Bayesian model configuration, selection and averaging in complex regression contexts

Aliaksandr Hubin

Dissertation presented for the degree of
Philosophiae Doctor (PhD)

UiO : University of Oslo

Department of Mathematics
University of Oslo
July 2018

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2035

ISSN 1501-7710

Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

Preface

This thesis is submitted for the degree of Doctor of Philosophy in mathematics at the University of Oslo, Norway. The research described herein was conducted under the supervision of Professor Geir Olve Storvik at the Department of Mathematics between August 2014 and July 2018. I will without doubt be looking back at these years as an enjoyable and challenging period in my life, during which I both matured and learned a lot, scientifically and personally. In this thesis I was working on the exciting topic of Bayesian model selection and averaging in various regression contexts from simple linear models to very complicated and reach deep regressions. Late in the work on deep Bayesian regression models I observed that numerous attempts to tackle Bayesian deep learning were done in the machine learning community and the whole NIPS 2017 session was devoted to Bayesian deep learning. The statistical and machine learning schools are still quite different and hopefully for me the developed approaches were novel even in the context of massive research on the field in the machine learning community. Yet I was disappointed with the lack of communication between the two schools. I hope the presented in this thesis work could bridge some gaps in the obviously emerging field of study. And I believe it could be of interest for both mathematicians and computer scientists. In order to make the thesis available to a broader audience, a very gentle and consistent introduction to statistical modeling and inference is given, summarizing the topics required for comfortable reading through the papers for people with a general mathematics (computer science) background.

Oslo, July 2018
Aliaksandr Hubin

Acknowledgements

Upon completing this thesis, I would like to share some most sincere acknowledgements. First and foremost, I would like to express the deepest gratitude to my main supervisor Geir Olve Storvik for his most active and helpful involvement in this work. The dissertation greatly benefited from his wise supervision, constructive criticism, advice, and comments. Secondly, I would like to gratitude my collaborator Florian Frommlet, who is a coauthor of the three articles in this dissertation, for numerous discussions, advice and help. I would also like to gratefully thank my parents Valiantsina Hubina and Aliaksandr Hubin for all the love, support and inspiration they shared with me. In these often rather lonely days of living far away from home numerous voice-ip-based conversations and occasional visits and trips made my life significantly brighter. This list would not be full without my sister, Volha Paliatayeva, and my aunt and uncle, Elvira and Mikalai Krutsko, who have always been inspiring examples of establishing prominent academic careers, and who have given a lot of support and shared the experience with me. Furthermore, I am extremely thankful to my uncle, Aliaksandr Tsikhanau, and my aunt, Natalia Stakhouskaya, for their wise supportive advice and love. Last but not least, I cannot even express my gratitude and affection to other members of my family, colleagues, and friends. Sometimes you did not even realize that you all supported and inspired me very much. Mentioning all of you would make the list far too long, but each of you is indeed dear to me in your own special way. Additionally, I would like to acknowledge my co-supervisors Ole Christian Lingjærde, Paul Grini and Melinka Butenko for the discussions we had together, which helped a lot to set the direction of the research and learn new things. It is also important to mention my friends and colleagues Olav Nikolai Breivik, Riccardo De Bin, and Jonas Moss for the interesting and thought-provoking discussions throughout the work. Finally, I would like to thank CELS for fully financially supporting this work throughout four years and NORBIS for the opportunity to spend fantastic three months of scientific exchange in Vienna.

List of papers and manuscripts

Paper I

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

Paper II

Hubin, A., Storvik G., Frommlet F. (2018). A novel algorithmic approach to Bayesian Logic Regression. *Submitted for publication*.

Paper III

Hubin, A., Storvik G., Frommlet F. (2018). Deep Bayesian regression models. *Submitted for publication*.

Paper IV

Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). A comprehensive study of Bayesian approaches to Genome-Wide Association Studies. *Manuscript*.

Paper V

Hubin, A., Storvik G. (2016). Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA). *Technical report; arXiv:1611.01450*.

Contents

1 Introduction

Regression models are powerful tools for answering numerous scientific questions in both natural and social sciences. These days they have also become widely appreciated in business related applications via the data science discipline. Traditionally, scientists have been very carefully specifying adequate models and choosing explanatory variables. The orthodox statistical approach discourages both collecting data of too many variables and relying on automatic procedures to detect the important ones (Burnham and Anderson, 2002). Instead, expert based knowledge of the field should guide the model building process such that only a moderate number of models are considered when answering specific research questions or hypotheses. However, in modern data analysis the number of explanatory variables becomes often so huge that it is almost impossible to carry preselection by hand. At the same time, development of computational resources allows to resolve the automated model selection and model averaging problems accurately and within a reasonable amount of time.

Statisticians traditionally dealt mainly with linear models due to their transparency and low computational complexity. However, it is often the case that linear relations between the explanatory variables and the response are not sufficient for high quality inference or predictions. Aitkin (2011) in this context said: "It is fair to say that the frequentist paradigm is coming to the end of its useful life; one sign of a dying paradigm is the proliferation of new “flexible” methods untrammelled by the paradigm: regression trees, with their recipes for growing and pruning, and the grandiose claims once made for neural networks, now made for support vector machines". Indeed, nonlinear effects and complex functional interactions between the explanatory variables can often significantly improve both the predictive and the inferential performance of the models. Nonlinear relations are for example handled by classification and regression trees, fractional polynomials, random forests, logic regressions, neural networks, etc. Whilst some efforts on model selection in classification and regression trees and logic regressions have been already done (Kooperberg and Ruczinski, 2005; Kooperberg et al., 2007; Fritsch and Ickstadt, 2007; Fritsch, 2006; Chen and Guestrin, 2016; Lambert et al., 2007), the topic remains rather undiscovered in the context of neural networks. Recently, particular cases of neural network based analysis called deep learning procedures have become extremely popular and highly successful in a variety of real world applications (Goodfellow et al., 2016). These algorithms apply iteratively nonlinear transformations aiming at optimal prediction of response variables. Each transformation yields another hidden layer of features, which are also called neurons. The architecture of a deep neural network then includes the specification of the nonlinear intra-layer transformations, the number of layers, the number of features at each layer and the connections between the neurons. The resulting model is fitted by applying some optimization procedure (e.g. stochastic gradient decent), in the space of parameters, with the goal of meeting a particular objective like minimization of some loss function, or maximization of the likelihood. However, as mentioned above, model selection across deep regression models remains an open and yet

undiscovered research area, raising exciting mathematical and computational challenges. The traditional deep learning approach assumes that the architecture is set manually, but recently approaches to select an optimal architecture of neural networks algorithmically have started to appear (Zoph et al., 2017; Pham et al., 2018; Elsken et al., 2017).

All of the models have particular kinds of strength and weaknesses leading to the performance limitation of single models. In practice, researchers and analysts are starting to actively use model averaging and model stacking (using output of some models as inputs for others), which often yields boosted predictive performance. At the same time, such heuristic procedures often lead to overfitting issues and totally uninterpretable results. It is hence of a particular importance to rigorously generalize these approaches into a single and well defined statistical model, which could allow to combine benefits and reduce weaknesses of different existing approaches. Constructing of such a meta model will definitely make a step towards satisfying Albert Einstein's famous quote (Einstein, 1934): "It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience." Furthermore, novel efficient and theoretically sound algorithms are required to fit such a powerful and general model.

Traditionally in statistics two major fundamental paradigms are frequentist and Bayesian statistics. Both of them induce different ways to make inference. Additionally, different model selection approaches arise within these paradigms. Model selection within them often differs in terms of both criteria and algorithms addressed, thought some overlaps are also present. Let us first address the paradigms as such and their development. In both philosophical literature (Zabell, 2005; Jeffrey, 1956; Reichenbach, 1935; Carnap, 1952; Ramsey, 1931; Keynes, 1921) and statistical literature (Jeffreys, 1939; Carnap, 1950) there have been a lot of discussions of frequentist and Bayesian approaches. Both of them are extremely influential in modern scientific community. They have been widely accepted by the statisticians and scientists and have formed a solid basis for becoming statistical paradigms (namely Bayesian and frequentist statistics) (Aitkin, 2011). Both of the approaches have also been sufficiently innovative to attract numerous followers not only among pure statisticians but also among natural and social scientists as well as industrial practitioners, which is proven by the amount of published articles or software packages, where either of them is used. One of the evolutionary developments within the discussed statistical paradigms is regression. Both linear and nonlinear regressions are currently targeted within Bayesian and frequentist statistics. Many, however, prefer the Bayesian approach for these purposes. Aitkin (2011) says the following regarding the issue: "The frequentist difficulties are familiar; for our purposes it is sufficient to point to the major successes of MCMC in complex unbalanced crossed and nested multilevel GLMMs, and the widespread adoption of multiple imputation, with its steady development towards a fully Bayesian analysis with incomplete data." In particular, the advantages of Bayesian approaches, when posterior model probabilities are used, are mainly high interpretability of the obtained results in the sense that for model selection one is simply searching for the posterior mode in the model space. At the same time, model averaging becomes highly interpretable from the point of view of classical probability theory, since corresponding model averaged probabilities of quantities of interest are simply marginalized out from the whole model space utilizing poste-

rior model probabilities. Moreover, whilst doing this, both model and parameters' uncertainties are properly handled. For example, variable importance can be computed as marginal posterior probability of this covariate to be present in the models. In order to take advantage of these fundamentally important properties of the Bayesian approach, this thesis will be build within the Bayesian framework and target an actual problem of variable selection within linear and nonlinear regressions. The main goal, thus, is to evaluate posterior probabilities of the models within the model space of interest, defined uniquely by the subsets of explanatory variables (features), within a particular regression context.

In particular, Paper I addresses Bayesian generalized linear mixed models. The main contribution of this paper is development of a mode jumping Monte Carlo Markov chain algorithm (MJMCMC) for model selection and model averaging, which allows to efficiently sample in the multimodal space. Paper II addresses Bayesian logic regressions. 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). The paper also generalizes the algorithm developed in Paper I to the domain of logic regression models, where the model space cannot be pre-specified in advance. This is achieved by means of a genetically modified mode jumping Monte Carlo Markov chain (GMJMCMC). It has to be noted that GMJMCMC is not a proper Metropolis-Hastings algorithm in the sense that its stationary distribution does not coincide with the target distribution of interest, however it guarantees exploring the whole model space asymptotically. This is sufficient for using alternative estimates of the target distribution based on the Bayes formula. Paper III introduces a class of deep Bayesian regression models (DBRM), which combines and generalizes classes of linear models, generalized linear models, generalized linear mixed models, classification and regression trees, multivariate adaptive regression splines, artificial neural networks, logic regressions and fractional polynomials into a powerful and broad Bayesian framework and addresses model selection and model averaging within the defined class of models. The GMJMCMC algorithm, developed in Paper II, is adapted to DBRM. Furthermore, a reversible version of GMJMCMC (RGMJMCMC), which is a proper Metropolis-Hastings algorithm, for fitting DBRM is developed in Paper III. Paper IV is an application of Bayesian model selection procedures, including those developed in Papers I, II and III, to genome wide association studies (GWAS). Finally, Paper V is focused on the comparison of different computational approaches to marginalizing parameters from the likelihood function in order to obtain the marginal likelihood (MLIK). Computing marginal likelihood is fundamentally important for the algorithms developed and used in Papers I-IV, since using MLIKs allows to avoid model selection within the joint space of parameters and models and rather work in the marginal space of models, which gives significant computational benefits. To summarize, *the main goal of this thesis is to suggest efficient and scalable Bayesian approaches for model selection and model averaging in linear and nonlinear Bayesian regression contexts*, that can be used in numerous applications in science, business and technology.

The remainder of this thesis is structured as follows: Chapter 2 gives a gentle introduction to statistical modeling and discusses such important models as linear regressions, generalized linear models, generalized linear mixed models, generalized additive models and artificial neural networks; Chapter 3 addresses statistical inference from the Bayesian and frequentist per-

spectives and discusses such popular inference methods as maximal likelihood, generalized method of moments, generalized least squares method, minimal divergence based methods, Monte Carlo Markov chains, variational Bayes, approximate Bayesian computing, and integrated nested Laplace approximations; Chapter 4 addresses model selection and validation concepts, describes the most popular model selection criteria in the Bayesian and frequentist settings, algorithms for Bayesian model selection and model averaging are also addressed there; Chapter 5 consists of summaries of each of the five articles constituting this thesis; Finally, Chapter 6 summarizes the contribution of the thesis, gives directions of further work, and discusses a few selected topics in more detail.

2 Statistical modeling

In this chapter statistical modeling concepts are addressed, and some of the most widely used models in modern statistical science are discussed.

Let $m(\boldsymbol{\theta}, \mathcal{D})$ be a general statistical model for data $\mathcal{D} = \{\mathbf{y}, \mathbf{X}\}$ on n realizations over $p + 1$ components of a random variable ξ defined on a Kolmogorov probability space $(\boldsymbol{\Omega}, \mathcal{F}, \mathcal{P})$ (Kolmogorov), for which the model likelihood $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ is constructed, where $\boldsymbol{\theta}$ is a vector of size k of unknown parameters and $\mathbf{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ with $\boldsymbol{x}_i, i \in \{1, ..., n\}$ are p dimensional vectors, and hence $\mathbf{X} \in \mathcal{R}^{n \times p}$. Without loss of generality, $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ will be further denoted as $p(\mathcal{D}|\boldsymbol{\theta})$, when considering the conditional likelihood of $\mathbf{y}$. In a very broad sense, a model $m(\boldsymbol{\theta}, \mathcal{D})$ can be parametric, semiparametric (for both cases $k > 0$) or non-parametric ($k = 0$). Note that some sources also consider the case, when $k = \infty$ as non-parametric. A model can belong to either a Bayesian or a frequentist domain. Among the most widely addressed statistical models, one can mention linear regression models (LM), generalized linear models (GLM), generalized linear mixed model (GLMM), classification and regression trees (CART), artificial neural networks (ANN), hidden Markov models (HMM), state space models (SSM), and generalized additive models (GAM). Models overlap between each other in different ways. For example, LM is a subclass for GLM, which is a subclass of ANN and GLMM, however GLMM is not a subclass of ANN as well as ANN is not a subclass of GLMM. GLM is at the same time a subclasses of GAM, whilst ANN strictly speaking is not. HMM are a subclass of SSM and so is GLMM. In the next sections the brief description of the most relevant classes of models for this thesis will be presented.

