Bayesian Analysis (2020) **15**, Number 1, pp. 263–333

A Novel Algorithmic Approach to Bayesian Logic Regression (with Discussion)

Aliaksandr Hubin[*,§], Geir Storvik[†], and Florian Frommlet[‡]

Abstract. Logic regression was developed more than a decade ago as a tool to construct predictors from Boolean combinations of binary covariates. It has been mainly used to model epistatic effects in genetic association studies, which is very appealing due to the intuitive interpretation of logic expressions to describe the interaction between genetic variations. Nevertheless logic regression has (partly due to computational challenges) remained less well known than other approaches to epistatic association mapping. Here we will adapt an advanced evolutionary algorithm called GMJMCMC (Genetically modified Mode Jumping Markov Chain Monte Carlo) to perform Bayesian model selection in the space of logic regression models. After describing the algorithmic details of GMJMCMC we perform a comprehensive simulation study that illustrates its performance given logic regression terms of various complexity. Specifically GMJMCMC is shown to be able to identify three-way and even four-way interactions with relatively large power, a level of complexity which has not been achieved by previous implementations of logic regression. We apply GMJMCMC to reanalyze QTL (quantitative trait locus) mapping data for Recombinant Inbred Lines in *Arabidopsis thaliana* and from a backcross population in *Drosophila* where we identify several interesting epistatic effects. The method is implemented in an R package which is available on github.

Keywords: logic regression, Bayesian model averaging, mode jumping Monte Carlo Markov Chain, genetic algorithm, QTL mapping.

1 Introduction

Logic regression (not to be confused with logistic regression) was developed as a general tool to obtain predictive models based on Boolean combinations of binary covariates (Ruczinski et al., 2003). Its primary application area is epistatic association mapping as pioneered by Ruczinski et al. (2004) and Kooperberg and Ruczinski (2005) although already early on the method was also used in other areas (Keles et al., 2004; Janes et al., 2005). Important contributions to the development of logic regression were later made by the group of Katja Ickstadt (Fritsch, 2006; Schwender and Ickstadt, 2008), which also provided a comparison of different implementations of logic regression (Fritsch and Ickstadt, 2007). Schwender and Ruczinski (2010) gave a brief introduction with various applications and potential extensions of logic regression. Recently a systematic comparison of the performance of logic regression and a more classical regression approach

[*] Department of mathematics, University of Oslo, aliaksah@math.uio.no
[†] Department of mathematics, University of Oslo, geirs@math.uio.no
[‡] Department of Medical Statistics (CEMSIIS), Medical University of Vienna, florian.frommlet@meduniwien.ac.at
[§] Norwegian Computing Center, aliaksandr.hubin@nr.no

https://doi.org/10.1214/18-BA1141

based on Cockerham's coding to detect interactions illustrated the advantages of logic regression to detect epistasic effects in QTL mapping (Malina et al., 2014). Given the potential of logic regression to detect interpretable interaction effects in a regression setting it is rather surprising that it has not yet become wider addressed in applications.

Originally logic regression was introduced together with likelihood based model selection, where simulated annealing served as a strategy to obtain one "best" model (see Ruczinski et al., 2003, for details). However, assuming that there is one "best" model disregards the problem of model uncertainty. Whilst this approach works well in simulation studies, it seems to be quite an unrealistic assumption in real world applications, where there often is no "true" model. Hence Bayesian model averaging, which implicitly takes into account model uncertainty, becomes important. Bayesian versions of logic regression combined with model exploration include Monte Carlo logic regression (MCLR) (Kooperberg and Ruczinski, 2005) and the full Bayesian version of logic regression (FBLR) by Fritsch (2006). Both MCLR and FBLR use Markov Chain Monte Carlo (MCMC) algorithms for searching through the space of models and parameters. Inference is then based on a large number of models instead of just one model as in the original version of logic regression. MCLR utilizes a geometric prior on the size of the model (defined through the number of logic terms and their complexity). All models of the same size get the same prior probability while larger models implicitly are penalized. Regression parameters are marginalized out, significantly simplifying computational complexity. In contrast FBLR is performed on a joint space of parameters and models. FBLR uses multivariate normal priors for regression parameters, while model size is furnished with a slightly different prior serving similar purposes as the MCLR prior. In case of a large number of binary covariates these MCMC based methods might require extremely long Markov chains to guarantee convergence which can make them infeasible in practice. Additionally both of them utilize simple Metropolis-Hastings settings which, together with the fact that the search space is often multimodal, increases the probability that they are stuck in local extrema for a significant amount of time.

In this paper we propose a new approach for Bayesian logic regression including model uncertainty. We introduce a novel prior for the topology of logic regression models which is slightly simpler to compute than the one used by MCLR and which still shows excellent properties in terms of controlling false discoveries. We consider two different priors for regression coefficients: Jeffreys' prior and the robust g-priors as a state of the art choice for priors of regression coefficients in variable selection problems. For Jeffreys' prior computing the marginal likelihoods can be performed with the Laplace approximation as in BIC (Bayesian information criterion) and similar model selection criteria. For the robust g-prior the marginal likelihood is efficiently computed using the integrated Laplace approximation (Li and Clyde, 2018).

The main contribution of this paper is the proposed search algorithm, named GMJMCMC, which provides a better search strategy for exploring the model space than previous approaches. GMJMCMC combines genetic algorithm ideas with the mode jumping Markov Chain Monte Carlo (MJMCMC) algorithm (Tjelmeland and Hegstad, 2001; Hubin and Storvik, 2018) in order to be able to jump between local modes in the model space. After formally introducing logic regression and describing the GMJMCMC algorithm in detail we will present results from a comprehensive simulation study. The performance of GMJMCMC is compared with MCLR and FBLR in case of logistic models (binary responses) and additionally analyzed for linear models (quantitative responses). Models of different complexities are studied which allows us to illustrate the potential of GMJMCMC to detect higher order interactions. Finally we apply our logic regression approach to perform QTL mapping using two publicly available data sets. The first study is concerned with the hypocotyledonous stem length in *Arabidopsis thaliana* using Recombinant Inbred Line (RIL) data (Balasubramanian et al., 2009), the second one considering various traits from backcross data of *Drosophila Simulans* and *Drosophila Mauritana* is presented in the web supplement (Hubin et al., 2018b). The method is implemented as an R package which is freely available on GitHub at http://aliaksah.github.io/EMJMCMC2016/, where one can also find examples of further logic regression applications.

2 Methods

2.1 Logic regression

The method of logic regression (Ruczinski et al., 2003) was specifically designed for the situation where covariates are binary and predictors are defined as logic expressions operating on these binary variables. Logic regression can be applied in the context of the generalized linear model (GLM) as demonstrated in Malina et al. (2014). It can also be easily expanded to the domain of generalized linear mixed models (GLMM), but to keep our presentation as simple as possible we will focus here on generalized linear regression models.

Consider a response variable $Y \in \mathbb{R}$, together with m binary covariates $X_1, X_2, \dots, X_m$. Our primary example will be genetic association studies where, depending on the context, each binary covariate, X_j, $j \in \{1, 2, \dots, m\}$, can have a different interpretation. In QTL mapping with backcross design or recombinant inbred lines X_j simply codes the two possible genetic variants. In case of intercross design or in outbred populations different X_j will be used to code dominant and recessive effects (see for example Malina et al., 2014). We will adapt the usual convention that a value 1 corresponds to logical TRUE and a value 0 to logical FALSE where the immediate interpretation in our examples is that a specific marker is associated with a trait or not. Each combination of the binary variables X_j with the logical operators $\wedge$ (AND), $\vee$ (OR) and X^c (NOT X), is called a logic expression (for example $L = (X_1 \wedge X_2) \vee X_3^c$). Following the nomenclature of Kooperberg and Ruczinski (2005) we will refer to logic expressions as *trees*, whereas the primary variables contained in each tree are called *leaves*. The set of leaves of a tree L will be denoted by $v(L)$, that is for the specified example above we have $v(L) = \{X_1, X_2, X_3\}$.

We will study logic regression in the context of the generalized linear model (GLM, see McCullagh and Nelder (1989)) of the form

$$Y \quad \sim \quad \mathfrak{f}(y \mid \mu(\boldsymbol{X}); \phi)\,, \tag{1}$$

$$h\left(\mu(\boldsymbol{X})\right) \quad = \quad \alpha + \sum_{j=1}^{q} \gamma_j \beta_j L_j, \tag{2}$$

where $\mathfrak{f}$ denotes the parametric distribution of Y belonging to the exponential family with mean $\mu(\boldsymbol{X})$ and dispersion parameter ϕ. The function h is an appropriate link function, α and $\beta_j, j \in \{1, \ldots, q\}$ are unknown regression parameters, and γ_j is the indicator variable which specifies whether the tree L_j is included in the model. For the sake of simplicity we abbreviate by $\mu(\boldsymbol{X})$ the complex dependence of the mean μ on $\boldsymbol{X}$ via the logic expressions L_j according to (2). Our primary examples are linear regression for quantitative responses and logistic regression for dichotomous responses but the implementation of our approach works for any generalized linear model.

We will restrict ourselves to trees with no more than C_{max} leaves. Consequently the total number of trees q will be finite. The considered models are restricted to include no more than k_{max} trees. The vector of binary random variables $M = (\gamma_1, \ldots, \gamma_q)$ fully characterizes a model in terms of which logical expressions are included. Here we go along with the usual convention in the context of variable selection that 'model' refers to the set of regressors and does not take into account the specific values of the non-zero regression coefficients.

Bayesian model specification

For a fully Bayesian approach one needs prior specifications for the model topology characterized by the index vector M as well as for the coefficients α and β_j belonging to a specific model M. This is a common approach in Bayesian model selection, used for example in Clyde et al. (2011) or Hubin and Storvik (2018). We start with defining the prior for M by

$$p(M) \propto \mathbb{I}\left(|M| \leq k_{max}\right) \prod_{j=1}^{q} \rho(\gamma_j). \tag{3}$$

Here $|M| = \sum_{j=1}^{q} \gamma_j$ is the number of logical trees included in the model and k_{max} is the maximum number of trees allowed per model. The factors $\rho(\gamma_j)$ are introduced to give smaller prior probabilities to more complex trees. Specifically we consider

$$\rho(\gamma_j) = a^{\gamma_j c(L_j)} \tag{4}$$

with $0 < a < 1$ and $c(L_j) \geq 0$ being a non-decreasing measure for the complexity of the corresponding logical trees. In case of $\gamma_j = 0$ it holds that $\rho(\gamma_j) = 1$ and thus the prior probability for model M only consists of the product of $\rho(\gamma_j)$ for all trees included in the model. It follows that if M and M' are two vectors only differing in one component, say $\gamma'_j = 1$ and $\gamma_j = 0$, then

$$\frac{p(M')}{p(M)} = a^{c(L_j)} < 1$$

showing that larger models are penalized more. This result easily generalizes to the comparison of more different models and provides the basic intuition behind the chosen prior.

The prior choice implies a distribution for the model size $|M|$ which can be interpreted as a multiple-testing penalty (Scott and Berger, 2008). For $k_{max} = q$ and a constant complexity value on all trees, $|M|$ follows a binomial distribution. With varying complexity measures, $|M|$ follows the *Poisson binomial* distribution (Wang, 1993) which is a unimodal distribution with $E[|M|] = \sum_{j=1}^{q} p_j$ and $\mathrm{Var}[|M|] = \sum_{j=1}^{q} p_j(1-p_j)$ where $p_j = a^{c(L_j)}/(1 + a^{c(L_j)})$. A truncated version of this distribution is obtained for $k_{max} < q$.

The choices of a and the complexity measure $c(L_j)$ are crucial for the quality of the model prior. Let $N(s)$ be the total number of trees having s leaves. Choosing $a = e^{-1}$ and $c(L_j) = \log N(s_j)$ as long as the number of leaves is not larger than C_{max} results for $\gamma_j = 1$ in

$$a^{c(L_j)} = \frac{1}{N(s_j)}\ , \quad s_j \leq C_{max}\ .$$

Therefore the multiplicative contribution of a specific tree of size s to the model prior will be indirectly proportional to the total number of trees $N(s)$ having s leaves as long as $s \leq C_{max}$. Given that $N(s)$ is rapidly growing with the tree size s this choice gives smaller prior probabilities for larger trees. The resulting penalty closely resembles the Bonferroni correction in multiple testing as discussed for example by Bogdan et al. (2008) in the context of modifications of the BIC.

The number $N(s)$ will in practice be difficult to compute. To compute a rough approximation of $N(s)$ we ignore logic expressions including the same variable multiple times. Then there are $\binom{m}{s}$ possibilities to select variables. Each variable can undergo logic negation giving s binary choices and furthermore there are $s-1$ logic symbols $(\vee, \wedge)$ to be chosen resulting in 2^{2s-1} different expressions. However, due to De Morgan's law half of the expressions provide identical logic regression models. This gives

$$N(s) \approx \binom{m}{s}\, 2^{2s-2}. \tag{5}$$

Using this approximation, for a model of size $k = |M|$ the full model prior is of the form

$$P(M) \propto \mathbb{I}\,(k \leq k_{max}) \prod_{r=1}^{k} \frac{\mathbb{I}\,(s_{j_r} \leq C_{max})}{\binom{m}{s_{j_r}} 2^{2s_{j_r}-2}}\ , \tag{6}$$

where $j_1, \ldots, j_k$ refer to the k trees of model M.

We will next discuss priors for the parameters given a specific model M. The GLM formulation (1) includes a dispersion parameter ϕ, which for example in case of the linear model is connected with the variance term σ^2 for the underlying normal distribution. If a GLM has a dispersion parameter then for the sake of simplicity we will adapt the commonly used improper prior (Li and Clyde, 2018; Bayarri et al., 2012)

$$\pi(\phi) = \phi^{-1}. \tag{7}$$

If a GLM does not include a dispersion parameter (like logistic regression) then one simply sets $\phi = 1$.

Concerning the intercept α and the regression coefficients β_j, where $j \in \{j_1, \ldots, j_{|M|}\}$ correspond to the non-zero coefficients of model M, we will consider two different types of priors, simple Jeffreys' priors and robust g-priors. Jeffreys' prior (Jeffreys, 1946, 1961; Gelman et al., 2013) assumes for the parameters of the model an improper prior distribution of the form

$$\pi_\alpha(\alpha)\pi_\beta(\boldsymbol{\beta}) = |\mathcal{J}_n(\alpha, \boldsymbol{\beta})|^{\frac{1}{2}}, \tag{8}$$

where $\mathcal{J}_n(\alpha, \boldsymbol{\beta})$ is the observed information.

To obtain model posterior probabilities one needs to evaluate the marginal likelihood of the model $P(Y \mid M)$ by integrating over all parameters of the model which is often a fairly difficult task. The greatest advantage of Jeffreys' prior is that this integral can be approximated simple and accurate through the Laplace approximation. In case of the Gaussian model choosing Jeffreys' prior (8) for the coefficients and the simple prior (7) for the variance term yields that the Laplace approximation becomes exact (Raftery et al., 1997) and gives a marginal likelihood of the simple form

$$P(Y \mid M) \propto P(Y \mid M, \hat{\theta})\; n^{\frac{|M|}{2}}, \tag{9}$$

where $\hat{\theta}$ refers to the maximum likelihood estimates of all parameters involved. On the log scale this exactly corresponds to the BIC model selection criterion (Schwarz, 1978) when using a uniform model prior. In case of logistic regression the marginal likelihood under Jeffreys' prior becomes approximately (9) with an error of order $O(n^{-1})$ (Tierney and Kadane, 1986; Claeskens and Hjort, 2008). Barber et al. (2016) also describe that Laplace approximations of the marginal likelihood yield very accurate results and can be trusted in Bayesian model selection problems.

Although there are many situations in which selection based on BIC like criteria works well, within the Bayesian literature using Jeffreys' prior for model selection has been widely criticized for not being consistent once the true model coincides with the null model (all $\gamma_j = 0$, Bayarri et al., 2012). A large number of alternative priors have been studied, see for example Li and Clyde (2018) who give a comprehensive review on the state of the art of g-priors. In a recent paper Bayarri et al. (2012) gave theoretical arguments in case of the linear model recommending the *robust* g-prior, which is consistent in all situations and yields errors diminishing significantly faster than other prior choices. Thus we will introduce the robust g-prior as an alternative to Jeffreys' prior.

Our description of robust g-priors follows Li and Clyde (2018) who consider an improper constant prior for the intercept, $P(\alpha) \propto 1$, and a mixture g-prior for the regression coefficients $\beta_j, j \in \{j_1, \ldots, j_{|M|}\}$ of the form

$$P(\boldsymbol{\beta} \mid g) \sim N_{|M|}\left(\mathbf{0}, g \cdot \phi \mathcal{J}_n(\boldsymbol{\beta})^{-1}\right). \tag{10}$$

Here $\mathcal{J}_n(\boldsymbol{\beta})$ is the subblock of the full observed information matrix $\mathcal{J}_n(\alpha, \boldsymbol{\beta})$ related to $\boldsymbol{\beta}$ and g itself is assumed to be distributed according to the so called truncated Compound Confluence Hypergeometric (tCCH) prior

$$P\left(\frac{1}{1+g}\right) \sim tCCH\left(\frac{a}{2}, \frac{b}{2}, r, \frac{s}{2}, v, \kappa\right). \tag{11}$$

This family of mixtures of g-priors includes a large number of priors discussed in the literature, see Li and Clyde (2018) for more details. The recommended robust g-prior is a particular case with the following choice of parameters:

$$a = 1, b = 2, r = 1.5, s = 0, v = \frac{n+1}{|M|+1}, \kappa = 1.$$

Under this prior specification precise integrated Laplace approximations of the marginal likelihood for GLM are given by Li and Clyde (2018), whilst exact values are available for Gaussian models (Li and Clyde, 2018; Bayarri et al., 2012).

2.2 Computing posterior probabilities

Given prior probabilities for any logic regression model M the model posterior probability can be computed according to Bayes formula as

$$P(M \mid Y) = \frac{P(Y \mid M)P(M)}{\sum_{M' \in \Omega} P(Y \mid M')P(M')} , \tag{12}$$

where $P(Y \mid M)$ denotes the integrated (or marginal) likelihood for model M and Ω is the set of all models in the model space. The sum in the denominator involves a huge number of terms and it is impossible to compute all of them. Classical MCMC based approaches (like MCLR and FBLR) overcome this problem by estimating model posteriors with the relative frequency with which a specific model M occurs in the Markov chain. In case of an ultrahigh-dimensional model space (like in case of logic regression) this is computationally extremely challenging and might require chain lengths which are prohibitive for practical applications.

An alternative approach makes use of the fact that most of the summands in the denominator of (12) will be so small that they can be neglected. Considering a subset $\Omega^* \subseteq \Omega$ containing the most important models we can therefore approximate (12) by

$$P(M \mid Y) \approx \tilde{P}(M \mid Y) = \frac{P(Y \mid M)P(M)}{\sum_{M' \in \Omega^*} P(Y \mid M')P(M')} . \tag{13}$$

To obtain good estimates we have to search in the model space for those models that contribute significantly to the sum in the denominator, that is for those models with large posterior probabilities or equivalently with large values of $P(Y \mid M)P(M)$. In Frommlet et al. (2012) specific memetic algorithms were developed to perform the model search for linear regression. Here we will rely upon the GMJMCMC algorithm, which is described in the next section. For now we assume that some method for computing the marginal likelihood $P(Y \mid M)$ is available. The details of such computation depend on the prior specifications of the parameters of a particular model and are given for the examples in the experimental sections.

Based on model posterior probabilities one can easily obtain an estimate of the posterior probability for a logic expression L_j to be included in a model (also referred to as the marginal inclusion probability) by

$$\tilde{P}(L_j \mid Y) = \sum_{M \in \Omega^* : \gamma_j = 1} \tilde{P}(M \mid Y).^{1} \tag{14}$$

Inference on trees can then be performed by means of selecting those trees with a posterior probability being larger than some threshold probability π_C. In case of exploratory studies where the main aim is to discover many potentially interesting features to be explored in further studies it can be reasonable to use low threshold values on $\tilde{P}(L_j \mid Y)$. High threshold values can be used if false discoveries need to be avoided. In general the threshold can be specified through a decision theoretic framework, including the aim of controlling false discovery rates, see (Wakefield, 2007).

A threshold of 0.5 corresponds to the median probability model of Barbieri et al. (2004) which under certain circumstances has greater predictive power than the most probable model. However, one of the criteria for the median probability model to be optimal in the linear Gaussian case, the graphical model structure criterion, will not always be valid in cases where one makes restrictions on the number of trees that can be included. The graphical model structure criterion requires that the median probability model results in a legal model. Consider the case with three covariates x_1, x_2, x_3 but with $k_{max} = 2$ and the posterior probabilities for models $\boldsymbol{\gamma} = (1,1,0)$, $\boldsymbol{\gamma} = (1,0,1)$ and $\boldsymbol{\gamma} = (0,1,1)$ each equal to $1/3$. Then all marginal inclusion probabilities are $2/3$ and the median probability model includes all variables which then has a model size larger than k_{max}. The median probability model can however still be a useful model to consider even in cases where the optimality results do not apply.

2.3 The GMJMCMC algorithm

To fix ideas consider first a variable selection problem with q potential covariates to enter a model. Recall that γ_j needs to be 1 if the j-th variable is to be included into the model and 0 otherwise. A model M is thus specified by the vector $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_q)$ and the general model space Ω is of size 2^q. If this discrete model space is multimodal in terms of model posterior probabilities then simple MCMC algorithms typically run into problems by staying for too long in the vicinity of local maxima. Recently, the mode jumping MCMC procedure (MJMCMC) was proposed by Hubin and Storvik (2018) to overcome this issue in a model selection setting.

MJMCMC is a proper MCMC algorithm equipped with the possibility to jump between different modes within the discrete model space. The key to the success of MJMCMC is the generation of good proposals of models which are not too close to the current state. This is achieved by first making a large jump (changing many model components) and then performing local optimization within the discrete model space to obtain a proposal model. Within a Metropolis-Hastings setting a valid acceptance probability is then constructed using symmetric backward kernels, which guarantees that the resulting Markov chain is ergodic and has the desired limiting distribution (Tjelmeland and Hegstad, 2001; Hubin and Storvik, 2018).

