | -4.12 |
| γ₁₃ | 0.15 | -0.24 | -2.39  | -9.62 | -1.35  | -8.62 | -1.15 | -0.49 | -4.76 | 0.28 | -4.32 |
| γ₁₁ | 0.16 | -0.24 | -2.33  | -8.69 | -1.21  | -7.95 | -1.13 | -0.38 | -4.59 | -0.10 | -4.44 |
| γ₁₅ | 0.17 | -0.23 | -1.93  | -7.64 | -1.06  | -9.59 | -0.78 | -0.58 | -4.15 | -0.19 | -3.74 |
| γ₅  | 0.48 | 0.00  | -1.15  | -14.18 | -0.47  | -11.97 | -0.25 | -0.29 | -0.94 | 0.46 | -1.17 |
| γ₉  | 0.51 | -0.10 | 0.78   | 13.11 | 0.23   | 11.96 | -0.32 | -1.79 | -2.20 | -0.22 | -1.53 |
| γ₂  | 0.54 | -0.07 | -1.21  | -18.43 | -0.50  | -14.64 | 0.34  | 1.73  | 0.29  | 0.35 | -0.25 |
| γ₁  | 0.74 | 0.18  | 2.12   | 4.88  | 1.04   | 3.99  | 1.19  | -0.23 | 3.39  | 0.41 | 3.69 |
| γ₃  | 0.91 | 0.25  | 1.60   | -1.79 | 0.91   | 0.03  | 1.56  | -0.40 | 3.59  | -0.14 | 4.00 |
| γ₄  | 1.00 | 0.01  | 0.00   | -5.94 | 0.00   | -2.49 | 0.00  | 0.01  | 0.01  | -0.02 | 0.01 |
| I(γ) | 0.00 | 1.52 | 7.54 | 36.68 | 3.38 | 34.85 | 3.04 | 3.72 | 22.57 | 2.95 | 20.57 |
| C(γ) | 1.00 | 0.99 | 0.89 | 0.89 | 0.95 | 0.95 | 0.72 | 0.72 | 0.74 | 0.74 |
| Eff | 2¹⁵ | 3276 | 1906 | 1906 | 3212 | 3212 | 3276 | 400 | 400 | 416 | 416 |
| Tot | 2¹⁵ | 3276 | 3276 | 3276 | 6046 | 6046 | 3276 | 3276 | 3276 | 3276 | 3276 |

Table C.3: Bias for the 100 simulated runs of every algorithm on the simulated data; the values reported in the table are Bias × 10² for Δ = γ_j and Bias × 10⁵ for I(γ), eq (19). C(γ) is as defined in (20). Tot is the total number of generated proposals, while Eff is the number of unique models visited during the iterations of the algorithms. RM corresponds to using the re-normalization procedure (13) while MC corresponds to using the MC procedure (18).
## C.2 Example 2

| Proposal | Optimizer | Frequency | Type 1 | Type 4 | Type 3 | Type 5 | Type 6 | Type 2 |
|----------|-----------|-----------|--------|--------|--------|--------|--------|--------|
| $q_g$    | -         | $g = 0.9836$ | 0.1176 | 0.3348 | 0.2772 | 0.0199 | 0.2453 | 0.0042 |
| $S$      | -         | -         | $\{2, 2\}$ | 2     | $\{2, 2\}$ | 1    | 1     | 15     |
| $\rho_i$ | -         | -         | $\hat{p}(\gamma_i|y)$ | -    | -    | -    | -    | $\hat{p}(\gamma_i|y)$ |
| $q_l$    | -         | 0.0164    | 0      | 1      | 0      | 0      | 0      | 0      |
| $S$      | -         | -         | -      | 4      | -      | -      | -      | -      |
| $\rho_i$ | -         | -         | -      | -      | -      | -      | -      | -      |
| $q_o$    | SA        | 0.5553    | 0.0788 | 0.3942 | 0.1908 | 0.1928 | 0.1385 | 0.0040 |
| $q_o$    | GREEDY    | 0.2404    | 0.0190 | 0.3661 | 0.2111 | 0.2935 | 0.1046 | 0.0044 |
| $q_o$    | MTMCMC    | 0.2043    | 0.2866 | 0.1305 | 0.2329 | 0.1369 | 0.2087 | 0.0040 |
| $S$      | -         | -         | $\{2, 2\}$ | 2     | $\{2, 2\}$ | 1    | 1     | 15     |
| $\rho_i$ | -         | -         | $\hat{p}(\gamma_i|y)$ | -    | -    | -    | -    | $\hat{p}(\gamma_i|y)$ |
| $q_r$    | -         | -         | 0      | 0      | 0      | 0      | 0      | 1      |
| $S$      | -         | -         | -      | -      | -      | -      | -      | 15     |
| $\rho_i$ | -         | -         | -      | -      | -      | -      | -      | 0.0010 |

