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

May 20, 2016

Abstract

Generalized linear mixed models (GLMM) are addressed 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 we introduce and implement efficient mode jumping MCMC algorithms for calculating posterior probabilities of the models for generalized linear models with a random effect. Marginal likelihoods of models, given the specific choice of priors and any choice of covariates, can be efficiently calculated using the integrated nested Laplace approximations approach (INLA) for the class of models addressed, however for some particular cases exact results are also available. We further apply the suggested algorithm to some simulated data, the famous U.S. crime data, protein activity data and real epigenetic data and compare its performance to some of the existing approaches like BAS, RS or MC^3.

Keywords: Bayesian variable selection; Generalized linear mixed models; Local metaheuristics; 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 reasonably chosen responses and 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 [1] these models have proven to be efficient in numerous applications including the simple banking scoring problems [16] and insurance claims modeling [8], studies on the course of illness in schizophrenia and linking diet with heart diseases [29], analyzing sophisticated astrophysical data [9], and genomics [21].

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 with calculation of the posterior distributions of parameters within GLMM is resolved efficiently with the INLA approach [28]. INLA requires the models to satisfy some properties: the latent model consisting of hidden variables can be of large dimension, however it should have numerous conditionally independent components, so its precision matrix is required to be sparse. The dimension of the hyperparameter vector should be relatively small (regression coefficients are not included into the vector of hyperparameters). INLA then efficiently calculates the marginal posterior distribution of parameters that can be summarized by means, variances and quantiles, as well as the marginal likelihood (MLIK) and other model selection criteria (DIC, WAIC), in addition to some popular predictive measures (CPO, PIT). Note that for simpler GLM with some choices of priors (usually conjugate) it is possible to obtain exact results efficiently [7]. Notice that it is possible to expand the number of possible models that can be fitted using the INLA methodology by means of combining it with MCMC [15].

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 *Deviance Information Criterion* (DIC) and *Watanabe-Akaike Information Criterion* (WAIC), *Marginal Likelihoods of The Data* (MLIK) or *Posterior Model Probability* (PMP) should get addressed. An explicit applied overview of numerous Bayesian model selection criteria is given by Piironen and Vehtari [24]. A more theoretical study is given by Gelman et al. [12]. 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 [12] can be viewed as prediction and finding the best model in terms of the inference on the true model. WAIC and PMP correspondingly can be viewed as natural model selection criteria in these contexts. Since in our research we concentrate on making inference on the true model, PMP is going to be addressed as the criterion for variable selection.

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. George and McCulloch [13] 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 [14] also addresses MCMC algorithms to estimate the posterior distribution over models. However, he 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. Clyde et al. [7] suggest a Bayesian adaptive sampling algorithm as an alternative to MCMC, which for small number of variables carries out full enumeration whilst for larger ones when the enumeration is not available allows to carry out perfect sampling without replacement. The authors show that their algorithm can under some conditions outperform standard MCMC. Song and Liang [30] address the case when there is by far more 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. Yet another approach for Bayesian model selection is addressed by Bottolo et al. [4], 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. Frommlet et al. [11] also apply an efficient memetic genetic search algorithm with greedy improvements after the cross over for exploration of the model space. Hans et al. [17] introduce a slightly different approach addressed as a shotgun stochastic search, which allows parallel evaluation of the proposals, however, the authors do not pay enough attention to efficient ways to escape from local optima, which relates the method to simple multiple try MCMC algorithms.

In this paper we introduce a new approach of searching for the best models in terms of posterior model probabilities (PMP). This approach is based on MCMC which manages to efficiently explore the search space by means of introducing locally optimized proposals and thus being able to jump between modes and escape from local optima. Locally optimized proposals in the spaces of continuous variables have been suggested previously by Tjelmeland and Hegstad [32]. We generalize their idea to discrete settings. In this context suggested algorithms can be seen not only as a model selection tool, but also as a novel tool for efficient MCMC inference on the posterior distributions of discrete parameters. 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 MCMC algorithms with locally improved proposals, are 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; important proofs of the balance equations for the suggested algorithms and detailed pseudo-codes are given 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 a vector of p covariates X_{ti} 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 δ_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:

$$\begin{aligned} Y_t|\mu_t &\sim \mathfrak{f}(y|\mu_t), t \in \{1, ..., T\} && (1)\\ \mu_t &= g^{-1}\left(\eta_t\right) && (2)\\ \eta_t &= \beta_0 + \textstyle\sum_{i=1}^{p} \gamma_i \beta_i X_{ti} + \delta_t && (3)\\ \boldsymbol{\delta} &= (\delta_1, ..., \delta_T) \sim N_T\left(\mathbf{0}, \boldsymbol{\Sigma}_b\right). && (4) \end{aligned}$$

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

$$\begin{aligned} \gamma_i &\sim Binom(1, q) && (5)\\ q &\sim Beta(\alpha_q, \beta_q) && (6)\\ \beta_i|\gamma_i &\sim \mathbb{1}(\gamma_i = 1) N(\mu_\beta, \sigma_\beta^2) && (7)\\ \boldsymbol{\psi} &\sim \varphi(\boldsymbol{\psi}), && (8) \end{aligned}$$

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

Let $\boldsymbol{\gamma} = (\gamma_1, ...\gamma_p)$, 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 marginal posterior probabilities (PMP) - $p(\boldsymbol{\gamma}|\mathbf{y})$, where $\mathbf{y}$ is the observed data. We are also interested in inference on the simultaneous posterior distribution of the vector of all parameters $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\psi}\}$ and the model $\boldsymbol{\gamma}$, namely $p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{y})$. Depending on the (1) we either use the popular INLA approach [28] based on integrated nested Laplace approximations of the posterior for calculating posterior marginal likelihoods $p(\mathbf{y}|\boldsymbol{\gamma})$ (MLIK) and $p(\boldsymbol{\theta}|\boldsymbol{\gamma}, \mathbf{y})$ or obtain exact results [7], for given $\boldsymbol{\gamma}$, and then use efficient MCMC algorithms to find $p(\boldsymbol{\gamma}|\mathbf{y})$. In order to infer on $p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{y})$, we will address the fact that $p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{y}) = p(\boldsymbol{\theta}|\boldsymbol{\gamma}, \mathbf{y})p(\boldsymbol{\gamma}|\mathbf{y})$. Posterior marginal probabilities $p(\boldsymbol{\gamma}|\mathbf{y})$, as stated above, are obtained by means of an MCMC walk, namely $p(\boldsymbol{\gamma}|\mathbf{y})$ can be rewritten with respect to Bayes formula as

$$p(\boldsymbol{\gamma}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}' \in \Omega_{\boldsymbol{\gamma}}} p(\mathbf{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')}, \tag{9}$$

which keeps the trade-off between $p(\mathbf{y}|\boldsymbol{\gamma})$ and the prior knowledge about the models incorporated through $p(\boldsymbol{\gamma})$. Also notice that in case $p(\boldsymbol{\gamma}) = p(\boldsymbol{\gamma}'), \forall \boldsymbol{\gamma}, \boldsymbol{\gamma}' \in \Omega_{\boldsymbol{\gamma}}$ maximization of PMP becomes equivalent to maximization of MLIK. In any case, in order to calculate $p(\boldsymbol{\gamma}|\mathbf{y})$ we have to iterate through the whole $\Omega_{\boldsymbol{\gamma}}$, which becomes computationally infeasible for sets of extremely large cardinality. That is why we aim at approximating $p(\boldsymbol{\gamma}|\mathbf{y})$ by means of exploration of some subspace $\mathbb{V}$ of $\Omega_{\boldsymbol{\gamma}}$ [7], namely:

$$\widehat{p}(\boldsymbol{\gamma}|\mathbf{y}) = \frac{\mathbb{1}(\gamma \in \mathbb{V})p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}' \in \mathbb{V}} p(\mathbf{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')}. \tag{10}$$

Notice that in (10) low $p(\mathbf{y}|\boldsymbol{\gamma})$ induce both low values of the numerator and small contributions to the denumerator in (10), hence low $p(\mathbf{y}|\boldsymbol{\gamma})$ will have no influence on posterior marginal probabilities of other models. On the other hand, models with high values of $p(\mathbf{y}|\boldsymbol{\gamma})$ are important to be addressed. This means that modes and near modal values of marginal likelihood are particularly important for construction of reasonable $\mathbb{V} \subset \Omega_{\boldsymbol{\gamma}}$ 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 denumerator 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 $\Omega_{\boldsymbol{\gamma}}$ growing exponentially fast with respect to the number of covariates and the fact that $\Omega_{\boldsymbol{\gamma}}$ 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 Δ the posterior distribution within our notation becomes

$$p(\Delta|\mathbf{y}) = \sum_{\boldsymbol{\gamma} \in \Omega_{\boldsymbol{\gamma}}} p(\Delta|\boldsymbol{\gamma}, \mathbf{y})p(\boldsymbol{\gamma}|\mathbf{y}), \tag{11}$$

whilst a model averaged expectation of a parameter Δ correspondingly is

$$\mathrm{E}[\Delta|\mathbf{y}] = \sum_{\boldsymbol{\gamma} \in \Omega_{\boldsymbol{\gamma}}} \mathrm{E}[\Delta|\boldsymbol{\gamma}, \mathbf{y}]p(\boldsymbol{\gamma}|\mathbf{y}). \tag{12}$$

For instance such an important parameter as posterior marginal inclusion probability $p(\gamma_j|\boldsymbol{y})$ can be expressed for any $j \in \{1, ..., p\}$ as

$$p(\gamma_j|\boldsymbol{y}) = \sum_{\boldsymbol{\gamma}' \in \Omega_{\boldsymbol{\gamma}}} \mathbb{1}(\gamma_j' = 1)p(\boldsymbol{\gamma}'|\mathbf{y}), \tag{13}$$

which can be approximated using (10) as

$$\widehat{p}(\gamma_j|\boldsymbol{y}) = \sum_{\boldsymbol{\gamma}' \in \Omega_{\boldsymbol{\gamma}}} \mathbb{1}(\gamma_j' = 1)\widehat{p}(\boldsymbol{\gamma}'|\mathbf{y}), \tag{14}$$

giving a measure for assessing importance of the covariates involved into the search procedure. Both (10) and (14) are consistent, but only asymptotically unbiased.