[1] Here by $P(L_j \mid Y)$ we mean $P(\gamma_j = 1 \mid Y)$.

The MJMCMC algorithm requires that all of the covariates defining the model space are known in advance and are all considered at each iteration of the algorithm. In case of logic regression the covariates are trees and a major problem in this setting is that it is quite difficult to fully specify the space Ω. In fact it is even difficult to specify q, the total number of feasible trees. To solve this problem we present an adaptive algorithm called Genetically Modified MJMCMC (GMJMCMC), where MJMCMC is embedded in the iterative setting of a genetic algorithm. In each iteration only a given set $\mathcal{S}$ of trees (of fixed size d) is considered. Each $\mathcal{S}$ then induces a separate *search space* for MJMCMC. In the language of genetic algorithms $\mathcal{S}$ is the *population*, which dynamically evolves to allow MJMCMC exploring different reasonable parts of the unfeasibly large total search space.

To be more specific, we consider different populations $\mathcal{S}_1, \mathcal{S}_2, \ldots$ where each $\mathcal{S}_t$ is a set of d trees. For each given population a fixed number of MJMCMC steps is performed. Since the MJMCMC algorithm is specified in full detail in Hubin and Storvik (2018), we will concentrate here on describing the evolutionary dynamics yielding subsequent populations $\mathcal{S}_t$. Utilization of the approximation (13) in combination with exact or approximated marginal likelihoods allows us to compute posterior probabilities for all models in Ω^* which have been visited at least once by the algorithm. Consequently we do not need a proper MCMC (an algorithm with convergence towards the target distribution) which is needed if model posterior probabilities are estimated by the relative frequency of how often a model has been visited. In principle it is possible to construct a proper MCMC algorithm which aims at simulating from extended models of the form $P(M, \mathcal{S} \mid Y)$ having $P(M \mid Y)$ as a stationary distribution. This version of the algorithm is considered in (Hubin et al., 2018a) where the main idea is to perform both forward and backward swaps between populations in order to obtain a reversible Markov chain.

The algorithm is initialized by first running MJMCMC for a given number of iterations N_{init} on the set of all binary covariates $X_1, \ldots, X_m$ as potential regressors, but not including any interactions. The first $d_1 < d$ members of population $\mathcal{S}_1$ are then defined to be the d_1 covariates with largest marginal inclusion probability. In our current implementation we select the d_1 leaves which have marginal posterior probabilities (estimated from the first N_{init} iterations) larger than ρ_{min}, thus d_1 is not pre-specified but is obtained in a data driven way. For later reference we denote this set of d_1 leaves by $\mathcal{S}_0$. The remaining $d - d_1$ members of $\mathcal{S}_1$ are obtained by forming logic expressions from the leaves of $\mathcal{S}_0$ where trees are generated randomly by means of the crossover operation described below. In practice one first has to choose some k_{max} which will depend on the expected number of trees to enter the model in the problem one studies. The choice of d can then be guided by the results of Theorem 1 given below.

After $\mathcal{S}_1$ has been initialized MJMCMC is performed for a fixed number of iterations N_{expl} before the next population $\mathcal{S}_2$ is generated. This process is iterated for T_{max} populations $S_t, t \in \{1, \ldots, T_{max}\}$. The d_1 input trees from the initialization procedure remain in all populations $\mathcal{S}_t$ throughout our search. Other trees from the population $\mathcal{S}_t$ with low marginal inclusion probabilities (below a threshold ρ_{min}) will be substituted by trees which are generated by crossover, mutation and reduction operators to be described in more detail below.

Let D_t be the set of trees to be deleted from $\mathcal{S}_t$. Then $|D_t|$ replacement trees must be generated instead. Each replacement tree is generated randomly by a *crossover* operator with probability P_c and by a *mutation* operator with probability $P_m = 1 - P_c$. A *reduction* operator is applied if *mutation* or *crossover* gives a tree larger than the maximal tree size C_{max}.

Crossover: Two *parent trees* are selected from $\mathcal{S}_t$ with probabilities proportional to the approximated marginal inclusion probabilities of trees in $\mathcal{S}_t$. Then each one of the parents is inverted with probability P_{not} by the logical not ^{c} operator, before they are combined with a $\wedge$ operator with probability P_{and} and with a $\vee$ operator otherwise. Hence the crossover operator gives trees of the form $L_{j_1} \wedge L_{j_2}$ or $L_{j_1} \vee L_{j_2}$ where either L_{j_i} or $L^c_{j_i}$ is in $\mathcal{S}_t$ for $i = 1, 2$.

Mutation: One parent tree is selected from $\mathcal{S}_t$ with probability proportional to the approximated marginal inclusion probabilities of trees in $\mathcal{S}_t$, whilst the other parent tree is selected uniformly from the set of $m - d_1$ leaves which did not make it into the initial population $\mathcal{S}_0$. Then just like for the crossover operator each of the parents is inverted with probability P_{not} by the logical not ^{c} operator, before they are combined with a $\wedge$ operator with probability P_{and} and with a $\vee$ operator otherwise. The mutation operator gives trees of the form $L_{j_1} \wedge X$ or $L_{j_1} \vee X$ where either L_{j_1} or $L^c_{j_1}$ is in $\mathcal{S}_t$ and X or X^c is in D_0.

Reduction: A new tree is generated from a tree by deleting a subset of leaves, where each leave has a probability of ρ_{del} to be deleted. The pruning of the tree is performed in a natural way meaning that the 'closest' logical operators of the deleted leaves are also deleted. If the deleted leave is not on the boundaries of the original tree the operation is resulting in obtaining two separated subtrees. The resulting subtrees are then combined in a tree with a $\wedge$ operator with probability P_{and} or with a $\vee$ operator otherwise.

For all three operators it holds that if the newly generated tree is already present in $\mathcal{S}_t$ then it is not considered for $\mathcal{S}_{t+1}$ but rather a new replacement tree is proposed instead. The pseudo-code **Algorithm 1** describes the full GMJMCMC algorithm. For each iteration t the initial model for the next MJMCMC run is constructed by randomly selecting trees from $\mathcal{S}_t$ with probability P_{init}. For the final population $\mathcal{S}_{T_{max}}$, MJMCMC is run until M_{fin} unique models are visited (within $\mathcal{S}_{T_{max}}$). M_{fin} should be sufficiently large to obtain good MJMCMC based approximations of the posterior parameters of interest based on the final search space $\mathcal{S}_{T_{max}}$.

The following result is concerned with consistency of probability estimates of GMJMCMC when the number of iterations increases.

Theorem 1. *Assume Ω^* is the set of models visited through the GMJMCMC algorithm where $d - d_1 \geq k_{max}$. Assume further the marginal likelihoods are calculated without errors. Then the model estimates based on* (13) *will converge to the true model probabilities as the number of iterations $T_{\max}$ goes to ∞.*

Proof. Note that the approximation (13) will provide the exact answer if $\Omega^* = \Omega$. It is therefore enough to show that the algorithm in the limit will have visited all possible

Algorithm 1 GMJMCMC.

```
Run the MJMCMC algorithm for $N_{init}$ iterations on $X_1, \dots, X_m$ and define $\mathcal{S}_0$ as the set of $d_1$ variables among them with the largest estimated marginal inclusion probabilities.
Generate $d - d_1$ trees by randomly selecting crossover operations of elements from $\mathcal{S}_0$ and add those trees to the set $\mathcal{S}_0$ to obtain $\mathcal{S}_1$.
Run the MJMCMC algorithm within search space $\mathcal{S}_1$.
for $t = 2, \dots, T_{max}$ do
    Delete trees within $\mathcal{S}_{t-1} \backslash \mathcal{S}_0$ which have estimated inclusion probabilities less than $\rho_{min}$.
    Add new trees which are generated by crossover, mutation or reduction operators until the having again a set of size $d$, which becomes $\mathcal{S}_t$.
    Run the MJMCMC algorithm within search space $\mathcal{S}_t$.
end for
```

models. Since $\mathcal{S}_0$ is generated in the first step and never changed, we will consider it to be fixed.

Define M_{S_t} to be the last model visited by the MJMCMC algorithm on search space $\mathcal{S}_t$. Then the construction of $\mathcal{S}_{t+1}$ only depends on $(\mathcal{S}_t, M_{S_t}, \boldsymbol{X})$ while $M_{\mathcal{S}_{t+1}}$ only depends on $\mathcal{S}_{t+1}$. Therefore $\{(\mathcal{S}_t, M_{\mathcal{S}_t}, \boldsymbol{X})\}$ is a Markov chain. Assume now $\mathcal{S}$ and $\mathcal{S}'$ are two populations differing in one component with $L \in \mathcal{S}$, $L' \in \mathcal{S}'$, $L \neq L'$. Define L_{sub} to be any tree that is a subtree of both L and L' (where a subtree is defined as a tree which can be obtained by reduction) and $\mathcal{S}_{sub}$ to be the search space where L is substituted with L_{sub} in $\mathcal{S}$. Then it is possible to move from S to S_{sub} in l steps using first *mutations* and *crossovers* to grow a tree L^* of size larger than C_{max}, which can undergo *reduction* (note that although only trees that have low enough estimated marginal inclusion probabilities can be deleted, there will always be a positive probability that marginal inclusion probabilities are estimated to be smaller than the threshold ρ_{min}) to get to L_{sub}. Further, assuming the difference in size between L_{sub} and L' is r, a move from S_{sub} to S' can be performed by r steps of *mutations* or *crossovers.* Two search spaces which differ in s trees can be reached by s combinations of the moves described above. Since also any model within a search space can be visited, the Markov chain $\{(\mathcal{S}_t, M_{\mathcal{S}_t}, \boldsymbol{X})\}$ is irreducible. Since the state space for this Markov chain is finite, it is also recurrent, and there exists a stationary distribution with positive probabilities on every model. Thereby, all states, including all possible models of maximum size d, will eventually be visited.

When $d_1 > 0$, some restrictions on the possible search spaces are introduced. However, when $d - d_1 \geq k_{max}$, any model of maximum size k_{max} *will* eventually be visited. □

Remark 1. If $d - d_1 < k_{max}$, then every model of size up to $d - d_1$ plus some of the larger models will eventually be visited, although the model space will get some additional constraints. In practice it is more important that $d - d_1 \geq k^*$, where k^* is the size of the true model. Unfortunately neither k^* nor d_1 are known in advance, and one has to make reasonable choices of k_{max} and d depending on the problem one analyses.

Remark 2. The result of Theorem 1 relies on exact calculation of the marginal likelihood $P(Y \mid M)$. Apart from the linear model, the calculation of $P(Y \mid M)$ is typically based on an approximation, giving similar approximations to the model probabilities. How precise these approximations are will depend on the type of method used. The current implementation includes Laplace approximations, integrated Laplace approximations, and integrated nested Laplace approximations. In principle other methods based on MCMC outputs (Chib, 1995; Chib and Jeliazkov, 2001) could be incorporated relatively easily resulting however in longer runtimes.

Parallelization

Due to our interest in exploring as many *unique* high quality models as possible and doing it as fast as possible, running multiple parallel chains is likely to be computationally beneficial compared to running one long chain. The process can be embarrassingly parallelized into B chains using several CPUs (Central processing units), GPUs (graphics processing units) or clusters. If one is mainly interested in model probabilities, then Equation (13) can be directly applied with Ω^* now being the set of unique models visited within all runs. However, we suggest a more memory efficient approach. If some statistic Δ is of interest, one can utilize the following posterior estimates based on weighted sums over individual runs:

$$\tilde{P}(\Delta \mid Y) = \sum_{b=1}^{B} w_b \tilde{P}_b(\Delta \mid Y). \tag{15}$$

Here w_b is a set of weights which will be specified below and $\tilde{P}_b(\Delta \mid Y)$ are the posteriors obtained with formula (4) from run b of GMJMCMC.

Due to the irreducibility of the GMJMCMC procedure it holds that for $\sum_b w_b = 1$ we obtain $\lim_{T_{max}\to\infty} \tilde{P}(\Delta \mid Y) = P(\Delta \mid Y)$ where T_{max} is the number of iterations within each run. Thus for any set of normalized weights the approximation $\tilde{P}(\Delta \mid Y)$ converges to the true posterior probability $P(\Delta \mid Y)$. Therefore in principle any normalized set of weights w_b would work, like for example $w_b = \frac{1}{B}$. However, uniform weights have the disadvantage to potentially give too much weight to posterior estimates from chains that have not quite converged. In the following heuristic improvement w_b is chosen to be proportional to the posterior mass detected by run b,

$$w_b = \frac{\sum_{M' \in \Omega_b^*} P(Y \mid M')P(M')}{\sum_{b=1}^{B} \sum_{M' \in \Omega_b^*} P(Y \mid M')P(M')} \,.$$

This choice indirectly penalizes chains that cover smaller portions of the model space. When estimating posterior probabilities using these weights we only need, for each run, to store the following quantities: $\tilde{P}_b(\Delta \mid Y)$ for all statistics Δ of interest and $s_b = \sum_{M' \in \Omega_b^*} P(Y \mid M')P(M')$ as a '*sufficient*' statistic of the run. There is no further need of data transfer between processes.

Alternatively (as mentioned above) one might use (4) directly to approximate $P(\Delta \mid Y)$ based on the totality Ω^* of unique models explored through all of the parallel chains. This procedure might give in some cases slightly better precision than the weighted sum approach (15), but it is still only asymptotically unbiased. Moreover keeping track of all models visited by all chains requires significantly more storage in the quick memory and RAM and requires significantly more data transfers across the processes. Consequently this approach is not part of the current implementation of GMJMCMC.

The consistency result of Theorem 1 also holds in case of the suggested embarrassing parallelization. Moreover it holds that even when the number of iterations per chain is finite that letting the numbers of chains B go to infinity yields consistency of the posterior estimates as shown in Theorem A.1 in the web supplement. The main practical consequence is that running more chains in parallel allows for having a smaller number of iterations within each thread.

Choice of algorithmic parameters Apart from the number of parallel chains, the GMJMCMC algorithm relies upon the choice of a number of tuning parameters which were described above. Section A of the web supplement presents the values that were used in the following simulation study and in real data analysis.

3 Experiments

3.1 Simulation study

The GMJMCMC algorithm was evaluated in a simulation study divided into two parts. The first part considered three scenarios (numbered 1–3) with binary responses and the second part three scenarios (4–6) with quantitative responses. For each scenario we generated $N = 100$ datasets according to a regression model described by Equations (1) and (2) with $n = 1000$ observations and $p = 50$ binary covariates. The covariates were assumed to be independent and were simulated for each simulation run as $X_j \sim \text{Bernoulli}(0.3)$ for $j \in \{1, \dots, 50\}$ in the first two scenarios and as $X_j \sim \text{Bernoulli}(0.5)$ for $j \in \{1, \dots, 50\}$ in the last four scenarios. All computations were performed on the Abel cluster.[2]

For Scenarios 3, 5 and 6 the effect sizes (β_j's) for higher order interactions might seem unrealistically large compared to real applications. To obtain more realistic scenarios with moderate effect sizes and still sufficient power to detect larger trees one would have to increase the sample sizes. However, this would be quite challenging computationally for a simulation study. In the section on sensitivity analysis additional simulations for Scenario 5 illustrate which effect sizes are needed with a sample size of $n = 1000$ for GMJMCMC to detect trees of different size. Furthermore, we demonstrate that increasing the sample size by a factor 10 and reducing the effect sizes by a factor $1/\sqrt{10}$ yields approximately the same power. This relationship indicates which sample sizes

[2] The Abel cluster node (http://www.uio.no/english/services/it/research/hpc/abel/) with 16 dual Intel E5-2670 (Sandy Bridge, 2.6 GHz.) CPUs and 64 GB RAM under 64 bit CentOS-6 is a shared resource for research computing.

would be necessary in practice to detect higher order interactions with smaller effect sizes.

Binary responses

The responses of the first three scenarios were sampled as modes of Bernoulli random variables with individual success probability π specified according to

$$\begin{aligned}\mathbf{S.1}: \operatorname{logit}(\pi) &= -\,0.7 + L_1 + L_2 + L_3,\\ \mathbf{S.2}: \operatorname{logit}(\pi) &= -\,0.45 + 0.6\ L_1 + 0.6\ L_2 + 0.6\ L_3,\\ \mathbf{S.3}: \operatorname{logit}(\pi) &= \quad 0.4 - 5\ L_1 + 9\ L_2 - 9\ L_3,\end{aligned}$$

where the corresponding logic expressions are provided in Table 1. The first two scenarios with models including only two-way interactions were copied from Fritsch (2006) except that we deliberately did not specify the trees in lexicographical order. The reason for this is that for some procedures (like stepwise search) it might be an algorithmic advantage if the effects are specified in a particular order. The second scenario is slightly more challenging than the first one due to the smaller effect sizes. The third scenario is more demanding with a model including three-way and four-way interactions. As mentioned above the corresponding regression coefficients were chosen rather large to make sure that these higher order trees can be detected for the given sample size. In practice when interested in smaller effects one would need larger sample sizes.

For the binary response scenarios GMJMCMC was compared with FBLR (Fritsch, 2006) and MCLR (Kooperberg and Ruczinski, 2005), where GMJMCMC was run with Jeffreys' prior as well as with the robust g-prior. For GMJMCMC the default setting of the maximal number of leaves per tree is $C_{max} = 5$. For Scenarios 1 and 2 we additionally report the results for $C_{max} = 2$, which were the values used in the original study of Fritsch (2006) and which we also used here for MCLR and FBLR. For Scenario 3 we set $C_{max} = 5$ for all three approaches. The maximal number of trees per model was set to $k_{max} = 10$ for GMJMCMC and FBLR whereas for MCLR it is only possible to specify a maximum of $k_{max} = 5$. This is apparently due to the complexity of prior computations in MCLR. Apart from the specification of C_{max} and k_{max} we used for all 3 algorithms their default priors. In all scenarios we used $d = 15$ for the population size in GMJMCMC.

GMJMCMC was run until up to 1.6×10^6 models were visited in the first two scenarios and up to 2.7×10^6 models were visited for the third scenario (divided approximately equally on 32 parallel runs). The length of the Markov chains for FBLR and MCLR were chosen to be 2×10^6 for the first two scenarios and 3×10^6 for the third scenario.

By default a tree is classified as detected if the (estimated) marginal inclusion probability is larger than 0.5. This corresponds to the median probability model of Barbieri et al. (2004). To evaluate the performance of the different algorithms we estimated the following metrics:

Individual power – the power to detect a particular tree from the data generating model;

Overall power – the average power over all true trees;

FP – the expected number of false positive trees;

FDR – the false discovery rate of trees;

WL – the total number of wrongly detected leaves.

Further computational details are given in Section B.1 of the web supplement.

	FBLR	MCLR	GMJMCMC	
Scenario 1			**Jef.**	**R. g**
$L_1 = X_1^c \wedge X_4$	0.30	≤ 0.67	0.99 (0.97)	1.00 (0.98)
$L_2 = X_5 \wedge X_9$	0.42	≤ 0.61	0.99 (1.00)	0.96 (0.95)
$L_3 = X_{11} \wedge X_8$	0.33	≤ 0.59	0.95 (0.91)	0.53 (0.77)
Overall Power	0.35	≤ 0.62	0.98 (0.96)	0.84 (0.90)
FP	3.88	≥ 2.70	0.08 (0.25)	1.01 (0.63)
FDR	0.77	≥ 0.06	0.03 (0.06)	0.25 (0.16)
WL	1	0	0 (0)	0 (0)
Scenario 2				
$L_1 = X_1^c \wedge X_4$	0.32	≤ 0.66	0.98 (0.97)	0.98 (0.97)
$L_2 = X_5 \wedge X_9$	0.40	≤ 0.67	0.99 (0.99)	0.94 (0.96)
$L_3 = X_{11} \wedge X_8$	0.37	≤ 0.60	0.96 (0.86)	0.54 (0.76)
Overall Power	0.36	≤ 0.64	0.98 (0.94)	0.82 (0.90)
FP	3.83	≥ 2.58	0.10 (0.38)	1.08 (0.66)
FDR	0.75	≥ 0.06	0.03 (0.09)	0.27 (0.16)
WL	1	1	0 (0)	0 (0)
Scenario 3				
$L_1 = X_2 \wedge X_9$	0.93	≤ 0.93	1.00	1.00
$L_2 = X_7 \wedge X_{12} \wedge X_{20}$	0.04	≤ 0.67	0.91	0.56
$L_3 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.00	≤ 0.19	1.00	0.56
Overall Power	0.32	≤ 0.60	0.97	0.71
FP	6.40	≥ 2.98	0.15	1.74
FDR	0.54	≥ 0.06	0.04	0.39
WL	90	72	1	0

Table 1: Results for the three simulation scenarios for binary responses. Power for individual trees, overall power, expected number of false positives (FP) and FDR are compared between FBLR, MCLR and GMJMCMC using either Jeffreys' prior (Jef.) or the robust g-prior (R.g.). For GMJMCMC the default $C_{max} = 5$ is used. For the first two scenarios we also present results for $C_{max} = 2$ (inside parentheses) corresponding to the parameters used by MCLR and FBLR. All algorithms were tuned to use approximately the same computational resources. In case of MCLR we can only provide upper bounds for the power and lower bounds for FP. We also report the total number of wrongly detected leaves (WL) over all simulation runs.

A summary of the results for the first three simulation scenarios is provided in Table 1. In all three scenarios, MCLR performed better than FBLR, even when taking into account the positively biased summary statistics of MCLR (see Section B.1 in the web supplement). On the other hand, GMJMCMC clearly outperformed MCLR and FBLR both in terms of power and in terms of controlling the number of false positives, where using Jeffreys' prior gave slightly better results than using the robust g-prior.

In the first two scenarios GMJMCMC with Jeffreys' prior worked almost perfectly both for $C_{max} = 5$ and $C_{max} = 2$. In the few instances where it did not detect the true tree it reported instead the two corresponding main effects. Note however that in case of $C_{max} = 5$ there were several instances where GMJMCMC detected $L_i^c \wedge L_j^c$ with $(1 \leq i < j \leq 3)$, which according to De Morgan's law is equivalent to $L_i + L_j$ and was therefore counted as true positive both for L_i and L_j. GMJMCMC with the robust g-prior had a few more instances where pairs of singletons were reported instead of the correct two-way interaction, especially when $C_{max} = 5$ was used. FBLR and MCLR were also good at detecting the true leaves in these simple scenarios, but GMJMCMC was much better in terms of identifying the exact logical expressions.