2.1 Linear models

The most widely known and addressed statistical model, with applications in numerous areas of science and business, is definitely the linear regression model (Freedman, 2009; Rencher and Christensen, 2012; Yan and Su, 2009; Seal, 1967), which is also known as Gaussian regression. Conditional independence between observations $y_i, i \in \{1, ..., n\}$ is assumed in linear regression, letting the likelihood function factorize easily. Then the relation between the explanatory variables and the observations is modeled via the following equations:

$$p(y_i|\mu_i, \sigma^2) = N(\mu_i, \sigma^2), \tag{2.1}$$

$$\mu_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}. \tag{2.2}$$

Here $\beta_j \in \mathcal{R}, j \in \{0, ..., p\}$, are the regression coefficients and σ^2 is the variance of the responses, which is assumed to be common across all of the observations. Hence, for the linear

regression models $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma^2\}$. The model is simple and easy to estimate and interpret. However the assumptions of this model are rather strict. Generalized linear models allow to get rid of the assumption of the Gaussian distribution of the responses, whilst generalized linear mixed models additionally allow to relax on the assumption of conditional independence of the observations. These models are briefly described in the following sections.

2.2 Generalized linear models

Generalized linear models (McCullagh and Nelder, 1989) broaden the class of linear regression models by assuming various distributions from the exponential family on $\mathbf{y}$. This allows to adjust to different data types, including among others, binomial, Poisson, gamma, or exponential distributions of observations. The generalized linear models are of the form:

$$p(y_i|\mu_i, \phi) = \mathfrak{f}(y|\mu_i, \phi), \quad \mu_i = g^{-1}(\eta_i), \tag{2.3}$$

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}. \tag{2.4}$$

Here, similarly to the linear regression, $\beta_j \in \mathcal{R}, j \in \{0, ..., p\}$, are the regression coefficients and ϕ is the dispersion parameter, which is assumed to be common across all of the observations. The link function $g(\cdot)$ is introduced in order to link the linear predictor η_i and the mean parameter of the corresponding distribution of interest $\mathfrak{f}$. Similarly to the linear regression context, for GLMs $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \phi\}$.

GLM hence is a very broad and powerful class of statistical models, which can be used in more applications than linear regression. However, it is still limited by several strong assumptions. Similarly to linear models, one of the major disadvantages of the generalized linear models is the assumption on the independence of the observations. Secondly, the models are linear in terms of the relations of the explanatory variables and the linear predictor. The first issue is resolved by GLMMs, whilst the second one is typically handled by GAMs and ANNs.

2.3 Generalized linear mixed models

To relax on the assumption of conditional independence and homoscedasticity of the dispersion of the observations, consider the following generalized linear mixed model (McCulloch and Neuhaus, 2001):

$$p(y_i|\mu_i, \phi) = \mathfrak{f}(y|\mu_i, \phi), \quad \mu_i = g^{-1}(\eta_i), \tag{2.5}$$

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \delta_i, \tag{2.6}$$

$$\boldsymbol{\delta} = (\delta_1, ..., \delta_n) \sim N_n(\mathbf{0}, \boldsymbol{\Sigma}_\delta). \tag{2.7}$$

Here it is assumed that $\mathfrak{f}(y|\mu,\phi)$ is a density/distribution from the exponential family with the corresponding link function $g(\cdot)$. $\beta_j \in \mathcal{R}, j \in \{0,...,p\}$, are the regression coefficients and ϕ is the dispersion parameter. The unexplained variability of the responses and the correlation structure between them is addressed through the latent Gaussian variables δ_i, with a specified parametric covariance matrix structure, defined through $\boldsymbol{\Sigma}_\delta = \boldsymbol{\Sigma}_\delta(\boldsymbol{\psi}) \in \mathcal{R}^{n\times n}$. Here $\boldsymbol{\psi}$ are parameters describing the correlation structure. The vector of parameters of the model is of the form $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\psi}, \phi\}$. Even though GLMMs are very powerful and broad, they still rely upon the linear nature of the relations between the explanatory variables and linear predictors.

2.4 Generalized additive models with extensions

Generalized additive model (GAM) (Hastie and Tibshirani, 1990) is essentially a generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some explanatory variables. The interest focuses on inference about these smooth functions and their parameters. GAM was specially developed to combine properties of generalized linear models with additive models. Similarly to GLM and GLMM, the model relates a response variable, $\mathbf{y}$, and explanatory variables $\mathbf{X}$. An exponential family distribution along with a link function g are specified linking the expected value of $\mathbf{y}$ to the explanatory variables via various functional structures $F_j(\cdot|\boldsymbol{\omega_j})$ with parameters $\boldsymbol{\omega_j}$ These functional structures can be functions with a specified parametric form like a polynomial or regression spline of a variable. They can also be specified non-parametrically or semi-parametrically. This flexibility on one hand provides the potential for better fit to the data than purely parametric models, but on the other hand often results in some loss of interpretability. A GAM model has a form:

$$y_i|\mu_i \sim \mathfrak{f}(y|\mu_i,\phi), \quad i \in \{1,...,n\}, \tag{2.8}$$

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{p} F_j(x_{ij}|\boldsymbol{\omega_j}). \tag{2.9}$$

The vector of parameters of GAMs is hence $\boldsymbol{\theta} = \{\cup_{j=1}^{p}\boldsymbol{\omega_j}, \beta_0, \phi\}$. GAMs can also incorporate latent Gaussian variables extending the class to generalized additive mixed models (GAMM) (Fahrmeir and Lang, 2001). GAMM will not be described in detail, since this generalization is equivalent to the extension of GLM to GLMM described in the previous section. Instead, more general additive models, allowing for $F_j(\cdot|\boldsymbol{\omega_j})$ to depend jointly on all covariates (i.e. $F_j(\cdot|\boldsymbol{\omega_j}) : \mathcal{R}^p \to \mathcal{R}$), will be introduced. Such models have a form:

$$y_i|\mu_i \sim \mathfrak{f}(y|\mu_i,\phi), \quad i \in \{1,...,n\}, \tag{2.10}$$

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{q} F_j(\boldsymbol{x}_i|\boldsymbol{\omega_j}), \tag{2.11}$$

with parameters $\boldsymbol{\theta} = \{\cup_{j=1}^{q}\boldsymbol{\omega_j}, \beta_0, \phi\}$. If one relaxes on the assumption of smoothness of

$F_j(\cdot|\boldsymbol{\omega_j})$, then logic regressions (LR) will fall into this category. Provided that all covariates are binary, the logic regressions, addressed in Paper II, take the functional form (2.10)-(2.11) with a linear predictor of the form:

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{q} \omega_j L_j, \tag{2.12}$$

where $\omega_j \in \mathcal{R}, j \in \{1, ..., q\}$, are the regression coefficients for the corresponding L_j, which are all possible logical expressions based on the input covariates. Logical expression are combinations of the binary variables $\boldsymbol{x}_j$ with the logical operators $\wedge$ (AND), $\vee$ (OR) and $\boldsymbol{x}^c$ (NOT $\boldsymbol{x}$), for example $L = (\boldsymbol{x}_1 \wedge \boldsymbol{x}_2) \vee \boldsymbol{x}_3^c$. The extended GAM also includes classification and regression trees (CART), artificial neural networks (if one relaxes on that ƒ belongs to the exponential family), and the deep Bayesian regression model (DBRM), developed in Paper III of this thesis. In the section to follow a detailed specification of artificial neural networks is given.

2.5 Artificial neural networks

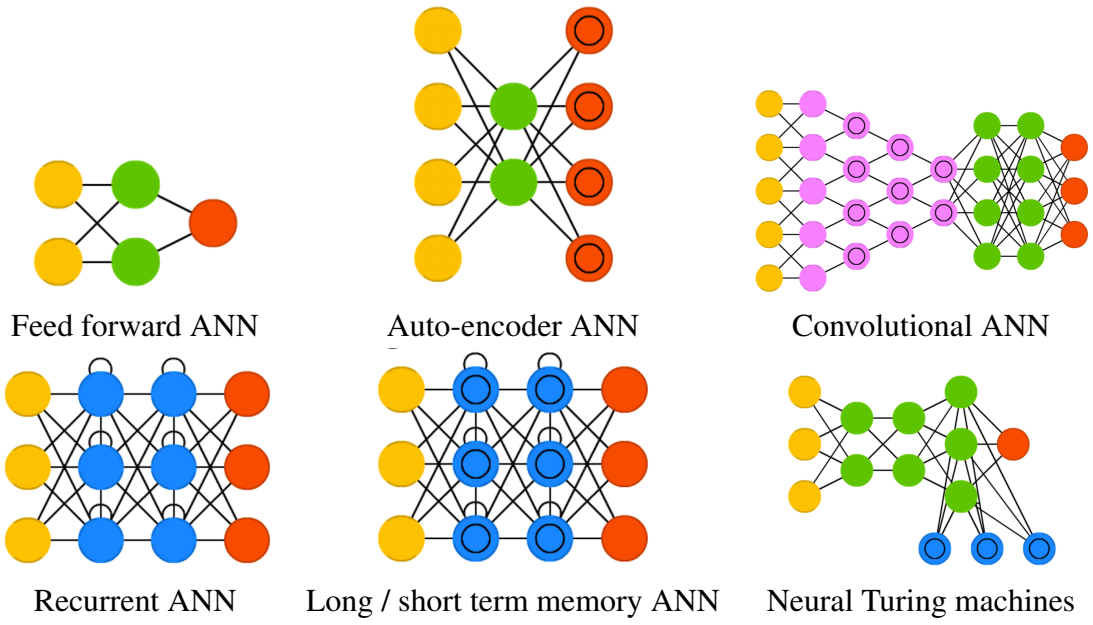

Figure 2.1: Here the most popular ANN architectures are presented. In the graphs each node represents a neuron based on a weighted sum of an input vector transformed by an activation function to produce an output. Yellow and red colored neurons are input and output nodes correspondingly. Pink colored neurons apply weighted inputs using a predefined kernel matrix. Green neurons are fully connected ones, where the sparsity has to be decided by the estimator of the zero values of some of the weights. Blue neurons are recurrent and they append their values from previous pass to the input vector. Blue neuron with circle inside a neuron corresponds to a memory cell. Red neuron with circle inside a neuron corresponds to the output, which coincides with the corresponding input (used only in auto-encoders). These architectures are originally presented in van Veen (2016).

In this section artificial neural networks (ANN) (Schalkoff, 1997) will be briefly described, following the notation from Polson et al. (2017). Similarly to the previously addressed models,

ANN links the observations $\mathbf{y}$ and explanatory variables $\mathbf{X}$ (note that in some ANNs $\mathbf{X}$ also includes $\mathbf{y}$ or even coincides with $\mathbf{y}$, which will be discussed further). ANNs do this via a functional mapping of the form (2.10), but possibly with a multidimensional mean parameter $\boldsymbol{\mu}_i = \boldsymbol{\mu}_i(\boldsymbol{x}_i), \boldsymbol{\mu}_i \in \mathcal{R}^r, i \in \{1, ..., n\}$, which can be written as:

$$y_i \sim \mathfrak{f}\left(\boldsymbol{\mu}_i, \phi\right). \tag{2.13}$$

The output y_i can be continuous, discrete or mixed, one or several dimensional, and does not necessarily belong to the exponential family. To construct the vector of mean parameters $\boldsymbol{\mu}_i$ of the distribution of interest, one builds a sequence of building blocks of hidden layers. Let $\sigma_j^{(l)}$ be univariate functions (further referred to as *activation functions*), where $l \in \{1, ..., L\}$ is the index of the layer, L is the number of layers (further referred to as *depth*), $j \in \{1, ..., p^{(l)}\}$ is the index of a hidden variable (further referred to as a *neuron*) from layer l, constructed by the corresponding activation function, and $p^{(l)}$ is the number of neurons in layer l (further referred to as *width* of a layer), here $p^{(1)} = p$ and $p^{(L)} = r$. Activation functions $\sigma_j^{(l)}$ are assumed to be differentiable almost everywhere and have non-constant partial derivatives in all $l \in \{1, ..., L-1\}$. For the last layer, $\sigma_j^{(L)}, j \in \{1, ..., r\}$ are allowed to be identity functions and, in case y_i belongs to the exponential family with $r = 1$, take a form of GLM link functions, described in detail in the previous sections. To construct a neuron j in layer $l+1$ for observation i, denoted as $z_{ij}^{(l+1)}$, a semi-affine transformation is used:

$$z_{ij}^{(l+1)} = \sigma_j^{(l)} \left(\beta_{0j}^{(l)} + \sum_{k=1}^{p^{(l)}} \beta_{kj}^{(l)} z_{ik}^{(l)} \right). \tag{2.14}$$

Here $\beta_{kj}^{(l)}$ are the slope coefficients for the inputs $\boldsymbol{z}_{ik}^{(l)}$ (note that $\boldsymbol{z}_{ik}^{(1)} = x_{ik}$) of the l-th layer and intercepts, and $p^{(l)}$ is the number of hidden units (neurons) at layer l. The mean vector $\boldsymbol{\mu}_i$ of ANN $y_i \sim \mathfrak{f}\left(\boldsymbol{\mu}_i, \phi\right)$ with L layers is a composite function of the form:

$$\boldsymbol{\mu}_i = \left(\vec{\sigma}^{(L)} \circ \ldots \circ \vec{\sigma}^{(1)}\right)(\boldsymbol{x}_i), \tag{2.15}$$

where $\circ$ is the composition operator, i.e. $f \circ g(x) = f(g(x))$ and $\vec{\sigma}^{(l)}(\cdot), l \in \{1, ..., L\}$ are the vector functions consisting of the corresponding univariate activations. The vector of parameters of ANNs hence is of the form $\boldsymbol{\theta} = \{\cup_{l=1}^{L-1} \cup_{j=1}^{p^{(l+1)}} \cup_{k=0}^{p^{(l)}} \beta_{kj}^{(l)}, \phi\}$. In the machine learning community parameters $\beta_{kj}^{(l)}$ are referred to as *weights* of the neural networks. In practice, when strictly monotonous and bounded activations are used, ANNs with at least one hidden layer and containing a finite number of neurons satisfy the conditions of universal approximation theorem (Hornik, 1991) and can approximate continuous functions on compact subsets of the Euclidean space.