3 MONTE CARLO MARKOV CHAIN

MCMC algorithms [26] and their extensions have been extremely popular for the exploration of the model space 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 make simulations transitions in the combined space of both models and parameters. In our research, however, we are using INLA or other available methods for within model calculations and thus obtain marginal likelihoods $p(\mathbf{y}|\boldsymbol{\gamma})$ of the models. Having obtained $p(\mathbf{y}|\boldsymbol{\gamma})$ 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 $p(\mathbf{y}|\boldsymbol{\gamma})$ too often. Moreover we would like to take advantage of the recent advances in the 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 implementations of these extensions of MCMC to generate mode jumping proposals. Finally we discuss tuning of the parameters of the search and opportunities for parallel computing of the suggested algorithms.

3.1 Theoretical background of MCMC

In the MCMC approach as described by Robert and Casella [26], Metropolis-Hastings algorithms are addressed as a class of methods for drawing from a complicated target distribution, which in our setting will be $\pi(\boldsymbol{\gamma}) = p(\boldsymbol{\gamma}|\mathbf{y})$. For a move from any state $\boldsymbol{\gamma}$ to any other state $\boldsymbol{\gamma}'$ we need a transition kernel $\mathsf{T}(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$ to generate an ergodic Markov Chain that has $\pi(\boldsymbol{\gamma})$ as a stationary distribution. It is sufficient for the transition kernel to be irreducible, aperiodic, and π-invariant [18]. We then aim to simulate this chain to obtain a dependent approximate sample from the target distribution. Thus we need to have developed a method that somehow manages to switch between different values in $\Omega_{\boldsymbol{\gamma}}$ based on some proposal distribution $q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})$ satisfying the irreducibility property. The Metropolis-Hastings algorithm [26] accepts the proposed $\boldsymbol{\gamma}' = \boldsymbol{\gamma}^*$ with probability

$$r_m(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}^*)q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^*)}{\pi(\boldsymbol{\gamma})q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})}\right\} \tag{15}$$

and otherwise remains in the old state $\boldsymbol{\gamma}' = \boldsymbol{\gamma}$.

Storvik [31] and Chopin et al. [5] describe high potential flexibility in choices of proposals by means of generating additional auxiliary states allowing cases where $q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})$ is not directly available. The auxiliary states can be for example chains generated by some local optimizers allowing for jumps to alternative modes. There are different ways to address this flexibility, in particular Storvik [31] shows that the detailed balance equation under some conditions is satisfied for the general case addressed further. Assume the current state to be $\boldsymbol{\gamma} \sim \pi(\boldsymbol{\gamma})$. Generate $(\boldsymbol{\chi}^*, \boldsymbol{\gamma}^*) \sim q(\boldsymbol{\chi}^*, \boldsymbol{\gamma}^*|\boldsymbol{\gamma})$ and consider $\boldsymbol{\chi}|\boldsymbol{\gamma}, \boldsymbol{\chi}^*, \boldsymbol{\gamma}^* \sim h(\boldsymbol{\chi}|\boldsymbol{\gamma}, \boldsymbol{\chi}^*, \boldsymbol{\gamma}^*)$ for some arbitrary chosen $h(\cdot|\cdot)$. $\boldsymbol{\chi}$ and $\boldsymbol{\chi}^*$ are then auxiliary states. Accept $\boldsymbol{\gamma}' = \boldsymbol{\gamma}^*$ with the following acceptance probability

$$r_m(\boldsymbol{\chi}, \boldsymbol{\gamma}; \boldsymbol{\chi}^*, \boldsymbol{\gamma}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}^*)h(\boldsymbol{\chi}^*|\boldsymbol{\gamma}^*, \boldsymbol{\chi}, \boldsymbol{\gamma})q(\boldsymbol{\chi}, \boldsymbol{\gamma}|\boldsymbol{\gamma}^*)}{\pi(\boldsymbol{\gamma})h(\boldsymbol{\chi}|\boldsymbol{\gamma}, \boldsymbol{\chi}^*, \boldsymbol{\gamma}^*)q(\boldsymbol{\chi}^*, \boldsymbol{\gamma}^*|\boldsymbol{\gamma})}\right\}, \tag{16}$$

or remain in the previous state otherwise. Then $\boldsymbol{\gamma}' \sim \pi(\boldsymbol{\gamma}')$. In a typical setting $\boldsymbol{\chi}^*$ is generated first, followed by $\boldsymbol{\gamma}^*$. The extra $\boldsymbol{\chi}$ is needed in order to calculate a legal acceptance probability, relating to a symmetric reverse move. Within the described procedure we are generating samples from the target distribution, i.e. the posterior model probability - $p(\boldsymbol{\gamma}|\mathbf{y})$, which can then be approximated as

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

whilst posterior marginal inclusion probabilities (13) for $j \in \{1, ..., p\}$ can be estimated as

$$\widetilde{p}(\gamma_j|\mathbf{y}) = \frac{\sum_{i=1}^{W} \mathbb{1}(\boldsymbol{\gamma}_j^{(i)} = 1)}{W} \xrightarrow[W\to\infty]{d} p(\gamma_j|\mathbf{y}), \tag{18}$$

where W is the total amount of MCMC samples. Although estimates (17) and (18) are both asymptotically consistent and unbiased, (10) and (13) will often be preferable estimators from the practical standpoint, since convergences of the MCMC based approximations (17) and (18) are much slower, moreover MCMC based approximations are expected to have by far more noise.

Notice that in the settings (16) different local learning and optimization routines seem to be a natural and efficient choice for the auxiliary states. This choice 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. We will consider in the following part of the subsection four different options: local SA optimization suggested by Yeh et al. [35], local greedy optimization introduced by Tjelmeland and Hegstad [32], and local multiple try MCMC methods described by Liu et al. [20] as well as mixtures of them. Later we describe possibilities for generalizations of the ideas to be able to use multiple cores.

3.2 Locally optimized mode jumping proposals

We address various options for generating proposals. For the simplest statement of the problem with no auxiliary states we can consider simple proposals as swaps of variables

$\gamma_j, j \in \{i_i, ..., i_S\}$, where $S \sim Unif\{\zeta, ..., \eta\}$ and $\{i_i, ..., i_S\}$ are uniform samples from $\{1, ..., p\}$ without replacement. This implies that in (15) the proposal probability for switching from $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma}^*$ becomes symmetric, which simplifies calculation of the acceptance probability:

$$q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}) = \frac{1}{\binom{p}{S}(\eta - \zeta + 1)} = q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^*) \tag{19}$$

Other possibilities for proposals are summarized in Table 1.

Proposal $q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})$ **probability**	**Label**
$\frac{\prod_{i\in\{i_i,...,i_S\}}\rho_i}{\binom{p}{S}(\eta-\zeta+1)}$	*Random change with random size of the neighborhood*
$\frac{\prod_{i\in\{i_i,...,i_S\}}\rho_i}{\binom{p}{S}}$	*Random change with fixed size of the neighborhood*
$\frac{1}{\binom{p}{S}(\eta-\zeta+1)}$	*Swap with random size of the neighborhood*
$\binom{p}{S}^{-1}$	*Swap with fixed size of the neighborhood*
$\frac{1-\mathbb{1}\left(\sum_i^p(\gamma_i)=p\right)}{p-\sum_i^p\gamma_i+\mathbb{1}\left(\sum_i^p(\gamma_i)=p\right)}$	*Uniform addition of a covariate*
$\frac{1-\mathbb{1}\left(\sum_i^p(\gamma_i)=0\right)}{\sum_i^p\gamma_i+\mathbb{1}\left(\sum_i^p(\gamma_i)=p\right)}$	*Uniform deletion of a covariate*

Table 1: Types of proposals suggested for the moves between the models during MCMC procedure. Here S is either a deterministic or a random ($S \sim Unif\{\zeta, ..., \eta\}$) size of the neighborhood; ρ_i is the probability of inclusion of variable γ_i, which can be either deterministic or adaptive (adaptively updated approximations of the marginal inclusion probabilities (14) or (18)).

All of these proposals 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 the modes and as a result both slow convergence of MCMC and poor exploration of $\Omega_{\boldsymbol{\gamma}}$. 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. Ideas for generation of good proposals based on local optimization techniques have been first suggested by Tjelmeland and Hegstad [32] and Yeh et al. [35], whilst multiple try MCMC methods with local optimization have been described by Liu et al. [20]. All of these methods fall into the category of generating auxiliary states for proposals.