Table C.4: Other tuning parameters of MTMCMC for all proposal types ($q_g$, $q_l$, $q_o$, and $q_r$) in example 2; see Table C.2 for details. Notice that for MJMCMC* reported in the example only proposals of type 4 are used.
| Par | True | TOP | MJMCMC | BAS | MC<sup>3</sup> | RS | MJMCMC* |
|-----|------|-----|--------|-----|-------------|----|----------|
| Δ   |   j  | -RM | -MC   | RM  | MC          | RM | RM       |
| γ₈  | 0.16 | -3.51 | -6.54 | -10.28 | -5.09 | -9.64 | -5.19 | 5.37 | -3.20 | 4.96 | -3.06 | 6.23 | 9.06 |
| γ₁₃ | 0.16 | -3.34 | -7.44 | -10.12 | -5.57 | -9.94 | -6.25 | 7.46 | 2.86 | 8.06 | 2.65 | 6.38 | 10.54 |
| γ₁₄ | 0.19 | -3.24 | -8.27 | -11.69 | -6.28 | -11.93 | -6.19 | 5.27 | -1.86 | 5.37 | -2.03 | 7.15 | 10.91 |
| γ₁₂ | 0.22 | -3.27 | -6.82 | -12.91 | -5.54 | -13.15 | -3.08 | 3.00 | -5.82 | 3.76 | -5.06 | 5.29 | 10.93 |
| γ₅  | 0.23 | -2.56 | -6.21 | -12.71 | -4.55 | -13.35 | -1.80 | -4.79 | -12.98 | -4.28 | -12.72 | 5.39 | 10.90 |
| γ₉  | 0.23 | -3.27 | -9.45 | -15.67 | -7.35 | -16.11 | -9.26 | 4.53 | -2.45 | 4.33 | -2.10 | 7.68 | 11.06 |
| γ₇  | 0.29 | -2.31 | -4.15 | -12.04 | -3.41 | -12.36 | -2.24 | -0.47 | -9.41 | -1.00 | -9.56 | 3.91 | 10.10 |
| γ₄  | 0.30 | -1.57 | -5.82 | -18.74 | -3.67 | -17.10 | 0.85 | -12.67 | -21.79 | -13.24 | -21.45 | 4.63 | 13.22 |
| γ₆  | 0.33 | -1.92 | -8.49 | -19.07 | -6.09 | -18.84 | -3.06 | 8.99 | 7.16 | 10.09 | 6.81 | 5.87 | 15.43 |
| γ₁  | 0.34 | -2.51 | -11.25 | -21.94 | -7.25 | -20.29 | -8.42 | 22.36 | 25.10 | 23.32 | 24.63 | 7.58 | 12.97 |
| γ₃  | 0.39 | -0.43 | 3.51 | -7.20 | 2.09 | -4.43 | 4.98 | -21.11 | -30.20 | -21.13 | -29.92 | 2.99 | 12.66 |
| γ₂  | 0.57 | 1.58 | 5.66 | -8.73 | 3.71 | -7.51 | 13.73 | -30.41 | -37.52 | -29.05 | -37.12 | 5.11 | 14.04 |
| γ₁₁ | 0.59 | 0.58 | 2.86 | 11.75 | 2.13 | 15.32 | -3.95 | 10.67 | 21.68 | 10.29 | 21.23 | 2.77 | 12.77 |
| γ₁₀ | 0.77 | 3.25 | 7.50 | -2.57 | 5.91 | 2.33 | 15.42 | -21.22 | -19.06 | -20.01 | -19.55 | 6.41 | 14.27 |
| γ₁₅ | 0.82 | 3.48 | 9.17 | 0.22 | 6.85 | 3.65 | 14.50 | -69.61 | -76.81 | -69.14 | -76.30 | 6.75 | 14.76 |
| I(γ) | 0.00 | 11.44 | 15.49 | 17.75 | 9.28 | 18.78 | 11.86 | 27.33 | 44.10 | 27.15 | 42.58 | 16.31 | 24.51 |
| C(γ) | 1.00 | 0.86 | 0.58 | 0.58 | 0.71 | 0.71 | 0.66 | 0.10 | 0.10 | 0.10 | 0.10 | 0.60 | 0.60 |
| Eff | 2<sup>15</sup> | 3276 | 1909 | 1909 | 3237 | 3237 | 3276 | 829 | 829 | 1071 | 1071 | 3264 | 3264 |
| Tot | 2<sup>15</sup> | 3276 | 3276 | 3276 | 5936 | 5936 | 3276 | 3276 | 3276 | 3276 | 3276 | 4295 | 4295 |