Depending on the structure and sparsity of the weights and activation functions, artificial neural networks can be classified as dense, convolutional, or recurrent. Dense ANN have fully connected layers and are quite rarely addressed due to their complexity and regularization difficulties. Convolutional ANN (CNN) and recurrent ANN (RNN) allow for sparse representations. CNN are typically used for the conditionally *i.i.d.* observations, whilst recurrent RNN typically

have $\mathbf{y}$ included into $\mathbf{X}$ and are used for space-state modeling. Depending on the number of layers ANN can be either deep or shallow. Recently, deep architectures have become extremely successful in applications (LeCun et al., 2015; Goodfellow et al., 2016). The most successful applications include image, text, sound, and video analysis, where huge amounts of data are available. An example of the artwork created by the convolutional neural network described in Gatys et al. (2015) is presented in Figure 2.2. Pascanu et al. (2013) and Montúfar and Morton (2015) show the advantage of representing functions with deep architectures, furthermore, Poggio (2016) provides theoretical results on the conditions when deep learning is preferable to shallow learning. In case $\mathbf{y}$ and $\mathbf{X}$ coincide, ANNs are called auto-encoders. Auto-encoders are typically used to compress high-dimensional data into a set of lower-dimensional features without significant loss of the information. In case $\mathbf{X}$ is a random noise of some simple structure (e.g. multivariate Gaussian), the ANNs are called generative and allow to sample from complex distributions based on the samples of $\mathbf{X}$. In particular, generative adversarial networks (GAN) (Goodfellow et al., 2014) today form a very popular research direction in machine learning (Goodfellow, 2016). The research on ANN is developing very rapidly inducing numerous new names for artificial neural networks having different configurations. In Figure 2.1 one can find illustrations of several popular architectures of ANN including feed-forward architectures, auto-encoders, convolutional, and recurrent ANNs. At the same time choice of a particular architecture of an ANN remains a state-of-the-art technique and depends significantly upon the application. The state-of-the-art artificial neural networks are specified by the sparsity of weights, depth, width for each layer and activation functions (this combination is further referred to as *architecture* of ANN). A modern researcher has to spend a significant amount of time to find a suitable architecture for the problem at hand.

Now consider a model of a type (2.10)-(2.11) with $F_j(\boldsymbol{x}_i|\boldsymbol{\omega_j}), j \in \{1, ...q\}$ associated to all possible architectures of ANNs of depth up to L_{max}, based on a set of available activation functions and weights. If one could fit such a GAM, the state-of-the-art techniques will no longer be required to get a powerful universal approximator for *any* regression or classification problem. Yet, this represents a complicated problem inducing numerous mathematical challenges. The deep Bayesian regression models, addressed in Paper III of this thesis, are designed to resolve these challenges efficiently. Additional discussion to the topic is given in Section 6.1 of this thesis.

Figure 2.2: A style-transferred picture, originally taken by the camera of my cellphone on my mother's 67th birthday (January 25, 2017) from the window of room 824 (before renovation) in Niels Henrik Abel's house, where the major part of work on this thesis was performed. The style-transfer was done using the design of the neural network described by Gatys et al. (2015) implemented in Morugin (2015).

3 Statistical inference

In this chapter some differences between Bayesian and frequentist inference from a mathematical point of view will be discussed, and some popular Bayesian and frequentist approaches for inference will be addressed. The notation in this chapter is based on an abstraction of a general model $m(\boldsymbol{\theta}, \mathcal{D})$ (or simply m).

3.1 The frequentist paradigm

From the frequentist point of view, one is typically interested in point estimates of a parameter $\boldsymbol{\theta}$ based on the model m and data $\mathcal{D}$. The parameter belongs to a certain parameter space Θ, i.e. $\boldsymbol{\theta} \in \Theta$. Then based on the obtained estimates $\hat{\boldsymbol{\theta}}$ one is trying to make inference on model m. In particular, people are typically interested in not only the estimates themselves, but also in getting confidence intervals and confidence distributions for the parameter of interest, and in the predictions for the unobserved data, based on the estimated values of $\boldsymbol{\theta}$ for the model $m(\boldsymbol{\theta}, \mathcal{D})$. Different estimators can be used to obtain estimates of $\boldsymbol{\theta}$. By an estimator here one considers a function $A(\mathcal{D}) : \mathcal{R}^{(p+1)\times n} \rightarrow \mathcal{R}^k$. The most common estimators include the maximum likelihood estimator (MLE) (Davidson et al., 1993), the generalized method of moments estimator (GMM) (Hall, 2005), the generalized least squared error estimator (GLSE) (Davidson et al., 1993; Maddala and Lahiri, 1992), and the minimal divergence estimator (MD) (Basu et al., 1998). Different estimators have different fundamental properties (Hansen, 1982; Self and Liang, 1987), which may drive the choice of them in practical applications. These properties typically include consistency, strong consistency, unbiasedness, asymptotic unbiasedness, efficiency, asymptotic efficiency, and asymptotic normality. In particular, one can say that an estimate $\hat{\boldsymbol{\theta}} = A(\mathcal{D})$ is *consistent* if $\hat{\boldsymbol{\theta}} \overset{\text{Pr}}{\rightarrow} \boldsymbol{\theta}_0$; *strongly consistent* if $\hat{\boldsymbol{\theta}} \overset{a.s.}{\rightarrow} \boldsymbol{\theta}_0$; *unbiased* if $E[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}_0, \forall n \geq 1$; *asymptotically unbiased* if $E[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0] \underset{n\rightarrow\infty}{\rightarrow} \mathbf{0}, \forall n \geq 1$; *asymptotically normal* if $\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \underset{n\rightarrow\infty}{\overset{d}{\rightarrow}} N_k(\mathbf{0}, \Sigma)$; *efficient* if $\hat{\boldsymbol{\theta}}$ is unbiased and $\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}\in\Theta}(|\Sigma_{\hat{\boldsymbol{\theta}}}|)$; *asymptotically efficient* if its asymptotic covariance matrix is a lower bound for all consistent asymptotically normal estimators. Some most widely used methods for constructing estimators are briefly described in the following sections.

Maximum likelihood estimators

The maximum likelihood (ML) (Davidson et al., 1993) method estimates the parameters by maximizing the joint likelihood of the data given the parameters $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ as a function of a parameter $\boldsymbol{\theta} \in \Theta$, which is denoted as $p(\mathcal{D}|\boldsymbol{\theta})$. Hence the maximum likelihood estimate, $\hat{\boldsymbol{\theta}}_{ML}$, is given by

$$\hat{\boldsymbol{\theta}}_{ML} = \text{argmax}_{\boldsymbol{\theta}\in\Theta} p(\mathcal{D}|\boldsymbol{\theta}). \tag{3.1}$$

For analytically and numerically simplicity one usually maximizes the logarithm of the likelihood, which is equivalent to the problem (3.1), since the logarithm is a monotonously increasing function. If analytical maximization is not possible one usually applies gradient descent based numerical optimization routines like Newton's method, Newton-Raphson method or Robbins-Monro method (Robbins and Monro, 1951). Under some regularity conditions maximum likelihood estimators are consistent, asymptotically normal, asymptotically efficient, however they might be biased. This method is by far the most popular under the frequentist paradigm and is used in thousands of applications, including econometrics (Cramer, 1989), biology (Yang, 2007), medical imaging (Shepp and Vardi, 1982), physics (Banaszek, 1998), and even linguistics (Pagel, 2000).

Generalized method of moments

The generalized method of moments (GMM) (Hall, 2005) is a general method for estimating parameters in statistical models. Normally, it is used in the context of semi-parametric models, where the parameter $\boldsymbol{\theta}$ of interest is finite-dimensional and the full shape of the distribution function of the data is not known. In such cases standard maximum likelihood estimation is not applicable. The method requires a number of moment conditions to be available for the model $m(\boldsymbol{\theta}, \mathcal{D})$. The moment conditions are functions of the model parameters and the data. Expectation of these functions take a zero value at the true values of the parameters. The GMM method relies upon minimization of a norm of sample averages of these moment conditions. The method works as follows. Suppose the first k moments of the model $m(\boldsymbol{\theta}, \mathcal{D})$ exist and are defined as the integrals over the whole data $\mathcal{X}$, $\mu_i(\boldsymbol{\theta}) = \int_{\mathcal{X}} f_i(x, \boldsymbol{\theta}) p(x|\boldsymbol{\theta}) \mathrm{d}x, i \in \{1, ..., k\}$, where $f_i(x, \boldsymbol{\theta})$ is the appropriate link function, such that $\mu_i(\boldsymbol{\theta})$ takes a value of zero at the true value of the parameter $\boldsymbol{\theta}_0$. Strongly consistent numerical estimates of the moments can be constructed as $\hat{\mu}_i(\boldsymbol{\theta}, \mathcal{D}) = \frac{1}{n} \sum_{j=1}^{n} f_i(\boldsymbol{d}_j, \boldsymbol{\theta})$, where $\boldsymbol{d}_j = \{y_j, x_{j1}, ..., x_{jp}\}$. Then one resolves numerically (using Newton's or Newton-Raphson methods) or analytically the system of equations $\hat{\mu}_i(\boldsymbol{\theta}, \mathcal{D}) = \mu_i(\boldsymbol{\theta}), i \in \{1, ..., k\}$ with respect to $\boldsymbol{\theta}$ to obtain $\hat{\boldsymbol{\theta}}_{GMM}$. Under some extra regularity conditions the generalized method of moments estimators are consistent, asymptotically normal, and efficient in the class of all estimators that do not use any extra information aside from those contained in the moment conditions. Applications of the generalized method of moments include, for example, finance (Ferson et al., 1994), hydrology (Kitanidis, 1988), and material science (Frenklach and Harris, 1987).

Minimal divergence estimators

Minimal divergence estimators (MD) (Basu et al., 1998) in general rely upon the following idea. Let φ be a proper closed convex function from $(-\infty, +\infty)$ to $[0, +\infty)$ with $\varphi(1) = 0$ and such that its domain $\Omega_\varphi := \{x \in \mathcal{R}$ such that $\varphi(x) < \infty\}$ is an interval with endpoints $a_\varphi < 1 < b_\varphi$ (which may be finite or infinite). For two measures P_α and P_θ, the φ-divergence is defined by

$$\phi(\alpha, \theta) := \int_{\mathcal{X}} \varphi \left(\frac{dP_\alpha}{dP_\theta}(x) \right) \, dP_\theta(x).$$

The basic property of $\varphi-$ divergences states that when φ is strictly convex on a neighborhood of $x = 1$, then

$$\phi(\alpha, \theta) = 0 \text{ if and only if } \alpha = \theta.$$

One can refer to Liese and Vajda (2007) for a complete study of those properties. Generally speaking, $\phi(\alpha, \theta)$ and $\phi(\theta, \alpha)$ are not equal. Hence, φ-divergences usually are not distances, but rather represent some difference between two measures. One of the most important properties of divergences between distributions of random variables is the invariance property with respect to a common smooth change of variables, which is also the case in MLE estimators. Among the most popular distance measures satisfying this property one can mention the Kullback-Leibler (KL), modified Kullback-Leibler (KL_m), χ^2, modified χ^2 (χ^2_m), Hellinger (H), and L_1 divergences, which are respectively associated to the convex functions $\varphi(x) = x \log x - x + 1$, $\varphi(x) = -\log x + x - 1$, $\varphi(x) = \frac{1}{2}(x-1)^2$, $\varphi(x) = \frac{1}{2}(x-1)^2/x$, $\varphi(x) = 2(\sqrt{x}-1)^2$ and $\varphi(x) = |x-1|$. All these divergences except the L_1 belong to the class of "power divergences" (Liese and Vajda, 2007), originally defined in Rényi (1961). Different properties of an $\hat{\boldsymbol{\theta}}_{MD}$ are dependent on the φ measures addressed. For example, for the (KL) divergence the estimator coincides with $\hat{\boldsymbol{\theta}}_{ML}$ and hence is consistent, asymptotically normal, and asymptotically efficient. Minimal divergence estimators also have a broad list of applications, including physics (Naudts, 2004) and econometrics (Ullah, 1996).