The third scenario is more complex than the previous ones but nevertheless GMJMCMC with Jeffreys' prior performed almost perfectly. GMJMCMC with the robust g-prior had more difficulties to correctly identify the three-way and four-way interaction. Both FBLR and MCLR had severe problems to detect the true logic expressions and they also reported a considerable number of wrongly detected leaves. For a more in depth discussion of these simulation results we refer to Section B.1 of the web supplement.

Continuous responses

Responses were simulated according to a Gaussian distribution with error variance $\sigma^2 = 1$ and the following three models for the expectation:

$$
\begin{aligned}
\mathbf{S.4} &: E(Y) = 1{+}1.43\ L_1 + 0.89\ L_2 + 0.7\ L_3,\\
\mathbf{S.5} &: E(Y) = 1{+}1.5\ L_1 + 3.5\ L_2 + 9\ L_3 + 7\ L_4,\\
\mathbf{S.6} &: E(Y) = 1{+}1.5\ L_1 + 1.5\ L_2 + 6.6\ L_3 + 3.5\ L_4\\
&\qquad +9\ L_5 + 7\ L_6 + 7\ L_7 + 7\ L_8.
\end{aligned}
$$

The logic expressions used in the three different scenarios are provided in Table 2. Scenario 4 is similar to the first two scenarios for binary responses and contains only two-way interactions. The models of the last two scenarios both include trees of size 1 to 4, where Scenario 5 has one tree of each size. Scenario 6 is the most complex one with two trees of each size, resulting in a model with 20 leaves in total.

For scenarios with Gaussian observations we were only able to study the performance of GMJMCMC since the other approaches cannot handle continuous responses (MCLR has an implementation but that did not work properly). For these scenarios the settings of GMJMCMC were adapted to the increasing complexity of the model. We used $k_{max} =$

Scenario 4	**Jeffreys**	**Robust g**
$L_1 = X_5 \wedge X_9$	1.00	1.00
$L_2 = X_8 \wedge X_{11}$	0.99	1.00
$L_3 = X_1 \wedge X_4$	0.97	0.98
Overall Power	0.99	0.99
FP	0.01	0.00
FDR	0.005	0.00
WL	0	0
Scenario 5	**Jeffreys**	**Robust g**
$L_1 = X_{37}$	1.00	1.00
$L_2 = X_2 \wedge X_9$	1.00	0.99
$L_3 = X_7 \wedge X_{12} \wedge X_{20}$	0.96	1.00
$L_4 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.89	0.90
Overall Power	0.96	0.97
FP	0.37	0.28
FDR	0.06	0.04
WL	2	5
Scenario 6	**Jeffreys**	**Robust g**
$L_1 = X_7$	0.95	0.99
$L_2 = X_8$	0.98	0.99
$L_3 = X_2 \wedge X_9$	0.98	0.99
$L_4 = X_{18} \wedge X_{21}$	0.96	0.95
$L_5 = X_1 \wedge X_3 \wedge X_{27}$	1.00	1.00
$L_6 = X_{12} \wedge X_{20} \wedge X_{37}$	0.95	0.96
$L_7 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.32	0.45
$L_8 = X_{11} \wedge X_{13} \vee X_{19} \wedge X_{50}$	0.21 (0.93)	0.16 (0.85)
Overall Power	0.79 (0.88)	0.81 (0.90)
FP	4.28 (2.05)	4.24 (1.96)
FDR	0.38 (0.19)	0.36 (0.16)
WL	3	7

Table 2: Results for the three simulation scenarios for linear regression. Power for individual trees, overall power, expected number of false positives (FP), FDR and the total number of wrongly detected leaves (WL) are given for parallel GMJMCMC. The four estimates in parentheses for Scenario 6 refer to results obtained when counting an equivalent logic expression of L_8 as true positive as explained in the text.

$10, 10$ and 20, and $d = 15, 20$ and 40, respectively, for the three scenarios thus allowing for models larger than twice the size of the data generating model and populations at least twice the size of the number of correct leaves involved. Furthermore, the total number of models visited by GMJMCMC before it stopped was increased to 3.5×10^6 for Scenario 6. C_{max} is set to 5 for all three of these scenarios. Otherwise all parameters of GMJMCMC were set as described for the binary responses.

Table 2 summarizes the results and further details are provided in Section B.2 of the web supplement. Scenario 4 illustrates that given a sufficiently large sample size GMJMCMC can reliably detect two-way interactions with effect sizes smaller than one standard deviation. Both Jeffreys' prior and the robust g-prior worked almost perfectly in terms of power. In this simple scenario even the type I error was almost perfectly controlled with false discovery rates equal to 0.005 for Jeffreys' prior and 0 for the robust g-prior. Interestingly the only false discovery over all 100 simulation runs was of the form $X_1 \wedge X_4 \vee X_8 \wedge X_{11}$ and is equal to $L_3 \vee L_2$. One might argue to which extent such a combination of trees should actually be counted as a false positive, a question which is further elaborated in Section B.2 of the web supplement and in the Discussion section.

The remaining two scenarios are way more complex due to the higher order interaction terms involved. In Scenario 5 the power to detect any of the four trees was very large, with only slightly smaller power for the four-way interaction. The robust g-prior had only a rather small advantage compared with Jeffreys' prior both in terms of power (overall 97% against 96%) and in terms of type I error (FDR of 4% against 6%). For both priors the majority of false positive results were connected to detecting subtrees of true trees and in all simulation runs there were only 2 wrongly detected leaves for Jeffreys' prior and 5 wrongly detected leaves for the robust g-prior.

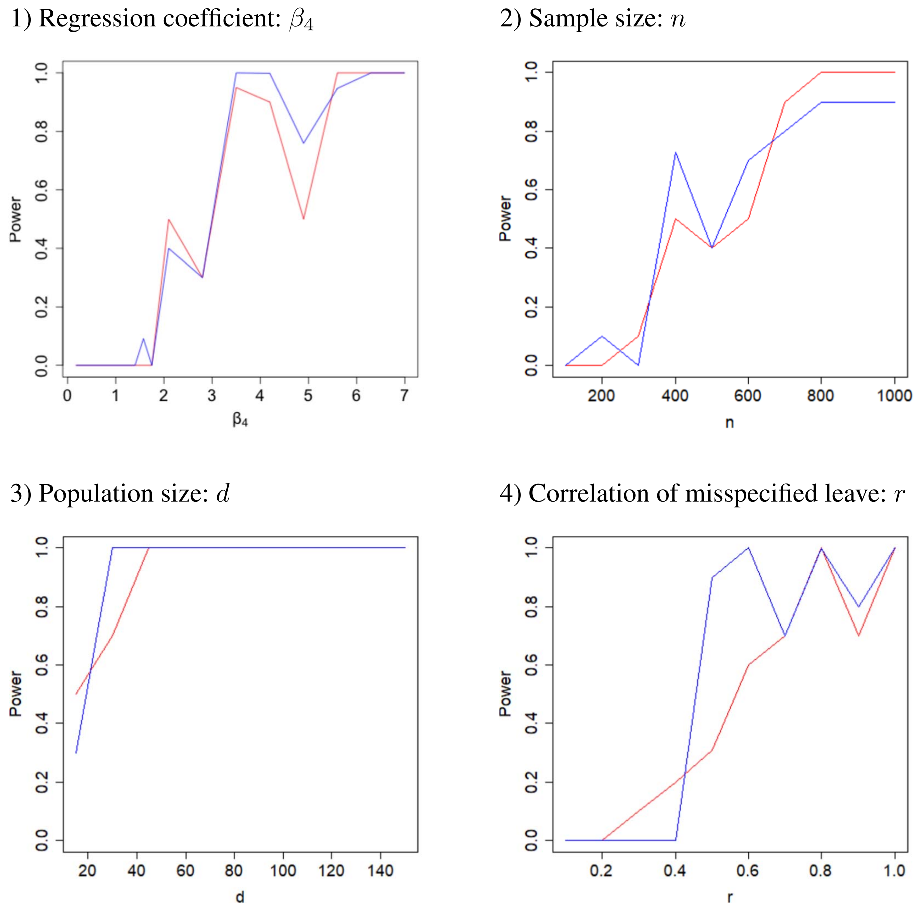
For the last scenario we again observed large power for all true trees up to order three. For the final two expressions L_7 and L_8 of order for the results became slightly more ambiguous with power estimated to 0.32 and 0.21, respectively, for Jeffreys' prior and 0.45 and 0.16 for the robust g-prior. However, among the false positive detections we very often found the expressions $X_{11} \wedge X_{13}$, $X_{19} \wedge X_{50}$ as well as $X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$. In fact in 72 simulation runs for Jeffreys' prior and 69 simulation runs for the robust g-prior all of these three expressions were detected. According to the logic equivalence

$$L_8 = X_{11} \wedge X_{13} + X_{19} \wedge X_{50} - X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$$

one might actually consider these findings as true positives. The numbers in parentheses in Table 2 were based on taking such similarities into account, resulting in much higher power. Among the remaining false positive detections more than two thirds were subtrees of true trees or trees with misspecified logical operators but consisting of leaves corresponding to a true tree. Thus again the vast majority of false detections points towards true epistatic effects where the exact logic expression was not identified. Interestingly like in Scenario 5 GMJMCMC with the robust g-prior detected again a larger number of wrong leaves than with Jeffreys' prior.

Sensitivity analysis

We performed sensitivity analysis for the power to detect trees in Scenario 5 based on $\tilde{P}(L_j|Y) > 0.5$ for $j \in \{1, \ldots, 4\}$. Figure 1 presents the results for the four-way interaction L_4. Results for the trees with fewer leaves are provided in Figures S1–S3 of the web supplement. Specifically we wanted to study how the power is effected by the following factors:



1) Regression coefficient: β_4

2) Sample size: n

3) Population size: d

4) Correlation of misspecified leave: r

Figure 1: Dependence of power to detect L_4 for Jeffreys' prior (red) and the robust g-prior (blue) when varying different parameters as specified above each plot. Parameters which are not explicitly varied are kept fixed at the levels from the original Scenario 5, except for the first plot where all four coefficients $\beta_1 \ldots, \beta_4$ are simultaneously varied by multiplying with the same factor.

1. A change in the corresponding **coefficients** β_j, where all coefficients are varied simultaneously by multiplying them with a factor $K \in \{0.05, 0.1, 0.2, \ldots, 1\}$ and all other parameters are kept constant.

2. A change in the **sample size** n, where the sample size n is varied from 100 to 1000 and all other parameters are kept constant.

3. A change in the **population size** d, where the population size d is varied from 15 to 150 and all other parameters are kept constant.

4. A **misspecified leave** within L_4, where the misspecified leave is substituted by a correlated leave with the correlation r varying from 0.1 to 1.

In cases 2, 3 and 4 the relevant parameters were increased uniformly in 10 steps, in all cases k_{max} was set to 20. For computational reasons the sensitivity analysis was performed only using 10 simulation runs for each parameter value, both for Jeffreys' prior and for the robust g-prior. This number of repetitions is not sufficient to give high resolution estimates of the power but it is enough to illustrate the general dependence on each of the considered parameters.

The first two plots of Figure 1 illustrate how the power to detect L_4 changes when either the regression coefficient β_4 or the sample size n are varied. With a sample size of 1000 the power seems to deteriorate only for effect sizes smaller than 4, whereas for the large effect size of Scenario 5 a sample size of $n = 600$ still seems to provide reasonable power to detect L_4. The first plots of Figures S1–S3 of the web supplement show that for the lower order trees a sample size of $n = 1000$ is sufficient to obtain reasonable power for much smaller effect sizes. Notably the three-way interaction L_3 can be detected with large power already for $\beta_3 = 1$ which is of the same order as the standard deviation of the error term.

To reach sufficient power to detect four-way interactions with smaller regression coefficients one would have to increase the sample size. For many statistical models there is the notion that when decreasing the effect size by a factor $1/K$ one would roughly have to increase the sample size by a factor K^2 to end up with the same power. Figure S4 from the web supplement indicates that this relationship also holds for the logic regression approach and together with the results from the first plot of Figure 1 one can induce that a sample size of $n > 10000$ is needed to have sufficient power to detect four-way interactions with regression coefficients which are of the order of the error standard deviation.

The third plot of Figure 1 is concerned with the influence of the population size d from the GMJMCMC algorithm on the power to detect L_4. Corresponding plots for the trees of lower size, for which the power is almost always equal to one, are provided in the web-supplement. As one can see for both priors power to detect L_4 grows gradually from 0 to 1 when d changes from 15 to 45. For values of $d > 30$ the power remains stable at 1. This illustrates the statement of Theorem 1, according to which one requires $d - d_1 \geq k_{max}$ to have an irreducible algorithm in the restricted space of logic regression models. In these simulations we have $k_{max} = 20$ and $d_1 = 10$. Hence according to Theorem 1 a population size $d \geq 30$ is sufficient for asymptotic irreducibility of the GMJMCMC algorithm. For $d - d_1 < k_{max}$ irreducibility is no longer guaranteed and hence we cannot expect the approximations of the model posteriors to be precise in all cases, specifically when the model size of the data generating model is larger than $d - d_1$.

The final plot of Figure 1 considers the effect of misspecification of one leave. This setting is motivated by genetic association studies, where it often happens that not a causal SNP (single nucleotide polymorphism) itself is genotyped but rather a strongly correlated tag SNP. As long as the correlation of the misspecified leave to the original leave is larger than 0.5 there appears to be no dramatic loss of power which indicates that a certain amount of model misspecification can be tolerated by our method.

Phenotype	Chr	Marker expression	$\tilde{P}(L \mid Y)$	Signif.
Blue Light	4	X44606688	0.767	***
Blue Light	5	X44607250	0.335	**
Blue Light	2	X21607656	0.309	**
Blue Light	$4\wedge2$	$\text{X44606688}\wedge\text{X44606810}$	0.203	*
Red Light	2	MSAT2.36	0.441	**
Red Light	2	PHYB	0.353	**
Red Light	$2\wedge1$	$\text{PHYB}^c\wedge\text{X44606541}$	0.112	*
Red Light	2	X21607013	0.092	*
Far Red Light	4	MSAT4.37	0.302	**
Far Red Light	4	NGA1107	0.302	**
White Light	5	X44606159	0.632	***
White Light	1	X21607165	0.427	**

Table 3: Potential additive and epistatic QTL for hypocytol length under different light conditions for Arabidopsis thaliana. Recombinant inbreed line data set taken from Balasubramanian et al. (2009). The last column shows the level of confidence with *** corresponding to $\tilde{P}(L \mid Y) > 0.5$, ** to $\tilde{P}(L \mid Y) > 0.3$ and $*$ to $\tilde{P}(L \mid Y) > 0.05$.

3.2 Analysis of Arabidopsis data

According to our simulation results there is no large difference in the performance of GMJMCMC between using Jeffreys' prior or the robust g-prior. On the other hand the clear computational advantage of Jeffreys' prior seems to justify to omit the robust g-prior for analyzing real data. Hence in this section we only use Jeffreys' prior for GMJMCMC. Furthermore we used $k_{max} = 15$ and $d = 25$ which allows for way more complex models than we would expect to see.

Balasubramanian et al. (2009) mapped several different quantitative traits in *Arabidopsis thaliana* using an advanced intercross-recombinant inbred line (RIL). Their data is publicly available as supporting information of their PLOS ONE article (Balasubramanian et al., 2009) which also gives all the details of the breeding scheme and the measurement of the different traits. We consider here only the hypocytol length in *mm* under different light conditions.[3] Genotype data is available for 220 markers distributed over the 5 chromosomes of Arabidopsis thaliana with 61, 39, 43, 31 and 46 markers, respectively. Balasubramanian et al. (2009) had genotyped 224 markers but we dismissed 4 markers which had identical genotypes with other markers. The amount of missing genotype data is relatively small with a genotype rate of 93.9% and most importantly the data contains only homozygotes (AA:49.6% vs. BB:50.4%). This means that the RIL population contains no heterozygote markers and logic regression can be directly applied using the genotype data as Boolean variables. Missing data were imputed using the R package R-QTL (http://www.rqtl.org/).

The imputed data was then analyzed with our algorithm GMJMCMC to detect potential epistatic effects and the results are summarized in Table 3. Under blue light Bala-

[3] Data obtained from the second to fifth column of the file http://journals.plos.org/plosone/article/file?type=supplementary&id=info:doi/10.1371/journal.pone.0004318.s002.

subramanian et al. (2009) reported 4 potential QTL's, the strongest one on chromosome 4 in the regions of marker X44606688 and three further fairly weak QTL on chromosomes 2, 3 and 5. Our analysis based on logic regression confirmed X44606688 and also detected those markers on chromosomes 2 and 5, though with a posterior probability slightly below 0.5. There was also some indication of a two-way interaction between the strong QTL on chromosome 4 and the QTL on chromosome 2.

Under red light the original interval mapping analysis reported the region of MSAT2.36 as a strong QTL on chromosome 2 and x44607889 as a weaker QTL on chromosome 1. Our logic regression analysis distributes the marker posterior weights on three different markers on chromosome 2 which are all in the neighborhood of MSAT2.36. Additionally there is some rather small posterior probability for an epistatic effect between this region and a marker on chromosome 1 which is rather close to x44607889. Finally both for Far Red Light and for White Light our analysis essentially yielded the same results as the interval mapping analysis, when observing that under the first condition the posterior probability was again almost equally distributed between the neighboring markers MSAT4.37 and NGA1107. In summary the sample size in this data set might be slightly too small to detect epistatic effects, although under the first two light conditions there was at least some indication for a two-way interaction.

We have analyzed a second data set concerned with QTL mapping for Drosophila where we compare logic regression with a more traditional approach to modeling epistasis. Further details and results are presented in Section D of the web supplement.

4 Discussion

We have introduced GMJMCMC as a novel algorithm to perform Bayesian logic regression and compared it with the two existing methods MCLR (Kooperberg and Ruczinski, 2005) and FBLR (Fritsch, 2006). The main advantage of GMJMCMC is that it is designed to identify more complex logic expressions than its predecessors. Our approach differs both in terms of prior assumptions and in algorithmic details. Concerning the prior of regression coefficients we compared the simple Jeffreys' prior with the robust g-prior. Jeffreys' prior in combination with the Laplace approximation coincides with a BIC-like approximation of the marginal likelihood, which was also used by MCLR. The robust g-prior has some very appealing theoretical properties for the linear model. However, in our simulation study it gave only slightly better results than Jeffreys' prior for the linear model and in case of logistic regression actually performed worse in terms of power to detect the trees of the data generating logic regression model. With respect to the model topology we chose a prior which is rather similar to the one suggested by Fritsch (2006) for FBLR, but instead of using a truncated geometric prior for the number of leaves of a tree we suggest a prior which penalizes the complexity of a tree indirectly proportionally to the total number of trees of a given size. The motivation behind this prior is to control the number of false positive detections of trees in a similar way to how the Bonferroni correction works in multiple testing.

GMJMCMC has the capacity to explore a much larger model search space than MCLR and FBLR because it manages to efficiently resolve the issue of not getting stuck in local extrema, a problem that both MCLR and FBLR have in common. In logic regression the marginal posterior probability function is typically multi-modal in the space of models, with a large number of extrema which are often rather sparsely located. Additionally, the search space for logic regression is extremely large, where even computing the total number of models is a sophisticated task. As discussed in more detail in Hubin and Storvik (2018), in such a setting simple MCMC algorithms often get stuck in local extrema, which significantly slows down their performance and convergence might only be reached after run times which are infeasible in practice.

The success of GMJMCMC relies upon resolving the local extrema issue, which is mainly achieved by combining the following two ideas. First, when iterating through a fixed search space S, GMJMCMC utilizes the MJMCMC algorithm (Hubin and Storvik, 2018) which was specifically constructed to explore multi-modal regression spaces efficiently. Second, the evolution of the search spaces is governed within the framework of a genetic algorithm where a population consists of a finite number of trees forming the current search space. The population is updated by discarding trees with low estimated marginal posterior probability and generating new trees with a probability depending on the approximations of marginal inclusion probabilities from the current search space. The aim of the genetic algorithm is to converge towards a population which includes the most important trees. Finally the performance of GMJMCMC is additionally boosted by running it in parallel with different starting points. Irreducibility of the proposals both for search spaces and for models within the search spaces guarantees that asymptotically the whole model space will be explored by GMJMCMC and global extrema will at some point be reached under some weak regularity conditions. Clearly the genetic algorithm used to update search spaces results in a Markov chain of model spaces.

One important question in the context of logic regression is concerned with how to define true positive and false positive detections in simulations. We adapted a rather strict point of view which might be called an 'exact tree approach': Only those detected logic expressions which were logically equivalent with trees from the data generating model were counted as true positives. While this seems to be a natural definition there are certain pitfalls and ambiguities that occur in logic regressions which might speak against this strict definition. Apart from the more obvious logic equivalences according to Boolean algebra, for example due to De Morgan's laws or the distributive law, there can be slightly more hidden logic identities in logic regression. For example the expressions $(X_1 \vee X_2) - X_1$ and $X_2 - (X_1 \wedge X_2)$ give identical models. We have seen a less trivial example including four-way interactions in Scenario 6 of our simulation study, where the data generating tree L_8 is equivalent to the expression $X_{11} \wedge X_{13} + X_{19} \wedge X_{50} - X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$ consisting of three trees. Furthermore, different logic expressions can be highly correlated even when they are not exactly identical.

Especially the results from the most complex Scenario 6 impose the question whether the exact tree approach is slightly too strict to define false positives. Subtrees of true trees give valuable information even if they are not describing the exact interaction. Often combinations of several subtrees and trees with misspecified logical operators can give expressions which are very close to the correct interaction term. For Scenario 6 we reported two possible summaries of the simulation results, one based strictly on the exact tree approach and the other one counting simultaneous detections of $X_{11} \wedge X_{13}$, $X_{19} \wedge X_{50}$ and $X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$ also as true positives. This was slightly ad hoc and we believe that good reporting of logic regression results is an area which needs further research. The output of MCLR takes a step in that direction, where only the leaves of trees are reported and if a tree has been detected then also all its subtrees are reported. However, in our opinion MCLR throws away too much information. We believe that several different layers of reporting might be more desirable, for example the exact tree approach, the MCLR approach and then something in between which does not reduce trees completely to their set of leaves. We have started to think more systematically in that direction and leave this topic open for another publication.