Table C.5: Bias for the 100 simulated runs of every algorithm on the Crime data; the values reported in the table are Bias $\times 10^2$ for $\Delta = \gamma_j$ and Bias $\times 10^5$ for I(γ). See the caption of Table C.3 for further details.
In the addressed data set the true regression parameters were chosen to be \( \beta_0 = 99 \) for the intercept, and for the slope coefficients \( \beta = (-4, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 2, 0, 37.1, 0, 0, 50, -0.00005, 10, 3, 0) \). What concerns the covariates, \( X_1 \) and \( X_3 \) are factors from a group with 3 levels, \( X_4 \) and \( X_6 \) are correlated to them factors from another group with 3 levels, \( X_7 \) and \( X_8 \) are two jointly dependent through copulas exponentially distributed variables with rate 0.3, \( X_9, X_{10} \) and \( X_{11} \) are all uniformly distributed with range from -1 to 10 and also jointly dependent through copulas, \( X_{12}, X_{13}, X_{14} \) and \( X_{15} \) are multivariate normal with a zero mean, standard deviation of 0.2 and some covariance structure, \( X_{16} \) represents some seasonality incorporated by the sinus transformation of the radiant representation of some angle equal to the corresponding ordering numbers of observations, \( X_{17} \) is the quadratic trend associated to the squared value of positions of observations, \( X_{19} = (-4 + 5X_1 + 6X_3)X_{15} \) and \( X_{20} = (-4 + 5X_1 + 6X_3)X_{11} \), finally to avoid over specification 2 layers from the mentioned above groups of factors were replaced with some auxiliary covariates \( X_2 = (X_{10} + X_{14}) \times X_9 \) and \( X_5 = (X_{11} + X_{15}) \times X_{12} \). The linear predictor is drawn as \( \eta \sim N(\beta'X, 0.5) \), whilst the observations \( Y \) are independent Bernoulli variables with the probability of success modeled by a logit transformation of the linear predictor, namely \( Y \sim Bernoulli \left( p = \frac{\exp(\eta)}{1 + \exp(\eta)} \right) \).  

### Table C.6: Other tuning parameters of MTMCMC for all proposal types (\( q_g, q_l, q_o, \) and \( q_r \)) in example 3; see Table C.2 for details.