We expand and generalize local optimization ideas in a rather flexible and efficient way, based on (16). Thus, once the MCMC procedure is stuck in some local mode $\boldsymbol{\gamma}$, we can carry out locally optimized proposals to escape from it reasonably fast. For generating such locally optimized proposals we first make a big jump to a new region of interest with respect to kernel $\mathsf{q}_\mathsf{l}(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})$ that can be for example of form (19), then we carry out

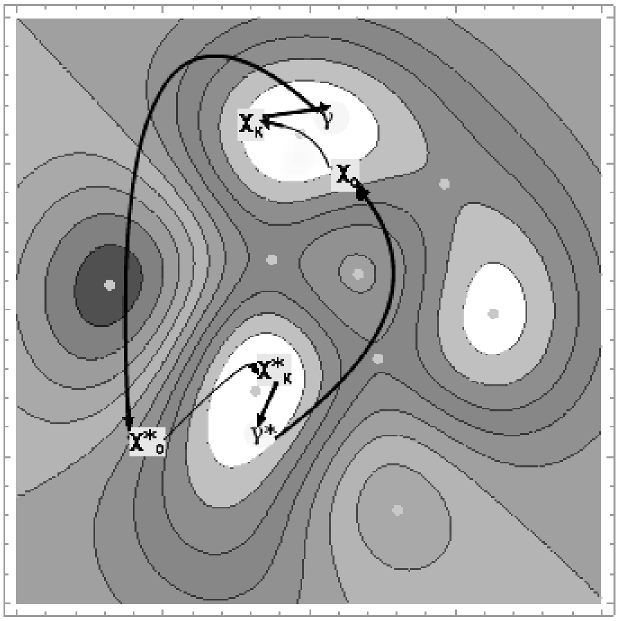

Figure 1: Locally optimized proposals

some local optimization of the target distribution $\pi(\boldsymbol{\gamma}^*)$ with chosen transition kernels $\mathsf{Q}_\mathsf{o}(\boldsymbol{\chi}_i^*|\boldsymbol{\chi}_{i-1}^*), i \in \{1,...,k\}$, which can be either stochastic or deterministic, and finally make a randomization $\boldsymbol{\gamma}^* \sim \mathsf{q}_\mathsf{r}(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)$ with a kernel based on a small neighborhood. For the reverse move we correspondingly first make a symmetric big jump $\mathsf{q}_\mathsf{l}(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)$, followed by the same type of local optimization $\mathsf{Q}_\mathsf{o}(\boldsymbol{\chi}_i|\boldsymbol{\chi}_{i-1}), i \in \{1,...,k\}$, and finally the probability of transition from the point at the end of optimization to the initial solution $\boldsymbol{\gamma}$ is calculated with respect to the randomizing kernel $\mathsf{q}_\mathsf{r}(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)$. The whole procedure is visualized in Figure 1. Then acceptance probabilities with respect to (16) are calculated and the move to a new state is either accepted or rejected. A convenient choice of the $h(\boldsymbol{\chi}|\boldsymbol{\gamma},\boldsymbol{\gamma}^*,\boldsymbol{\chi}^*)$ function allowing to store very little of the information from the local optimization routine is to consider it of a form $h(\boldsymbol{\chi}|\boldsymbol{\gamma},\boldsymbol{\gamma}^*,\boldsymbol{\chi}^*) = \mathsf{h}(\boldsymbol{\chi}|\boldsymbol{\gamma},\boldsymbol{\gamma}^*)$:

$$\mathsf{h}(\boldsymbol{\chi}|\boldsymbol{\gamma},\boldsymbol{\gamma}^*) = \mathsf{q}_\mathsf{l}(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)\left[\prod_{i=1}^{k}\mathsf{Q}_\mathsf{o}\left(\boldsymbol{\chi}_i|\boldsymbol{\chi}_{i-1}\right)\right]. \tag{20}$$

Then (16) reduces to

$$r_m(\boldsymbol{\gamma},\boldsymbol{\gamma}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}^*)\mathsf{q}_\mathsf{r}(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)}{\pi(\boldsymbol{\gamma})\mathsf{q}_\mathsf{r}(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)}\right\}. \tag{21}$$

Here we only need to consider transitions at the very last step of the particular local optimization routine and 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. Furthermore, the acceptance probability (21) disregards the intermediate states for the acceptance of the moves, which focuses on the performance of the proposed mode in comparison to the current state and the last step of the optimization procedure. Note that all solutions visited during the local optimization procedure can be added to $\mathbb{V} \subset \Omega_\gamma$ and we are not loosing these iterations in terms of the exploration of the space of models Ω_γ, unless we visit the same solutions several times. Revisiting of the models is not a serious issue once we are not stuck in the local modes and once we are able to store the results of estimations of models when the computational cost of estimation is high.

In principle, various kinds of heuristics [19] and methaheuristics [3] 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 procedure, however most of them are pretty expensive computationally. In this paper we address *accept the first improving neighbor*, *forward* and *backward regression*, *simulated annealing*, and *local MCMC* approaches. Pseudo codes for them can be found in the supplementary materials of the paper. Note that the two last methods are non-deterministic and thus their transition kernel might well be used for the randomization at the end of the suggested procedure, whilst for the deterministic approaches we need to perform some stochastic randomization at the end to achieve better mixtures and ensure that the global chain is irreducible. Furthermore, notice that a mixture of local optimization routines and proposals with respect to some kernel can be addressed in the mode jumping MCMC procedure. As a rule of thumb based on suggestions of Tjelmeland and Hegstad [32] and our own experience we recommended that in not more than 5% of the cases no mode jumping is 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.

3.3 Tuning parameters of the search

In practice tuning parameters of the local optimization routines such as in general the choice of the neighborhood, generation of proposals within it, the cooling schedule for *simulated annealing* [23] 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. Additionally tuning of the probabilities of addressing different local optimizers and different proposals in the mixture can be often beneficial. Without any doubt 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. Generally speaking, MCMC is required to be homogeneous in tuning parameters of the algorithm, hence parameters of the chain can either be adjusted during burn-in or at the so-called regeneration times [34]. The latter provides numerous additional complications, hence in our practical implementation we apply a simple step-wise greedy optimization of such tuning parameters of the search as probabilities of choosing a particular local optimization tool with a particular neighborhood $\mathsf{s}(\mathfrak{m}|\boldsymbol{\chi}_0), \mathfrak{m} \in \mathfrak{M}$ by means of greedily updating them with the objective to maximize the captured mass penalized with complexity of the local optimizers. This can be generalized for improving all of the tuning parameters of the search including the sizes of the neighborhood for proposals, cooling schedule in simulated annealing procedures and etc. Even though such greedy optimization based only on the models obtained during the burn-in steps often works well in practice, it does not generally speaking guarantee optimality of the obtained values for the rest of the search space. We, however, leave these issues for further research. Additional literature review on search parameter tuning can be found in [22].

3.4 Generalizations for multiple cores

In the context of Big Data in terms of the increasing number of potential explanatory variables it is important to be able to utilize multiple cores and GPUs of either local machines or supercomputers in parallel to get the model selection results reasonably fast. In this section we describe how the global MCMC and locally optimized proposals can be carried out in parallel.

3.4.1 Multiple try MCMC

A popular generalization of MCMC is a so-called multiple-try MCMC [20] or MTMCMC, aimed to improve computational speed in MCMC methods. MTMCMC allows to use several cores simultaneously. The idea of the method is to allow generating K trial proposals $\boldsymbol{\gamma}_1^*, ...\boldsymbol{\gamma}_K^*$ in parallel. Then within a trial set $\boldsymbol{\gamma}^* \in \{\boldsymbol{\gamma}_1^*, ..., \boldsymbol{\gamma}_K^*\}$ is selected with probability proportional to some weights $w(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*), i \in \{1, ..., K\}$, which are of the form

$$w(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*) = \pi(\boldsymbol{\gamma}_i^*) q(\boldsymbol{\gamma}|\boldsymbol{\gamma}_i^*) \lambda(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*), i \in \{1, ..., K\}, \tag{22}$$

where $\lambda(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*), i \in \{1, ..., K\}$ is some symmetric and non-negative function. In the reversed move $\boldsymbol{\gamma}_1, ...\boldsymbol{\gamma}_{K-1}$ are generated conditioning on $\boldsymbol{\gamma}^*$ from the proposal $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^*)$ and $\boldsymbol{\gamma}_K = \boldsymbol{\gamma}$. Acceptance probabilities for MTMCMC then can be of a form

$$r_m(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = \min\left\{1, \frac{w(\boldsymbol{\gamma}, \boldsymbol{\gamma}_1^*) + \cdots + w(\boldsymbol{\gamma}, \boldsymbol{\gamma}_K^*)}{w(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}_1) + \cdots + w(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}_K)}\right\}. \tag{23}$$

Moreover, weights $w(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*), i \in \{1, ..., K\}$ can be chosen to drive MTMCMC to accept better solutions in terms of some non-negative measure $\mathsf{G}(\cdot)$ to be maximized. Such adjustments can be done through defining the symmetric $\lambda(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*), i \in \{1, ..., K\}$ as

$$\lambda(\boldsymbol{\gamma}, \boldsymbol{\gamma}_i^*) = \left(\frac{1 + \mathsf{G}(\boldsymbol{\gamma}) + \mathsf{G}(\boldsymbol{\gamma}_i^*)}{c}\right)^{\alpha}, \tag{24}$$

where $i \in \{1, ..., K\}$, c and α are some calibrating constants to be properly tuned. The sequence of MTMCMC moves then gives the target distribution $\pi(\boldsymbol{\gamma}) = p(\boldsymbol{\gamma}|\mathbf{y})$. In a global algorithm with no locally optimized proposals we can use acceptance probabilities (23) instead of (15). Advantages of this approach are that we are forcing the objective function $\mathsf{G}(\cdot)$ to be maximized in terms of the choice of $\boldsymbol{\gamma}^*$ and that we take advantage of using multiple cores for the multiply try MCMC. Notice that MTMCMC can also be addressed as a local optimization strategy, which is expected to allow us to find the local mode faster in comparison to the method based on a single core, then transition probabilities of local MTMCMC based on (23) are used as $\mathsf{Q_o}(\boldsymbol{\chi}_i|\boldsymbol{\chi}_{i-1}), i \in \{1, ..., k\}$ in (21). Ideas for parallelizing other local optimizers are described in subsection 3.4.2.