This paper has had a focus on model selection and selection of features of interest. The method is however directly applicable to prediction as well. One can approximate the posterior probability of some parameter/variable Δ via model averaging by

$$\tilde{P}(\Delta \mid Y) = \sum_{M \in \Omega^*} P(\Delta \mid M, Y)\tilde{P}(M \mid Y),$$

where Δ might be for example the predictor of unobserved data based on a specific set of covariates. Given estimates of posterior model probabilities, other prediction procedures such as the median probability model (Barbieri et al., 2004) or the posterior weighted median (Clarke et al., 2013) can also easily be applied.

Supplementary Material

Supplementary Material for: A novel algorithmic approach to Bayesian Logic Regression (DOI: 10.1214/18-BA1141SUPP; .pdf). https://github.com/aliaksah/EMJMCMC2016/tree/master/supplementaries/Bayesian%20Logic%20Regression.

References

Balasubramanian, S., Schwartz, C., Singh, A., Warthmann, N., Kim, M., Maloof, J., Loudet, O., Trainer, G., Dabi, T., Borevitz, J., Chory, J., and Weigel, D. (2009). "QTL mapping in new Arabidopsis thaliana advanced intercross-recombinant inbred lines." *PLoS One*, 4(2). 265, 283

Barber, R. F., Drton, M., and Tan, K. M. (2016). *Laplace Approximation in High-Dimensional Bayesian Regression*, 15–36. Cham: Springer International Publishing. MR3616262. 268

Barbieri, M. M., Berger, J. O., et al. (2004). "Optimal predictive model selection." *The annals of statistics*, 32(3): 870–897. MR2065192. doi: https://doi.org/10.1214/009053604000000238. 270, 276, 286

Bayarri, M. J., Berger, J. O., Forte, A., García-Donato, G., et al. (2012). "Criteria for Bayesian model choice with application to variable selection." *The Annals of statis-*

tics, 40(3): 1550–1577. MR3015035. doi: https://doi.org/10.1214/12-AOS1013. 267, 268, 269

Bogdan, M., Ghosh, J. K., and Tokdar, S. T. (2008). "A comparison of the Simes-Benjamini-Hochberg procedure with some Bayesian rules for multiple testing." *IMS Collections, **Vol. 1**, Beyond Parametrics in Interdisciplinary Research: Fetschrift in Honor of Professor Pranab K. Sen, edited by N. Balakrishnan, Edsel Peña and Mervyn J. Silvapulle*, 211–230. MR2462208. doi: https://doi.org/10.1214/193940307000000158. 267

Chib, S. (1995). "Marginal likelihood from the Gibbs output." *Journal of the American Statistical Association*, 90(432): 1313–1321. MR1379473. 274

Chib, S. and Jeliazkov, I. (2001). "Marginal likelihood from the Metropolis–Hastings output." *Journal of the American Statistical Association*, 96(453): 270–281. MR1952737. doi: https://doi.org/10.1198/016214501750332848. 274

Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. MR2431297. doi: https://doi.org/10.1017/CBO9780511790485. 268

Clarke, J. L., Clarke, B., Yu, C.-W., et al. (2013). "Prediction in M-complete Problems with Limited Sample Size." *Bayesian Analysis*, 8(3): 647–690. MR3102229. doi: https://doi.org/10.1214/13-BA826. 286

Clyde, M. A., Ghosh, J., and Littman, M. L. (2011). "Bayesian adaptive sampling for variable selection and model averaging." *Journal of Computational and Graphical Statistics*, 20(1): 80–101. MR2816539. doi: https://doi.org/10.1198/jcgs.2010.09049. 266

Fritsch, A. (2006). "A Full Bayesian Version of Logic regression for SNP Data." Ph.D. thesis, Diploma Thesis. 263, 264, 276, 284

Fritsch, A. and Ickstadt, K. (2007). "Comparing Logic Regression Based Methods for Identifying SNP Interactions." *Springer Berlin / Heidelberg, Lecture Notes in Computer Science*, 4414: 90–103. MR2291281. doi: https://doi.org/10.1080/09332480.2006.10722798. 263

Frommlet, F., Ljubic, I., Arnardottir, H., and 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): Article 2. MR2944873. doi: https://doi.org/10.1515/1544-6115.1793. 269

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC. MR3235677. 268

Hubin, A. and Storvik, G. (2018). "Mode jumping MCMC for Bayesian variable selection in GLMM ." *Computational Statistics and Data Analysis*. MR3820324. doi: https://doi.org/10.1016/j.csda.2018.05.020. 264, 266, 270, 271, 285

Hubin, A., Storvik, G., and Frommlet, F. (2018a). "Deep Bayesian regression models." *arXiv preprint arXiv:1806.02160*. Submitted for publication. 271

Hubin, A., Storvik, G., and Frommlet, F. (2018b). "Supplementary Material for: A novel algorithmic approach to Bayesian Logic Regression." *Bayesian Analysis*. doi: https://doi.org/10.1214/18-BA1141SUPP. 265

Janes, H., Pepe, M., Kooperberg, C., and Newcomb, P. (2005). "Identifying target populations for screening or not screening using logic regression." *Statistics in Medicine*, 24: 1321–1338. MR2134561. doi: https://doi.org/10.1002/sim.2021. 263

Jeffreys, H. (1946). "An invariant form for the prior probability in estimation problems." *Proceedings of the Royal Society of London. Series A*, 186(1007): 453–461. MR0017504. doi: https://doi.org/10.1098/rspa.1946.0056. 268

Jeffreys, H. (1961). *Theory of probability*. Oxford University Press, London. 268

Keles, S., van der Laan, M., and Vulpe, C. (2004). "Regulatory motif finding by logic regression." *Bioinformatics*, 20: 2799–2811. 263

Kooperberg, C. and Ruczinski, I. (2005). "Identifying Interacting SNPs Using Monte Carlo Logic Regression." *Genetic Epidemiology*, 28: 157–170. 263, 264, 265, 276, 284

Li, Y. and Clyde, M. A. (2018). "Mixtures of g-priors in generalized linear models." *Journal of the American Statistical Association*, (just-accepted). 264, 267, 268, 269

Malina, M., Ickstadt, K., Schwender, H., Posch, M., and Bogdan, M. (2014). "Detection of epistatic effects with logic regression and a classical linear regression model." *Statistical Applications in Genetics and Molecular Biology*, 13(1): 83–104. MR3159119. doi: https://doi.org/10.1515/sagmb-2013-0028. 264, 265

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models. 2nd Edition*. Chapman and Hall, London. MR3223057. doi: https://doi.org/10.1007/978-1-4899-3242-6. 265

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). "Bayesian model averaging for linear regression models." *Journal of the American Statistical Association*, 92(437): 179–191. MR1436107. doi: https://doi.org/10.2307/2291462. 268

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). "Logic regression." *Journal of Computational and Graphical Statistics*, 12(3): 474–511. MR2002632. doi: https://doi.org/10.1198/1061860032238. 263, 264, 265

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2004). "Exploring Interactions in High-Dimensional Genomic Data: An Overview of Logic Regression, with Applications." *Journal of Multivariate Analysis*, 90: 178–195. MR2086341. doi: https://doi.org/10.1016/j.jmva.2004.02.010. 263

Schwarz, G. (1978). "Estimating the dimension of a model." *The Annals of Statistics*, 6: 461–464. MR0468014. 268

Schwender, H. and Ickstadt, K. (2008). "Identification of SNP interactions using logic regression." *Biostatistics*, 9: 187–198. 263

Schwender, H. and Ruczinski, I. (2010). "Logic Regression and Its Extensions." *Advances in Genetics*, 72: 25–45. 263

Scott, J. G. and Berger, J. O. (2008). "Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem." *Annalls of Statistics*, 38(5): 2587–2619. MR2722450. doi: `https://doi.org/10.1214/10-AOS792`. 266

Tierney, L. and Kadane, J. B. (1986). "Accurate Approximations for Posterior Moments and Marginal Densities." *Journal of the American statistical association*, 81(393): 82–86. MR0830567. 268

Tjelmeland, H. and Hegstad, B. K. (2001). "Mode jumping proposals in MCMC." *Scandinavian Journal of Statistics*, 28(1): 205–223. MR1844357. doi: `https://doi.org/10.1111/1467-9469.00232`. 264, 270

Wakefield, J. (2007). "A Bayesian measure of the probability of false discovery in genetic epidemiology studies." *The American Journal of Human Genetics*, 81(2): 208–227. 270

Wang, Y. H. (1993). "On the number of successes in independent trials." *Statistica Sinica*, 295–312. MR1243388. 267

Acknowledgments

The first two authors gratefully acknowledge the financial support of the *CELS project at the University of Oslo*, http://www.mn.uio.no/math/english/research/groups/cels/index.html.

Invited Discussion

Ingo Ruczinski[*], Charles Kooperberg[†], and Michael LeBlanc[‡]

The logic regression project started some 20 years ago as part of the doctoral dissertation of Ingo Ruczinski. The initial motivation was indeed to develop and implement a method specifically to detect epistatic interactions in genetic studies measuring single nucleotide polymorphisms (SNPs). However, it soon occurred to us that logic regression could also be useful for many other data types and settings, particularly in medical studies where often many binary data are collected. In its final version, the dissertation contained two applications of logic regression: 1) a genetic association study using SNP data from the Genetic Analysis Workshop (GAW), and 2) a medical study to infer which brain regions affected by infarcts influence the cognitive state of patients. Logic regression was quickly adopted by the community after the first open source software release, with many applications analyzing data with predictors other than SNPs. Moreover, various groups also developed new methodology extending the original logic regression framework (including Bayesian versions of logic regression), which is particularly rewarding for its creators. The algorithm introduced here by Hubin, Storvik and Frommlet (HSF hereafter) is an improvement for Bayesian model selection in the space of logic regression models. We would like to congratulate the authors and say "thank you" for their contribution to the field, and would like to offer a few additional thoughts and perspectives.

We completely agree with the authors that the search algorithm in the original Bayesian version of logic regression (Kooperberg and Ruczinski, 2005) had room for improvement. We also experienced that the Markov Chain Monte Carlo (MCMC) algorithm can get stuck in a particular part of the model space, and we are not surprised that the method and implementation put forth by HSF based on "mode jumping", the Genetically modified Mode Jumping Markov Chain Monte Carlo (GMJMCMC) algorithm, performs better in this regard. Our implementation of MCMC was in essence a modification of the simulated annealing algorithm we developed to maximize the (frequentist) likelihood function. Great care in setting the parameters for the annealing scheme is required, since the algorithm for the search of the global optimum can easily get trapped in local extrema as well. The main contribution by HSF is a greatly improved search strategy to explore an extremely "ragged" likelihood landscape. We think that a non-Bayesian equivalent of GMJMCMC can also be used for maximizing such a likelihood, similar to simulated annealing. We discuss this in more detail further below.

In the abstract, HSF motivate the GMJMCMC algorithm by stating that logic regression *"has been mainly used to model epistatic effects in genetic association studies, which is very appealing due to the intuitive interpretation of logic expressions to describe the interaction between genetic variations. Nevertheless logic regression has (partly due*

[*] Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, ingo@jhu.edu
[†] Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, clk@fhcrc.org
[‡] Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, mleblanc@fhcrc.org

to computational challenges) remained less well known than other approaches to epistatic association mapping." We believe the strong focus on epistasis in the abstract is not necessary, and it might actually deter practitioners who do not work with genetic data. The Boolean combinations of binary variables used in logic regression indeed lend themselves to genetic analyses, be it as recorded mutations, markers in breeding studies, or bi-allelic SNPs recorded as two binary variables in dominant and recessive coding. In our initial efforts, we used data from genetic candidate studies, which at the time recorded between a few dozen and a few hundred markers. The early 2000s, when the logic regression methodology was published and the software released (Kooperberg et al., 2001; Ruczinski et al., 2003), saw the emergence of commercial SNP genotyping arrays (Bumgarner, 2013), typing tens of thousands of markers at a time. Trying to infer higher-order epistatic interactions (with logic regression or otherwise) in data generated from even these first generation arrays is futile, given the size of the search space to explore these interactions and the ensuing multiple comparisons problem. Even when only pairwise interactions are considered, 100,000 markers yield 4.5 billion possible SNP-SNP interactions! We note that this is also reflected in the data HSF present: the simulation study comprises of 50 markers, the Arabidopsis data have 220 markers, and the Drosophila data in the supplementary materials have 45 markers. Logic regression and similar algorithms are suitable for data of this dimensionality, but certainly not for modern genome scans with millions of markers typed or sequenced. But this is not a limitation of the GMJMCMC algorithm – logic regression is still being used in many settings, particularly in the medical literature (see for example the Introduction of Tietz et al. (2019) for a number of recent examples), and these applications will also benefit from advancements such as GMJMCMC.

The GMJMCMC algorithm puts a prior on model size (equations (3) and (6) in HSF) and therefore depends on a definition of model complexity. This is not a trivial issue, since the predictors in logic regression models are Boolean combinations of binary covariates, and the number of parameters in the model is the same regardless how complex the Boolean terms are. In addition, equivalent Boolean terms can have different expressions and therefore also a different number of binary predictors, for example the Boolean expressions $X_1 \wedge (X_2 \vee X_3)$ and $(X_1 \wedge X_2) \vee (X_1 \wedge X_3)$. HSF write the model prior (equation (3)) as a product of terms, introduced to give smaller probabilities to more complex trees (subject to the total number of trees not exceeding the number of trees allowed). These terms are chosen so the multiplicative contribution of a logic tree of a given size is inversely proportional to the number of possible trees having the same number of leaves. To estimate this number and to deal with the thorny issue of tree complexity, HSF propose to ignore Boolean terms that include the same binary covariate multiple times. This is a very reasonable proposal we believe, and allows for a straightforward estimate of the number of possible trees to be incorporated in the model prior. These terms are also chosen so larger models are penalized more (i.e. the prior probability of a model is always larger than the prior probability of any other model it is nested in). Using Jeffreys' prior for the regression parameters, the authors highlight that the model posterior probabilities can be calculated using the Laplace approximation, and discuss the relationship with the Bayesian Information Criterion. We would like to add that this approach also allows for an alternative model selection strategy in the original logic regression approach as introduced in Ruczinski et al.

(2003), where we try to obtain a model that best explains the observed data. Logic regression uses simulated annealing to find optimal models (according to the objective function used) for a variety of possible model sizes, and then employs cross-validation or permutation tests to select the suitable size for the model. Jeffreys' prior proposed by HSF for generalized linear models leads to an objective function that in essence corresponds to a penalized likelihood. Thus, when adopting the framework of HSF, one could dramatically save on CPU time by only running one annealing chain using the above described posterior probability terms as the objective function, without the need for cross-validation or permutation tests for model size selection. We note that our software allows for the specification of one's own objective function, as described in the software manual (`https://CRAN.R-project.org/package=LogicReg`).

In addition to numerous options to define complexity for logic regression models, we concur with the authors that it is also not clear-cut how the performance of algorithms to detect Boolean interactions should be evaluated (HSF, p. 23). In their simulation study, HSF classify a tree as detected if the marginal inclusion probability is estimated to be at least 50%, and report various metrics (the power to detect a particular tree from the data generating model, the average power over all true trees, the expected number of false positive trees, the false discovery rate of trees, and the total number of wrongly detected leaves) to evaluate the performance of the algorithm. Since *"Jeffreys' prior for model selection has been widely criticized for not being consistent once the true model coincides with the null model"* (HSF, p. 6) the authors also evaluate the GMJMCMC algorithm using the robust g-prior. While no dramatic differences are observed in their simulation based on non-null models, it appears that Jeffreys' prior performs a bit better for the logistic models than the robust g-prior according to the above mentioned metrics. So which one to choose in practice? For the analysis of the Arabidopsis data the authors argue that *"the clear computational advantage of Jeffreys' prior seems to justify to omit the robust g-prior for analyzing real data"* (HSF, p. 21). We suggest a simple two-step procedure for the practitioner that circumvents the need to make this decision. The original logic regression framework offers an easily executed permutation test to determine whether there is any signal in the data (the "null model test" in Ruczinski et al., 2003), which answers the questions whether the assumption of a non-null model is correct. If there is a signal, simply proceed with the GMJMCMC algorithm using Jeffreys' prior.

It was a bit surprising to us that the authors simulated independent binary predictors to assess the performance of the algorithms. In real data we commonly see dependent random variables (e.g. genetic markers can be highly correlated due to linkage disequilibrium), and Bayesian approaches are particularly suitable to address the ensuing model uncertainty as the notion of one "best" model is very questionable due to the correlation structure between the binary variables (the sensitivity analyses presented in HSF Figure 1 and the supplementary materials speak to that to some degree). We also wonder if in the here presented simulation study, especially for the models with large effect sizes, the original logic regression approach as introduced in Ruczinski et al. (2003) might have been a more suitable approach than for example Markov Chain logic regression (MCLR)? Due to the independence of the predictors and the large effect sizes used, it would not be surprising to us if the original annealing based approach would consistently detect the underlying interactions in the simulation study. As mentioned above and discussed by HSF (p. 23), it could also be debated whether any algorithm employed really needs to detect the exact Boolean trees, or simply harnesses the power to explore binary interactions to detect the leaves involved in these Boolean trees (this, we argue, would be the case for example in genetic association studies). Thus, in addition to the total number of wrongly detected leaves (i.e. the specificity), we think the number of correctly detected leaves (i.e. the sensitivity) could also be of interest (and if all leaves are consistently detected due to the large effect sizes, a maybe more challenging simulation could be considered).

To end, we have a few more technical questions for the authors. In many biomedical applications we want to adjust for some predictors (additively) in the model, such as the age and body mass index of subjects, or some principal components to correct for genetic heterogeneity in association studies. Is this easily accommodated in the implementation of the GMJMCMC algorithm, specifying a prior for the corresponding parameters similar to the one for the intercept? For linear models as the ones presented in this manuscript one could of course also regress these variables out, and use the residuals as dependent variables to search for the Boolean expressions of the binary predictors, but that strategy is not possible for generalized linear models with non-linear link functions, such as the logistic model in the simulation study.

Another practical question is how the relevant parameters in the GMJMCMC algorithm should be chosen to obtain dependable results. Clearly, this depends on the problem at hand – more predictors require a longer run, but also a more complex data structure (i.e. higher order interactions) demands a longer search. The authors use the default tuning parameters of the implementation of the underlying Mode Jumping Markov Chain Monte Carlo (MJMCMC) algorithm in all simulations and data analyses presented, but use a range of values for the parameters related to the genetic algorithm of the GMJMCMC algorithm (HSF, Supplementary Table A.1). Do the authors have some general guidance how to choose these? A simulated annealing approach like the one implemented in logic regression – in theory – converges to the optimal solution as long as the chain is aperiodic and irreducible (van Laarhoven and Aarts, 1987). We do not have infinite CPU time in practice however, so rely on some observable metrics to guide the annealing algorithm in logic regression. We implemented the search as a sequence of Metropolis-Hastings algorithms by keeping the temperature fixed for a chain, and then gradually decreasing the temperature to generate a sequence of limiting distributions converging to the optimum. In our implementation we suggest to monitor the acceptance probabilities of the proposed moves in each of the chains: these probabilities have to be essentially 100% early on at higher temperatures when almost every move has to be accepted, and slowly have to converge to 0% as only moves that improve the score should be accepted for very low temperatures (and once the optimum has been reached, the acceptance probability for any move at that temperature should be zero in essence). Further, these types of Metropolis-Hastings based simulated annealing approaches also undergo a phase transition (van Laarhoven and Aarts, 1987), and the variance of the scores visited in a chain should be constant before dropping to zero. Are there similar metrics for the GMJMCMC algorithm that could be considered to guide the selection of the critical parameters such as M_{fin} and T_{max}, and therefore the resulting chain length?

References

Bumgarner, R. (2013). "Overview of DNA microarrays: types, applications, and their future." *Current Protocols in Molecular Biology*, Chapter 22: Unit 22.1. 291

Kooperberg, C. and Ruczinski, I. (2005). "Identifying interacting SNPs using Monte Carlo logic regression." *Genetic Epidemiology*, 28: 157–170. 290

Kooperberg, C., Ruczinski, I., LeBlanc, M., and Hsu, L. (2001). "Sequence analysis using logic regression." *Genetic Epidemiology*, 21 Suppl 1: S626–S631. MR2002632. doi: https://doi.org/10.1198/1061860032238. 291

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). "Logic regression." *Journal of Computational and Graphical Statistics*, 12(3): 475–511. MR2002632. doi: https://doi.org/10.1198/1061860032238. 291, 292

Tietz, T., Selinski, S., Golka, K., Hengstler, J. G., Gripp, S., Ickstadt, K., Ruczinski, I., and Schwender, H. (2019). "Identification of interactions of binary variables associated with survival time using survivalFS." *Archives of Toxicology*, 93: 585–602. 291

van Laarhoven, P. J. and Aarts, E. H. (1987). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers. MR0904050. doi: https://doi.org/10.1007/978-94-015-7744-1. 293

Invited Discussion

Malgorzata Bogdan[*,‡], Blazej Miasojedow[†], and Jonas Wallin[‡]

First of all we would like to congratulate the authors for a very interesting and important article. Logic regression model introduced in Ruczinski (2000); Ruczinski et al. (2003, 2004) is a Generalized Linear Model (GLM) where individual predictors take form of logic expressions dependent on binary explanatory variables. This model arises naturally in the context of identifying epistatic effects in genetic studies. Following Bateson and Mendel (1909), biological epistasis is usually understood as a phenomenon in which "a variant or allele at one locus [. . .] prevents the variant or allele at another locus from manifesting its effect" (see Cordell, 2002), or more generally as a situation when the effect of one allele can only be observed when a second allele is also present. Such epistatic effects can be naturally expressed using logic expressions of the binary variables dependent on the genotypes of genetic markers. While each logic expression can be also represented in the form of the regular linear model, this usually requires many main effects and lower interaction terms. For example, a single "logic interaction" involving four variables $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$ in classical representation takes the form

$$x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 - x_1x_3x_4 - x_2x_3x_4 - x_1x_2x_3 - x_1x_2x_4 + x_1x_2x_3x_4 \ , \quad (0.1)$$

and its natural interpretation is lost in the large number of classical interaction terms. Moreover, the possible causal influence of this "logic interaction" is practically impossible to recover by the regular linear model, where the regression coefficients by each component of (0.1) are estimated separately.