| Proposal | Optimizer | Frequency | Type 1 | Type 4 | Type 3 | Type 5 | Type 6 | Type 2 |
|----------|-----------|-----------|--------|--------|--------|--------|--------|--------|
| \( q_g \) | -         | 0.9820    | 0.1179 | 0.3357 | 0.2779 | 0.0200 | 0.2459 | 0.0021 |
| \( S \)  | -         | \( \{1,1\} \) | 1      | \( \{1,1\} \) | 1      | 1      | 20     |
| \( \rho_i \) | -         | -         | \( \tilde{p}(\gamma_i|y) \) | -      | -      | -      | -      |
| \( q_l \) | -         | 0.0180    | 0      | 1      | 0      | 0      | 0      | 0      |
| \( S \)  | -         | -         | -      | 5      | -      | -      | -      | -      |
| \( \rho_i \) | -         | -         | -      | -      | -      | -      | -      | -      |
| \( q_o \) | SA        | 0.5042    | 0.0636 | 0.3249 | 0.1571 | 0.2288 | 0.2246 | 0.0009 |
| \( q_o \) | GREEDY    | 0.2183    | 0.0160 | 0.3085 | 0.1779 | 0.2474 | 0.2493 | 0.0007 |
| \( q_o \) | MTMCMC    | 0.2774    | 0.2879 | 0.3016 | 0.1582 | 0.1107 | 0.1401 | 0.0013 |
| \( S \)  | -         | \( \{1,1\} \) | 1      | \( \{1,1\} \) | 1      | 1      | 20     |
| \( \rho_i \) | -         | -         | \( \tilde{p}(\gamma_i|y) \) | -      | -      | -      | -      |
| \( q_r \) | -         | -         | 0      | 0      | 0      | 0      | 0      | 1      |
| \( S \)  | -         | -         | -      | -      | -      | -      | -      | 20     |
| \( \rho_i \) | -         | -         | -      | -      | -      | -      | -      | 0.0010 |
| Par | True | TOP  | MJMCMC | BAS | MCBAS | RS  |
|-----|------|------|--------|-----|-------|-----|
| $\Delta$ | $\pi_j$ | - | RM  | MC  | RM  | MC  | RM  | RM  | RM  | MC  |
| $\gamma_6$ | 0.29 | 0.00 | -7.23 | -14.89 | -4.48 | -16.40 | -6.46 | -3.59 | -5.96 | 0.23 |
| $\gamma_8$ | 0.31 | 0.00 | -5.97 | -13.94 | -3.89 | -16.57 | -5.57 | -2.85 | -5.28 | -0.35 |
| $\gamma_{12}$ | 0.35 | 0.00 | -4.07 | -8.12 | -2.56 | -11.65 | -4.20 | -1.82 | -3.80 | 0.06 |
| $\gamma_{15}$ | 0.35 | 0.00 | -3.66 | -8.85 | -2.21 | -12.04 | -4.58 | -1.35 | -3.25 | -0.28 |
| $\gamma_2$ | 0.36 | 0.00 | -4.60 | -14.71 | -2.81 | -16.80 | -5.39 | -2.19 | -3.51 | 0.04 |
| $\gamma_{20}$ | 0.37 | 0.00 | -4.16 | -8.38 | -2.46 | -12.03 | -3.30 | -1.75 | -4.07 | -0.12 |
| $\gamma_3$ | 0.40 | 0.00 | -8.99 | -19.22 | -5.58 | -21.72 | -9.73 | -4.63 | -6.69 | 0.23 |
| $\gamma_{14}$ | 0.44 | 0.00 | 1.08 | 7.12 | 0.51 | 7.63 | 3.68 | -0.62 | -0.99 | 0.22 |
| $\gamma_{10}$ | 0.44 | 0.00 | -2.68 | -7.62 | -1.68 | -11.89 | -4.79 | -0.29 | -1.19 | 0.13 |
| $\gamma_5$ | 0.46 | 0.00 | -1.74 | -10.78 | -0.88 | -12.29 | -3.93 | 0.57 | 0.55 | -0.23 |
| $\gamma_9$ | 0.61 | 0.00 | 0.32 | -2.29 | 0.00 | -1.24 | 3.78 | 0.22 | 1.99 | -0.11 |
| $\gamma_4$ | 0.88 | 0.00 | 5.61 | 6.20 | 3.71 | 6.13 | 6.60 | 5.54 | 7.58 | -0.45 |
| $\gamma_{11}$ | 0.91 | 0.00 | 5.36 | 6.47 | 3.87 | 6.84 | 4.64 | 3.01 | 4.29 | -0.28 |
| $\gamma_1$ | 0.97 | 0.00 | 1.86 | 0.98 | 1.32 | 1.17 | 2.43 | 1.94 | 2.28 | -0.31 |
| $\gamma_{13}$ | 1.00 | 0.00 | 0.00 | -0.33 | 0.00 | -0.29 | 0.00 | 0.00 | 0.00 | -0.3 |
| $\gamma_7$ | 1.00 | 0.00 | 0.00 | -0.41 | 0.00 | -0.36 | 0.00 | 0.00 | 0.00 | -0.27 |
| $\gamma_{16}$ | 1.00 | 0.00 | 0.00 | -0.33 | 0.00 | -0.31 | 0.00 | 0.00 | 0.00 | -0.17 |
| $\gamma_{17}$ | 1.00 | 0.00 | 0.00 | -0.38 | 0.00 | -0.35 | 0.00 | 0.00 | 0.00 | -0.17 |
| $\gamma_{18}$ | 1.00 | 0.00 | 0.00 | -0.37 | 0.00 | -0.32 | 0.00 | 0.00 | 0.00 | -0.19 |
| $\gamma_{19}$ | 1.00 | 0.00 | 0.00 | -0.40 | 0.00 | -0.32 | 0.00 | 0.00 | 0.00 | -0.34 |
| $\mathbf{I}(\gamma)$ | 0.00 | 0.00 | 1.05 | 1.76 | 0.55 | 2.02 | 1.08 | 0.50 | 1.15 | 0.26 |
| Cap | 1.00 | 1.00 | 0.72 | 0.72 | 0.85 | 0.85 | 0.74 | 0.85 | 0.68 | 0.68 |
| Eff | $2^{20}$ | 10000 | 5148 | 5148 | 9988 | 9988 | 10000 | 10000 | 1889 | 1889 |
| Tot | $2^{20}$ | 10000 | 9998 | 9998 | 19849 | 19849 | 10000 | 10000 | 10000 | 10000 |