3.4.2 Parallel computing in local optimizers

General principles of utilizing multiple cores in local optimization are provided in the overview [10]. So, as one of the proper alternatives 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 returned by them. In these sequential calculations, however, we need to recalculate the transition probabilities as if the corresponding solutions were proposed in the sequence they are returned by the processing units and are either accepted or rejected. More strictly at each of the optimization steps $i \in \{0, ..., k\}$ we first generate K trial proposals $\boldsymbol{\chi}_{i,1}^*, ..., \boldsymbol{\chi}_{i,K}^*$ in parallel. Parameters of these proposals are also estimated in parallel. They are returned in some order $\boldsymbol{\chi}_{i,j_1}^*, ..., \boldsymbol{\chi}_{i,j_K}^*$, where $\{j_1, ..., j_K\}$ is some permutation of $\{1, ..., K\}$. Proposals are then either accepted or rejected in this order having the transition probabilities adjusted. Thus, in the first step transition $\mathsf{Q_o}(\boldsymbol{\chi}_{i,j_1}^*|\boldsymbol{\chi}_{i-1}^*), i \in \{0, ..., k\}$ is addressed and either $\boldsymbol{\chi}_{i,j_1}^*$ or $\boldsymbol{\chi}_{i-1}^*$ is selected as $\boldsymbol{\chi}_{i,1}^\flat$. In the second step transition probability $\mathsf{Q_o}(\boldsymbol{\chi}_{i,j_2}^*|\boldsymbol{\chi}_{i,1}^\flat)$ is addressed and $\boldsymbol{\chi}_{i,2}^\flat$ is determined. Proceeding in this manner we obtain the sequence $\boldsymbol{\chi}_{i,1}^\flat, ...\boldsymbol{\chi}_{i,K}^\flat$ and consider $\boldsymbol{\chi}_i^* = \boldsymbol{\chi}_{i,K}^\flat$. Finally, randomization step is performed in order to obtain better mixing and guarantee ergodicity of the global Monte Carlo Markov chain, namely we generate $\boldsymbol{\gamma}^* \sim q(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)$ from the appropriate randomizing kernel as discussed in the previous section. Finally, the acceptance probability for $\boldsymbol{\gamma}^*$ is calculated with respect to (21) and $\boldsymbol{\gamma}'$ either becomes $\boldsymbol{\gamma}^*$ or remains to be $\boldsymbol{\gamma}$. The described procedure enables to utilize multiple cores and thus reach local modes faster.

4 EXPERIMENTS

In this section we are going to apply the described algorithms further addressed as MJMCMC to some data sets and analyze the results. Initially we address examples from [7] to compare the performance of our approach to some existing algorithms such as BAS and competing MCMC methods (MC^3, RS) [25]. BAS carries out sampling without repetition from the space of models with respect to the adaptively updated marginal inclusion probabilities, whilst both MC^3 and RS are simple MCMC procedures based on the standard Metropolis-Hastings procedure with proposals chosen correspondingly as an inversion or a random change of one coordinate in $\boldsymbol{\gamma}$ at a time [7]. For the cases when full enumeration of the model space is possible we additionally compare all of the aforementioned approaches to the "unbeatable" TOP method that consists of the best quantile of models in terms of the posterior probability for the corresponding number of addressed models $\|\mathbb{V}\|$ and can not by any chance be outperformed in terms of the posterior mass captured. The examples addressed include a simulated data set with 100 observations and 15 covariates, a famous U.S. Crime Data [25] 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 the protein activity data with 96 observations and 88 covariates [6]. Finally in the last example we aim at addressing an Arabadopsis study, based on the real data set suggested by collaborating biologists. This example involves 17 explanatory variables and 1502 observations.

Following Clyde et al. [7] types of approximations for model probabilities (10) and marginal inclusion probabilities (14) based on the Bayes theorem are further referred to as RM 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 100 replications of each algorithm as described by Clyde et al. [7].

4.1 Example 1

In this experiment we compare MJMCMC to BAS and competing MCMC methods (MC^3, RS) using simulated data with $p = 15$ and $T = 100$, which we analyze by the linear regression model. The exact posterior model probabilities may be obtained by enumeration of the model space in this case. Notice that all columns of the design matrix except the ninth

Par	True	TOP	MJMCMC				BAS		MC^3		RS	
Δ	π_j	-	RM	MC	RM	MC	eplnp	unif	MC	RM	MC	RM
γ_{12}	0.09	-0.29	-2.11	-4.95	-1.19	-5.47	-1.23	-1.35	-0.14	-4.21	0.35	-3.80
γ_{14}	0.10	-0.28	-2.12	-6.58	-1.12	-6.07	-1.14	-1.25	-0.23	-4.23	0.05	-3.89
γ_{10}	0.11	-0.28	-2.30	-6.89	-1.30	-7.64	-1.14	-1.30	-0.10	-4.23	0.11	-4.02
γ_8	0.12	-0.27	-1.96	-6.16	-1.08	-7.69	-0.97	-1.11	0.36	-3.94	-0.51	-3.81
γ_6	0.13	-0.25	-2.24	-8.03	-1.26	-8.33	-1.05	-1.27	-0.65	-4.64	0.06	-4.24
γ_7	0.14	-0.25	-2.05	-7.45	-1.28	-8.37	-1.04	-1.18	-0.13	-4.41	0.08	-4.12
γ_{13}	0.15	-0.24	-2.39	-9.62	-1.35	-8.62	-1.15	-1.24	-0.49	-4.76	0.28	-4.32
γ_{11}	0.16	-0.24	-2.33	-8.69	-1.21	-7.95	-1.13	-1.28	-0.38	-4.59	-0.10	-4.44
γ_{15}	0.17	-0.23	-1.93	-7.64	-1.06	-9.59	-0.78	-0.92	-0.58	-4.15	-0.19	-3.74
γ_5	0.48	0.00	-1.15	-14.18	-0.47	-11.97	-0.25	-0.38	-0.29	-0.94	0.46	-1.17
γ_9	0.51	-0.10	0.78	13.11	0.23	11.96	-0.32	-0.26	-1.79	-2.20	-0.22	-1.53
γ_2	0.54	-0.07	-1.21	-18.43	-0.50	-14.64	0.34	0.27	1.73	0.29	0.35	-0.25
γ_1	0.74	0.18	2.12	4.88	1.04	3.99	1.19	0.91	-0.23	3.39	0.41	3.69
γ_3	0.91	0.25	1.60	-1.79	0.91	0.03	1.56	1.30	-0.40	3.59	-0.14	4.00
γ_4	1.00	0.01	0.00	-5.94	0.00	-2.49	0.00	0.00	0.01	0.01	-0.02	0.01
$\mathbf{I}(\boldsymbol{\gamma})$	0.00	1.52	7.54	36.68	3.38	34.85	3.04	3.29	3.72	22.57	2.95	20.57
Cap	1.00	0.99	0.89	0.89	0.95	0.95	0.95	0.95	0.72	0.72	0.74	0.74
Eff	2^{15}	3276	1906	1906	3212	3212	3276	3276	400	400	416	416
Tot	2^{15}	3276	3276	3276	6046	6046	3276	3276	3276	3276	3276	3276

Table 2: Bias for the 100 simulated runs of every algorithm on the simulated data; the values reported in the table are Bias $\times 10^2$ for $\Delta = \gamma_j$ and Bias $\times 10^5$ for $\mathbf{I}(\boldsymbol{\gamma})$

were generated from independent standard normal random variables and then centered. As stated by Clyde et al. [7] the ninth column was constructed so that its correlation with the second column was approximately 0.99. The regression parameters were chosen as $\beta_0 = 2$, $\boldsymbol{\beta} = (0.48, 8.72, 1.76, 1.87, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0)$ with $\phi = 1$. For the parameters in each model Zellner's g-prior with $g = T$ is addressed. This leads to the marginal likelihood of the model to be proportional to:

$$p(\mathbf{y}|\boldsymbol{\gamma}) \propto (1+g)^{(T-P-1)/2}(1+g[1-R^2_\gamma])^{-(T-1)/2}, \tag{25}$$

where R^2_γ is the usual coefficient of determination of a linear regression model; with this scaling, the marginal likelihood of the null model is 1.0. To complete the prior specification, we use (5) with $q = 0.5$. This leads to a model space with two main modes, which is certainly an over-simplified example, where even rather simple approaches might well work pretty good. We compare the chosen algorithms on approximately 3276 iterations (corresponding to 10% of the model space) with 100 replications of each algorithm. Additionally we address