Generalized least squared estimators

In generalized least squared estimators (Davidson et al., 1993; Maddala and Lahiri, 1992) one minimizes some distance between an estimator $A(\mathcal{D})$ and the objective parameter function $\tau(\boldsymbol{\theta})$. One could think of different measures for this distance. For instance, one could consider the probability that the estimator is close to the objective parameter function, or one could use an average measure of closeness like the mean absolute deviation, but it is usually mathematically more convenient to consider an average squared deviation, the mean squared error (MSE), namely $E\left[(A(\mathcal{D}) - \tau(\boldsymbol{\theta}))^2\right]$, which can be evaluated as $\hat{E}\left[(A(\mathcal{D}) - \tau(\boldsymbol{\theta}))^2\right] = \frac{1}{n}\sum_{i=1}^{n}(A(\boldsymbol{d}_i) - \tau(\boldsymbol{\theta}))^2$. The estimator $A(\mathcal{D})$ minimizing this quantity is called the generalized least squared estimator and is denoted as $\hat{\boldsymbol{\theta}}_{GLS}$. Properties of generalized least squared estimators depend on $\tau(\boldsymbol{\theta})$ and the distance functions. The standard least squared estimator, coinciding with minimization of the mean squared distance between the parametric means of the observations and the observations themselves, namely $\hat{\boldsymbol{\theta}}_{LS}$, is unbiased and efficient. If, in addition, the likelihood model is Gaussian, then simple least squares estimators are also ML estimators, yielding additionally properties of the latter. Applications of generalized least squared estimators include, for instance, physics (Wahba, 1965), bioinformatics (Kim et al., 2004), and marketing (Srinivasan and Mason, 1986).

3.2 The Bayesian paradigm

From the perspective of the Bayesian paradigm, inference on the defined $m(\boldsymbol{\theta}, \mathcal{D})$ is mainly based upon posterior probabilities of parameters of interest $\boldsymbol{\theta}$ (Box and Tiao, 2011). In order to derive the posterior one has to define the prior believes, incorporated via a prior distribution $p(\boldsymbol{\theta})$. From the orthodox Bayesian standpoint, the prior distribution has to be specified without

access to the actual data by means of taking into consideration problem based domain only. However there have been numerous debates upon this and a significant part of Bayesian statisticians use informative priors for both obtaining better inference and reducing computational complexity of obtaining posteriors of interest (Gelman, 2009). Many also use improper priors, corresponding to prior penalties, for which there does not exist a normalizing constant (Wahba, 1978). Improper priors can also be either informative or not. In order to get to the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ of interest one applies Bayes theorem:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}, \tag{3.2}$$

where $p(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood under model $m(\boldsymbol{\theta}, \mathcal{D})$, and $p(\mathcal{D})$ is the normalizing constant. Note that improper priors do not necessarily lead to improper posteriors and hence improper priors that lead to existence of finite normalizing constant $p(\mathcal{D})$ can be used alongside with the proper priors. In order to get posterior estimates minimizing a certain loss function, the parameters are obtained by means of minimizing this loss under the obtained posterior distribution of the vector of parameters $\boldsymbol{\theta}$. The most widely addressed Bayesian point estimates include posterior mean, median, and mode. These estimates correspond to using L_2, L_1 and L_0 loss functions, respectively. Bayesian analogues of confidence intervals are credibility intervals, which are build upon the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ to evaluate the uncertainty of $\boldsymbol{\theta}$.

The computational effort for obtaining posterior in the form of equation (3.2) depends on the total complexity of the model $m(\boldsymbol{\theta}, \mathcal{D})$. By using a conjugate prior $p(\boldsymbol{\theta})$ to the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$, an explicit analytical formula can be obtained for the posterior. Unfortunately this is rarely possible in practice, hence calculation or approximations of the posterior should be done by means of implementing numerical procedures. Computing $p(\mathcal{D}) = \int_\Theta p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ is often infeasible by exact integration for models with relatively high dimension of the parameter space Θ. Fortunately $p(\mathcal{D})$ is a constant with respect to the unknown parameters of interest, and there exist methods for estimating the posterior distribution without evaluating $p(\mathcal{D})$. The most popular approaches to estimating the posterior include Markov chain Monte Carlo (MCMC) algorithms, such as Metropolis-Hastings algorithm and Gibbs sampler, integrated nested Laplace approximations (INLA), approximate Bayesian computation (ABC), and variational Bayes (VB). These approaches will be briefly described in the following sections.

Markov Chain Monte Carlo

The most widely addressed approaches for obtaining posterior distributions are based upon Markov chain Monte Carlo (MCMC) methods (Robert and Casella, 2005). One typically uses either a Gibbs sampler or a Metropolis-Hastings algorithm. It is also possible to combine them by using a Gibbs sampler with Metropolis-Hastings steps. Let $p(\boldsymbol{\theta}|\mathcal{D})$ be the target distribution of interest, and $\boldsymbol{\theta} \in \mathcal{R}^k$, consisting of $l \leq k$ non-overlapping components of different dimension sizes, $\boldsymbol{\theta} = (\theta_1, ..., \theta_l)$. The Gibbs sampler iterates trough the components of $\boldsymbol{\theta}$ and samples from the corresponding full conditional distributions $p(\theta_j|\boldsymbol{\theta}_{-j}, \mathcal{D})$. For each iteration, $u = 1, ..., W$,

each $\theta_j^{(u)}$ is sampled from

$$p(\theta_j^{(u)}|\boldsymbol{\theta}_{-j}^{(u)}, \mathcal{D}), \tag{3.3}$$

where $\boldsymbol{\theta}_{-j}^{(u)}$ represents the adjunction $\theta_j^{(u)}$ to $\boldsymbol{\theta}^{(u)}$:

$$\boldsymbol{\theta}_{-j}^{(u)} = (\theta_1^{(u)}, ..., \theta_{j-1}^{(u)}, \theta_{j+1}^{(u-1)} ..., \theta_l^{(u-1)}). \tag{3.4}$$

The Gibbs sampler requires the availability of the posterior conditional distributions defined in (3.3). When the full conditional distribution is not available for some components of the parameter, a Metropolis-Hastings step can be used for sampling from them. For making a Metropolis-Hastings step, a proposal for a new sample, $\theta_j^{*(u)}$ corresponding to θ_j is drawn from a proposal distribution $\theta_j^{*(u)} \sim q(\theta_j^{*(u)}|\theta_j^{(u-1)})$ of arbitrary form (proposals can also depend on more than one previous state, as shown in Storvik (2011)). The proposed value $\theta_j^{*(u)}$ is accepted as $\theta_j^{(u)}$ with probability

$$\alpha(\theta_j^{*(u)}|\theta_{-j}^{(u-1)}, \mathcal{D}) = \min\left(1, \frac{p(\theta_j^{*(u)}|\boldsymbol{\theta}_{-j}^{(u-1)}, \mathcal{D}) q(\theta_j^{(u-1)}|\theta_j^{*(u)})}{p(\theta_j^{(u-1)}|\boldsymbol{\theta}_{-j}^{(u-1)}, \mathcal{D}) q(\theta_j^{*(u)}|\theta_j^{(u-1)})}\right). \tag{3.5}$$

Otherwise $\theta_j^{(u)} = \theta_j^{(u-1)}$.

For high-dimensional problems, when the likelihood function is often non-concave, the exploration of the posterior distribution can be extremely slow due to both dimensionality curse and getting stuck at the local extrema for a long time. Numerous approaches are suggested for addressing these issues. For example, the Mode Jumping MCMC (Tjelmeland and Hegstad, 1999) algorithm introduces a valid MCMC algorithm capable of jumping between local extrema, the Metropolis adjusted Langevin algorithm (MALA) (Roberts and Stramer, 2002) utilizes the gradient of the posterior for fast exploration, and RMALA (Girolami and Calderhead, 2011) utilizes a Riemann manifold in the MALA algorithm for a dynamic selection of the step sizes of the proposals. Multiple try MCMC methods with local optimization are described by Liu et al. (2000), while Yeh et al. (2012) propose local annealing approaches. These methods fall into the category of generating auxiliary states for proposals (Storvik, 2011; Chopin et al., 2013). Even though MCMC algorithms are very general and can be applied to numerous problems, adapting an MCMC approach to a particular high-dimensional problem often requires some advanced techniques. Examples of such techniques are presented in Papers I and III of this thesis.

Variational Bayes

Variational Bayes (VB) (Jordan et al., 1999) is another well known approach to approximate intractable posterior distributions $p(\boldsymbol{\theta}|\mathcal{D})$. As mentioned above, traditional Markov Chain Monte Carlo (MCMC) algorithms draw samples from a discrete-time Markov chain, whose stationary distribution is the target distribution. They often face scalability problems for high-dimensional data. In contrast to MCMC, variational Bayes tackles the problem from the Kullback-Leibler

divergence KL $(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}))$ minimization standpoint, where $q(\boldsymbol{\theta})$ comes from a class of analytically tractable distributions for the problem at hand, referred to as the variational family. For the popular mean-field approximation, the vector of parameters is divided into sub vectors. Similarly to the Gibbs sampler case, $\boldsymbol{\theta} = (\theta_1, ..., \theta_l)$, with $l \leq k$. The variational distribution is assumed to be independent across sub-vectors, i.e. it factorizes as $q(\boldsymbol{\theta}) = \prod_{j=1}^{l} q_j(\theta_j)$. The distribution $p^*(\boldsymbol{\theta}|\mathcal{D})$ in the variational family, which is the closest to the target distribution according to the Kullback-Leibler (KL) divergence is then used to approximate the target distribution of interest. Hence, for the case addressed above $p^*(\boldsymbol{\theta}|\mathcal{D}) = \prod_{j=1}^{l} q_j^*(\theta_j)$. Let $E_{q_{-j}}\left[\log\left(p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\right)\right] = \int_{\boldsymbol{\theta}_{-j}\in\Theta_{-j}} \log\left(p(\boldsymbol{\theta}|\mathcal{D})p(\mathcal{D})\right) \prod_{i\neq j} q_i(\theta_i) \mathrm{d}\boldsymbol{\theta}_{-j}$ denote the expectation with respect to all terms except θ_j, then

$$q_j^*(\theta_j) \propto \exp\left(E_{q_{-j}}\left[\log\left(p(\boldsymbol{\theta}|\mathcal{D})p(\mathcal{D})\right)\right]\right) \tag{3.6}$$

Thus, the optimal solution $p^*(\boldsymbol{\theta}|\mathcal{D})$ directly depends on data $\mathcal{D}$, the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. In case $q_j^*(\theta)$ do not have defined forms, an expectation maximization (EM) like recursive algorithm that utilizes coordinate descent or alternating minimization techniques can be applied for the given optimization problem. This algorithm has a guarantee of convergence to a local extremum. However, it does not guarantee convergence to a global extremum for an arbitrary likelihood function. Factorization here is crucial for the final results. A key role is played by the number of factors and precise subdivision. In particular, the independence assumption should be validated when performing factorization in order to not face underestimating of variability issues, which are common for the VB approach. At the same time, one should bear in mind the trade off between the number of factors and the computational complexity. Here fewer factors lead to significantly better results in the presence of dependence between the sub-vectors. On the other hand, such an approach is more computationally demanding when it comes to optimization.

Variational inference has various applications in latent variable models such as mixture models (Humphreys and Titterington, 2000), hidden Markov models (MacKay, 1997), graphical models (Attias, 2000), and, most recently, deep Bayesian neural networks (Graves, 2011). Due to the fast convergence properties of the variational objective, variational inference algorithms are typically orders of magnitude faster in high-dimensional problems than MCMC algorithms (Ahmed et al., 2012). Additionally, efficient subsampling techniques are suggested (Gal, 2016), making the application of variational Bayes feasible for large in terms of the number of observations data samples. At the same time, a general statistical theory qualifying the statistical properties of a variational solution is not well developed yet. Recently Alquier et al. (2016) and Yang et al. (2017) introduced a modified objective function, allowing for an inverse-temperature parameter. They also obtained some general guarantees for the variational solution under this modified objective function.

Approximate Bayesian computation

For the models, for which it is impossible or impractical to apply a likelihood function of data given model parameters, Approximate Bayesian Computation (ABC) (Marin et al., 2012) can be applied. The aim is to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ without using a likelihood

function $p(\mathcal{D}|\boldsymbol{\theta})$. This is achieved by generating observation from the prior and comparing them to the actual data using a distance metric. The latter is important for situations, in which the dimension k of the vector of parameters $\boldsymbol{\theta}$ is large compared to the sample size n, and when there is for some reason no explicit form of the likelihood function. Due to this property, ABC has recently been gaining popularity in various applications including cosmology, genetics, and ecology (Beaumont, 2010; Beaumont et al., 2002; Weyant et al., 2013).

In the most general ABC approach, a sample from the prior $p(\boldsymbol{\theta})$ is drawn. Then new data $\mathcal{D}'$ with respect to the sampled parameters is generated from a given model. The sampled parameter vector $\boldsymbol{\theta}'$ is accepted into the posterior if its distance metric $\rho\left(\tau(\mathcal{D}), \tau(\mathcal{D}')\right)$ based on the summary statistics τ of the original data $\mathcal{D}$ and sampled data $\mathcal{D}'$ is lower than some chosen threshold $\epsilon > 0$. The choice of ϵ is crucial for ABC. On one hand, for too small values of ϵ the acceptance rate will be low, and a large number of runs will be required. On the other hand, if ϵ is too large, then the approximated posterior will be much broader than the true posterior, giving in the extreme case no information at all.