Logic regression seems to be particularly useful for the analysis of outbred populations (like humans), where the number of genetic variants is often much larger than in controlled populations (like e.g. domesticated animals or experimental crosses). Also, it can be applied in a much wider context, like e.g. for the natural representation of the joint influence of general qualitative variables or for the model selection for discrete multicolored graphical models, like the Potts model, in the spirit of (Miasojedow and Rejchel, 2018; Banerjee et al., 2008; Höfling and Tibshirani, 2009; Ravikumar et al., 2010). In case of multicolored graphical models logic expressions can naturally describe dependence between nodes of the graph.

Application of logic regression in real life problems requires solving complex computational and statistical issues, resulting from the large number of possible logic expression models and the possibility of writing a single logic expression in many equivalent tautological forms. For example, the logicFS program of Schwender and Ickstadt (2008) uses simulated annealing (Kirkpatrick et al., 1983) to maximize the likelihood function

[*] Department of Mathematics, University of Wroclaw, Plac Grunwaldzki 2/4, 50-384 Wroclaw, Poland, malgorzata.bogdan@uwr.edu.pl

[†] Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland, B.Miasojedow@mimuw.edu.pl

[‡] Department of Statistics, Lund University, Box 743, 220 07 Lund, Sweden, jonas.wallin@stat.lu.se

over all logic regression models with a given number of leaves. After selecting the "best" model, each logic expression is transformed into a disjunctive normal form (DNF) i.e., OR combination of AND combinations (i.e. prime implicants or logic interactions). The importance of individual interactions is estimated by repeating the whole procedure using many bootstrap samples from the original data and taking into account both the frequency with which a given interaction appears in bootstrap replications as well as its contribution to a total model likelihood.

The disadvantage of the importance measures proposed in Schwender and Ickstadt (2008) is that their values depend on the size of the data set and there exist no natural thresholds which would allow to separate important interactions from false predictors. However, these importance measures can be used for ranking the potential interactions. Concerning model selection strategies, Malina et al. (2014) use logicFS importance measures to build a GLM model including a moderate number of most important interactions. Then the "statistically significant" interactions are selected using the backward elimination procedure based on the multiplicity adjusted p-values. The multiplicity adjustment takes into account that the number of interactions in the space searched by logicFS increases with the interaction complexity.

In Hubin et al. (2020) the issue of identifying important logic interactions is addressed within a Bayesian framework, where the importance of a given logic expression is measured by the sum of posterior probabilities of GLM models which contain this expression as one of predictors. The algorithm in Hubin et al. (2020) calculates the posterior probability for a GLM model M by an approximation to the Bayes rule. The marginal likelihood of the data given M is calculated using the analytical formulas or Laplace approximations for Jeffreys' or robust g-priors. This allows to avoid computational burden of Markov Chain Monte Carlo (MCMC) search over the space of model parameters. Similarly as in Bogdan et al. (2004); Baierl et al. (2006), the prior for each model M depends on its complexity and is selected in such a way that the prior expected numbers of logic expressions of different lengths are approximately the same and do not depend on the number of predictors m. Since the number of complex interactions increases with m at a higher rate than the number of simple interactions, this effectively introduces the additional penalty on the model complexity, which depends on m. The arguments presented in Bogdan et al. (2008b,a) illustrate that this penalty is related to the Bonferroni-type correction for multiplicity, similar to the multiple testing correction used in Malina et al. (2014).

To calculate the posterior probability of a model M the authors use the Bayes rule

$$P(M|Y) = \frac{P(Y|M)P(M)}{\sum_\Omega P(Y|M)P(M)} \ , \tag{0.2}$$

where Ω contains all possible logic regression models. Since it is not possible to visit all these models, the main computational challenge relies on designing a search algorithm which can visit most of the likely models, thus well approximating the denominator of (0.2). Similar problem appears also when fitting regular regression models and in Frommlet et al. (2012a) it was approached by the application of the genetic algorithm supplied with the "local" research in the neighborhood of promising models. In Hubin et al. (2020) the authors propose an iterative algorithm, where in each iteration some new predictors are formed using the specifically designed crossover, mutation and reduction operators on the selected set of logic expressions and then apply the Mode Jumping MCMC (MJMCMC) of Hubin and Storvik (2018) to search the space of GLM models based on these predictors.

While we believe that the article of Hubin et al. (2020) is an interesting and important contribution to the research on the logic regression, we are rather reserved with respect to the proposed algorithm.

In Section 2.3 of Hubin et al. (2020) it is mentioned that a proper MCMC algorithm is not needed if the main purpose is to visit many highly probable models. We agree with the authors and believe that the reversibility of MJMCMC is actually not desired, since it creates unnecessary loops and increases the time of visiting many distinct models. In our opinion a better performance could be obtained by constructing an irreducible and well mixing algorithm of walking over the space of GLM models. In the recent years non-reversible MCMC algorithms received large attention (see e.g. Bouchard-Côté et al., 2018; Bierkens et al., 2019) due to the fact that non-reversible chains are able to explore the state space much faster than the reversible algorithms. For example, let us consider a uniform distribution on the set $0, 1, \dots, N$. In this case the standard reversible MCMC algorithm reduces to a random walk. Hence, after n steps the expected number of explored states is proportional to $\sqrt{n}$ and the number of moves to explore the whole space is proportional to N^2. Instead, we could construct a simple, non-reversible algorithm; i.e. we remember the direction of the previous move and go in the same direction until we hit 0 or N, where the direction is reversed. Then we can explore the whole space in at most $2N$ steps. In case of the problem discussed in Hubin et al. (2020), the construction of the non-reversible MCMC algorithm would be rather simple, since the convergence to the stationary measure is not needed. The only requirement is that the algorithm is irreducible and aperiodic. One solution here would be to define the global and local moves and accept the new state with probability $(\pi(y)/\pi(x))^\alpha$ with some $\alpha > 0$. The parameter α would control the permissible deviations of the posterior with respect to its maximum. Another solution could rely on storing the visited states in a priority queue, with priority proportional to the posterior probability. Then the elements from the queue could be modified by some kernel and placed back to the queue. Such an approach would allow us to explore the space starting from the more promising candidates. Also, this method could be easily parallelized without the need of post processing.

Further, we are concerned with the lack of treatment for tautologies. It seems to us that this might lead to the dilution of the posterior probability among many tautological representations of a given interaction and the loss of power of identification of this interaction. While at the final stage of the algorithm this problem can be solved by post-processing of the output, it is not clear what is the impact of this dilution on elimination of interesting interactions at the earlier stages of the algorithm. It is also important to observe that the number of tautological representations increases with the interaction complexity. Thus, if one merges all tautologies to a single logical expression in a post-processing step, the total prior probability assigned to this unique expression effectively increases with its length and counterbalances the effect of the multiplicity correcting priors suggested by the authors.

The authors estimate the posterior probabilities of different models using (0.2). It is not clear to us why the sum in the denominator of (0.2) contains only $M_{fin} = 10000$ models based on d trees from the final stage of the algorithm. Why not use the information from the earlier stages? Further, in some of the reported simulation examples the authors use $d = 15$. Thus the final search is performed only over $d = 2^{15} = 32768$ models, which could be easily looked at without application of the MCMC algorithm.

Also, a huge random reduction of the final model space leads to substantially different results for different parallel runs of the algorithm. Therefore the authors aggregate results from different runs using a weighting scheme specified in (15) of their paper. In our opinion it seems more reasonable to estimate the posterior probabilities of different models simply by including all models visited in different runs in denominator of (0.2). Also, as we mentioned above, it seems to us that the priority queues would allow for some synchronization between different runs and more efficient search through the model space.

Concerning implementation issues – we observed that the denominator of (0.2) calculated by the currently implemented algorithm includes only the models accepted by MJMCMC. Taking into account that the acceptance rate is usually below 0.1, storing all the models proposed rather than only accepted would give a better estimate of the denominator of (0.2). Further, it seems to us that in the current implementation the denominator of (0.2) increases every time the model is accepted by MJMCMC, without checking if this model already appeared in the sum. However, the detailed analysis of the hidden duplication problems would require a more careful analysis of the code, which is rather difficult due to its structure.

The authors conclude that there is almost no difference between the results when the Jeffreys' or the robust g-prior is used when calculating the model marginal likelihood. However, it seems important to note that the simulations justifying this claim were performed using rather simple GLM models with independent predictors. Actually, it seems that many of the solutions proposed by the authors are specifically designed for this case. For example, consider the case when a given predictor is strongly correlated with other explanatory variables. Then the posterior probability of a "true" model including this predictor will be diluted between "neighboring" models and this predictor might easily miss the threshold for inclusion in the subsequent populations. As noted by the authors, the dilution of posterior probabilities actually occurred in the real data from the experimental recombinant inbred line, where the neighboring markers are rather strongly correlated. We simulated similar spatially correlated data and had a substantial difficulty with identifying a simple two-way logic interaction. Actually, the dilution issue seems to be even more problematic for interactions than for the main effects since the number of correlated interaction terms is substantially higher than the number of respective correlated markers.

Also, one of important features of the algorithm is the initial selection of d_1 important binary variables, which stay as the single trees in the spaces S_i in all iterations of the algorithm. The initial space S_1 is formed by including logic expressions dependent only on these predictors. Other variables can enter the search space only during the mutation, which occurs with a relatively low probability. Thus, the selection of these initial predictors effectively reduces the search space. This approach again seems to be very well suited for the situation when explanatory variables are independent but might lead to missing important predictors otherwise.

Another interesting property, worth studying, is the scaling of the algorithm with respect to the number of explanatory variables m. This number seems to hinder the speed of MJMCMC only at the first step, where d_1 important main effects are selected. However, the magnitude of m probably strongly influences the power of identifying logic interactions. Since the number of possible logic interactions increases rapidly with m, the prior probability for each of them quickly diminishes, which results in decrease of posterior probabilities.

To summarize: it appears to us that the usefulness of the proposed algorithm and the GLM logic regression model is rather restricted to the case when predictors are roughly independent and $n \gg m$. This is however still of a great value in genetic studies, where the raw data are often pre-processed and only relatively few candidate genetic markers are used for building more sophisticated predictive models. If such markers are sufficiently distant, they are almost not correlated. The candidate markers are usually selected using the prior biological knowledge. Since logic interactions usually have strong main effects components, the candidate markers could also be selected using the classical Genome Wide Association Studies (see e.g. Frommlet et al., 2012b or Brzyski et al., 2017).

References

Baierl, A., Bogdan, M., Frommlet, F., and Futschik, A. (2006). "On locating multiple interacting quantitative trait loci in intercross designs." *Genetics*, 173: 1693–1703. 296

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data." *Journal of Machine Learning Research*, 9: 485–516. MR2417243. 295

Bateson, W. and Mendel, G. (1909). *Mendel's principles of heredity*. Cambridge University Press: New York, G.P. Putnam's Sons. 295

Bierkens, J., Fearnhead, P., and Roberts, G. (2019). "The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data." *Annals of Statistics*, 47: 1288–1320. MR3911113. doi: https://doi.org/10.1214/18-AOS1715. 297

Bogdan, M., Ghosh, J., and Doerge, R. W. (2004). "Modifying the Schwarz Bayesian Information Criterion to locate multiple interacting quantitative trait loci." *Genetics*, 167: 989–999. 296

Bogdan, M., Frommlet, F., Biecek, P., Cheng, R., Ghosh, J., and Doerge, R. W. (2008a). "Extending the Modified Bayesian Information Criterion (mBIC) to dense markers and multiple interval mapping." *Biometrics*, 64: 1162–1169. MR2522264. doi: https://doi.org/10.1111/j.1541-0420.2008.00989.x. 296

Bogdan, M., Ghosh, J., and Żak-Szatkowska, M. (2008b). "Selecting explanatory variables with the modified version of Bayesian Information Criterion." *Quality and Reliability Engineering International*, 24: 627–641. 296

Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2018). "The Bouncy Particle Sampler: A Nonreversible Rejection-Free Markov Chain Monte Carlo Method." *Journal of the American Statistical Association*, 113(522): 855–867. MR3832232. doi: https://doi.org/10.1080/01621459.2017.1294075. 297

Brzyski, D., Peterson, C., Sobczyk, P., Candès, E., Bogdan, M., and Sabatti, C. (2017). "Controlling the rate of GWAS false discoveries." *Genetics*, 205: 61–75. 299

Cordell, H. (2002). "Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans." *Human Molecular Genetics*, 11: 2463–2468. 295

Frommlet, F., Ljubic, I., Arnardottir, H., and Bogdan, M. (2012a). "QTL Mapping Using a Memetic Algorithm with Modifications of BIC as Fitness Function." *Statistical Applications in Genetics and Molecular Biology*, 11: Art. 2. MR2944873. doi: https://doi.org/10.1515/1544-6115.1793. 296

Frommlet, F., Ruhaltinger, F., Twarog, P., and Bogdan, M. (2012b). "Modified versions of Bayesian Information Criterion for genome-wide association studies." *Computational Statistics and Data Analysis*, 56(5): 1038–1051. MR2897552. doi: https://doi.org/10.1016/j.csda.2011.05.005. 299

Höfling, H. and Tibshirani, R. (2009). "Estimation of sparse binary pairwise Markov networks using pseudolikelihoods." *Journal of Machine Learning Research*, 10: 883–906. MR2505138. 295

Hubin, A. and Storvik, G. (2018). "Mode jumping MCMC for Bayesian variable selection in GLMM." *Computational Statistics and Data Analysis*, 127: 281–297. MR3820324. doi: https://doi.org/10.1016/j.csda.2018.05.020. 297

Hubin, A., Storvik, G., and Frommlet, F. (2020). "A Novel Algorithmic Approach to Bayesian Logic Regression." *Bayesian Analysis*. 296, 297

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). "Optimization by simulated annealing." *Science*, 220: 671–680. MR0702485. doi: https://doi.org/10.1126/science.220.4598.671. 295

Malina, M., Ickstadt, K., Schwender, H., Posch, M., and Bogdan, M. (2014). "Detection of epistatic effects with logic regression and a classical linear regression model." *Statistical Applications in Genetics and Molecular Biology*, 13: 83–104. MR3159119. doi: https://doi.org/10.1515/sagmb-2013-0028. 296

Miasojedow, B. and Rejchel, W. (2018). "Sparse Estimation in Ising Model via Penalized Monte Carlo Methods." *Journal of Machine Learning Research*, 19(75): 1–26. URL http://jmlr.org/papers/v19/16-554.html MR3899777. 295

Ravikumar, P., Wainwright, M., and Lafferty, J. (2010). "High-dimensional Ising model selection using l_1-regularized logistic regression." *The Annals of Statistics*, 38: 1287–1319. MR2662343. doi: https://doi.org/10.1214/09-AOS691. 295

Ruczinski, I. (2000). “Logic Regression and Statistical Issues Related to the Protein Folding Problem.” Dissertation, Department of Statistics, University of Washington, Seattle, WA. MR2716929. 295

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). “Logic Regression.” *Journal of Computational and Graphical Statistics*, 12: 475–511. MR2002632. doi: https://doi.org/10.1198/1061860032238. 295

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2004). “Exploring Interactions in High-Dimensional Genomic Data: An Overview of Logic Regression, with Applications.” *Journal of Multivariate Analysis*, 90: 178–195. MR2086341. doi: https://doi.org/10.1016/j.jmva.2004.02.010. 295

Schwender, H. and Ickstadt, K. (2008). “Identification of snp interactions using logic regression.” *Biostatistics*, 9: 187–198. 295, 296

Invited Discussion

Holger Schwender[*] and Katja Ickstadt[†]

We congratulate Hubin, Storvik, and Frommlet to this nice paper providing an approach to (Bayesian) logic regression that substantially differs from the existing procedure for fitting logic regression models. Their valuable work contributes a new view on how to generate such models, and, therefore, on how to model and identify interactions, in particular, in genetic association studies. Moreover, their work enhances the use of logic regression due to the algorithmic improvement by their GMJMCMC (Genetically modified Mode Jumping Markov Chain Monte Carlo) procedure.

Since their methods differ from the common approaches to logic regression, we will briefly discuss in the following the original approach to logic regression that to some extent also underlies the other Bayesian logic regression procedures MCLR (Monte Carlo Logic Regression) and FBLR (Fully Bayesian Logic Regression) considered in the paper. We will also present another procedure called GPAS (Genetic Programming for Association Studies) that uses similar operators as GMJMCMC to move through the search space consisting of all possible models. Finally, we will mention how procedures based on logic regression can be employed to not just identify interactions associated with the outcome of interest, but also to rank these interactions by their importance and to guide statements on their relevance and significance.

1 Fitting Logic Regression Models

The starting point of the original logic regression developed by Ruczinski et al. (2003) is a regression model containing one logic expression consisting of one binary/logic variable as predictor. This model is then modified by either adding a new logic expression consisting of one binary variable as predictor to the model or by modifying the logic expression (or later, one of the logic expressions) already in the model by either changing one of the logic variables or logic operators in the logic expression or by adding or removing a logic variable to/from the model. This step is repeated until a score function assessing the fit of the logic regression model converges. E.g., when considering a binary response, the binomial deviance serves as score.

In both logic regression and GMJMCMC, tree-based structures are used to represent the logic expressions in the regression models or populations, respectively, and to modify these logic expressions to move through the space of all possible models. While GMJMCMC borrows ideas from genetic algorithms for the modification of the logic expressions, other, partly related moves directly embedded in the framework of logic trees are employed by logic regression for this purpose. In the nomenclature of logic

[*] Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany, holger.schwender@hhu.de
[†] Faculty of Statistics, TU Dortmund University, Dortmund, Germany, ickstadt@statistik.tu-dortmund.de

regression, replacing a logic variable or operator is called "Alternating a leaf" or "Alternating an operator", respectively. A logic variable can be added to a logic expression by "Splitting a leaf" or "Growing a branch" depending on the position in the logic tree at which this change should be made. Logic variables can also be removed from the models by the countermoves "Delete the leaf" and "Prune a branch" to the adding moves.

Thus, a logic tree can be modified at any level of this tree and not just by adding a new level to the hierarchy of the tree. This is in contrast to CART (Classification And Regression Trees, Breiman et al., 1984), the arguably most well-known tree-based classification and regression method, in which the trees have a hierarchical structure. Because of their non-hierarchical structure, logic trees do not only provide a much more concise representation of logic expressions compared to CART trees, but also a flexible framework to search for a logic regression model that best explains the considered response variable.

Depending on the search algorithm, all choices in all steps of logic regression are either made randomly (when the stochastic search algorithm simulated annealing is employed) or by selecting the modification of the currently considered logic regression model that leads to the largest improvement of the score (when a greedy search is used). As mentioned by Hubin et al., the standard search procedure in logic regression is simulated annealing. This stochastic search algorithm is based on Markov chains, where the probability of accepting a proposed new model is governed by a parameter called temperature that decreases the acceptance probability during the run of simulated annealing. As a result, many models are visited at the beginning of the search, but towards the end of the search it gets more and more unlikely that a modification to the logic regression model gets accepted when the proposed model has a worse score than the current model. In `LogicReg`, the `R` package in which the original logic regression is implemented (Kooperberg and Ruczinski, 2019), also a greedy search is implemented that has the drawback that it is not capable to escape from a local minimum. However, this greedy search works also well when it is put into an ensemble framework such as bagging (Breiman, 1996; see also Section 3 of this discussion).

As mentioned by Hubin et al., in particular in genetic association studies, there usually does not exist the one and only explanation for a response variable such as the disease status, but many competing models that fit the data almost equally well. Hubin et al., therefore, correctly argue that generating just a single best model – as the original logic regression does – ignores the problem of model uncertainty. For this reason, they disregard the original logic regression in their following discussion and consider only Bayesian versions of logic regression enabling Bayesian model averaging by generating and considering many competing logic regression models. It might be, at first sight, a drawback of the original logic regression that it generates just one model. However, as we will discuss in Section 3, it is straightforward to formulate an ensemble framework that can be employed to fit a large number of logic regression models, and thus, to analyze uncertainty in the models.

2 Genetic Programming for Association Studies

Another procedure more closely related to GMJMCMC than logic regression is GPAS (Genetic Programming for Association Studies) proposed by Nunkesser et al. (2007). As implied by its name, GPAS employs genetic programming instead of genetic algorithms as search procedure. As in GMJMCMC, logic expressions are represented by trees and crossover, mutation as well as reduction operators are used to generate in each iteration of the search algorithm a new population of logic expressions that are then evaluated to remove dispensable logic expressions from the population.

In GPAS, each logic expression is generated directly in disjunctive normal form, i.e. as OR-combination of AND-combinations of the logic variables, since the AND-combinations can (at least in a statistical sense) be interpreted as the interactions contained in the logic expression. Besides a crossover operator similar to the one of GMJMCMC, a logic variable that is part of a logic expression in the population of the current iteration of GPAS can be replaced by another logic variable. Also a new logic variable can be added either to an AND-combination or to the OR-combination as a start of a new AND-combination. As in logic regression, countermoves to these two operations are also part of the move set of GPAS. In each iteration of GPAS, two logic expressions are randomly selected to generate a new logic expression by performing a randomly chosen crossover operation on these two expressions. Moreover, five logic expressions – one expression for each of the described mutation operations – are chosen at random to apply a random modification to each of these expressions. In this way, a new population consisting of all logic expressions from the population of the current iteration and six new logic expressions is generated.

While in GMJMCMC a fixed number of MJMCMC (Mode Jumping Markov Chain Monte Carlo) iterations is performed for each population of the genetic algorithm to compute marginal inclusion probabilities and to remove logic expressions with a marginal inclusion probability below some threshold from the population, GPAS employs multi-objective optimization and domination selection in which logic expressions dominated by other logic expressions are removed from the population. Thus, multiple criteria are used to evaluate the performance of a logic expression. If this expression shows a worse value for one of these criteria than another logic expression in the population and not a better value for all the other criteria, then this expression is dominated by the other logic expression, and hence, removed from the population. Since GPAS has been developed for the association analysis of case-control data, the criteria considered in GPAS are the rate of correctly classified observations, the number of correctly predicted controls (in simulations, it has turned out to be beneficial to use the controls in two objectives), and the length of the logic expression.