Par	True	TOP	MJMCMC				BAS		MC^3		RS	
Δ	π_j	-	RM	MC	RM	MC	eplnp	unif	MC	RM	MC	RM
γ_{12}	0.09	0.29	2.11	5.31	1.19	5.73	1.23	1.35	2.77	4.27	2.14	3.83
γ_{14}	0.10	0.28	2.13	6.99	1.13	6.25	1.14	1.26	2.92	4.31	2.59	3.95
γ_{10}	0.11	0.28	2.31	7.41	1.31	7.74	1.15	1.31	3.06	4.31	2.40	4.07
γ_8	0.12	0.27	1.97	6.44	1.09	7.80	0.97	1.12	2.77	4.01	2.23	3.87
γ_6	0.13	0.25	2.25	8.87	1.27	8.46	1.05	1.28	3.12	4.74	2.72	4.31
γ_7	0.14	0.25	2.06	7.75	1.29	8.51	1.05	1.19	3.45	4.52	2.50	4.17
γ_{13}	0.15	0.24	2.42	9.98	1.36	8.79	1.15	1.24	3.50	4.87	2.44	4.38
γ_{11}	0.16	0.24	2.36	9.38	1.22	8.31	1.13	1.29	3.64	4.71	3.01	4.52
γ_{15}	0.17	0.23	1.96	9.38	1.08	9.73	0.78	0.93	3.92	4.27	3.32	3.84
γ_5	0.48	0.00	1.22	15.66	0.50	12.90	0.27	0.40	3.69	1.41	4.35	1.59
γ_9	0.51	0.10	1.15	16.35	0.38	12.92	0.37	0.39	16.70	5.62	6.93	2.08
γ_2	0.54	0.07	1.46	20.69	0.58	15.38	0.39	0.40	16.56	5.25	6.91	1.46
γ_1	0.74	0.18	2.15	6.43	1.06	5.97	1.20	0.92	4.10	3.55	4.51	3.90
γ_3	0.91	0.25	1.61	3.03	0.92	3.33	1.57	1.31	2.96	3.66	3.42	4.10
γ_4	1.00	0.01	0.00	6.08	0.00	2.66	0.00	0.00	0.01	0.01	0.17	0.01
$\mathbf{I}(\boldsymbol{\gamma})$	0.00	1.52	7.94	39.79	3.55	36.15	3.16	3.40	33.61	25.35	29.43	22.12
Cap	1.00	0.99	0.89	0.89	0.95	0.95	0.95	0.95	0.72	0.72	0.74	0.74
Eff	2^{15}	3276	1906	1906	3212	3212	3276	3276	400	400	416	416
Tot	2^{15}	3276	3276	3276	6046	6046	3276	3276	3276	3276	3276	3276

Table 3: Square root of the mean squared error (RMSE) from the 100 simulated runs of every algorithm on the simulated data; the values reported in the table are RMSE $\times 10^2$ for $\Delta = \gamma_j$ and RMSE $\times 10^5$ for $\mathbf{I}(\boldsymbol{\gamma})$

the cases where algorithms visit 10% unique models in $\mathbb{V}$ ($\|\mathbb{V}\| \approx 3276$). The latter seems to be a bit more relevant because the cost of a proposal of a model is by far cheaper than the cost of estimation of parameters of a model (which we store and do not re-estimate), moreover unlike other MCMC methods, MJMCMC does not have the habit to get stuck in the local modes and thus the number of unique models addressed within the same amount of proposed models is much higher for MJMCMC.

In Tables 2 - 7 we report the biases and root mean squared errors for the marginal inclusion probabilities - π_j and the model probabilities - $\mathbf{I}(\boldsymbol{\gamma})$ as well as the percentages of the posterior mass captured by the addressed algorithms (CAP) on 100 replications of each of them. We also report the means of total number of generated proposals (Tot) and the number of unique models (Eff) visited over these replications. As one can see from these tables MJMCMC seems to be by far outperforming simpler MCMC methods (including the thinned versions, which we did not include in the tables but which can be found in [7]) in terms of RM approximations of marginal posterior inclusion probabilities, individual model probabilities and the totally captured mass, however the MC approximations seem to be slightly poorer, which may be explained by that MJMCMC has not reached the stationary mode during these runs, which in turn is a result of that we use 2 adaptive types of proposals based on the current marginal inclusion probabilities. As soon as the marginal inclusions converge, the sampler reaches the stationary regime and the errors are expected to dramatically reduce. As we have noticed before whenever both MC and RM approximations are available one should address the latter since they always have less noise. However if we compare 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 its performance for this example becomes equivalent to BAS when we consider 6046 proposals with 3212 unique models visited. The latter is something we would expect, since here we are not facing a really multiple mode issue having just two modes of marginal log likelihood. All MCMC methods tend to revisit the same states from time to time and for such simple example can hardly 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 [33] and much later stated to be a test bed for evaluation of methods for model selection by Raftery et al. [25]. The data set consists of 48 observations on 15 covariates and the responses. We will compare performance of the algorithms based on a linear Bayesian regression model with a Zellner's g-prior, $g = 47$.

This is a more sophisticated example with much more local modes, which results in the fact that 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 (Tables 4 and 5). Whilst BAS on 3276 iterations slightly outperforms MJMCMC, on the same number of unique models estimated ($\|\mathbb{V}\|$) 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. As a matter of fact MJMCMC for this example even outperforms in terms of model bias and model RMSE approximated by (10) the artificial TOP method associated to 3276 best models in terms of posterior model probabilities. The latter how-

Par	True	TOP	MJMCMC				BAS		MC^3		RS	
Δ	π_j	-	RM	MC	RM	MC	eplnp	unif	MC	RM	MC	RM
γ_8	0.16	-3.51	-6.54	-10.28	-5.09	-9.64	-5.19	-6.31	5.37	-3.20	4.96	-3.06
γ_{13}	0.16	-3.34	-7.44	-10.12	-5.57	-9.94	-6.25	-7.17	7.46	2.86	8.06	2.65
γ_{14}	0.19	-3.24	-8.27	-11.69	-6.28	-11.93	-6.19	-7.36	5.27	-1.86	5.37	-2.03
γ_{12}	0.22	-3.27	-6.82	-12.91	-5.54	-13.15	-3.08	-5.14	3.00	-5.82	3.76	-5.06
γ_5	0.23	-2.56	-6.21	-12.71	-4.55	-13.35	-1.80	-3.88	-4.79	-12.98	-4.28	-12.72
γ_9	0.23	-3.27	-9.45	-15.67	-7.35	-16.11	-9.26	-8.31	4.53	-2.45	4.33	-2.10
γ_7	0.29	-2.31	-4.15	-12.04	-3.41	-12.36	-2.24	-2.85	-0.47	-9.41	-1.00	-9.56
γ_4	0.30	-1.57	-5.82	-18.74	-3.67	-17.10	0.85	-0.67	-12.67	-21.79	-13.24	-21.45
γ_6	0.33	-1.92	-8.49	-19.07	-6.09	-18.84	-3.06	-5.21	8.99	7.16	10.09	6.81
γ_1	0.34	-2.51	-11.25	-21.94	-7.25	-20.29	-8.42	-7.13	22.36	25.10	23.32	24.63
γ_3	0.39	-0.43	3.51	-7.20	2.09	-4.43	4.98	3.51	-21.11	-30.20	-21.13	-29.92
γ_2	0.57	1.58	5.66	-8.73	3.71	-7.51	13.73	8.38	-30.41	-37.52	-29.05	-37.12
γ_{11}	0.59	0.58	2.86	11.75	2.13	15.32	-3.95	-1.14	10.67	21.68	10.29	21.23
γ_{10}	0.77	3.25	7.50	-2.57	5.91	2.33	15.42	10.33	-21.22	-19.06	-20.01	-19.55
γ_{15}	0.82	3.48	9.17	0.22	6.85	3.65	14.50	11.64	-69.61	-76.81	-69.14	-76.30
$\mathbf{I}(\boldsymbol{\gamma})$	0.00	11.44	15.49	17.75	9.28	18.78	11.86	10.94	27.33	44.1	27.15	42.58
Cap	1.00	0.86	0.58	0.58	0.71	0.71	0.66	0.67	0.1	0.1	0.1	0.1
Eff	2^{15}	3276	1909	1909	3237	3237	3276	3276	829	829	1071	1071
Tot	2^{15}	3276	3276	3276	5936	5936	3276	3276	3276	3276	3276	3276

Table 4: 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 $\mathbf{I}(\boldsymbol{\gamma})$

ever should not be seen as a significant result. We believe it is only observed because 3276 models for this example is not enough to generate unbiased estimates for posterior model probabilities. Simply by chance MJMCMC visits those models that caused smaller model bias than the best 3276 models in Ω_γ for this example.