To reduce the computational cost of ABC, it is important to minimize the number of simulations. Several algorithms have been suggested for accelerating the ABC approach. One of the major suggestions is to combine ABC and standard Monte Carlo Markov chain (MCMC) algorithms (Marin et al., 2012), sequential Monte Carlo (Sisson et al., 2007) or population Monte Carlo (Akeret et al., 2015). These methods explore the parameter space more efficiently than the basic ABC algorithm. At the same time, they use the available information in a limited way, as the choice of a new point is only informed by the previous one by the Markov property of the chains. It is often the case that every simulation sample comes at a high computational cost and hence an efficient method would aim to utilize the whole history when accepting or rejecting samples. To sum up, ABC becomes a reasonable approach in the cases when the tractable form of the likelihood function $p(\mathcal{D}|\boldsymbol{\theta})$ is not available. Otherwise other approaches are more suitable. For more details on ABC, see Csilléry et al. (2010).

Another interesting remark on ABC is its similarity to the generalized fiducial inference (Hannig, 2009). The fiducial paradigm is an alternative to the Bayesian and frequentist fundamental approaches for inference on a parameter of interest. This approach provides the fiducial distribution of parameters of interest without assuming any prior. Hence, it tries to resolve the subjectiveness of the choice of priors in Bayesian statistics, making it similar to the objective Bayes approaches (Berger et al., 2006). Even though fiducial inference is an active area of research in the modern statistical science, it will not be discussed in more detail in this dissertation. It also has to be mentioned that both fiducial inference and ABC seem to be partially reinvented in the machine learning community in what is called generative models (Jaakkola and Haussler, 1999).

Integrated nested Laplace approximations

Within hierarchical models with latent Gaussian structures, integrated nested Laplace approximations (INLA) for efficient inference on the posterior distribution (Rue et al., 2009) can be used. Consider $\boldsymbol{\theta} = (\boldsymbol{\eta}, \boldsymbol{z})$, where $\boldsymbol{z}$ is the vector of all the latent Gaussian variables and $\boldsymbol{\eta}$ is the vector of hyperparameters of $\boldsymbol{z}$. In the regression settings, typically, the vector of slope co-

efficients $\boldsymbol{\beta}$ is a part of $\boldsymbol{z}$ allowing to make $\boldsymbol{\eta}$ low-dimensional, which is important to facilitate computations. The INLA approach is based on two steps. First the marginal posterior of the hyperparameters is approximated by

$$p(\boldsymbol{\eta}|\mathcal{D}) \propto \frac{p(\boldsymbol{z},\boldsymbol{\eta},\mathcal{D})}{p(\boldsymbol{z}|\boldsymbol{\eta},\mathcal{D})} \approx \frac{p(\boldsymbol{z},\boldsymbol{\eta},\mathcal{D})}{\tilde{p}_G(\boldsymbol{z}|\boldsymbol{\eta},\mathcal{D})}\Big|_{\boldsymbol{z}=\boldsymbol{z}^*(\boldsymbol{\eta})}. \tag{3.7}$$

Here $\tilde{p}_G(\boldsymbol{z}|\boldsymbol{\eta},\mathcal{D})$ is the Gaussian approximation of $p(\boldsymbol{z}|\boldsymbol{\eta},\mathcal{D})$, and $\boldsymbol{z}^*(\boldsymbol{\eta})$ is the mode of the distribution $p(\boldsymbol{z}|\boldsymbol{\eta},\mathcal{D})$. The posterior mode of the hyperparameters is found by maximizing the corresponding Laplace approximation by some gradient descent method (like for example the Newton-Raphson routine). Then an area with relatively high posterior density of the hyperparameters is explored with either some grid based procedure or variational Bayes.

The second step involves the approximation of the latent variables for every set of the explored hyperparameters. Here computation complexity of the approximation depends on the likelihood type for the data $\mathcal{D}$. If it is Gaussian, the posterior of the latent variables is Gaussian, and the approximation is exact and fully tractable. In the case the likelihood is skewed or heavy tails are present, a Gaussian approximation of the latent variables tends to become inaccurate and another Laplace approximation should be used,

$$\tilde{p}_{\text{LA}}(z_i|\boldsymbol{\eta},\mathcal{D}) \propto \frac{p(\boldsymbol{z},\boldsymbol{\eta},\mathcal{D})}{\tilde{p}_{\text{GG}}(\boldsymbol{z}_{-i}|z_i,\boldsymbol{\eta},\mathcal{D})}\Big|_{\boldsymbol{z}_{-i}=\boldsymbol{z}^*_{-i}(z_i,\boldsymbol{\eta})}. \tag{3.8}$$

Here, $\tilde{p}_{\text{GG}}$ is the Gaussian approximation to $p(\boldsymbol{z}_{-i}|z_i,\boldsymbol{\eta},\mathcal{D})$ and $\boldsymbol{z}^*_{-i}(z_i,\boldsymbol{\eta})$ is the corresponding posterior mode. The full Laplace approximation of the latent fields defined in equation (3.8) is rather time consuming, hence lower order Laplace approximations are often used instead. Once the posterior distribution of the latent variables given the hyperparameters is approximated, the uncertainty in the hyperparameters can be marginalized out (Rue et al., 2009);

$$\tilde{p}(z_i|\mathcal{D}) = \sum_k \tilde{p}_{\text{LA}}(z_i|\boldsymbol{\eta}_k,\mathcal{D})\tilde{p}(\boldsymbol{\eta}_k|\mathcal{D})\Delta_k, \tag{3.9}$$

where Δ_k is the area weight corresponding to the grid exploration of the posterior distribution of the hyperparameters. INLA methodology is particularly important for this thesis, since INLA based marginal likelihoods are used in some of the examples of Papers I and III, whilst its accuracy is discussed in Paper V.

4 Model selection and validation

For most of applications, several models $m_1, ...m_M \in \Omega$ can be considered simultaneously, defining a meta level consisting of a set of possible models, also called an ensemble of all possible models or a model space. Typically, either only one of them is selected or all models get some mass in the probability distributions of different models from the defined model space. Furthermore, the purpose of statistical modeling is to give insight through data, and it is important to give an objective scientific reasoning of whether the insight is achieved. Hence the criteria for good models should rely upon their capability of describing well the observed data (being as close to the "true" model as possible and/or giving high predictive power) and being not too complex. Numerous model selection criteria have been suggested in the literature. Here a short critical overview of the most popular of them will be given.

4.1 Model selection criteria

In a traditional statistical model selection setting, criteria take care of the trade-off between the goodness of fit of the model and the complexity of the model. Different model selection criteria use different fit measures and penalties for the complexities. The details on some of them are described further.

AIC and BIC

The most popular model selection criteria are the *Akaike information criterion* **(AIC)** and the *Bayesian information criterion* **(BIC)** (Gelman et al., 2014). AIC is based on the trade-off between the log likelihood of the model at the point MLE estimate of the parameters and the number of parameters in the model. Mathematically this is expressed as

$$\mathsf{AIC} = -2 \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}_{ML}) + 2k, \tag{4.1}$$

where k is the dimensionality of the vector of parameters of the model. In linear models, where ML estimators can be applied for parameter estimation, it works well. However, in cases beyond linear models, k cannot simply be computed. k can then be evaluated as the effective number of parameters, which is an approximation to the number of 'unconstrained' parameters, such that a parameter is counted if it is estimated with no constraints or prior information, and is not counted if it is fully constrained or all the information about the parameter comes from the prior. Penalties based on the effective number of parameters are used in WAIC and DIC, described below in this section.

The BIC criterion is also based on a penalized log-likelihood function at the ML estimate, but

with a different penalty term. It is defined as

$$\mathsf{BIC} = -2\log p(\mathcal{D}|\hat{\boldsymbol{\theta}}_{ML}) + k\log n, \tag{4.2}$$

where k is again the number of parameters of the model and n is the sample size. Just like AIC, BIC can not handle complex collections of models present in high-dimensional variable selection (or feature selection). Whilst AIC is more suitable for predictions, BIC has the property of being consistent, meaning that it is able to capture almost surely the true model when n goes to infinity, under the condition that linear models with fixed numbers of parameters are addressed and that the true model is within the model space and does not coincide with the null model.

WAIC and DIC

In a fully Bayesian context the traditional model selection criteria, described above, often become biased. To deal with this issue, Bayesian equivalents like the *Deviance Information Criterion* **(DIC)** and the *Watanabe-Akaike Information Criterion* **(WAIC)** have been developed. According to Gelman et al. (2014), WAIC provides a good measure for both fit of the existing data with a proper penalty on the complexity. Unlike AIC and BIC, WAIC has the property of averaging over the posterior distribution of the parameters rather than conditioning on point estimate. This is especially relevant for prediction purpose, since WAIC is evaluating the predictions based on the unobserved new data in a Bayesian context. WAIC is a fully Bayesian approach for estimating the so called out of sample expectation, giving a computationally convenient approximation to cross-validation. The aim is estimating the expectation of the joint log posterior predictive density,

$$E[\log p(\widetilde{\mathcal{D}}|\mathcal{D})]. \tag{4.3}$$

where $\mathcal{D}$ and $\widetilde{\mathcal{D}}$ are the old and new data sets, respectively. However, expression (4.3) does not seem computationally feasible for the cases without conditional independence in the sample of posterior predictive observations. To resolve this, an artificial measure $W[\log p(\widetilde{\mathcal{D}}|\mathcal{D})]$ is suggested to estimate the out of sample expectation of the model. This measure is simply the sum of expectations of log point-wise posterior densities,

$$W[\log p(\widetilde{\mathcal{D}}|\mathcal{D})] = \sum_{i=1}^{n} E[\log p(\widetilde{\boldsymbol{d}}_i|\mathcal{D})], \tag{4.4}$$

where $\widetilde{\boldsymbol{d}}_i$ are individual observations from the new data set $\widetilde{\mathcal{D}}$. Note that, under conditional independence of $\widetilde{\boldsymbol{d}}_i|\mathcal{D}, \widetilde{\boldsymbol{d}}_i \in \widetilde{\mathcal{D}}$, equation (4.4) becomes equivalent to equation (4.3). One estimates expression (4.4) by means of first computing the posterior marginalized log likelihood of the data,

$$\sum_{i=1}^{n} \log p(\boldsymbol{d}_i|\mathcal{D}) = \sum_{i=1}^{n} \log \int_{\Theta} p(\boldsymbol{d}_i|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) \mathrm{d}\boldsymbol{\theta}, \tag{4.5}$$

where $\boldsymbol{d}_i$ are observations from the old data set $\mathcal{D}$. The integral in (4.5) is typically evaluated numerically, for example by Laplace approximations or Monte Carlo simulations (comparison of different approaches of approximating the marginal likelihood is given in Paper V of this thesis). Then a penalty, based on the effective number of parameters, in the form of equation

(4.6) or (4.7) is added to equation (4.5) in order to avoid overfitting. In the former choice, the effective number of parameters is estimated as

$$p_{\mathsf{WAIC},1} = 2\sum_{i=1}^{n} \log E_{\boldsymbol{\theta}|\mathcal{D}}[p(\boldsymbol{d}_i|\boldsymbol{\theta})] - 2E_{\boldsymbol{\theta}|\mathcal{D}}[\log p(\boldsymbol{d}_i|\boldsymbol{\theta})], \tag{4.6}$$

whereas in the latter option the variance of individual terms in the log-predictive density scale is addressed,

$$p_{\mathsf{WAIC},2} = \sum_{i=1}^{n} \mathrm{Var}_{\boldsymbol{\theta}|\mathcal{D}}[\log p(\boldsymbol{d}_i|\boldsymbol{\theta})]. \tag{4.7}$$

Thus, WAIC is defined as

$$\mathsf{WAIC}_i = -2\log \widehat{p}(\widetilde{\mathcal{D}}|\mathcal{D}) + 2\widehat{p}_{\mathsf{WAIC},i}, i \in \{1,2\}. \tag{4.8}$$

For Gaussian linear models with large sample sizes, known variances, and uniform prior distribution on the coefficients, $p_{\mathsf{WAIC},1}$ and $p_{\mathsf{WAIC},2}$ become approximately equal to the number of parameters in the model. In general, they act as the estimates of the effective number of parameters.

DIC appears to be the most popular predictive measure of choice in Bayesian applications nowadays, although, unlike WAIC, it relies on a point estimate of the parameter. It can be interpreted as the Bayesian version of AIC, since it replaces the ML estimate of the parameters with the posterior means (PM) in the log likelihood approximations. It also uses the effective number of parameters instead of the exact number k, resulting in a better handling of complex models. Thus, the DIC criterion is defined as

$$\mathsf{DIC}_i = -2\log p(\mathcal{D}|\hat{\boldsymbol{\theta}}_{PM}) + 2\widehat{p}_{\mathsf{DIC},i}, i \in \{1,2\}, \tag{4.9}$$

where $\hat{\boldsymbol{\theta}}_{PM} = E[\boldsymbol{\theta}|\mathcal{D}]$ and $p_{\mathsf{DIC},i}$ are penalties on the efficient number of parameters, defined as

$$p_{\mathsf{DIC},1} = 2\log p(\mathcal{D}|\hat{\boldsymbol{\theta}}_{PM})) - 2E_{\boldsymbol{\theta}|\mathcal{D}}[\log p(\mathcal{D}|\boldsymbol{\theta}))] \tag{4.10}$$

and

$$p_{\mathsf{DIC},2} = 2\mathrm{Var}_{\boldsymbol{\theta}|\mathcal{D}}[\log p(\mathcal{D}|\boldsymbol{\theta})]. \tag{4.11}$$

Here $p_{\mathsf{DIC},1}$ and $p_{\mathsf{DIC},2}$ both give the effective number of parameters in the limit. For linear models with uniform prior distributions, both of these measures for effective sample size reduce to k. $p_{\mathsf{DIC},1}$ is more numerically stable but $p_{\mathsf{DIC},2}$ has the advantage of always being positive

The four criteria addressed in this section are not the only model selection criteria used in practice. In particular, in Paper IV of this thesis mAIC2 and mBIC2 model selection criteria, capable of controlling the false discovery rate, are addressed. Among other criteria one can mention EBIC, mAIC and mBIC (Frommlet et al., 2012).