Because of the similarities in the operators considered in GPAS and GMJMCMC and the differences in the selection processes, it would be valuable to compare these two procedures, which might perhaps even enhance GMJMCMC. Furthermore, as indicated in FBLR, the representation of logic expressions in their disjunctive normal form could also be employed to formulate the full model prior $P(M)$ for GMJMCMC.

3 Assessing the Importance of Identified Interactions

After identifying potentially interesting interactions, i.e. interactions that potentially influence the outcome of interest and are important for a correct prediction of the outcome, with search procedures such as GMJMCMC, GPAS, or logic regression, the importance of these interactions for correctly predicting the outcome should be measured to differ between interactions associated with this outcome and interactions found almost at random. Such a measure can then also be used to rank the interactions by their importance, which is often of particular interest.

Hubin et al. consider for this purpose the marginal inclusion probability $\tilde{P}\big(L_j \mid Y\big)$, i.e. the estimated probability for a logic expression L_j to be included in a model. This probability is, however, determined on the same data on which the models are built, which usually results in a positively biased estimate of the importance of L_j. It would, thus, be preferable to quantify the importance of a particular logic expression or interaction based on new/independent data.

One approach for such a quantification on independent data that was originally developed by Breiman (2001) for measuring the importance of variables in Random Forests is (in a modified version) also used in logicFS (logic Feature Selection; Schwender and Ickstadt, 2008). In logicFS, the original logic regression is applied to B bootstrap samples drawn from the considered data set, resulting in B logic regression models. The logic expressions in each of these models are, afterwards, transformed into disjunctive normal forms to identify the interactions composing these models by the conjunctions/AND-combinations in these disjunctive normal forms. The importance of each of the interactions is then quantified by considering the respective out-of-bag observations, i.e. the observations that do not belong to the respective bootstrap sample and were, therefore, not used in the fitting of the respective model. For the quantification of the importances, the predictive power of each of the B logic regression models is determined, e.g., by the number of correctly out-of-bag observations. Afterwards, the interaction for which the importance should be computed is removed from all the logic regression models, and again, the predictive power of each of the now reduced models is determined. The mean difference between the predictive powers of the models before and after an interaction has been removed from them can then be employed to quantify the improvement in the prediction due to this interaction and thus as a measure of the importance of this interaction for a good prediction of the outcome. Using a permutation test, the importances of the interactions can also be tested (Schwender et al., 2011).

Since logicFS also results in several logic regression models, a comparison of the performance of GMJMCMC and logicFS as well as their importance measures might be valuable.

4 Effect Sizes in Genetic Association Studies

It is much appreciated that Hubin et al. consider the same scenarios as Fritsch (2006). In the other scenarios, they, however, consider effect sizes, in particular, for interactions of higher order from which they admit that they “might seem unrealistically large compared to real applications." In our opinion, this is a large understatement. E.g., in Scenario S.3, odds ratios of $\exp(5) \approx 148$ and $\exp(9) \approx 8103$ are, even for epidemiological risk factors such as smoking, implausibly vast.

In genetic association studies, individual noteworthy SNPs (Single Nucleotide Polymorphisms) seldomly show an odds ratio larger than 1.5 (see, e.g., Golka et al., 2011). Interactions of SNPs might have a substantially higher impact on diseases, but the effect sizes do by far not reach the effect sizes considered in the simulation study. E.g., Selinski et al. (2017) identified a combination of four SNPs showing an odds ratio of 2.59 in a subgroup of urinary bladder cancer patients (where the odds ratios of the individual SNPs ranged between 1.1 and 1.3).

Even though the sensitivity analysis of Hubin et al. gives some insight in the performance of GMJMCMC for a range of effect sizes and sample sizes, the effect sizes in the simulated analysis in our opinion could have been chosen a bit more realistically.

Considering these unrealistically large effect sizes, however, does not really diminish this nice, well-thought-out paper, and in particular, not the proposed method. It is a very welcome contribution to the association analysis of interactions in genetic association studies and opens the field for further research in this direction.

References

Breiman, L. (1996). "Bagging Predictors." *Machine Learning*, 26: 123–140. 303

Breiman, L. (2001). "Random Forests." *Machine Learning*, 45: 5–32. MR3874153. 305

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA. MR0726392. 303

Fritsch, A. (2006). "A Full Bayesian Version of Logic Regression for SNP Data." Master's thesis, Faculty of Statistics, TU Dortmund University. 305

Golka, K., Selinski, S., Lehmann, M. L., Blaszkewicz, M., Marchan, R., Ickstadt, K., Schwender, H., Bolt, H. M., and Hengstler, J. G. (2011). "Genetic variants in urinary bladder cancer: collective power of "wimp SNPs"." *Archives of Toxicology*, 85: 539–554. 306

Kooperberg, C. and Ruczinski, I. (2019). *LogicReg: Logic Regression*. R package version 1.6.1. URL `https://CRAN.R-project.org/package=LogicReg`. 303

Nunkesser, R., Bernholt, T., Schwender, H., Ickstadt, K., and Wegener, I. (2007). "Detecting High-Order Interactions of Single Nucleotide Polymorphisms Using Genetic Programming." *Bioinformatics*, 23: 3280–3288. 304

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). "Logic Regression." *Journal of Computational and Graphical Statistics*, 12: 475–511. MR2002632. doi: `https://doi.org/10.1198/1061860032238`. 302

Schwender, H. and Ickstadt, K. (2008). "Identification of SNP Interactions Using Logic Regression." *Biostatistics*, 9: 187–198. 305

Schwender, H., Ruczinski, I., and Ickstadt, K. (2011). "Testing SNPs and sets of SNPs for importance in association studies." *Biostatistics*, 12: 18–32. 305

Selinski, S., Blaszkewicz, M., Ickstadt, K., Gerullis, H., Otto, T., Roth, E., Volkert, F., Ovsiannikov, D., Moormann, O., Banfi, G., Nyirady, P., Vermeulen, S. H., Garcia-Closas, M., Figueroa, J. D., Johnson, A., Karagas, M. R., Kogevinas, M., Malats, N., Schwenn, M., Silverman, D. T., Koutros, S., Rothman, N., Kiemeney, L. A., Hengstler, J. G., and Golka, K. (2017). "Identification and replication of the interplay of four genetic high-risk variants for urinary bladder cancer." *Carcinogenesis*, 38: 1167–1179. 306

Contributed Discussion

Grégoire Clarté[*] and Christian P. Robert[†]

While logic regression is not to be confused with logistic regression, the distinction may be proved more delicate than stated. For one thing, as the central object of interest is a generalised linear model (or rather a family of such models) based on a vector of binary covariates, it covers the special case of logistic regression. For another thing, it does not very clearly slit from a standard generalised linear model—or generalised analysis of variance model—when all covariates are dummy variables. Culling the number of total covariates (trees) away from the exponential of exponential number of possible covariates defined by logical combinations appears to be a significant component of the approach but this selection of potential (sub-)models remains obscure. If this primary selection is to be data-dependent, there could be a connection with variable length Markov chain models (Bühlmann and Wyner, 1999).

1 Prior Issues

With respect to the prior modeling adopted in the paper, it mostly relies on a rather standard decomposition in variable indicators—to signify whether or not some trees are included in the regression (and hence the model)—. The prior modelling on these indicators is purely a complexity penalisation in that it is only function of the number of active trees, hence not accounting for a possible specificity of some covariates, as for instance when dealing with imbalanced binary covariates (many more 1's than 0's, say).

> "...using Jeffreys' prior for model selection has been widely criticized for not being consistent once the true model coincides with the null model."

A central issue with the prior modelling adopted in the paper is its loose handling of improper priors. It is well-known that the use of improper priors is debatable for model choice settings and hence that they should be best avoided altogether, to wit the Lindley-Jeffreys paradox (Lindley, 1957; DeGroot, 1973, 1982). Let us first recall that Jeffreys (1939) distinguishes between estimation and testing reference priors (Bayarri and Garcia-Donato, 2007; Robert et al., 2009). Not only does the paper adopt the notion of a same, improper, prior on the GLM scale parameter, which is a position advocated in some part of the Bayesian literature (Berger et al., 1998), but it also seems to be using an improper prior on each set of model parameters (further undifferentiated between models). Because the priors operate on different (sub)sets of parameters, we wonder whether or not this jeopardises the later discourse on the posterior probabilities of the different models, since such probabilities are not meaningful from a probabilistic viewpoint. Such a prior construct indeed implies there is no joint distribution and no marginal density. In some cases, it may even be that $p(y|M)$ becomes infinite. Referring

[*] CEREMADE, Univeristé Paris Dauphine, clarte@ceremade.dauphine.fr
[†] CEREMADE, Univeristé Paris Dauphine, xian@ceremade.dauphine.fr

to a “simple Jeffreys”’ prior in this setting is therefore anything but simple as Jeffreys (1939) himself shied away from using improper priors on the parameter of interest.

We therefore find it surprising that this fundamental and well-known difficulty with improper priors in hypothesis testing is not even alluded to in the paper, the above quote being a much milder criticism, core setting thus seems to be flawed. Now, the numerical comparisons run in the paper between Jeffreys’ prior and a regular g-prior exhibit close numerical proximity and we wonder at the reason if the Bayes factor is defined up to an arbitrary constant. Could it be that the culling and selection processes end up having a similar number of covariates and hence ignore the overall impact of the prior? Or is it rather a consequence of recoursing to a Laplace approximation of the marginal likelihood since it completely escapes the problem lack of definition of the said marginal?

2 Algorithmic Aspects

Methinks the proposed strategy is fruitful in a discrete space; we agree with the authors that contrary to Metropolis-Hastings-like methods, it does not involve repeated computations of the same quantity (which can be expensive, especially when involving marginal likelihoods). However, even a limited number of computations of these marginal likelihoods may constitute a real challenge, while the solutions mentioned in the paper are not necessarily the most efficient (Geyer, 1993; Gutmann and Hyvärinen, 2012).

> “. . . we do not need a proper MCMC (an algorithm with convergence towards the target distribution) which is needed if model posterior probabilities are estimated by the relative frequency of how often a model has been visited.”

While we have not read the referred article on MJMCMC in detail, a first comment is that the name itself is somewhat unsuitable, as indeed the algorithm does not sample from a distribution but only explores its surface. There is no proper sampling part in the algorithm, as quantities are computed over Ω^* with integrals of the form

$$\sum_{x\in\Omega^*} f(x)\pi(x)/\mathcal{Z}^*\,,$$

where π the target and $\mathcal{Z}^* = \sum_{x\in\Omega^*}\pi(x)$ is an approximation of the normalizing constant. Finding such a set Ω^* is the main goal of the method developed in the paper.

> “. . . hereby, all states, including all possible models of maximum sized, will eventually be visited.”

In a self-avoiding mode, keeping track of all the previous states visited by the chain ensures that those states will never be visited again. As we are in a discrete setting, this implies that once a mode has been visited the algorithm is constrained to eventually visit another mode, even if the potential between the modes is almost zero. The set Ω^* is then built sequentially, removing states with too low posterior probability to add more interesting states which neighbours have been recently visited.

GMJMCMC starts from this idea to develop a more complex algorithm in which the previous exploration technique is used inside a subset of models which is then updated. We however wonder whether or not the algorithm is not wasting time overall by exploring some parts of an already explored section of the space. More generally, it seems to us that the genetic layer in the algorithm has solely been added to constrain the exploration to smaller spaces, hence are wondering of the efficiency gain brought by this addition.

This method may in the end suffer from several flaws. First, it does not provide the theoretical security of (asymptotic) unbiasedness that is attained with MCMC method. However, it could be of interest to study the variance of such estimators, as Markov chains with poor mixing properties can have huge variance in a multi-modal context. For example, assuming the function f is primarily supported by points outside of Ω^*, it is clear that the estimation is inefficient; however, an MCMC algorithm will similarly be inefficient in the sense that low probability states will also be underexplored, leading to a massive variance estimator. In our opinion, the main issue is to ensure that Ω^* is well-chosen, that is to say, that it contains the right amount of points. Several parameters account for this in the algorithm, choosing these parameters may be a tough calibration problem—especially the choice of the cutoff parameter ρ_{min}, even though it could be chosen as a "quantile of posterior probability" or be adaptive.

References

Bayarri, M. and Garcia-Donato, G. (2007). "Extending conventional priors for testing general hypotheses in linear models." *Biometrika*, 94: 135–152. MR2367828. doi: https://doi.org/10.1093/biomet/asm014. 308

Berger, J., Pericchi, L., and Varshavsky, J. (1998). "Bayes factors and marginal distributions in invariant situations." *Sankhya A*, 60: 307–321. MR1718789. 308

Bühlmann, P. and Wyner, A. J. (1999). "Variable length Markov chains." *The Annals of Statistics*, 27(2): 480–513. MR1714720. doi: https://doi.org/10.1214/aos/1018031204. 308

DeGroot, M. (1973). "Doing what comes naturally: Interpreting a tail area as a posterior probability or as a likelihood ratio." *Journal of the American Statistical Association*, 68: 966–969. MR0362639. 308

DeGroot, M. (1982). "Discussion of Shafer's 'Lindley's paradox'." *Journal of the American Statistical Association*, 378: 337–339. MR0664677. 308

Geyer, C. (1993). "Estimating normalizing constants and reweighting mixtures in Markov chain Monte Carlo." Technical Report 568, School of Statistics, Univ. of Minnesota. MR1341319. doi: https://doi.org/10.2307/1390763. 309

Gutmann, M. U. and Hyvärinen, A. (2012). "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics." *Journal of Machine Learning Research*, 13(1): 307–361. MR2913702. 309

Jeffreys, H. (1939). *Theory of Probability*. Oxford: The Clarendon Press, first edition. MR0187257. 308, 309

Lindley, D. (1957). "A statistical paradox." *Biometrika*, 44: 187–192. MR0087273. doi: https://doi.org/10.1093/biomet/44.1-2.179. 308

Robert, C., Chopin, N., and Rousseau, J. (2009). "Theory of Probability revisited (with discussion)." *Statistical Science*, 24(2): 141–172 and 191–194. MR2655841. doi: https://doi.org/10.1214/09-STS284. 308

Rejoinder

Aliaksandr Hubin[*], Geir Storvik[†], and Florian Frommlet[‡]

1 Introduction

We would like to begin this rejoinder with expressing our sincere gratitude to all of the discussants for their interesting and thought-provoking comments and remarks. We also feel heartily thankful to the editorial board of Bayesian Analysis for giving us the opportunity to publish our paper entitled "A novel algorithmic approach to Bayesian logic regression" (Hubin et al., 2020a) as a discussion article. Logic regression is a tool to model non-linear relationships between binary covariates and some response variable by constructing predictors as Boolean combinations. The number of possible logic expressions grows exponentially with the number of binary variables involved, making the model search significantly harder with the increasing complexity of Boolean combinations. Due to Boolean equivalence, it is in fact almost impossible to specify the full model space a priori even for a relatively small number of covariates.

Our primary goal is to identify those logic expressions which are associated with the response variable. To this end, we want to estimate posterior probabilities of logic expressions within the framework of generalized linear models. The major contributions of our paper are two-fold: Firstly, we have introduced novel model priors for Bayesian logic regression (BLR), which yield good power to detect important logic expressions while controlling the number of false positive discoveries. Secondly, we have introduced a novel genetically modified mode jumping Markov chain Monte Carlo (GMJMCMC) algorithm to efficiently explore the space of logic regression models.

The main idea of GMJMCMC is to embed the mode jumping Markov chain Monte Carlo (MJMCMC) algorithm (Hubin and Storvik, 2018) into the iterative setting of a genetic algorithm. Populations for the genetic algorithm consist of relatively small sets of logic expressions. Any such subset forms a well defined model space which allows to run MJMCMC. The population is then regularly updated in such a way that the algorithm is guaranteed to be irreducible in the model space of all logic regression models. This is required for asymptotic unbiasedness of the estimated posterior probabilities, as we will discuss in more detail below. Although GMJMCMC is not a proper MCMC algorithm (in the sense that its stationary distribution does not coincide with the target distribution of interest), renormalized estimates of the posterior probabilities are readily available.

The discussants have pointed out several interesting extensions and open problems. We have structured the rejoinder according to different topics while trying to address all

[*] Norwegian Computing Center, aliaksandr.hubin@nr.no

[†] Department of Mathematics, University of Oslo, geirs@math.uio.no

[‡] Department of Medical Statistics (CEMSIIS), Medical University of Vienna, florian.frommlet@meduniwien.ac.at

the points raised by the discussants. We also provide several interesting extensions of the model. Finally, we give a brief tutorial on the relevant part of our R-package EMJMCMC `http://aliaksah.github.io/EMJMCMC2016/` dealing with BLR. This should facilitate the practical application of the methodology developed in Hubin et al. (2020a).

2 Applications of Bayesian logic regression

We very much appreciate that Ruczinski et al. (2020) have pointed out important applications of logic regression outside of genetics. Our emphasis on genetic applications was not meant to indicate limitations of the usefulness of logic regression in other areas but rather reflects our own previous research interests. Also, the applications mentioned by Bogdan et al. (2020) in the context of multicolored graphical models sound quite interesting. We are however more sceptical whether logic regression, in whichever form, will ever be applicable directly to association studies of outbred populations, *where the number of genetic variants is much larger than in controlled populations.* For large numbers of binary covariates, already the number of pairwise logic expressions becomes prohibitively large to apply logic regression, both in terms of algorithmic feasibility and in terms of having sufficient power while controlling type I error. Realistic applications of logic regression (with the aim of identifying true logic expressions) will most likely be restricted to applications with a few hundred binary covariates unless technologic advances allow one day to efficiently resolve the $\mathcal{NP}$ hard combinatorial problem of model search. However, one might consider bagging and boosting to obtain scalable versions of logic regression for prediction.

In this rejoinder, we also discuss extensions of Bayesian logic regression, allowing for non-binary predictors and latent Gaussian variables to be included into the model. This could further extend the applications of Bayesian logic regression methodology to such fields as epidemiology, spatio-temporal statistics, environmetrics and econometrics. For example, in Hubin et al. (2020c), a model with latent Gaussian processes (where a subset of predictors are binary) was used for the analysis of DNA methylation. The paper discusses the potential of using logic expressions of the binary predictors as a direction for further research. With the extensions of BLR provided in this rejoinder, it would become feasible to perform logic regression in the settings of Hubin et al. (2020c).

Both Ruczinski et al. (2020) and Bogdan et al. (2020) commented on the lack of correlated regressors in our simulation studies. This was mainly due to the fact that for the sake of comparison we wanted to use the scenarios from Fritsch (2006). For the more complex scenarios, we simply extended these scenarios by adding logic expressions of a higher order. In our sensitivity analysis, though, we considered one scenario with correlations (but only for one *true* leaf), where reasonably good results were obtained when the correlation of a mis-specified leaf r was being varied from 0.1 to 1. However, specifically to respond to the remarks from Bogdan et al. (2020), who hypothesised that our approach would only work under independence, we will provide some additional simulation results, where we consider regressors with different correlation structures.

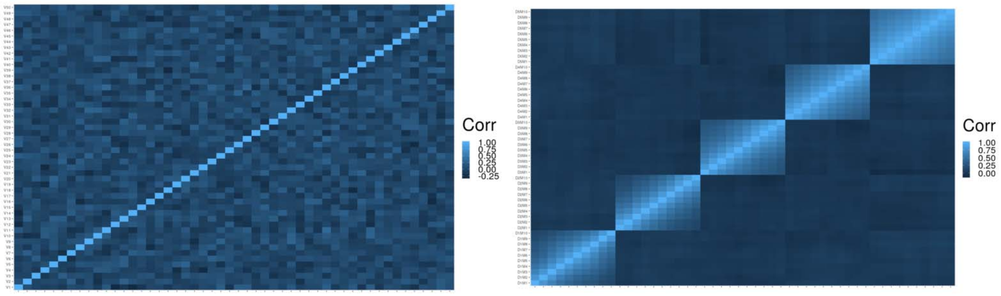


Figure 1: Correlation structure of the simulated covariates with a general correlation structure (left) and from QTL back-cross (right).

2.1 Simulation study with correlated regressors

In this study, we simulate the data using $p = 50$ regressors with two different types of correlation structure: The first one is rather general and uses fairly weak correlations, whereas the second one is typical for QTL mapping and gives very strong correlations. For the first scenario, we consider covariates which are marginally distributed according to $X_j \sim \text{Bernoulli}(0.5), j \in \{1, \ldots, 50\}$. The correlation matrix is obtained using the approach from Joe (2006), which allows to generate positive definite matrices where all pairwise correlations are i.i.d. from a Beta distribution $B(a, a)$ linearly transformed to the interval (-1, 1). The parameter of the Beta distribution equals $a = alphad + (p-2)/2$, where $alphad > 0$ can be chosen. In our case, for $p = 50$ and $alphad = 5/2$ it holds that the pairwise correlation lies between -0.2 and 0.2 with probability 0.85 and between -0.3 and 0.3 with probability 0.97. Correlations with an absolute value larger than 0.4 are extremely unlikely. Multivariate binary random variables $X_j, j \in \{1, \ldots, p\}$ with such correlation structures are then simulated by thresholding normal distributions as described by Leisch et al. (1998). A typical correlation structure of covariates generated by this approach is shown in the left panel of Figure 1.

The second scenario is based on the classical back-cross design for QTL mapping. We used the R/QTL package (Broman et al., 2003) to generate a map of 5 chromosomes of different lengths ranging from 100 cM to 40 cM with 10 equidistant markers per chromosome, see Figure 2. For experimental populations, there is a direct relationship between the genetic distance between markers on the same chromosome and their correlation as described in any textbook on QTL mapping (Chen, 2016). The corresponding correlation structure from simulated genotypes of $n = 1000$ individuals from a back-cross design is illustrated by the heatmap in the right panel of Figure 1. One can see that the correlations between markers on the same chromosome are very strong, getting close to 0.9 for neighbouring markers.