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 negative feature. Those original 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, which is by the way rather common for simpler MCMC strategies. This can be clearly seen from

Par	True	TOP	MJMCMC				BAS		MC^3		RS	
Δ	π_j	-	RM	MC	RM	MC	eplnp	unif	MC	RM	MC	RM
γ_8	0.16	3.51	6.57	10.68	5.11	10.29	5.21	6.36	6.49	3.49	5.87	3.31
γ_{13}	0.16	3.34	7.46	10.54	5.60	10.19	6.26	7.20	8.62	3.39	8.83	3.05
γ_{14}	0.19	3.24	8.30	12.43	6.30	12.33	6.20	7.39	6.58	2.55	6.22	2.46
γ_{12}	0.22	3.27	6.87	13.61	5.57	13.64	3.10	5.21	5.81	6.23	4.93	5.27
γ_5	0.23	2.56	6.30	13.45	4.59	13.65	1.84	4.02	6.07	13.05	5.13	12.77
γ_9	0.23	3.27	9.49	16.21	7.40	16.21	9.27	8.37	5.99	2.99	5.70	2.60
γ_7	0.29	2.31	4.37	13.63	3.45	12.73	2.28	2.96	4.74	9.61	3.46	9.70
γ_4	0.30	1.57	6.18	19.22	3.79	17.31	0.99	1.20	13.24	21.84	13.53	21.48
γ_6	0.33	1.92	8.61	19.71	6.14	19.49	3.11	5.30	10.19	7.47	10.99	7.12
γ_1	0.34	2.51	11.32	22.68	7.29	20.50	8.43	7.20	22.89	25.19	23.63	24.71
γ_3	0.39	0.43	3.95	11.13	2.38	6.99	5.02	3.78	21.48	30.24	21.39	29.94
γ_2	0.57	1.58	5.92	13.21	3.82	9.03	13.78	8.66	30.81	37.57	29.27	37.15
γ_{11}	0.59	0.58	3.57	13.49	2.37	15.94	4.04	2.18	11.88	21.79	11.16	21.31
γ_{10}	0.77	3.25	7.62	7.28	5.97	4.78	15.45	10.46	21.83	19.18	20.53	19.65
γ_{15}	0.82	3.48	9.23	4.45	6.89	5.85	14.50	11.75	69.68	76.81	69.19	76.30
$\mathbf{I}(\boldsymbol{\gamma})$	0.00	11.44	16.83	24.92	10.00	22.22	12.47	11.65	34.39	45.68	34.03	44.18
Cap	1.00	0.86	0.58	0.58	0.71	0.71	0.66	0.67	0.1	0.1	0.1	0.1
Eff	2^{15}	3276	1909	1909	3237	3237	3276	3276	829	829	1071	1071
Tot	2^{15}	3276	3276	3276	5936	5936	3276	3276	3276	3276	3276	3276

Table 5: Square root of the mean squared error (RMSE) from the 100 simulated 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 $\mathbf{I}(\boldsymbol{\gamma})$

Tables 2 - 7 where for the same number of iterations simpler MCMC methods give a much lower number of original models visited and also a somewhat smaller mass captured.

4.3 Example 3

In the third example, based on the logistic regression, we face a slightly more sophisticated case in comparison to the previous examples. Firstly we are using a generated data set with $p = 20$ and thus 1048576 potential models to be explored and secondly we are addressing the GLM case with conditionally independent Bernoulli observations with a logit link to the covariates. Additionally in this example $T = 2000$, which makes estimation of a single model significantly slower than in the previous examples.

Par	True	TOP	MJMCMC				BAS	MCBAS	RS	
Δ	π_j	-	RM	MC	RM	MC	RM	RM	RM	MC
γ_6	0.29	0.00	-7.23	-14.89	-4.48	-16.40	-6.46	-3.59	-5.96	0.23
γ_8	0.31	0.00	-5.97	-13.94	-3.89	-16.57	-5.57	-2.85	-5.28	-0.35
γ_{12}	0.35	0.00	-4.07	-8.12	-2.56	-11.65	-4.20	-1.82	-3.80	0.06
γ_{15}	0.35	0.00	-3.66	-8.85	-2.21	-12.04	-4.58	-1.35	-3.25	-0.28
γ_2	0.36	0.00	-4.60	-14.71	-2.81	-16.80	-5.39	-2.19	-3.51	0.04
γ_{20}	0.37	0.00	-4.16	-8.38	-2.46	-12.03	-3.30	-1.75	-4.07	-0.12
γ_3	0.40	0.00	-8.99	-19.22	-5.58	-21.72	-9.73	-4.63	-6.69	0.23
γ_{14}	0.44	0.00	1.08	7.12	0.51	7.63	3.68	-0.62	-0.99	0.22
γ_{10}	0.44	0.00	-2.68	-7.62	-1.68	-11.89	-4.79	-0.29	-1.19	0.13
γ_5	0.46	0.00	-1.74	-10.78	-0.88	-12.29	-3.93	0.57	0.55	-0.23
γ_9	0.61	0.00	0.32	-2.29	0.00	-1.24	3.78	0.22	1.99	-0.11
γ_4	0.88	0.00	5.61	6.20	3.71	6.13	6.60	5.54	7.58	-0.45
γ_{11}	0.91	0.00	5.36	6.47	3.87	6.84	4.64	3.01	4.29	-0.28
γ_1	0.97	0.00	1.86	0.98	1.32	1.17	2.43	1.94	2.28	-0.31
γ_{13}	1.00	0.00	0.00	-0.33	0.00	-0.29	0.00	0.00	0.00	-0.3
γ_7	1.00	0.00	0.00	-0.41	0.00	-0.36	0.00	0.00	0.00	-0.27
γ_{16}	1.00	0.00	0.00	-0.33	0.00	-0.31	0.00	0.00	0.00	-0.17
γ_{17}	1.00	0.00	0.00	-0.38	0.00	-0.35	0.00	0.00	0.00	-0.17
γ_{18}	1.00	0.00	0.00	-0.37	0.00	-0.32	0.00	0.00	0.00	-0.19
γ_{19}	1.00	0.00	0.00	-0.40	0.00	-0.32	0.00	0.00	0.00	-0.34
$\mathbf{I}(\boldsymbol{\gamma})$	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 6: Bias for the 100 simulated runs of every algorithm on the simulated data of experiment 3; the values reported in the table are Bias $\times 10^2$ for $\Delta = \gamma_j$ and Bias $\times 10^5$ for $\mathbf{I}(\boldsymbol{\gamma})$

We are using the AIC-prior [7] for the regression coefficients of the linear predictor leading to the following approximation of the log marginal likelihood:

$$p(\mathbf{y}|\boldsymbol{\gamma}) \propto -\frac{1}{2}\left(D(y) + 2\sum_{i=1}^{p}\gamma_i\right), \tag{26}$$

where $D(y)$ is the deviance for the logistic regression model.

The true regression parameters were chosen to be $\beta_0 = 99$ for the intercept, and for the slope coefficients $\boldsymbol{\beta} = (-4, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1.2, 0, 37.1, 0, 0, 50, -0.00005, 10, 3, 0)$. 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 of factors 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 tranformation 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 $X_2 = (X_{10} + X_{14})X_9$ and $X_5 = (X_{11} + X_{15})X_{12}$. The linear predictor is then 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 $Y \sim Bernoulli\left(p = \frac{\exp(\eta)}{1+\exp(\eta)}\right)$.

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 Tables 6 and 7 MJMCMC for the same number of estimated models outperforms pure BAS by far in terms of posterior mass captured biases and rmses of marginal inclusion probabilities and model probabilities. It also, unlike simpler RS, does not get stuck in the local modes and manages to explore a greater amount of important models for the same amount of visited models. Notice that even for almost two times less originally visited models in $\mathbb{V}$, comparing to BAS, MJMCMC gives almost the same results. Also it is worth mentioning that MJMCMC for the given number of unique models visited did not outperfrom the combination of MCMC and BAS, that is recommended by Clyde et al. [7] for larger model spaces; both of them gave almost 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 [6] 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).

Par	True	TOP	MJMCMC				BAS	MCBAS	RS	
Δ	π_j	-	RM	MC	RM	MC	RM	RM	RM	MC
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
$\mathbf{I}(\boldsymbol{\gamma})$	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 7: Square root of the mean squared error (RMSE) from the 100 simulated runs of every algorithm on the simulated data; the values reported in the table are RMSE $\times 10^2$ for $\Delta = \gamma_j$ and RMSE $\times 10^5$ for $\mathbf{I}(\boldsymbol{\gamma})$

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 88×2^{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 [7] adjusted, and those based on RM and MC approximations obtained by 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 88×2^{19} iterations providing 2^{19} models for estimating initial sampling probabilities and then BAS was run for the other 2^{19} iterations. MJMCMC

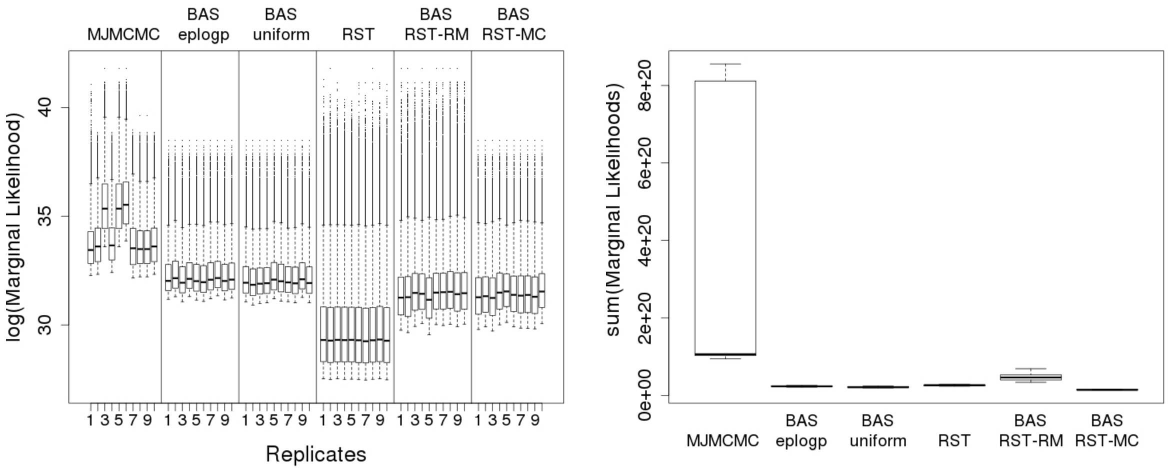

Figure 2: Comparions of the log marginal likelihood in the protein data of the top 100000 models (left) and boxplots of the posterior mass captured (right) obtained by MJMCMC, BAS-eplogp, BAS-uniform, thinned version of Random Swap (RST), BAS with Monte Carlo estimates of inclusion probabilities from the RST samples (BAS-RST-MC), and BAS renormalized estimates of inclusion probabilities (BAS-RST-RM) from the RST samples.