Cross-validation

Cross-validation investigates the predictive performance of the model by using parts of the data for inference, and parts of the data for comparing predicted values to observations. A K-fold cross-validation procedure divides the data $\mathcal{D}$ into K sets, and iteratively uses each sets as a test sets and the other as train (inference) sets. Ideally, different sets should be used for choosing the model, estimating the parameters and comparing predicted values to observations. In this thesis a 2-fold cross-validation procedure is used in three examples of Paper III.

4.2 Advances in Bayesian model selection

If one is interested in Bayes factors and posterior marginal model probabilities, models have to be differentiated. Hence, to take this into account, the notation of a model m from Ω with the prior probability distribution $p(m)$ is introduced.

The marginal likelihood (MLIK) is a likelihood function in which parameters have been marginalized out. In the context of Bayesian statistics, it may also be referred to as evidence or model evidence. Thus, for a model m one aims at calculating

$$p(\mathcal{D}|m) = p(\mathbf{y}|\mathbf{X}, m) = \int_{\Theta} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, m)p(\boldsymbol{\theta}|m)\mathrm{d}\boldsymbol{\theta}, \forall m \in \Omega. \tag{4.12}$$

Marginalization in equation (4.12) creates an alternative to penalties on the efficient number of parameters, since greater dimensions of $\boldsymbol{\theta}$ correspond not only to greater likelihoods, but typically also to smaller prior probabilities $p(\boldsymbol{\theta})$ in a greater dimensional setting. Note that MLIK can be approximated by the BIC model selection criterion, namely $p(m|\mathcal{D}) \approx \exp(-\frac{BIC_m}{2})$, as discussed in Claeskens et al. (2008).

The Bayes factor (BF) is closely related to marginal likelihoods. It is defined as the fraction of the MLIKs of a pair of models m and m',

$$\mathrm{BF} = \frac{p(\mathcal{D}|m)}{p(\mathcal{D}|m')}, \forall m, m' \in \Omega. \tag{4.13}$$

The Bayes factor is rather sensitive to the choice of the priors on $\boldsymbol{\theta}$, since the priors are integrated out in both numerator and denominator in (3.2). The Bayes factor should thereby be used with caution, and preferably only in the cases, when the user has some knowledge of the prior contributions to the marginal likelihoods.

Finally, the posterior marginal probabilities of the models $p(m|\mathcal{D}), m \in \Omega$, can be used in a Bayesian setting for both model selection and model averaging. These posteriors can be obtained by the Bayes formula as

$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)p(m)}{\sum_{m' \in \Omega} p(\mathcal{D}|m')p(m')}. \tag{4.14}$$

Then model selection is performed by choosing the posterior mode in the space of models, $\text{argmax}_{m\in\Omega} p(m|\mathcal{D})$, which is known as the maximum a posteriori probability criterion (MAP). However, if one is not interested in inference based on model selection, but would rather like to marginalize out all models to make inference on a quantity Δ of interest, then the posterior marginal probabilities can be used,

$$p(\Delta|\mathcal{D}) = \sum_{m\in\Omega} p(\Delta|\mathcal{D}, m)p(m|\mathcal{D}). \tag{4.15}$$

This is also known as Bayesian model averaging. In this thesis this approach is used for both inference and predictions in Papers I-IV.

There are different ways to describe the model space Ω. One can perform selection of the response distributions or link functions, but by far the most common notion to define Ω is induced by the context of variable or feature selection. With no loss of generality in defining Ω, the latter notion will be further addressed in this thesis, when the term model selection is addressed. To cover a broad class of statistical models consider an extended GAM (introduced in Section 2.4) of the form:

$$y_i|\mu_i \sim \mathfrak{f}(y|\mu_i, \phi), \quad i \in \{1, ..., n\}, \tag{4.16}$$

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j F_j(\boldsymbol{x}_i|\boldsymbol{\omega_j}), \tag{4.17}$$

where now in addition to the parameters, addressed in Section 2.4, $\gamma_j \in \{0, 1\}, j \in \{1, ..., q\}$ are defined as the binary indicators switching the corresponding features $F_j(\boldsymbol{x}_i|\boldsymbol{\omega_j})$ on and off. These indicators are defining individual models, $m = (\gamma_1, ..., \gamma_q)$. In this setting there are $|\Omega| = 2^q$ models in the model space Ω. To complete Bayesian specification of the model, priors on $\boldsymbol{\gamma}$ and the vector of parameters of the model $\boldsymbol{\theta} = \{\cup_{j=1}^{q} \boldsymbol{\omega_j}, \beta_0, \phi\}$ (conditionally on $\boldsymbol{\gamma}$) should be defined. There are many ways to specify priors on $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$. The most standard approach is a so called *"spike and slab"* prior, which assumes Bernoulli prior probabilities on the components of $\boldsymbol{\gamma}$, i.e. $p(\gamma_j = 1) = \pi, j \in \{1, ..., q\}$. This means that $\boldsymbol{\omega}_j$ are drawn from the *"slab"* density with probability π and, with probability $1 - \pi$, $\boldsymbol{\omega}_j$ are equal to zero (*"spike"*). The slab density as well as the priors on other parameters depend a lot upon applications (hence are not addressed in this very general introduction). Various choices of priors on both $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}|\boldsymbol{\gamma}$ are considered in Papers I-IV of this thesis.

In order to calculate $p(m|\mathcal{D})$ in the settings (4.16)-(4.17) one has to iterate through the whole model space Ω, which becomes computationally infeasible for large q. One can use MCMC for evaluating these posteriors. The ordinary MCMC estimate is based on a number of MCMC samples $m^{(u)}, u = 1, ..., W$:

$$\widetilde{p}(m|\mathcal{D}) = \frac{\sum_{u=1}^{W} \mathrm{I}(m^{(u)} = m)}{W} \xrightarrow[W\to\infty]{d} p(m|\mathcal{D}), \tag{4.18}$$

where $\mathrm{I}(\cdot)$ is the indicator function. An alternative, named the renormalized model (RM) esti-

mates by Clyde et al. (2011), is

$$\widehat{p}(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)p(m)}{\sum_{m' \in \mathbb{V}} p(\mathcal{D}|m')p(m')} \mathrm{I}(m \in \mathbb{V}), \tag{4.19}$$

where $\mathbb{V} \subseteq \Omega$ is the set of visited models during the search algorithm run (here the search algorithm does not *have* to be a proper MCMC). Although both (4.19) and (4.18) are asymptotically consistent (provided irreducibility of the search algorithm in Ω), (4.19) will often be the preferable estimator since convergence of the MCMC based approximation (4.18) is much slower, see Clyde et al. (2011).

One aims at approximating $p(m|\mathcal{D})$ by means of searching through some subspace $\mathbb{V}$ of Ω making the approximation (4.19) as precise as possible. Models with high values of $p(\mathcal{D}|m)p(m)$ are important to be addressed. This means that modes and near modal values of marginal likelihoods times the prior are particularly important for construction of reasonable $\mathbb{V} \subseteq \Omega$ and missing them can dramatically influence the estimates. Note that these aspects are just as important if the standard MCMC estimate (4.18) is to be used. A main difference is that while for using (4.18) the number of times a specific model is visited is important, for (4.19) it is enough that a model is visited at least once. In this context the denominator of (4.19), which should be as high as possible, becomes an extremely relevant measure for the quality of the search in terms of being able to capture whether the algorithm visits all of the modes, whilst the size of $\mathbb{V}$ should be low (in practice) in order to save computational time. At the same time, in order to guarantee that (4.19) is asymptotically unbiased, the addressed search algorithm has to be irreducible in Ω. Another important aspect of using (4.19) is the flexibility in parallelizing the search strategies. The process can be embarrassingly parallelized into several chains using several computational threads. If one is mainly interested in model probabilities, then equation (4.19) can be directly applied with $\mathbb{V}$ being the set of unique models visited within all runs. The posterior marginal inclusion probability $p(\gamma_j = 1|\mathcal{D})$ can be then approximated by

$$\widehat{p}(\gamma_j = 1|\mathcal{D}) = \sum_{m' \in \mathbb{V}} \mathrm{I}(\gamma_j = 1)\widehat{p}(m'|\mathcal{D}), \tag{4.20}$$

giving a measure for assessing importance of the covariates or features. Other parameters of interest can be estimated similarly.

The arguments described in this paragraph motivate the creation of efficient search algorithms to explore the model space Ω. Existing algorithms for estimating $\mathbb{V}$ are described in Section 4.3, whilst some novel algorithms for this problem are suggested Papers I-IV of this thesis. In practice $p(\mathcal{D}|m)$ may not be available analytically. One then has to rely on some precise approximations $\widehat{p}(\mathcal{D}|m)$. Such approximations introduce additional errors in (4.19) and (4.20), but in some cases they can be assumed to be small enough to be ignored. Computational comparison of various approaches approximating the marginal likelihood is addressed in detail in Paper V of this thesis.

4.3 Search algorithms

Model selection algorithms in the Bayesian settings allow to either search for the optimal model in terms of some of the mentioned above criteria or evaluate posterior model probabilities. A short summary of the algorithms aiming at the latter will be given further. The search algorithms with the goal of getting posterior distribution of models and other parameters of interest typically perform a search in the combined space of models *and* parameters and rely on MCMC. However, a few of them work on the marginal space of models, where parameters are marginalized out. Equation (4.14) can be approximated in an asymptotically unbiased way by means of (4.19) using the search algorithms exploring Ω irreducibly. These algorithms do not even have to be proper MCMC. This is a well known fact used in numerous articles and some of the approaches developed in Papers I-IV of this thesis.

Among the approaches working on the marginal space of models we can mention Bové and Held (2011), who consider an MCMC algorithm within the model space, but only allow local moves. This might be a severe limitation in cases where multiple sparsely located modes are present in the model space. Bivand et al. (2014) combine approximations of marginal likelihood with Bayesian model averaging within spatial models. Clyde et al. (2011) suggest a Bayesian adaptive sampling (BAS) algorithm as an alternative to MCMC allowing for perfect sampling without replacement. Another approach for Bayesian model selection working on the marginal space of models is described by Bottolo et al. (2011, 2010), who propose the moves of MCMC between local optima through a permutation based genetic algorithm that has a pool of solutions in a current generation suggested by parallel tempered chains. Song and Liang (2015) address the case when there is by far more explanatory variables than observations. They suggest a split and merge Bayesian model selection algorithm that first splits the set of covariates into a number of subsets, then finds relevant variables from these subsets and in the second stage merges these relevant variables and performs a new selection from the merged set. In general, this algorithm cannot guarantee convergence to a global optimum or find the true posterior distribution of the models. However, under some strict regularity conditions, it does so asymptotically. An efficient *mode jumping MCMC* for the model space exploration is suggested in Paper I of this thesis. Simulations show its computational competitiveness in comparison to other existing approaches.

There are also algorithms working on the joint space of models and parameters. Some of them are discussed in George and McCulloch (1997), who outline computational methods including Gray Code sequencing and standard MCMC for posterior evaluation and exploration of the space of models. They also comment on the infeasibility of exhaustive exploration of the space of models for moderately large problems as well as the inability of standard MCMC techniques to escape efficiently from local optima. Al-Awadhi et al. (2004) also work in this domain and consider using several MCMC steps within a new model to obtain good proposals within the combined parameter and model domain, while Yeh et al. (2012) propose local annealing approaches. Ghosh (2015) also uses MCMC algorithms to estimate the posterior distribution over models. She observes that the estimates of the posterior probabilities of individual models based on MCMC output are often not reliable because the number of MCMC samples is typically considerably smaller than the size of the model space. As a consequence, she considers

the median probability model of Barbieri et al. (2004) and shows that this algorithm can, under some conditions, outperform standard MCMC.

All of these works address only linear or generalized linear models. However equations (4.16)-(4.17) allow to address much broader classes of models. When these classes of models are addressed, less techniques are available due to the complexity and size of Ω. Some approaches are developed in the context of logic regression, described in Section 2.4. Important contributions to the development of logic regression are made by the group of Katja Ickstadt (Fritsch, 2006; Schwender and Ickstadt, 2008), which also provides a comparison of different implementations of logic regression (Fritsch and Ickstadt, 2007). Schwender and Ruczinski (2010) give a brief introduction with various applications and potential extensions of logic regression. 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. In Paper II a novel algorithmic approach for Bayesian logic regression is developed. The developed algorithm shows great performance in the simulation scenarios (compared to FBLR and MCLR). Proper Bayesian model selection and model averaging in more sophisticated classes of models such as ANN has never been done before (to the best knowledge of the author). Though, there are several heuristic approaches in the non-Bayesian settings, like Lari and Abadeh (2014); Arifovic and Gencay (2001); Zoph et al. (2017). Other recent approaches for the search of an optimal architecture of neural networks include Pham et al. (2018); Elsken et al. (2017). In Paper III of this thesis a novel approach for targeting complicated model classes is developed. Several applied examples show the developed approach to be prominent in both predictive and inferential examples.

5 Summary of papers

5.1 Paper I

In the first paper, Bayesian variable selection and model averaging in generalized linear mixed models (GLMM), discussed in Section 2.3, are addressed. The aim was to develop an efficient search algorithm across different marginal model configurations just as described in Section 4.3. The model configurations are here defined by various combinations of explanatory variables, inducing an NP-hard search problem. In a Bayesian setting, the marginal posterior distribution of the models, based on the observed data, can be viewed as a relevant measure for the model evidence. This distribution consists of discrete posterior probabilities (4.14) of individual models, which are proportional to the products of the model priors and marginal likelihoods of the corresponding models. The marginalization of the parameters simplifies the search significantly, since the algorithm does not have to explore the joint space of models and parameters any longer. At the same time, even in this discrete setting, efficient search algorithms have to be adapted for evaluating posterior distribution within a reasonable amount of time.