Response variables Y are simulated from the data generating model of Scenario 5 from Hubin et al. (2020a), where $Y \sim N(\mu, 1)$, with

$$\mu = 1 + 1.5L_1 + 3.5L_2 + 9L_3 + 7L_4. \qquad (1)$$

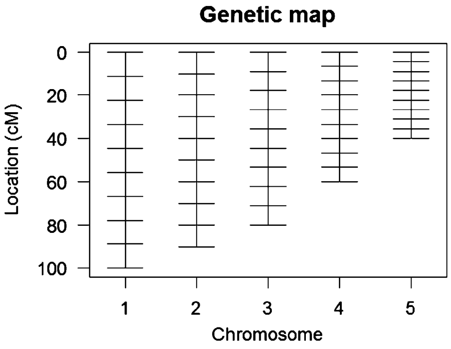


Figure 2: Genetic map of markers on five chromosomes of different length (given in centiMorgan). For the second scenario of our simulation study, these marker positions are used to simulate genotype data from a back-cross design. The closer markers on the same chromosome are lying the stronger will be the correlation of the corresponding genotype data.

The exact definition of the trees L_1–L_4 is given in Table 1 below and is equivalent to the definition in Table 2 of our original article. For the QTL mapping scenario, the responses were generated for each simulation replicate after randomly permuting the order of the genetic markers. In this way, we considered different patterns of correlations between the leaves of the data generating model. For both correlation structures, we generated $N = 100$ datasets with $n = 1000$ observations. Every data set was analysed with the Jeffreys' prior and with the robust g-prior using GMJMCMC with the same tuning parameters as in Scenario 5 of the original article.

We appreciate the comment of Schwender and Ickstadt (2020) that for the higher order interactions L_3 and L_4 the effect sizes are unrealistically large. However, as illustrated by our sensitivity analysis, if one wants to have sufficient power do detect more complex logic expressions with realistic effect sizes then one will need a much larger sample size. This would be potentially feasible for real data analysis (by means of simply collecting more observations) but not for a simulation study with hundreds of simulation runs. In any case, our goal here is to show that correlated regressors are not an impediment for our approach.

Table 1 summarizes the results of our simulations with correlated regressors. For the first scenario with the general structure, correlations are ranging between 0 and 0.5 in absolute values. Comparing the results with Table 2 from the original manuscript which was based on independent regressors the differences are relatively small. Only for L_4, there is a decrease in power, for Jeffreys' prior from 0.89 to 0.66 and for the robust g-prior from 0.9 to 0.66. On the other hand, the number of false positives increases for Jeffreys' prior from 37 to 78 and for the robust g-prior from 28 to 73. It has to be expected that the performance of logic regression becomes a little worse under correlation but GMJMCMC is still behaving very well for the scenario with a general correlation structure. Jeffreys' prior and robust g-prior perform almost equally well with only a very slight advantage of the latter.

	General		QTL	
	Jef.	**R. g**	**Jef.**	**R. g**
$L_1 = X_{37}$	1.00	1.00	0.83	0.85
$L_2 = X_2 \wedge X_9$	0.98	0.99	0.82	0.81
$L_3 = X_7 \wedge X_{12} \wedge X_{20}$	0.96	0.99	0.92	0.92
$L_4 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.66	0.66	0.20	0.24
Overall Power	0.90	0.91	0.69	0.71
FP	0.78	0.73	2.02	2.01
FDR	0.13	0.13	0.39	0.38
WL	9	6	108	98

Table 1: Results for the additional simulation scenarios with correlated binary covariates. Power for individual trees, overall power, the expected number of false positives (FP) and FDR are compared for GMJMCMC using either Jeffreys' prior or the robust g-prior under the general correlation structure and the correlation structure from QTL mapping with back-cross design.

The results in Table 1 for our second correlation structure from QTL mapping are based on the strict definition that only discoveries of trees from the data-generating model itself are counted as true positives. While there is some loss of power, the results for the first three logic expressions are still quite satisfactory. Only for L_4, the estimated power becomes unacceptably low. At the same time, the number of false positives, as well as the number of wrongly detected leaves, increases substantially. For QTL mapping, the correlation between neighboring markers often is so strong, that it becomes extremely difficult to distinguish between them. For that reason, in simulation studies for QTL-mapping, one often takes the approach that the detection of a marker strongly correlated with a QTL is still counted as a true positive. If we take such an approach and consider markers within a range of 15 cM as correct representatives of a leaf from the data generating model then we get slightly better results. In particular, the number of wrong leaves goes down from 108 to 50 for Jeffreys' prior and from 98 to 58 for the robust g-prior. Extending the window for defining true positives would further reduce the number of wrongly detected leaves.

3 Prior related aspects

Clarte and Robert (2020) criticize several aspects of our choice of priors. We fully agree that the use of improper priors is debatable and should be done with great care. We had stated this *explicitly* already in the original article. From a theoretical point of view, our preference would be mixtures of g-priors. As a representative, the robust g-prior is implemented within our package. However, given the strong popularity of the BIC criterion, we wanted to study the performance of this choice as well. Our description of BIC as an approximation of the marginal likelihood under Jeffreys' prior could indeed have included a discussion of its weak points. As Clarte and Robert (2020) remark, the good performance of the BIC choice is most likely connected with applying the Laplace approximation of the marginal likelihood. However, in the case of the Gaussian linear model, the approximation is exact.

The main empirical point, though, is that in all our examples from the original manuscript, the BIC measure as an approximation for the marginal density performed better than the analytical expression under the robust g-prior, both in terms of evaluation metrics and speed. Under the correlated designs provided in this rejoinder, the robust g-prior slightly outperforms Jeffreys' prior in terms of evaluation metrics but the BIC choice still performs rather well. Moreover, the running time of GMJMCMC under Jeffreys' prior (having all of the tuning parameters of the algorithm fixed) is still significantly shorter.

Note that the main contributions of our approach are: a) introducing novel model priors and b) a new search algorithm, whilst for the choice of the parameter priors and the calculation of the marginal densities we are using already established procedures. For example, our approach is fully compatible with integrated nested Laplace approximations (INLA) (Rue et al., 2009) and all of the parameter priors available there can be used. More generally, the R-package we have developed allows the users to easily specify their own method of calculating the marginal likelihood (whatever they prefer and/or what is available for their specific model: analytical integration, Monte Carlo based approximation, or other approximations) for their own choice of parameter priors. This flexibility allows extending the method easily to broader classes of logic regression models. In Section 5, for instance, we describe an extension to latent Gaussian models with both logic and non-logic covariates, where alternative types of parameter priors are possible and the marginal likelihood is computed via integrated nested Laplace approximations (INLA) (Rue et al., 2009).

Also note, that in both Bayarri et al. (2012) and Li and Clyde (2018), priors on models are indirectly obtained through priors on the regression parameters. In our approach, we include specific priors on model complexity as well. This is done via equations (3) and (4) in the main paper. The theoretical properties of combining model and parameter priors definitely require further distinguished research, which, we feel, lies slightly outside the scope of this rejoinder.

4 Algorithmic aspects

Given that one of the main contributions of this manuscript was the development of the GMJMCMC algorithm, it is no surprise that many comments of the discussants were concerned with the algorithm. We will start with replying to some questions which are simple to answer, then give a more detailed recap of the MJMCMC algorithm (Hubin and Storvik, 2018) and finally discuss some questions on the parameter settings of GMJMCMC.

Ruczinski et al. (2020) wondered whether covariates which are not logic can be easily combined with Boolean combinations in the model. The answer is *yes*. We will discuss this extension in Section 5.2 and provide an example in the tutorial in Section 6.3. Ruczinski et al. (2020) also suggested a simple two-stage approach where one first checks whether the covariates have any association with the response variable at all and one only then applies logic regression. This is, of course, a viable approach which can be easily adopted. In practice, this could save resources by avoiding to run computationally costly inference on BLR.

Whilst Bogdan et al. (2020) are *rather reserved with respect to the proposed algorithm*, we believe that most of their concerns are actually based on misunderstandings of the algorithm and we are glad to have the opportunity to clarify some of these points. The question of correlated regressors has been addressed in Section 2.1 of this rejoinder, where we have seen that GMJMCMC works reasonably well even when regressors are heavily dependent. Furthermore, Bogdan et al. (2020) were wondering about a *lack of treatment for tautologies*. This can be easily addressed because, in fact, our implementation of GMJMCMC is taking care of Boolean equivalence already when generating new trees. In particular, as we discuss in Section 2.3 of the paper, "*for all three operators it holds that if the newly generated tree is already present in $\mathcal{S}_t$ then it is not considered for $\mathcal{S}_{t+1}$ but rather a new replacement tree is proposed instead.*" What we do in practice is to check whether newly generated trees have correlation ±1 with any tree within $\mathcal{S}_t$, which for sufficiently large sample size will correspond to logic equivalence. Consequently, tautologies within a GMJMCMC chain are simply not allowed.

Bogdan et al. (2020) also wonder why *the sum in the denominator of (0.2) contains only $M_{fin} = 10000$ models based on d trees from the final stage of the algorithm.* This is indeed one of the implemented options in our package (though M_{fin} does not have to be 10000). The reason for this choice is to avoid having either too large and/or too densely filled hash tables (as a data structure), both of which become quite slow to handle. Whilst this introduces some undesired limitations, it remains an important pragmatic decision to make. The number of logic trees grows exponentially with the number of leaves involved and the number of models grows exponentially with the number of logic trees. Hence, even for the small examples with $p = 50$, the size of a hash table including all visited models and their statistics can become prohibitively large to be used in practice. That would be even more acute for larger p's. As an alternative, one could use the best N_H models from all T generations, where N_H is finite and of reasonable size. But in this case, when the hash table is filled, the worst models must be deleted to allow new ones to be included. In practice, this strategy would become extremely slow. One has to read from, write to and delete from the almost full hash table, which will be also very large. One would either have to create some novel hashing/dehashing functions which make this approach efficient or devise an alternative data structure which is especially designed for the problem at hand. Given the complexity of enumerating logic expressions due to logic equivalence and due to the super-exponential growth of the number of models with respect to the number of leaves involved, we would expect this to be a ground breaking task in the field of algorithms and data structures.

Bogdan et al. (2020) raised the question of why we need MJMCMC for the final population of GMJMCMC when for $d = 15$ full enumeration is feasible. The simple answer is that for many applications one needs much larger d to obtain reliable results, see for example the remark after Theorem 1 of Hubin et al. (2020a) and also Figure 1, panel 3, from the sensitivity analysis of Hubin et al. (2020a). For larger d, a full enumeration will no longer be possible, whilst we would like to offer a generally functioning algorithm. Bogdan et al. (2020) additionally say: *"Also, a huge random reduction of the final model space leads to substantially different results for different parallel runs of the algorithm. Therefore the authors aggregate results from different runs using a weighting scheme specified in equation (15) of their paper. In our opinion it seems more reasonable to estimate the posterior probabilities of different models simply by including all models visited in different runs in denominator of (0.2)."* We agree that in principle this is a reasonable approach, which we, in fact, suggested in Section 2.3 of our paper. There, however, we also discussed the drawback that this approach is computationally more costly because one has to transfer a large amount of information from different models between the cores. Finally, Bogdan et al. (2020) promote using synchronization between the cores via priority queues. Whilst we find the idea interesting, we are a little sceptical whether it would actually work. When compared to embarrassing parallelization, synchronization between the processes in practice often slows down the inference instead of speeding it up (Chai and Bose, 1993; Kukanov, 2008). There, of course, a lot depends on the back-end used for implementation. We currently do not have the capacity to try this approach ourselves, but we would like, by all means, to encourage Bogdan et al. (2020) or other future researchers of BLR to test this idea. We would be very happy if using synchronization via priority queues could lead to an objectively better and faster inference algorithm for BLR than GMJMCMC.

4.1 Mode jumping Markov Chain Monte Carlo

Both Bogdan et al. (2020) and Clarte and Robert (2020) seem to be slightly confused with respect to the MJMCMC algorithm (Hubin and Storvik, 2018), which we did not describe in detail in Hubin et al. (2020a). We thus briefly discuss the main ideas of MJMCMC to clarify certain misunderstandings.

In Hubin and Storvik (2018), a proper MCMC algorithm for the search through a fixed limited model space was proposed. The algorithm deals with the multimodality in the space of models through mode jumping proposals. The mode jumping MCMC (MJMCMC) algorithm relies upon the idea of making smart moves between local extrema with a reasonable frequency. Local MCMC is performed in the absolute majority of steps. For the rest, a large move in the model space (which is likely to hit a model with very low posterior probability) is made, followed by local optimization. The goal of the latter step is to reach a local optimum in a different part of the model space. Then the proposal is randomized around this optimum and the transition to the proposed model is either accepted or rejected according to a Metropolis-Hastings acceptance probability. The convergence properties of the suggested Markov chain is proven through a refinement of the results of Tjelmeland and Hegstad (2001). Its limiting distribution is shown to correspond to the marginal model posterior probabilities. Further extensions of the algorithm allowing for parallel computing and using mixtures of proposals were also suggested.

MJMCMC is described in more detail in Algorithm 1 below, where we consider $M = (\gamma_1, \dots \gamma_p)$ to be associated with models in the given discrete model space Ω (here $\gamma_j \in \{0, 1\}$ indicates whether covariate x_j is included in the model). We assume that marginal likelihoods $p(Y|M)$ are available for a given M, and then use MJMCMC to explore $p(M|Y)$. By Bayes formula

$$p(M|Y) = \frac{p(Y|M)p(M)}{\sum_{M' \in \Omega} p(Y|M')p(M')}. \tag{2}$$

In order to calculate $p(M|Y)$ we have to iterate through the whole model space Ω, which becomes computationally infeasible for large p. The ordinary Monte Carlo estimate is based on a number of MJMCMC samples $M^{(i)}, i = 1, \ldots, W$:

$$\widetilde{p}(M|Y) = \frac{1}{W} \sum_{i=1}^{W} \mathbb{I}(M^{(i)} = M) \xrightarrow[W \to \infty]{d} p(M|Y), \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function. An alternative named the renormalized model (RM) estimate by Clyde et al. (2011), is

$$\widehat{p}(M|Y) = \frac{p(Y|M)p(M)}{\sum_{M' \in \mathbb{V}} p(Y|M')p(M')} \mathbb{I}(M \in \mathbb{V}), \tag{4}$$

where now $\mathbb{V}$ is the set of ***all models visited at least once*** during the MJMCMC run. Assuming the Markov chain eventually will visit all possible models, also $\widehat{p}(M|Y)$ will converge to $p(M|Y)$. Note that this estimate also can utilize all models that are visited, not only those that have been accepted. This answers the comment of Bogdan et al. (2020), who presumed that we include only models accepted by MJMCMC into $\mathbb{V}$. Although both (4) and (3) are asymptotically consistent, (4) will often be the preferable estimator since the convergence of the MCMC based approximation (3) is typically much slower, see Clyde et al. (2011).

We now describe the MJMCMC algorithm in more detail. We aim at approximating $p(M|Y)$ by means of searching for some subspace $\mathbb{V}$ of Ω which makes the approximation (4) as precise as possible. Models with high values of $p(Y|M)$ are important to be included. This means that modes and near modal values of marginal likelihoods are particularly important for the construction of $\mathbb{V} \subset \Omega$ and missing them can dramatically influence our estimates. Note that these considerations are equally important for the standard MCMC estimate (3). The main difference is that when using (3) the number of times a specific model is visited is important, for (4) it is enough that a model is visited at least once. In this context, the denominator of (4) becomes an extremely relevant measure for the quality of the search. It should be as large as possible in order to capture the probability mass from all the local optima of the posterior distribution, whilst at the same time the size of $\mathbb{V}$ should be low in order to save computational time.

Algorithm 1 Mode jumping MCMC.

```
1: Generate a large jump $M_0^*$ according to a proposal distribution $q_l(M_0^*|M)$.
2: Perform a local optimization, defined through $M_k^* \sim q_o(M_k^*|M_0^*)$.
3: Perform a small randomization to generate the proposal $M^* \sim q_r(M^*|M_k^*)$.
4: Generate backwards auxiliary variables $M_0 \sim q_l(M_0|M^*)$, $M_k \sim q_o(M_k|M_0)$.
5: Put
       $M' = \begin{cases} M^* & \text{with probability } r_{mh}(M, M^*; M_k, M_k^*); \\ M & \text{otherwise,} \end{cases}$
   where
       $r_{mh}^*(M, M^*; M_k, M_k^*) = \min\left\{1, \frac{p(M^*|y)q_r(M|M_k)}{p(M|y)q_r(M^*|M_k^*)}\right\}.$   (5)
```

Algorithm 1 describes in detail the mode jumping step within the MJMCMC algorithm. In the first step, a large change in the model space is made through the proposal distribution q_l. This will typically lead to a model with little support in the data, so in step 2 a local optimization is performed in order to obtain a better model. Due to the need for a proper Metropolis-Hastings probability derived through a backwards move (step 4), a randomization, through q_r, of the local optima is needed for the reverse move back to the original model to be possible. Step 5 specifies the acceptance probability which is shown in Hubin and Storvik (2018) to satisfy the detailed balance equation with respect to $p(M|Y)$.

Hopefully, this detailed discussion of MJMCMC fully resolves the confusion of Clarte and Robert (2020), who, in their discussion, presume the following: *"While we have not read the refered article on MJMCMC in detail, a first comment is that the name itself is somewhat unsuitable, as indeed the algorithm does not sample from a distribution but only explores its surface."* We would like to emphasize that the MJMCMC ***is not*** incorporating any of the ideas of Tabu search algorithms (Glover et al., 1995), which are not allowing to return to the previously visited models. This should also clarify another misleading presumption by Clarte and Robert (2020): *"In a self-avoiding mode, keeping track of all the previous states visited by the chain ensures that those states will never be visited again. As we are in a discrete setting, this implies that once a mode has been visited the algorithm is constrained to eventually visit another mode, even if the potential between the modes is almost zero."*

Convergence of GMJMCMC

The MJMCMC algorithm, in the setting of BLR, only gives convergence within each of the restricted search spaces (populations) that it considers. We apply the MJMCMC as an inner iteration within the GMJMCMC algorithm where the space of models is dynamically modified. Given that the movement within and between the search spaces is irreducible with respect to the whole model space, which is shown in Theorem 1 of Hubin et al. (2020a), the GMJMCMC provides the estimates equivalent to (4). They also converge towards the right model probabilities. This fully resolves another concern from

Clarte and Robert (2020) who stated that the renormalized estimator of the marginal posterior model probabilities *"does not provide the theoretical security of (asymptotic) unbiasedness that is attained with MCMC method."*

4.2 Parameter settings

The choice of the tuning parameters for the algorithm is definitely an important problem as indicated by Ruczinski et al. (2020) and Clarte and Robert (2020). Whilst there is not (and cannot be) any uniformly best choice of tuning parameters of GMJMCMC, we will try to briefly indicate some strategies allowing to manually choose reasonable values of the most important tuning parameters of the algorithm. Regarding the choice of the population size d and the maximal number of variables in a model k_{max}, we give some guidance in Remark 1 after Theorem 1 in Hubin et al. (2020a): *"When* $d_1 > 0$ *(which is the* N_{init} *covariates with largest marginal inclusion probability in* $\mathcal{S}_1$*), some restrictions on the possible search spaces are introduced. However, when* $d - d_1 \geq k_{max}$*, any model of maximum size* k_{max} will *eventually be visited. If* $d - d_1 < k_{max}$*, then every model of size up to* $d - d_1$ *plus some of the larger models will eventually be visited, although the model space will get some additional constraints. In practice, it is more important that* $d - d_1 \geq k^*$*, where* k^* *is the size of the true model. Unfortunately, neither* k^* *nor* d_1 *are known in advance, and one has to make reasonable choices of* k_{max} *and* d *depending on the problem one analyses."* Also, note that we provide some sensitivity analysis of d in Section 3.1 of the main article.

Regarding the maximal depth of logic expressions C_{max}, one should use some prior knowledge on the complexity of logic expressions. It also depends upon the individual hypotheses the researcher has. At the same time, using unreasonably large C_{max} is prohibitive computationally and also unrealistic in terms of power to detect too complex trees.

When combining two Boolean expressions, first a decision is made whether it will be combined through an *and* or an *or* operation (with P_{and} specifying the probability for *and*) and thereafter a decision is made whether the logic *not* is applied to it (with probability P_{not}). In our experience, the actual values of these tuning parameters will not influence the result very much with respect to finding the right expressions within the equivalence classes. However, simpler expressions (within the equivalence classes) are usually obtained when choosing somewhat larger P_{and} and somewhat smaller P_{not}. We recommend the choice $P_{and} = 0.9$ and $P_{not} = 0.1$.

The tuning parameter ρ_{min} is used to determine which variables should be removed from the current population with probability one minus the current approximation of the marginal inclusion probability of these variables. ρ_{min} should be chosen in such a way that it is on the one hand possible to get rid of unimportant trees, while at the same time avoiding the deletion of potentially important trees. Concerning the question of Ruczinski et al. (2020) on the choice of M_{fin} and T_{max} and the resulting chain length, we provided some guidance in Theorem A.1 in Section A.2 of Hubin et al. (2020b). There, we proved convergence guarantees also for fixed T_{max} and M_{fin} when increasing the number of parallel chains of GMJMCMC. Thus, apparently, there exists a natural trade off: the more chains one can afford running in parallel the fewer resources could be used within each chain and vice versa – the less parallel chains one runs – the larger T_{max} and M_{fin} are required.

The choice of the tuning parameters for the examples from Hubin et al. (2020a) are provided in Section A.1 of Hubin et al. (2020b). These values might be considered for problems of similar dimensionality, effect sizes and correlations between covariates. At the same time, we cannot provide a strict stopping criterion for GMJMCMC or a general rule for the choice of its parameters. Experimental tuning for different applications might be beneficial. If one has enough computational resources, grid search or an adaptation of Bayesian optimization for the tuning parameters of GMJMCMC (Snoek et al., 2012) can be considered. Alternatively, one might consider some kind of adaptive learning of the algorithm's tuning parameters similarly to Hubin (2019). More details on these possibilities are beyond the scope of this rejoinder.

5 Various extensions of BLR and GMJMCMC

In this section, we briefly present extensions of the logic regression model. Some of these extensions are further discussed in the tutorial of Section 6 of the rejoinder. A more detailed description of the proposed extensions, including theoretical support and real applications, are material for a future publication.