was run until 2^{20} unique models were obtained. All of the algorithms were run on 10 replications. In Figure 3 one can see box-plots of the best 100000 models captured by the corresponding replications of the algorithms as well as posterior masses captured by them. 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 $8.56e + 20$ are reported in Figure 2, whilst those obtained by other methods can be found in [7]. 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 (it took around 60000 MJMCMC iterations to obtain these results). 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 2). The latter issue can be explained by that the method is adaptive in two ways: first, the frequencies of addressing various local optimizers are adjusted during the burn-in and thus, as we mentioned before, depend on the sequence of models visited during the burn-in; secondly, some of the proposals addressed in the mixture (Table 1) are adaptive and change with respect to the systematically updated (after every 50000 unique models are visited in our

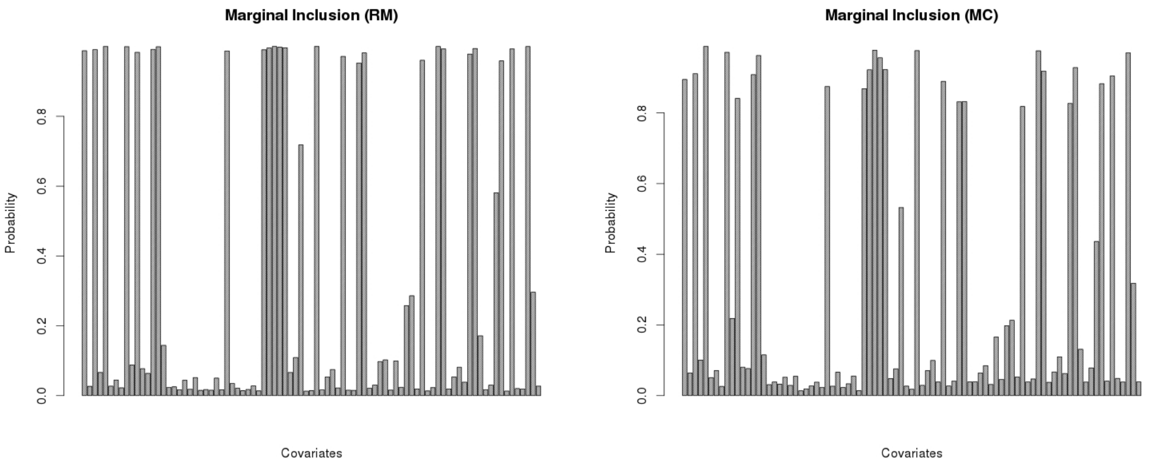

Figure 3: Comparions of RM (left) and MC (right) estimates of marginal posterior inclusion probabilities obtained by the best run of MJMCMC with $8.56\mathrm{e}+20$ posterior mass captured.

case) marginal inclusion probabilities. This on one hand allows to draw proposals that are expected to be of a good quality, however on the other hand it may introduce some tardiness in terms of escaping from the mode in terms of marginal inclusion probabilities (not to be misinterpreted with the PMP modes, which we are escaping by means of addressing local optimizers), faster escaping from these modes is a yet another challenge that can in principle be resolved (as a heuristic suggestion) by means of either truncating marginal inclusion probabilities or carrying out the square root transformation followed by re-normalization of them. Even without these adjustments, however, the results of all runs are asymptotically converging as has been shown in the previous experiments. Also note that for this case we are only allowing visiting $3.39\mathrm{e}-19\%$ of the total model space in the addressed replications, which might well be simply not enough to always converge to the same posterior mass captured.

4.5 Example 5

In this example we address an Arabadopsis study and illustrate how MJMCMC works for the GLMM models. Arabadopsis is a plant model organism with a lot of genomic/epigenomic data easily available [2]. We model a number of methylated reads $Y_t, \in \{1, ..., n_t\}$ per loci $t \in \{1, ..., T\}$ to be Poisson distributed with a with mean $\mu_t \in \mathbb{R}^+$ modeled via the log link to the chosen covariates $X_t = \{X_{t,1}, ..., X_{t,p}\}, t \in \{1, ..., T\}$ and a spatially correlated random effect, which is modeled via an $AR(1)$ process with parameter $\rho \in \mathbb{R}$, namely $\delta_t = \rho\delta_{t-1} + \epsilon_t \in \mathbb{R}, t \in \{1, ..., T\}$. Thus, we take into account spatial dependence structures of methylation rates along the genome as well as the variance of the observations

Par	True	TOP-B	TOP-R	EMJ-B	EMJ-R	EMJ-B	EMJ-R
γ_{14}	0.00	0.00	0.00	0.00	0.00	0.00	0.00
γ_1	0.00	-0.03	0.03	-0.66	0.80	-2.53	2.54
γ_4	0.00	-0.07	0.07	-2.79	3.09	-7.81	7.82
γ_6	0.01	-0.06	0.06	-3.93	4.36	-11.78	11.80
γ_7	0.01	-0.05	0.05	-4.18	4.50	-12.85	12.89
γ_8	0.01	-0.04	0.04	-5.69	6.30	-17.39	17.43
γ_9	0.01	-0.04	0.04	-6.51	7.40	-22.75	22.83
γ_5	0.01	-0.04	0.04	-6.69	7.84	-23.21	23.28
γ_{13}	0.07	-0.06	0.06	-9.43	13.91	-98.70	99.22
γ_{12}	0.09	-0.05	0.05	-11.44	13.16	-60.89	61.19
γ_{10}	0.14	-0.05	0.05	-5.16	8.55	-51.29	51.77
γ_{11}	0.50	0.00	0.00	-0.09	6.18	55.92	56.34
γ_2	1.00	0.00	0.00	0.00	0.00	0.00	0.00
γ_3	1.00	0.00	0.00	0.00	0.00	0.00	0.00
$\mathbf{I}(\boldsymbol{\gamma})$	0.00	0.04	0.04	1.11	1.48	8.43	8.48
Cap	1.00	1.00	1.00	1.00	1.00	0.98	0.98
Eff	16384	375	375	375	375	206	206
Tot	16384	375	375	511	511	224	224

Table 8: Bias and square root of the mean squared error (RMSE) from the 100 simulated runs of MJMCMC on the epigenetic data; the values reported in the table are BIAS and RMSE $\times 10^4$ for $\Delta = \gamma_j$; BIAS and RMSE $\times 10^5$ for $\mathbf{I}(\boldsymbol{\gamma})$; RM estimates only are reported

not explained by the covariates. The addressed covariates may be positions within a gene, indicators of underlying genetic structures, which chromosome the observed nucleobase belongs to, etc. We then put relevant priors for the parameters of the model in order to make a fully Bayesian inference. We use priors (5) for $\boldsymbol{\gamma}$ for the decision variables, (7) for the regression coefficients, whilst the following priors have been chosen for the latent Gaussian field:

$$\begin{pmatrix}\psi_1\\ \psi_2\end{pmatrix} \sim N_2(\mu_{\rho,\epsilon}, \Sigma_{\rho,\epsilon}) \tag{27}$$

$$\epsilon_t \sim N(0, \sigma_\epsilon^2) \tag{28}$$

where $\psi_1 = \log \frac{1}{\sigma^2_{\epsilon,t}}(1-\rho^2)$, $\psi_2 = \log \frac{1+\rho}{1-\rho}$ are scaled hyper-parameters of the latent model and ϵ_t are the error terms of $AR(1)$ model, which are a priori normally distributed with zero mean and variance ${\sigma_\epsilon}^2$. We have addressed 17 different covariates with the intercept. Among these covarites 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

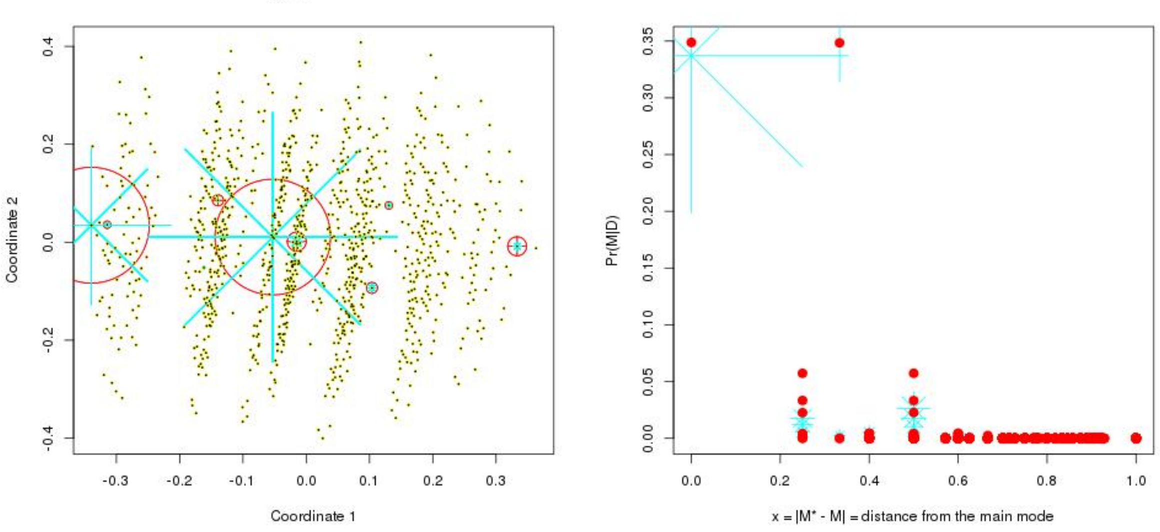

Figure 4: MDS plot [27] (left) of the best 1024 models in terms of PMP in the space of models (blue stars correspond to MC approximations, whilst red circles to RM approximations, sizes are proportional to posterior probabilities of models, yellow dots are the explored models, black dots are centers of the models) and plot of posterior probabilities (right) with respect to distance from the main mode (blue stars correspond to MC approximations, whilst red circles to RM approximations, sizes of blue stars are proportional to posterior MC based probabilities of models); estimates are obtained after 3000 unique models were visited.