Table C.7: Bias for the 100 simulated runs of every algorithm on the simulated data of experiment 3; the values reported in the table are $\text{Bias} \times 10^2$ for $\Delta = \gamma_j$ and $\text{Bias} \times 10^5$ for $\mathbf{I}(\gamma)$
## C.4 Example 4

| Proposal | Optimizer | Frequency | Type 1 | Type 4 | Type 3 | Type 5 | Type 6 | Type 2 |
|----------|-----------|-----------|--------|--------|--------|--------|--------|--------|
| $q_g$    | -         | $q = 0.9816$ | 0.0932 | 0.2654 | 0.2197 | 0.0158 | 0.1944 | 0.2116 |
| $S$      | -         | {1, 3}    | 3      | {1, 3} | 1      | 1      | 88     |
| $\rho_i$ | -         | $\tilde{p}(\gamma_i | y)$ | -      | -      | -      | -      | $\tilde{p}(\gamma_i | y)$ |
| $q_l$    | -         | 0.0164    | 0      | 1      | 0      | 0      | 0      |
| $S$      | -         | -         | -      | 20     | -      | -      | -      |
| $\rho_i$ | -         | -         | -      | -      | -      | -      | -      |
| $q_o$    | SA        | 0.5553    | 0.0633 | 0.3165 | 0.1532 | 0.1548 | 0.1112 | 0.2011 |
| $q_o$    | GREEDY    | 0.2404    | 0.0149 | 0.2871 | 0.1656 | 0.2302 | 0.0820 | 0.2201 |
| $q_o$    | MTMCMC    | 0.2043    | 0.2310 | 0.1052 | 0.1877 | 0.1103 | 0.1682 | 0.1980 |
| $S$      | -         | {1, 3}    | 3      | {1, 3} | 1      | 1      | 88     |
| $\rho_i$ | -         | $\tilde{p}(\gamma_i | y)$ | -      | -      | -      | -      | $\tilde{p}(\gamma_i | y)$ |
| $q_r$    | -         | -         | 0      | 0      | 0      | 0      | 0      | 1      |
| $S$      | -         | -         | -      | -      | -      | -      | -      | 88     |
| $\rho_i$ | -         | -         | -      | -      | -      | -      | -      | 0.0010 |