5.1 Predictions with BLR

As mentioned in the discussion section of Hubin et al. (2020a), our method is directly applicable to prediction as well. In particular, the standard Bayesian model averaging can be easily applied. Thus, one can approximate the posterior probability of some parameter/variable Δ via model averaging by

$$\hat{p}(\Delta \mid Y) = \sum_{M \in \mathbb{V}} p(\Delta \mid M, Y)\hat{p}(M \mid Y), \tag{6}$$

where Δ might be, for example, the predictor of unobserved data based on a specific set of covariates. Given estimates of model posterior probabilities, other prediction procedures such as the median probability model (Barbieri and Berger, 2004) or the most probable model can be also easily adopted, yielding:

$$\hat{p}(\Delta \mid Y) = p(\Delta \mid M^*, Y), \tag{7}$$

where M^* is the selected median probability or the most probable a posteriori model.

5.2 BLR with non-binary covariates

Responding to a question from Ruczinski et al. (2020), we can allow non-binary fixed effects to be included in the model. For this extension, we simply replace equation (2) in Hubin et al. (2020a) with:

$$\mathsf{h}\left(\mu\left(\boldsymbol{X}\right)\right) = \alpha + \sum_{j=1}^{q} \gamma_j \beta_j L_j + \sum_{j=q+1}^{q+q'} \gamma_j \beta_j z_{j-q}, \tag{8}$$

where $z_l, l \in \{1, \ldots, q'\}$ are non-binary covariates which are not allowed to form logic expressions. In this formulation of the Bayesian logic regression, the model includes $q+q'$ possible components. The priors on the additional components $\gamma_j, j \in \{q+1, \ldots, q+q'\}$ are of form (4) from Hubin et al. (2020a) with $c(z_{j-q}) = 1, j \in \{q+1, \ldots, q+q'\}$. This results in the following joint model prior:

$$p(M) \propto a^{\sum_{j'=q+1}^{q+q'} \gamma_{j'}} \prod_{j=1}^{q} a^{\gamma_j c(L_j)}. \tag{9}$$

In terms of model inference, the GMJMCMC is adopted, where modifications, mutations and reductions are only allowed for the Boolean terms.

5.3 BLR with non-binary covariates and latent Gaussian variables

We also mentioned in Hubin et al. (2020a) that it is straight-forward to extend our approach for generalized linear mixed models. Here, we will formally describe this extension by including both fixed effects for non-binary covariates and latent Gaussian variables, which can be used to model correlation structures between observations (in space and time) and over-dispersion. For this extension, we further update equation (8),

$$\mathsf{h}\left(\mu\left(\boldsymbol{X}\right)\right) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j L_j + \sum_{j=q+1}^{q+q'} \gamma_j \beta_j z_{j-q} + \sum_{k=1}^{r} \delta_{ik}, \tag{10}$$

where $z_l, l \in \{1, \ldots, q'\}$ are non-binary covariates which are not allowed to form logic expressions and $\boldsymbol{\delta_k} = (\delta_{1k}, \ldots, \delta_{nk}) \sim N_n\left(\mathbf{0}, \boldsymbol{\Sigma}_k\right)$ are latent Gaussian variables. The latent Gaussian variables with covariance matrices $\boldsymbol{\Sigma}_k$ allow to model different correlation structures between individual observations (e.g. auto-regressive models or various other spatio-temporal models). The matrices typically depend only on a few parameters, so that in practice one has $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_k(\boldsymbol{\psi}_k)$. Whilst the model priors (9) are still valid, parameter priors here need to be adjusted as

$$\boldsymbol{\beta}|\boldsymbol{\gamma} \sim N_{p_{\boldsymbol{\gamma}}}(\mathbf{0}, I_{p_\gamma} e^{-\psi_{\beta_\gamma}}), \tag{11}$$

$$\boldsymbol{\psi}_k \sim \pi_k(\boldsymbol{\psi}_k). \tag{12}$$

Here, all kind of hyper-parameters of priors compatible with INLA (Rue et al., 2009) can in principle be chosen. This allows to efficiently compute the marginal likelihoods of individual models using the INLA approach (Rue et al., 2009; Hubin and Storvik, 2016).

6 A tutorial on GMJMCMC for BLR

Finally, we provide a brief tutorial on how to apply our approach in practice. Our code should be run under Linux. One would need to incorporate some sort of extra *hacks* (see https://bit.ly/37tf3cm) to be able to run the code under Windows (due to the limitations of the standard ***parallel::mclapply*** R function which is applied within the library).

6.1 Installing the packages

We start by preparing the R environment for running our approach to BLR. The R-script below will install all packages that are needed to run the code. Depending on which R packages you have already installed, running this script might take a while. Then we install the EMJMCMC package from GitHub.

```r
#*******************************************************************
# install all packages which will be needed for the EMJMCMC package
source("https://raw.githubusercontent.com/aliaksah/EMJMCMC2016/master/
R/load_dependencies/loaddeps.R")
#*******************************************************************
# (currently works only under Linux)
install.packages("https://github.com/aliaksah/EMJMCMC2016/blob/master/
EMJMCMC_1.4.2_R_x86_64-pc-linux-gnu.tar.gz?raw=true",
repos = NULL, type="source")
#*******************************************************************
```

One might want to restart R before proceeding to have a clean environment. After having the package installed we can load EMJMCMC.

```r
#*******************************************************************
# load the EMJMCMC package
library(EMJMCMC)
#*******************************************************************
```

Additionally, we will need the following three packages for the tutorial, which you might have to install from CRAN.

```r
#*******************************************************************
# load other packages needed to simulate and illustrate data
# if necessary these packages first have to be installed from CRAN
library(clusterGeneration)
library(bindata)
library(ggplot2)
#*******************************************************************
```

6.2 Running BLR with weakly correlated covariates

We first generate some binary data with the general correlation structure from the first scenario of the simulation study above.

```
#****************************************************************
# set the seed
set.seed(040590)
# construct a correlation matrix for M = 50 variables
M = 50
m = clusterGeneration::rcorrmatrix(M,alphad=2.5)
# simulate 1000 binary variables with this correlation matrix
X = bindata::rmvbin(1000, margprob = rep(0.5,M), bincorr = m)
#****************************************************************
```

The following code generates the heat-map of Figure 1 which illustrates the non-trivial correlations of the simulated binary variables.

```
#****************************************************************
# prepare the correlation matrix in the melted format
melted_cormat = reshape2::melt(cor(X))
# plot the heat-map of the correlations
ggplot2::ggplot(data = melted_cormat,
ggplot2::aes(x=Var1, y=Var2, fill=value)) +
  ggplot2::geom_tile() +
  ggplot2::theme(axis.title.x = ggplot2::element_blank(),
                 axis.title.y =  ggplot2::element_blank(),
                 axis.text.x = ggplot2::element_blank())
#****************************************************************
```

Next, we simulate the responses according to Scenario 4 from Hubin et al. (2020a), but with correlated binary covariates.

```
#****************************************************************
# simulate Gaussian responses from a model with two-way interactions
# which is considered in S.4 of the paper
df = data.frame(X)
df$Y=rnorm(n = 1000,mean = 1+1.43*(df$X5*df$X9)+
        0.89*(df$X8*df$X11)+0.7*(df$X1*df$X4),sd = 1)
#****************************************************************
```

Before performing logic regression with GMJMCMC one might like to have a look at the documentation of the R function ***LogicRegr***:

```
#****************************************************************
help("LogicRegr")
#****************************************************************
```

The following code runs inference on BLR with 32 parallel threads of GMJMCMC, where we are first using the robust g-prior and then Jeffreys' prior. Depending on the cluster each of these might run for some time from several minutes to more than half an hour. If you are running the code on a home PC or a laptop, please reduce *ncores* parameter to something reasonable for your machine (e.g. set *ncores* = 3).

```
#*********************************************************************
# specify the initial formula
formula1 = as.formula(paste(colnames(df)[M+1],"~ 1 + ",
   paste0(colnames(df)[-c(M+1)],collapse = "+")))
#*********************************************************************
# Bayesian logic regression with the robust-g-prior
res4G = LogicRegr(formula = formula1, data = df,
   family = "Gaussian", prior = "G", report.level = 0.5,
   d = 15,cmax = 2,kmax = 15, p.and = 0.9, p.not = 0.1, p.surv = 0.2,
   ncores = 32)
#*********************************************************************
# Bayesian logic regression with the Jeffreys prior
res4J = LogicRegr(formula = formula1, data = df,
   family = "Gaussian", prior = "J", report.level = 0.5,
   d = 15, cmax = 2,kmax = 15, p.and = 0.9, p.not = 0.1, p.surv = 0.2,
   ncores = 32)
#*********************************************************************
```

We obtain the following results using the robust g-prior:

```
#*********************************************************************
# print the results for the robust g-prior
print(base::rbind(c("expressions","probabilities"),res4G$feat.stat))
     [,1]                [,2]
[1,] "expressions"       "probabilities"
[2,] "I(((X5))&((X9)))"  "1"
[3,] "I(((X1))&((X4)))"  "1"
[4,] "I(((X11))&((X8)))" "0.999999645314492"
#*********************************************************************
```

and rather similar results with the Jeffreys' prior:

```
#*********************************************************************
#print the results for the Jeffreys prior
print(base::rbind(c("expressions","probabilities"),res4J$feat.stat))
     [,1]                [,2]
[1,] "expressions"       "probabilities"
[2,] "I(((X11))&((X8)))" "0.999999774980675"
[3,] "I(((X1))&((X4)))"  "0.999999520871822"
[4,] "I(((X5))&((X9)))"  "0.999873046960372"
#*********************************************************************
```

6.3 Additional non-binary fixed effects and predictions

Ruczinski et al. (2020) asked whether it would be possible to include covariates in the model which are not a part of the logic expressions. Furthermore, Schwender and Ickstadt (2020) are interested in whether the model can be easily used for predictions. These options are currently not implemented in the ***LogicRegr*** function, which we would like to keep as simple as possible. At the same time, these tasks can be easily performed by a general call of the ***EMJMCMC::pinferunemjmcmc*** function which is available in our package. This routine is however much more advanced and requires, at this time, expert knowledge to be used.

First, we will generate an additional Poisson distributed covariate *age* which is then used as an additional additive effect in the data generating logic regression model. For the sake of brevity we perform the analysis here only with Jeffreys' prior.

```
#*******************************************************************
# simulate Gaussian responses from a model with two-way interactions
# and an age effect which is an extension of S.4 of the paper
Xp = data.frame(X)
Xp$age = rpois(1000,lambda = 34)
Xp$Y=rnorm(n = 1000,mean = 1+0.7*(Xp$X1*Xp$X4) +
0.89*(Xp$X8*Xp$X11)+1.43*(Xp$X5*Xp$X9) + 2*Xp$age, sd = 1)
#*******************************************************************
```

We will not only perform model inference but also show how to make predictions with the *EMJMCMC* package. To this end, we will randomly divide the data into a training set (900 observations) and a testing set (100 observations).

```
#*******************************************************************
teid  = sample.int(size =100,n = 1000,replace = F)
test  = Xp[teid,]
train = Xp[-teid,]
#*******************************************************************
```

The function ***pinferunemjmcmc*** has more capabilities than performing logic regression. First, one might want to see its arguments:

```
#*******************************************************************
help("pinferunemjmcmc")
#*******************************************************************
```

The following call of ***pinferunemjmcmc*** performs logic regression using 30 cores. Note that the non-binary covariate *is not* a part of the formula passed to the function, but is rather specified through $runemjmcmc.params\$latnames =$ “$I(age)$”. Also, one might expect this to run slightly longer than previous examples, particularly because keeping track of the β coefficients for prediction takes some additional time. Further, many of the input options used are explained in the help pages of ***pinferunemjmcmc***. If one is not interested in predictions, $runemjmcmc.params\$save.beta = F$, $predict = F$ and $test.data = NULL$ should be set (this will decrease inference time for the same training data sample and other tuning parameters fixed).

```r
#***********************************************************************
# specify the link function
g = function(x) x
#***********************************************************************
# specify the parameters of the custom estimator function
estimator.args = list(data = train, n = dim(train)[1],
  m =stri_count_fixed(as.character(formula1)[3],"+"),k.max = 15)
#***********************************************************************
# specify the parameters of gmjmcmc algorithm
gmjmcmc.params = list(allow_offsprings=1,mutation_rate = 250,
  last.mutation=10000, max.tree.size = 5, Nvars.max =15,
  p.allow.replace=0.9,p.allow.tree=0.01,p.nor=0.01,p.and = 0.9)
#***********************************************************************
# specify some advenced parameters of mjmcmc
mjmcmc.params = list(max.N.glob=10, min.N.glob=5, max.N=3, min.N=1,
  printable = F)
#***********************************************************************
# run the inference of BLR with a non-binary covariate and predicions
res.alt = pinferunemjmcmc(n.cores = 30, report.level =  0.2,
  num.mod.best = 100,simplify = T,predict = T,test.data = test,
  link.function = g,
  runemjmcmc.params = list(formula = formula1,latnames = c("I(age)"),
   data = train,estimator = estimate.logic.lm.jef,
   estimator.args =estimator.args,
   recalc_margin = 249, save.beta = T,interact = T,outgraphs=F,
   interact.param = gmjmcmc.params,
   n.models = 10000,unique = T,max.cpu = 4,max.cpu.glob = 4,
   create.table = F,create.hash = T,pseudo.paral = T,burn.in = 100,
   print.freq = 1000,
   advanced.param = mjmcmc.params))
#***********************************************************************
```

Below, a list of the logic expressions and non-logic covariates that were found to be of importance is listed. There, we clearly see that all features from the data-generative model are detected without any false positive discoveries.

```r
#***********************************************************************
print(base::rbind(c("expressions","probabilities"),res.alt$feat.stat))
     [,1]                 [,2]
[1,] "expressions"       "probabilities"
[2,] "I(((X5))&((X9)))"  "1"
[3,] "I(age)"            "0.999999999999998"
[4,] "I(((X11))&((X8)))" "0.999999990458405"
[5,] "I(((X1))&((X4)))"  "0.99999997999928"
#***********************************************************************
```

To assess the quality of prediction we use two criteria, RMSE $= \sqrt{n_p^{-1}\sum_{i=1}^{n_p}(\hat{Y}_i^* - Y_i^*)^2}$ and MAE $= n_p^{-1}\sum_{i=1}^{n_p}|\hat{Y}_i^* - Y_i^*|$, where Y_i^* are responses in the test data, $\hat{Y}_i^*$ are model averaged predictions of them, and n_p is the size of the test data set.

```
#*****************************************************************
print(sqrt(mean((res.alt$predictions-test$Y)^2)))
[1] "0.8835489"
print(mean(abs((res.alt$predictions-test$Y))))
[1] "0.6904736"
#*****************************************************************
```

We want to compare the performance of BLR in this example with a simple standard approach, namely ridge regression (Zou and Hastie, 2005), combined with model selection according to AIC. In the script below, we run ridge regression and perform prediction on the test data set.

```
#*****************************************************************
library(HDeconometrics)
ridge = ic.glmnet(x = train[,-51],y=train$Y,family = "gaussian",
alpha = 0)
predict.ridge = predict(ridge$glmnet,newx = as.matrix(test[,-51]),
type = "response")[,which(ridge$glmnet$lambda == ridge$lambda)]
print(sqrt(mean((predict.ridge-test$Y)^2)))
[1] "1.061406"
print(mean(abs((predict.ridge-test$Y))))
[1] "0.865467"
#*****************************************************************
```

We finally compute the evaluation metrics for prediction based on the expectations of the data-generative (true) model for the test data:

```
#*****************************************************************
tmean = 1+2*test$age+0.7*(test$X1*test$X4) +
0.89*(test$X8*test$X11)+1.43*(test$X5*test$X9)
print(sqrt(mean((tmean - test$Y)^2)))
[1] "0.8671786"
print(mean(abs((tmean - test$Y))))
[1] "0.6850737"
#*****************************************************************
```

We clearly see that for this specific example logic regression significantly outperforms the ridge regression baseline with respect to both RMSE and MAE. This is not surprising given that the data generative process has multiple non-linear effects. Moreover, the predictions obtained by the BLR model are extremely close to the predictions from the means of the data generative model.

7 Comparison with other approaches

Several other approaches were mentioned by the discussants. Ruczinski et al. (2020) mentioned that simulated annealing for logic regression could be equipped with a penalized likelihood criterion following from the priors used in our setting. Schwender and Ickstadt (2020) pointed out certain similarities of GMJMCMC with Genetic Programming for Association Studies as well as logic Feature Selection. Bogdan et al. (2020) mentioned the recently developed non-reversible MCMC algorithms as well as parallel tempering MCMC algorithms. It would be most interesting to compare all these different algorithms with GMJMCMC but we believe this would need substantial additional effort and goes far beyond the scope of this rejoinder. We leave these possibilities open as topics for further research.

8 Conclusions

We would like to thank once again all of the discussants for their valuable and insightful feedback. We are happy to have provoked so many questions, comments and remarks. We hope that we managed to shed light on the majority of them in this rejoinder. Moreover, we provided some useful extension of Bayesian logic regression method here. The discussions also motivate multiple directions for further research, which are outside the scope of this rejoinder. However, we hope this research will be in future performed in close collaboration with the discussants.

References

Barbieri, M. M. and Berger, J. O. (2004). "Optimal predictive model selection." *The Annals of Statistics*, 32(3): 870–897. MR2065192. doi: https://doi.org/10.1214/009053604000000238. 323

Bayarri, M. J., Berger, J. O., Forte, A., García-Donato, G., et al. (2012). "Criteria for Bayesian model choice with application to variable selection." *The Annals of Statistics*, 40(3): 1550–1577. MR3015035. doi: https://doi.org/10.1214/12-AOS1013. 317

Bogdan, M., Miasojedow, B., and Wallin, J. (2020). "Discussion of "A Novel Algorithmic Approach to Bayesian Logic Regression" by A. Hubin, G. Storvik and F. Frommlet." *Bayesian Analysis*. 313, 318, 319, 320, 331

Broman, K. W., Wu, H., Sen, Ś., and Churchill, G. A. (2003). "R/qtl: QTL mapping in experimental crosses." *Bioinformatics*, 19(7): 889–890. 314

Chai, J. S. and Bose, A. (1993). "Bottlenecks in parallel algorithms for power system stability analysis." *IEEE Transactions on Power Systems*, 8(1): 9–15. 319

Chen, Z. (2016). *Statistical methods for QTL mapping*. Chapman and Hall/CRC. MR3241239. 314

Clarte, G. and Robert, C. (2020). “A discussion on “A Novel Algorithmic Approach to Bayesian Logic Regression”.” *Bayesian Analysis*. 316, 319, 321, 322

Clyde, M. A., Ghosh, J., and Littman, M. L. (2011). “Bayesian adaptive sampling for variable selection and model averaging.” *Journal of Computational and Graphical Statistics*, 20(1): 80–101. MR2816539. doi: https://doi.org/10.1198/jcgs.2010.09049. 320

Fritsch, A. (2006). “A Full Bayesian Version of Logic regression for SNP Data.” Ph.D. thesis, Diploma Thesis. 313

Glover, F., Kelly, J. P., and Laguna, M. (1995). “Genetic algorithms and tabu search: hybrids for optimization.” *Computers & Operations Research*, 22(1): 111–134. 321

Hubin, A. (2019). “An adaptive simulated annealing EM algorithm for inference on non-homogeneous hidden Markov models.” In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, 1–9. 323

Hubin, A. and Storvik, G. (2016). “Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).” arXiv:1611.01450v1. 324

Hubin, A. and Storvik, G. (2018). “Mode jumping MCMC for Bayesian variable selection in GLMM.” *Computational Statistics & Data Analysis*, 127: 281–297. MR3820324. doi: https://doi.org/10.1016/j.csda.2018.05.020. 312, 317, 319, 321

Hubin, A., Storvik, G., and Frommlet, F. (2020a). “A novel algorithmic approach to Bayesian Logic Regression.” *Bayesian Analysis*. 312, 313, 314, 318, 319, 321, 322, 323, 324, 326

Hubin, A., Storvik, G., and Frommlet, F. (2020b). “Supplementary material for “A novel algorithmic approach to Bayesian Logic Regression”.” URL https://projecteuclid.org/download/suppdf_1/euclid.ba/1545296448. 322, 323

Hubin, A., Storvik, G., Grini, P., and Butenko, M. (2020c). “A Bayesian binomial regression model with latent Gaussian processes for modelling DNA methylation.” *Austrian Journal of Statistics*, 49–50. 313

Joe, H. (2006). “Generating random correlation matrices based on partial correlations.” *Journal of Multivariate Analysis*, 97(10): 2177–2189. MR2301633. doi: https://doi.org/10.1016/j.jmva.2005.05.010. 314

Kukanov, A. (2008). “Why a simple test can get parallel slowdown.” URL https://software.intel.com/en-us/blogs/2008/03/04/why-a-simple-test-can-get-parallel-slowdown. 319

Leisch, F., Weingessel, A., and Hornik, K. (1998). “On the generation of correlated artificial binary data.” 314

Li, Y. and Clyde, M. A. (2018). “Mixtures of g-priors in generalized linear models.” *Journal of the American Statistical Association*, 113(524): 1828–1845. MR3902249. doi: https://doi.org/10.1080/01621459.2018.1469992. 317

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2020). "Invited comment on Article by Hubin, Storvik and Frommlet." *Bayesian Analysis*. 313, 317, 322, 323, 328, 331

Rue, H., Martino, S., and 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. MR2649602. doi: https://doi.org/10.1111/j.1467-9868.2008.00700.x. 317, 324

Schwender, H. and Ickstadt, K. (2020). "Discussion of a novel algorithmic approach to Bayesian logic regression." *Bayesian Analysis*. 315, 328, 331

Snoek, J., Larochelle, H., and Adams, R. P. (2012). "Practical Bayesian optimization of machine learning algorithms." In *Advances in Neural Information Processing Systems*, 2951–2959. 323

Tjelmeland, H. and Hegstad, B. K. (2001). "Mode jumping proposals in MCMC." *Scandinavian Journal of Statistics*, 28(1): 205–223. MR1844357. doi: https://doi.org/10.1111/1467-9469.00232. 319

Zou, H. and Hastie, T. (2005). "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320. MR2137327. doi: https://doi.org/10.1111/j.1467-9868.2005.00503.x. 330