In this paper an MCMC algorithm for the search through the model space is suggested. The algorithm deals with its multimodality through mode jumping proposals. It is called mode jumping Markov chain Monte Carlo (MJMCMC). MJMCMC relies upon the idea of making smart moves between the local extrema with a reasonable frequency. In particular, local MCMC is performed in the absolute majority of the steps, while for the rest a large move in the model space (which is likely to hit a very low probability model) is made, followed by a local optimization. The goal of the latter step is to reach the local extremum in the new part of the model space. Then the proposal is randomized around this extremum and the transition to the proposed model is either accepted or rejected. The ergodicity of the suggested Markov chain is proven. Its limiting distribution is shown to correspond to the marginal posterior model probabilities. Further extensions of the algorithm allowing for parallel computing and using mixtures of proposals are suggested.

The performance of the suggested algorithm is compared to several existing approaches, like Random Swap MCMC and Bayesian adaptive sampling (Clyde et al., 2011), on both simulated and real data. The real examples include the famous U.S. crime data, a protein activity data set and an epigenetic data set. The algorithms are evaluated based on various performance measures of accuracy on the posterior probabilities. MJMCMC outperforms the competing approaches in many of the cases.

5.2 Paper II

The second paper addresses a more sophisticated model selection problem in the context of Bayesian logic regression. The logic regression model, described in Section 2.4, was initially

suggested as a tool to construct predictors from Boolean combinations of binary covariates, which means that the relations between the observations and the explanatory variables are not linear, unlike in the problem from the first paper. The number of possible logical expressions is exponential in the number of binary explanatory variables, making the search significantly harder. In fact, it is not possible anymore to specify the model space a priori. Nevertheless, the goal is to approximate posterior probabilities of logical expressions involved in the model by cleverly extending the MJMCMC algorithm.

The idea is to embed MJMCMC into the iterative setting of a genetic algorithm, where the populations are formed by different subsets of all possible logical expressions. For any fixed subset, a well defined model space is present, allowing to run MJMCMC on it. Changes in the population are made in a way to guarantee irreducibility of the algorithm in the model space of all logic regression models, which is required for asymptotic unbiasedness of the estimated posterior probabilities, as discussed in Section 4.2. Finally, an embarrassing parallelization strategy is suggested to speed up the inference by using multiple cores on local machines or clusters. The suggested algorithm is named a Genetically modified mode jumping Monte Carlo Markov chain (GMJMCMC). The GMJMCMC algorithm though is not a proper MCMC algorithm, since its stationary distribution does not coincide with the target distribution of interest. At the same time, renormalized estimates of the posterior probabilities, discussed in Section 4.2, are applicable. A novel model prior for the Bayesian logic regressions is also suggested. This prior allows to achieve a good trade off between power and false discovery rate.

The performance of GMJMCMC is evaluated and compared with other Bayesian approaches for logic regression, namely MCLR (Kooperberg and Ruczinski, 2005) and FBLR (Fritsch, 2006), in 6 advanced simulation scenarios. The simulation studies show great performance of the suggested approach to recover complex logical expressions with high power and low false discovery rate for 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. Finally, GMJMCMC is applied to analyze QTL mapping data for Recombinant Inbred Lines in *Arabidopsis thaliana* and from a backcross population in *Drosophila* where several interesting epistatic effects are identified.

5.3 Paper III

The third paper provides a generalization of the models and the algorithms of those suggested in Paper II. This generalization allows to work with all types of explanatory variables. A model that extends such classes of statistical models as GLM, GLMM, ANN, CART, MARS and fractional polynomials into a powerful and flexible Bayesian framework is suggested. The suggested model is named deep Bayesian regression model (DBRM). DBRM allows, in particular, to construct architectures of Bayesian ANN automatically. Additionally it allows for polynoms of neural networks or CART based on these polynoms (and other interesting mixtures of different known classes of models), giving a lot of flexibility in modeling nonlinearities between the observations and explanatory variables. Under weak regularity conditions DBRM satisfies the universal approximation theorem (Hornik, 1991). Overfitting issues are accurately handled

by using priors, which explicitly penalize complexities of the features and individual models in the DBRM model space. The link to Papers I and II is straightforward: the basic algorithm for fitting DBRM is a modification of GMJMCMC, which is adapted to work in a more flexible model space. In this paper a reversible version of GMJMCMC - RGMJMCMC is also suggested, allowing for the proper MCMC based estimates of posterior probabilities of interest in the space of deep Bayesian regression models.

In the experimental section of this paper properties of the suggested approach are tested. It is shown how deep Bayesian regression models can be used for inference and predictions in various applications. In particular, four examples are devoted to the former (inference) and three - to the latter (prediction). The predictive performance of DBRM is compared with those of various other statistical and machine learning approaches for problems of breast cancer prediction (whether the tissue is cancerous or not), asteroid classification (whether the asteroid is a potentially hazardous object or not) and spam classification (whether the email is a spam letter or not). In all of the studies DBRM performs very well, proving its high predictive ability. In three following examples inference on the data with known ground truth is performed. In particular, the simulation studies include one scenario, in which complex nonlinearities are involved, where DBRM manages to recover the truth with low FDR and high power. Then two ground physical laws (planetary mass law and the 3rd Kepler's law) are addressed. In both cases the laws are recovered in a closed form with high power and low FDR. Finally, a study on epigenetic data, where the underlying truth is not known, is performed. Some potentially interesting relations and the optimal structure of the spacial dependence between the observations are discovered in this example.

5.4 Paper IV

In the fourth paper Bayesian approaches to genome wide association studies (GWAS) are studied. In particular, various Bayesian algorithms (including GMJMCMC and CMJMCMC, which stands for convolutional MJMCMC), developed to search through the class of possible regression models, are compared. GMJMCMC used in this paper coincides with the versions developed in Papers II and III. CMJMCMC is a novel algorithm, which uses a 3 staged principal. First, a subset of all SNPs is preselected with respect to either marginal p-values or marginal correlations with the responses. Then this subset is divided into independent non-intersecting subsets of reasonably small cardinality and MJMCMCs are run until convergence for each of them (second stage). Finally, the best covariates with posteriors above a predefined threshold on the posterior probability are selected and MJMCMC is run until convergence on them (third stage). After two convolutions it is expected that the final search space contains enough good models for the high quality inference. However this greedy convolutional approach does not guarantee asymptotic exploration of the whole model space and hence even asymptotically does not guaranty unbiasedness of the obtained posteriors. The comparison also includes MOSGWA (Gola et al., 2013), an approach previously developed by some coauthors of the paper, and a frequentist benchmark (Bonferroni corrected marginal t-tests). Smart priors on the class of possible models, inducing mAIC2 and mBIC2 model selection criteria, are used to control FDR and FWER. A novel Bayesian method for estimating the proportion of the trait variance

explained by the genetic factors is also proposed. Different Bayesian approaches and a frequentist benchmark are compared on three advanced simulation scenarios. All of the Bayesian approaches significantly outperform the frequentist benchmark. This paper is a practical example of how the approaches developed in Papers I, II, and III can be applied to real world data with a high number (of order 10^4 and larger) of potential regressors. At the moment of submission of this thesis this paper is a work in progress. Some additional work has to be done before being able to submit it. In particular, in addition to the algorithms, described above, it would be of interest to add MOSGWA memetic algorithm (introduced in the paper) to the simulation study. Additionally piMASS approach (Guan and Stephens, 2011) should definitely be included in the comparison to be able to draw more sound and comprehensive conclusions. Recently variational Bayes based approaches (Carbonetto et al., 2012) have been shown efficient for Bayesian variable selection in GWAS and we should consider them as a part of the simulation study too. Finally, some consistency studies, based on real data should be performed. It is planned to address two challenging real data sets. First, the expression data from Stranger et al. (2007) will be reanalyzed for those nine genes, for which analysis with MOSGWA reported by Frommlet et al. (2012) gave models with at least three SNPs. This data set is extremely challenging due to the rather small sample size. As a second example STAMPEED data from the Northern Finland Birth Cohort 1966 (Sabatti et al., 2009) will be considered.

5.5 Paper V

All the approaches used in the first four papers are based on marginal posterior probabilities. To calculate them, one needs to first efficiently obtain the marginal likelihood or its accurate estimate. Note that the marginal likelihood itself is a well established model selection criterion in Bayesian statistics. In many complex models, including latent modeling approaches and Bayesian neural networks, the marginal likelihood is not tractable and, generally speaking, very difficult to compute. Different approximations are available. One recent promising approach for approximating the marginal likelihood is the Integrated Nested Laplace Approximation (INLA).

In this paper the approximation of the marginal likelihood obtained with INLA is compared to some alternative approaches (Laplace approximations, Chib's method (Chib, 1995), Chib and Jeliazkov's method (Chib et al., 1998), harmonic means, etc.) on a number of examples of different complexities. In particular, a simple linear latent model, a Bayesian linear regression model, logistic Bayesian regression models with probit and logit links, and a Poisson longitudinal generalized linear mixed model are considered. It is shown that INLA approach is fast, accurate and robust. Chib's and Chib and Jeliazkov's methods also perform very well but are significantly slower. The standard Laplace approximations are the fastest, but can sometimes be less precise. Nevertheless they become very accurate for large enough samples. Based on the results suggested in this paper, some of the mentioned above approximations of the marginal likelihood can be trusted and hence their use in practice is justified. This paper is a technical report, where important studies for justification of the methods, used in the other papers from this thesis, were performed. However it is not planned to submit it for publication as a separate article.

6 Discussion

In this PhD thesis important problems of Bayesian model selection and model averaging are addressed in various regression contexts. The approaches developed here are all based on the idea of marginalizing out parameters from the likelihood. This allows to work on the marginal space of models, which simplifies the search algorithms significantly. For the linear models an efficient mode jumping Monte Carlo Markov chain algorithm was suggested. The approach performed very well on simulated and real data. Further, the algorithm was extended to work with logic regressions, where one has a feature space consisting of various complicated logical expressions, which makes enumeration of all features computationally and memory infeasible in most of the cases. The genetically modified MJMCMC algorithm was suggested to resolve this issue. The algorithm combines the idea of keeping and updating the populations of highly predictive logical expressions combined with MJMCMC for the efficient exploration of the model space. Several simulation and real data studies show that logical expressions of high orders can be recovered with large power and low false discovery rate, which was not feasible with the existing algorithms for Bayesian logic regression. Moreover, the genetically modified MJMCMC approach (GMJMCMC) is adapted to estimate posterior model probabilities, perform Bayesian model averaging and selection within the class of deep Bayesian regression models (which, as suggested in this thesis, is an extension of various machine and statistical learning models like ANN, CART, logic regressions and GLMM). The reversible GMJMCMC, named RGMJMCMC, is also suggested. It makes transitions between the populations of variables in a way that satisfies the detailed balance equation. Based on several examples, it is shown that the DBRM approach can be efficient for both inference and prediction in various applications. In particular, two ground physical laws were recovered from the data with large power and low FDR. Three classification examples were also studied, where the comparison to other popular machine and statistical learning approaches was performed. Finally, a thorough study comparing different Bayesian approaches to GWAS (including the GMJMCMC based approach) was done. In particular, it was shown that the approaches suggested in the thesis may be successfully applied to data with a huge number of explanatory variables, if accurately tuned. The developed algorithms often require significant resources in terms of computational time, hence embarrassing parallelization of MJMCMC, GMJMCMC and RGMJMCMC was suggested and used in some of the examples. Within this parallelization one runs multiple independent chains of the algorithms, which all explore some (possibly intersecting) parts of the model space. Then the results are merged together using Bayes formula of a type (4.14) with the set of all unique models visited by the algorithms. It is in general recommended to use parallelization when possible. A memory efficient way to perform map and reduce steps within the parallelization is also proposed. The *EMJMCMC* R-package is developed and currently available from the GitHub repository (Hubin, 2018). The developed package gives to the user high flexibility in the choice of the methods to obtain marginal likelihoods and model selection criteria for the class of DBRM models.

6.1 Future work and extensions

There are several important ways to extend the work presented in this thesis. Many of the potential future extensions are described in detail in the discussion sections of the corresponding articles. Theoretical studies of the properties of the approaches suggested in this thesis could be of great interest. For example, proving consistency of deep Bayesian regression model (in terms of being able to recover the data generating model) is of a particular importance. Moreover, several extensions of the models, addressed in this thesis, could be suggested to allow even more flexibility. As mentioned in the discussion section of Paper I (and is not yet resolved), there is a particular interest in extending the approaches from Papers I-III towards automatic search through different probability distributions for the responses, hence further relaxing the model assumptions. Handling missing data is also not yet incorporated in the developed in this thesis approaches. It might be of a particular importance to carry out some research in this direction. However, in this discussion the most important future work from the author's perspective will be described. This work could be build upon the adaptation of the suggested models and algorithms to extremely large sample sizes and numbers of explanatory variables. All the approaches developed in the thesis rely on marginalization of parameters of individual models and working with the marginal space of model configurations. To be able to perform this, one needs to compute the marginal likelihood (4.12), which currently is among the major computational bottlenecks. In fact, the computation of the marginal likelihood for a dataset consisting of millions and billions of observations becomes almost infeasibly slow with the standard methods. At the same time, large data samples do not seem unreasonable any longer. Machine learning techniques have shown to be able to successfully deal with large data sets. For example, stochastic gradient descent, originally introduced as a stochastic approximation method in Robbins and Monro (1951), allows to train deep neural networks (often involving millions and billions of parameters too) on arbitrary large datasets. The success of modern stochastic gradient descent approaches relies upon the extensive use of parallel computing and efficient subsampling techniques. However, subsampling in a proper Bayesian way (with guarantees of the ergodicity of the corresponding Markov chain Monte Carlo) is significantly more demanding from both methodological and computational perspectives. Beaumont (2003); Andrieu and Roberts (2009) have shown that if one replaces $p(\boldsymbol{\theta}|\mathcal{D})$ described in detail in Section 3.2 by its unbiased estimate $\hat{p}(\boldsymbol{\theta}|\mathcal{D}) : E[\hat{p}(\boldsymbol{\theta}|\mathcal{D})] = p(\boldsymbol{\theta}|\mathcal{D})$, then all of the standard asymptotic properties of the standard Metropolis-Hastings algorithm are still valid. The question then is how to construct such $\hat{p}(\boldsymbol{\theta}|\mathcal{D})$ using subsampling techniques. It is very easy to construct the unbiased estimates of $\widehat{\log p}(\boldsymbol{\theta}|\mathcal{D})$ by means of, for example, importance sampling. Unfortunately $\exp\left(\widehat{\log p}(\boldsymbol{\theta}|\mathcal{D})\right)$ is not an unbiased estimate of $p(\boldsymbol{\theta}|\mathcal{D})$, which means that more advanced techniques should be applied.