Table C.8: Other tuning parameters of MTMCMC for all proposal types ($q_g, q_l, q_o,$ and $q_r$) in example 4; see Table C.2 for details.
## C.5 Example 5

| Proposal | Optimizer | Frequency | Type 1 | Type 4 | Type 3 | Type 5 | Type 6 | Type 2 |
|----------|-----------|-----------|--------|--------|--------|--------|--------|--------|
| $q_g$    | -         | $\rho = 0.9615$ | 0.1662 | 0.3323 | 0.1662 | 0.1662 | 0.1662 | 0.0029 |
| $S$      | -         | -         | \{1,1\} | 1      | \{1,1\} | 1      | 1      | 13     |
| $\rho_i$ | -         | -         | $\tilde{p}(\gamma_i|y)$ | -      | -      | -      | -      | $\tilde{p}(\gamma_i|y)$ |
| $q_l$    | -         | -         | 0.0385 | 0      | 1      | 0      | 0      | 0      |
| $S$      | -         | -         | -      | 4      | -      | -      | -      | -      |
| $\rho_i$ | -         | -         | -      | -      | -      | -      | -      | -      |
| $q_o$    | SA        | 0.5000    | 0.0657 | 0.3281 | 0.1588 | 0.2247 | 0.2209 | 0.0019 |
| $q_o$    | GREEDY    | 0.2500    | 0.0160 | 0.3083 | 0.1778 | 0.2472 | 0.2491 | 0.0014 |
| $q_o$    | MTMCMC    | 0.2500    | 0.2875 | 0.3012 | 0.1580 | 0.1105 | 0.1398 | 0.0026 |
| $S$      | -         | -         | \{1,1\} | 1      | \{1,1\} | 1      | 1      | 13     |
| $\rho_i$ | -         | -         | $\tilde{p}(\gamma_i|y)$ | -      | -      | -      | -      | $\tilde{p}(\gamma_i|y)$ |
| $q_r$    | -         | -         | 0      | 0      | 0      | 0      | 0      | 1      |
| $S$      | -         | -         | -      | -      | -      | -      | -      | 13     |
| $\rho_i$ | -         | -         | -      | -      | -      | -      | -      | 0.0010 |

Table C.9: Other tuning parameters of MTMCMC for all proposal types ($q_g, q_l, q_o$, and $q_r$) in example 5; see Table C.2 for details.
Table C.10: Bias of the mean squared error (BIAS) from the 100 simulated runs of MJMCMC on the epigenetic data; the values reported in the table are $\text{BIAS} \times 10^2$ for $\Delta = \gamma_j$; $\text{BIAS} \times 10^5$ for $I(\gamma)$.

| Par | True | TOP | MJMCMC | RS |
|-----|------|-----|--------|----|
|      |      |     |         |     |
| $\Delta$ | $\pi_j$ | RM | RM | MC | RM | MC |
| $\gamma_4$ | 0.0035 | -0.0005 | -0.0019 | 1.7361 | -0.0189 | 1.6397 |
| $\gamma_6$ | 0.0048 | -0.0006 | -0.0041 | 1.8155 | -0.0241 | 1.5437 |
| $\gamma_7$ | 0.0065 | -0.0006 | -0.0045 | 1.9763 | -0.0338 | 0.2191 |
| $\gamma_3$ | 0.0076 | -0.0007 | -0.0014 | 2.9714 | -0.0339 | 0.5167 |
| $\gamma_8$ | 0.0076 | -0.0007 | -0.0066 | 1.8370 | -0.0326 | 1.1101 |
| $\gamma_5$ | 0.0096 | -0.0007 | -0.0055 | 1.5439 | -0.0430 | 1.1780 |
| $\gamma_{11}$ | 0.0813 | -0.0007 | -0.0131 | -0.7623 | -0.1060 | 1.0394 |
| $\gamma_{12}$ | 0.0851 | -0.0006 | -0.0042 | -0.4290 | -0.0637 | 0.3118 |
| $\gamma_9$ | 0.1185 | -0.0008 | -0.0121 | -1.3414 | -0.1277 | -0.4439 |
| $\gamma_{10}$ | 0.3042 | -0.0006 | -0.0036 | -8.4912 | -0.0501 | 2.6866 |
| $\gamma_{13}$ | 0.9827 | -0.0002 | 0.0051 | -1.6177 | 0.0607 | -1.0082 |
| $\gamma_{11}$ | 1.0000 | 0.0007 | 0.0000 | -4.4528 | 0.0000 | -1.0018 |
| $\gamma_{12}$ | 1.0000 | 0.0000 | 0.0000 | -2.3865 | 0.0000 | -0.7782 |