generating two covariates X_2 and X_3 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 $X_4 - X_9$, we also include such 1D distance as a continuous covariate X_1. 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_α, M_β, M_γ, M_δ or M_0 inducing 4 additional covariates $X_{10} - X_{13}$. Finally, we have a continious predictor represented by an expression level for a nucleobase $X_{14} \in \mathbb{R}^+$. Thus 17 predictors with respect to the strict choice of the reference model in our example induced 14 covariates addressed, i.e. $p = 14$ and the cardinality of our search space $\Omega_{\boldsymbol{\gamma}}$ is $L = 2^p = 16384$ for this example. As one can see from Table 8 within just 375 unique models visited (2.29% of $||\Omega_\gamma||$) we are able to capture allmost full posterior mass for this problem. Ω_γ as shown in Figure 4 has very few sparsely located modes in a pretty large model space.

According to marginal inclusion probabilities, factors of weather the location is CGH or CHH are both extremely significant, a bit lower significance has a factor whether the location is in M_β group of genes. Additionally factors for M_α, M_δ and M_0 groups of genes have non-zero marginal inclusion probabilities, although they do not seem to have a high significance. In future it would be of an interest to obtain additional covariates such as whether a nucleobase belong to a particular part of the gene like promoter, intron or a coding region. Furthermore, it is definitely of interest to address factors whether a base is within a CpG island, and whether it belongs to a tranposone. Moreover interactions of these covariates may be interesting. On top of that alternative choice of the response distributions (e.g. binomial or negative binomial) and/or type of random effects ($AR(k)$, $ARMA(l, k)$) can potentially improve the inference.

5 SUMMARY AND DISCUSSION

In this article we introduced the MJMCMC approach for estimating posterior model probabilities and Bayesian model averaging and selection, which incorporates the ideas of MCMC with possibility of large jumps combined with local optimizers to generate proposals in the discrete space of models. Unlike standard MCMC methods, 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 BAS having the tendency to capture a higher posterior mass within the same amount of unique models visited. This can be explained by the fact that for problems with numerous covariates BAS requires good initial marginal inclusion probabilities to perform efficiently. The latter creates a possibility to combine the two approaches and let such methods as BAS use marginals obtained by MJMCMC as input, which significantly improves their performance.

The *EMJMCMC* R-package is developed and currently available from the GitHub repository: http://aliaksah.github.io/EMJMCMC2016/. The developed package gives a user high flexibility in the choice of methods to obtain marginal likelihoods and model selection criteria for the class of methods he addresses. Whilst the default choice is based on INLA [28], we also have adopted efficient C based implementations for the Bayesian linear regression, Bayesian logistic and Poisson regressions with g-priors, as well as AIC and BIC priors [7]. 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) - (12) for marginal inclusion and posterior model probabilities based on Bayes formula and obtained by MJMCMC, as noticed by Clyde et al. [7], 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 WAIC or DIC rather than PMP as the target distribution in MJMCMC. These choices are possible within the *EMJMCMC* package as well.

Based on the obtained in the experimental part results, we can claim MJMCMC to be a rather competitive novel algorithm that is addressing a much wider class of models (GLMM) than all of the competing approaches. In particular for this class of models one can incorporate a random effect, which both models the variability unexplained by the covariates and introduces spatial-temporal dependence for the 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 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 a more general case. In future it would be of an interest to extend the procedure to the level of selection of link functions, priors and response distributions. The latter is expected to provide new horizons in automation of model selection and thus expand opportunities for addressing properly defined statistical models within machine learning applications. It will also require even more accurate tuning of parameters of the search 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. The package includes the U.S. crime data. (EMJMCMC_ 1.2.tar.gz; GNU zipped tar file)

Data and code: Data (simulated and real) and *R OOP* 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 and pseudo codes for MJMCMC and local combinatorial optimizers. (appendix.pdf)

ACKNOWLEDGMENTS

We would like to thank CELS project at the University of Oslo for giving us the opportunity, inspiration and motivation to write this article.

References

[1] C. Ariti. Walter w stroup, generalized linear mixed models, modern concepts, methods and applications. *Statistical Methods in Medical Research*, 2014. doi: 10.1177/0962280214563202.

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

[3] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *Acm computing surveys*, pages 268–308, 2003.

[4] L. Bottolo, M. Chadeau-Hyam, D. I. Hastie, S. R. Langley, E. Petretto, L. Tiret, D. Tregouet, and S. Richardson. Ess++: a c++ objected-oriented algorithm for Bayesian stochastic search model exploration. *Bioinformatics*, 27(4):587–588, 2011.

[5] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. 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, 2013.

[6] M. Clyde, G. Parmigiani, and B. Vidakovic. Multiple shrinkage and subset selection in wavelets. *Biometrika*, 85(2):391–401, 1998. doi: 10.1093/biomet/85.2.391.

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

[8] M. David. Auto insurance premium calculation using generalized linear models. *Procedia Economics and Finance*, 20:147 – 156, 2015. Globalization and Higher Education in Economics and Business Administration - {GEBA} 2013.

[9] R. S. de Souza, E. Cameron, M. Killedar, J. Hilbe, R. Vilalta, U. Maio, V. Biffi, B. Ciardi, and J. D. Riggs. The Overlooked Potential of Generalized Linear Models in Astronomy - I: Binomial Regression. 2014.

[10] S. Ekioglu, P. Pardalos, and M. Resende. Parallel metaheuristics for combinatorial optimization. In R. Corra, I. Dutra, M. Fiallos, and F. Gomes, editors, *Models for Parallel and Distributed Computation*, volume 67 of *Applied Optimization*, pages 179–206. Springer US, 2002. ISBN 978-1-4419-5219-6.

[11] F. Frommlet, I. Ljubic, B. Arnardttir Helga, and M. Bogdan. Qtl mapping using a memetic algorithm with modifications of BIC as fitness function. *Statistical Applications in Genetics and Molecular Biology*, 11(4):1–26, 2012.

[12] A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014. ISSN 0960-3174.

[13] E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, pages 339–374, 1997.

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

[15] V. Gomez-Rubio and H. Rue. Markov chain monte carlo with INLA. Manuscript, 2016.

[16] L. Grossi and T. Bellini. Credit risk management through robust generalized linear models. In S. Zani, A. Cerioli, M. Riani, and M. Vichi, editors, *Data Analysis, Classification and the Forward Search*,

Studies in Classification, Data Analysis, and Knowledge Organization, pages 377–386. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-35977-7.

[17] C. Hans, A. Dobra, and M. West. Shotgun stochastic search for large p regression. *Journal of the American Statistical Association*, 102(478):507–516, 2007.

[18] F. P. Kelly. *Reversibility and stochastic networks*. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester, 1979. ISBN 0-471-27601-4.

[19] N. Kokash. An introduction to heuristic algorithms, 2009.

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

[21] S. Lobraux and C. Melodelima. Detection of genomic loci associated with environmental variables using generalized linear mixed models. *Genomics*, 105(2):69 – 75, 2015.

[22] G. Luo. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Working paper*.

[23] W. Michiels, E. Aarts, and J. Korst. *Theoretical aspects of local search*, volume 1. Springer, 2007.

[24] J. Piironen and A. Vehtari. Comparison of Bayesian predictive methods for model selection. *ArXiv e-prints*, 2015.

[25] A. E. Raftery, D. Madigan, and J. A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.

[26] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

[27] D. L. T. Rohde. Methods for binary multidimensional scaling. *Neural Comput.*, 14(5):1195–1232, May 2002. ISSN 0899-7667.

[28] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Sosciety*, 71(2): 319–392, 2009.

[29] A. Skrondal and S. Rabe-Hesketh. Some applications of generalized linear latent and mixed models in epidemiology: repeated measures, measurement error and multilevel modeling. *Norwegian Journal of Epidemology*, 13(2):265–278, 2003.

[30] Q. Song and F. Liang. 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, 2015.

[31] G. Storvik. On the flexibility of Metropolis-Hastings acceptance probabilities in auxiliary variable proposal generation. *Scandinavian Journal of Statistics*, 38:342–358, 2011.

[32] H. Tjelmeland and B. K. Hegstad. Mode jumping proposals in MCMC. *Scandinavian journal of statistics*, 28:205–223, 1999.

[33] W. Vandaele. Participation in illegitimate activities: Ehrlich revisited. *Deterrence and Incapacitation*, pages 270–335, 2007.

[34] S. K. S. Walter R. Gilks, Gareth O. Roberts. Adaptive Markov Chain Monte Carlo through regeneration. *Journal of the American Statistical Association*, 93(443):1045–1054, 1998.

[35] Y. T. Yeh, L. Yang, M. Watson, N. Goodman, and P. Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Transactions on Graphics*, 31(4): 56–58, 2012.