| I($\gamma$) | 0.0000 | 0.0279 | 0.1353 | 496.2205 | 1.4386 | 582.7996 |
| Cap | 1.0000 | 1.0000 | 0.9998 | 0.9998 | 0.9977 | 0.9977 |
| Eff | 8192 | 385 | 1758 | 1758 | 155 | 155 |
| Tot | 8192 | 385 | 3160 | 3160 | 10000 | 10000 |
Algorithm B.2 Mode jumping MCMC

1: procedure MJMCMC(N)
2: \[ \gamma \leftarrow \gamma_0 \] \hspace{1cm} \triangleright \text{define the initial state}
3: for \( t = 1, \ldots, N \) do
4: \hspace{1cm} if \( \text{Unif}[0; 1] \leq \varrho \) then \hspace{1cm} \triangleright \text{large jump with local optimization}
5: \hspace{1cm} \hspace{1cm} \varrho_t \sim P_t(\zeta) \hspace{1cm} \triangleright \text{choose large jump kernel}
6: \hspace{1cm} \hspace{1cm} \varrho_o \sim P_o(\zeta) \hspace{1cm} \triangleright \text{choose local optimizer}
7: \hspace{1cm} \hspace{1cm} \varrho_r \sim P_r(\zeta) \hspace{1cm} \triangleright \text{choose randomization kernel}
8: \hspace{1cm} \hspace{1cm} J^* \sim q_l(\cdot|\gamma) \hspace{1cm} \triangleright \text{Indices for large jump}
9: \hspace{1cm} \hspace{1cm} x_0^* \leftarrow \text{SWAP}(\gamma, J^*) \hspace{1cm} \triangleright \text{large jump}
10: \hspace{1cm} \hspace{1cm} x_k^* \sim q_o(\cdot|x_0^*) \hspace{1cm} \triangleright \text{local optimization}
11: \hspace{1cm} \hspace{1cm} \gamma^* \sim q_r(\cdot|x_k^*) \hspace{1cm} \triangleright \text{randomization around the mode}
12: \hspace{1cm} \hspace{1cm} x_0 \leftarrow \text{SWAP}(\gamma^*, J^*) \hspace{1cm} \triangleright \text{reverse large jump}
13: \hspace{1cm} \hspace{1cm} x_k \sim q_o(\cdot|x_0) \hspace{1cm} \triangleright \text{local optimization}
14: \hspace{1cm} \hspace{1cm} r \leftarrow r_m(x, \gamma; x^*, \gamma^*) \hspace{1cm} \triangleright \text{from (15)}
15: \hspace{1cm} \text{else} \hspace{1cm} \triangleright \text{ordinary proposal}
16: \hspace{1cm} \hspace{1cm} q_g \sim P_g(\zeta) \hspace{1cm} \triangleright \text{choose MTMCMC proposal kernel}
17: \hspace{1cm} \hspace{1cm} \gamma^* \sim q_g(\cdot|\gamma) \hspace{1cm} \triangleright \text{MTMCMC proposed solution}
18: \hspace{1cm} \hspace{1cm} r \leftarrow r_m(\gamma, \gamma^*) \hspace{1cm} \triangleright \text{from (B.2)}
19: \hspace{1cm} \text{end if}
20: \hspace{1cm} \text{if } \text{Unif}[0; 1] \leq r \text{ then} \hspace{1cm} \triangleright \text{accept the move}
21: \hspace{1cm} \hspace{1cm} \gamma \leftarrow \gamma^* \hspace{1cm} \triangleright \text{accept the move}
22: \hspace{1cm} \text{end if}
23: \hspace{1cm} \text{end for}
24: \text{end procedure}