Subsampling MCMC

There are several attempts to suggest subsampling techniques for MCMC using additional information from the data. These attempts are either approximate (Bardenet et al., 2014, 2017; Korattikara et al., 2014; Quiroz et al., 2014) or exact (Quiroz et al., 2016; Maclaurin and Adams, 2014; Liu et al., 2015). Maclaurin and Adams (2014), for instance, suggest the firefly

MCMC approach. The approach is based upon introducing auxiliary and latent binary indicators $z_i, i \in \{1, ...n\}$ indicating if realization of the random variable corresponding to observation i is considered for calculation of the likelihood in the given MCMC step or not. Consider the notation used in Section 3.2 with conditionally independent observations and extend $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})$ to

$$p(\boldsymbol{\theta}, \boldsymbol{z}|\mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta}) \left[\frac{p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta}) - B_i(\boldsymbol{\theta})}{p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})}\right]^{z_i} \left[\frac{B_i(\boldsymbol{\theta})}{p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})}\right]^{1-z_i} \tag{6.1}$$

where $z_i \in \{0, 1\}$ and $0 \leq B_i(\boldsymbol{\theta}) \leq p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})$. Obviously, in this settings $p(\boldsymbol{\theta}|\mathcal{D})$ is the marginal distribution of $p(\boldsymbol{\theta}, \boldsymbol{z}|\mathcal{D})$. Then a standard MCMC can be performed on the joint space of parameters $\boldsymbol{z}$ and $\boldsymbol{\theta}$, with

$$p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{z}) \propto \prod_{i=1}^{n} \left[p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta}) - B_i(\boldsymbol{\theta})\right]^{z_i} \left[B_i(\boldsymbol{\theta})\right]^{1-z_i}, \tag{6.2}$$

which only requires evaluation of $p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})$ for $z_i = 1$, and

$$p(\boldsymbol{z}|\mathcal{D}, \boldsymbol{\theta}) \propto \prod_{i=1}^{n} \left[\frac{p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta}) - B_i(\boldsymbol{\theta})}{p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})}\right]^{z_i} \left[\frac{B_i(\boldsymbol{\theta})}{p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta})}\right]^{1-z_i}, \tag{6.3}$$

which corresponds to simple binomial sampling. The main computational benefit of this approach is that if $B_i(\boldsymbol{\theta})$ are simple to calculate, it is enough to sample from a (small) fraction of *complex* components corresponding to z_i's at each iteration.

Another prominent subsampling idea is suggested in Welling and Teh (2011), who propose a stochastic gradient MCMC technique. Recall first stochastic gradient optimization (SGO) (Robbins and Monro, 1951), that guarantees convergence to a local extremum of the likelihood:

$$\boldsymbol{\theta}^{(u+1)} = \boldsymbol{\theta}^{(u)} + \frac{\varepsilon^{(u)}}{2} \left(\nabla \log p(\boldsymbol{\theta}^{(u)}) + \frac{n}{m} \sum_{i=1}^{m} \nabla \log p(y_{u,i}|\boldsymbol{x_{u,i}}, \boldsymbol{\theta}^{(u)})\right), \tag{6.4}$$

where one requires the regularity conditions

$$\sum_{u=1}^{\infty} \varepsilon^{(u)} = \infty, \quad \sum_{u=1}^{\infty} {\varepsilon^{(u)}}^2 < \infty \tag{6.5}$$

to be satisfied. Also the likelihood function has to be smooth with respect to the parameters of interest. Recall the standard Langevin dynamics MCMC procedure (Beskos et al., 2008) capable of drawing from the posterior,

$$\boldsymbol{\theta}^{(u+1)} = \boldsymbol{\theta}^{(u)} + \frac{\varepsilon}{2} \left(\nabla \log p(\boldsymbol{\theta}^{(u)}) + \sum_{i=1}^{n} \nabla \log p(y_i|\boldsymbol{x_i}, \boldsymbol{\theta}^{(u)})\right) + \eta^{(u)}, \quad \eta^{(u)} \sim N(0, \varepsilon) \tag{6.6}$$

Welling and Teh (2011) showed that a procedure combining the two approaches mentioned

above is drawing from the posterior distribution of a parameter of interest $\boldsymbol{\theta}$, when the regularity conditions (6.5) are satisfied. This modification is called stochastic gradient Langevin dynamics MCMC and has the following form:

$$\boldsymbol{\theta}^{(u+1)} = \boldsymbol{\theta}^{(u)} + \frac{\varepsilon^{(u)}}{2}\left(\nabla \log p(\boldsymbol{\theta}^{(u)}) + \frac{n}{m}\sum_{i=1}^{m} \nabla \log p(\boldsymbol{y_{u,i}}|\boldsymbol{x_{u,i}}, \boldsymbol{\theta}^{(u)})\right) + \eta^{(u)}, \tag{6.7}$$

where just like in the standard Langevin dynamics MCMC procedure $\eta^{(u)} \sim N(0, \varepsilon^{(u)})$.

Delayed acceptance MCMC

In addition, the delayed acceptance MCMC (Banterle et al., 2014) can be used to speed-up MCMC. It relies upon the following idea. Assume $\boldsymbol{\theta}^{(u)}$ is generated according to the standard MCMC procedure described in Section 3.2. The delayed acceptance MCMC suggests to accept $\boldsymbol{\theta}^{(u)}$ if it both is preliminary accepted with a probability

$$\min\{1, \tfrac{p(\boldsymbol{\theta}^{(u)}|\mathcal{D})}{p(\boldsymbol{\theta}^{(u-1)}|\mathcal{D})}\} \tag{6.8}$$

and is finally accepted with a probability

$$\min\{1, \tfrac{q_r(\boldsymbol{\theta}^{(u-1)}|\boldsymbol{\theta}^{(u)})}{q_r(\boldsymbol{\theta}^{(u)}|\boldsymbol{\theta}^{(u-1)})}\}. \tag{6.9}$$

One rejects the proposal if any of the criteria is not satisfied. This allows to gain computational efficiency when rejecting in the first stage, although in general the total acceptance rate will be smaller than without delayed acceptance (Banterle et al., 2015, remark 1). The delayed acceptance MCMC approach is addressed in detail in Papers I and III of this thesis. Combinations of the delayed acceptance MCMC and subsampling have been also suggested in several works (Quiroz et al., 2017; Payne and Mallick, 2014). However, these approaches are still severely limited because a full data set has to be used for all the accepted transitions in MCMC.

Divide and conquer MCMC

To facilitate computations, divide and conquer MCMC procedures (Neiswanger et al., 2013; Scott et al., 2016; Wang and Dunson, 2013; Li et al., 2017; Minsker et al., 2014) have also been proposed. These methods typically rely upon three steps. The first step is to split the data into a large number of smaller (possibly overlapping) data sets. The second stage is to perform inference on each smaller data set. In the third stage the results are combined together. Within these approaches computation is performed on small data sets, moreover, the second stage can be embarrassingly parallelized. However one typically ends up with inexact results, although some asymptotic results are becoming available. Consensus Monte Carlo algorithm (Scott et al., 2016), for instance, uses the following idea. Assume conditionally independent blocks $\boldsymbol{y_1}, ..., \boldsymbol{y_S}$. This allows to factorize the posterior as

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \prod_{s=1}^{S} p(\boldsymbol{y_s}|\boldsymbol{x_s}, \boldsymbol{\theta}) p(\boldsymbol{\theta})^{1/S}. \tag{6.10}$$

Then one can simulate $\boldsymbol{\theta}_{s,1}, ..., \boldsymbol{\theta}_{s,G}$ from $p_s(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{y_s}|\boldsymbol{x_s}, \boldsymbol{\theta})p(\boldsymbol{\theta})^{1/S}$ and combine the results as $\boldsymbol{\theta}_g = (\sum_s \boldsymbol{W}_s)^{-1} \sum_s \boldsymbol{W}_s \boldsymbol{\theta}_{s,g}$. This algorithm is approximate, but gives exact posteriors if $p_s(\boldsymbol{\theta}|\mathcal{D}), s = 1, ..., S$ are Gaussian and the weights are chosen as $\boldsymbol{W}_s = \text{Var}_{p_s(\boldsymbol{\theta}|\mathcal{D})}[\boldsymbol{\theta}]$. Otherwise the inference is approximate.

Unfortunately, none of the approaches described above is very efficient. Moreover, to the best knowledge of the author, no subsampling techniques in the context of proper Bayesian model selection and model averaging have been suggested, giving potentially interesting directions for the new research.

Variational inference in Bayesian deep learning

Another important extension of the performed work relates to developing methodology for approximating marginal likelihoods across all layers in deep Bayesian regression models (introduced in Paper III) or deep Bayesian neural networks. Consider the ANN model of a form (2.13) with parametric neurons (2.14) described in Section 2.5. Let $\boldsymbol{\theta}$ be a vector of all parameters of this ANN across all layers. Define a prior $p(\boldsymbol{\theta})$ for them. Consider also (for simplicity) ϕ fixed and known (hence excluded from $\boldsymbol{\theta}$). The most straight forward (and computationally feasible) solution to approximate the marginal likelihood (across all layers) in this settings is to rely upon variational approximations obtained by using efficient subsampling techniques, as shown in Gal (2016). Consider Gaussian variational families with independence across weights, resulting in $p(\boldsymbol{\theta}|\mathcal{D}, m) \approx \hat{p}_{\text{VB}}(\boldsymbol{\theta}|\mathcal{D}, m) = q_{\hat{\boldsymbol{\eta}}}(\boldsymbol{\theta})$ with

$$q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) = \prod_{i \in \{1,...,l\}} q_{\boldsymbol{\eta}_i}(\boldsymbol{\theta}_i), \tag{6.11}$$

where $q_{\boldsymbol{\eta}_i}(\boldsymbol{\theta}_i) = N(\mu_{m,i}, \sigma^2_{m,i})$. Here $\boldsymbol{\eta}_i = (\mu_{m,i}, \sigma^2_{m,i})$ and $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_l)$ with l being a total number of variational components. As discussed in Section 3.2, in the VB approach one aims at minimization of Kullback-Leibler divergence between the variational family distribution and the true posterior, namely,

$$\text{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}, m)) = \int_\Theta q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log \frac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathcal{D}, m)} \text{d}\boldsymbol{\theta}. \tag{6.12}$$

The minimization of this divergence is mathematically equivalent to the maximization of the so called *evidence lower bound* (ELBO) with respect to $\boldsymbol{\eta}$, where ELBO is

$$\mathcal{L}_{VI}(\boldsymbol{\eta}) := \int_\Theta q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log p(\mathcal{D}|\boldsymbol{\theta}, m) \text{d}\boldsymbol{\theta} - \text{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})). \tag{6.13}$$

Here ELBO is always smaller or equal than the log marginal likelihood of model m, i.e. $\mathcal{L}_{VI}(\boldsymbol{\eta}) \leq \log p(\mathcal{D}|m)$. The integration is replaced with optimization with respect to the variational parameters $\boldsymbol{\theta}$, exactly as described in Section 3.2. However note that even within this computationally simplified (compared to a proper MCMC approach) case, one requires to compute $\int_\Theta q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log p(\mathcal{D}|\boldsymbol{\theta}, m) d\boldsymbol{\theta}$, which is infeasible for very large samples (it requires calculations over the whole data). Clever subsampling allows to resolve this computational issue. From equation (6.13) and under the assumption of conditional independence of the observations, it

directly follows that

$$\mathcal{L}_{VI}(\boldsymbol{\eta}) = \sum_{i=1}^{N} \int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log p(y_i|\boldsymbol{\theta}, \boldsymbol{x}_i, m) \mathrm{d}\boldsymbol{\theta} - \mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})). \tag{6.14}$$

Here $\int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log p(y_i|\boldsymbol{\theta}, \boldsymbol{x}_i, m) \mathrm{d}\boldsymbol{\theta}$ are still not tractable for the Bayesian neural networks with more than one hidden layer. Additionally one still has to carry out computations over the whole data set, which is computationally expensive for large sample sizes n. The latter issue can be addressed by applying the so called mini-batch optimization, i.e. by optimizing

$$\hat{\mathcal{L}}_{VI}(\boldsymbol{\eta}) := \frac{N}{M} \sum_{i \in S} \int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log p(y_i|\boldsymbol{\theta}, \boldsymbol{x}_i, m) \mathrm{d}\boldsymbol{\theta} - \mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})), \tag{6.15}$$

where S is the set of indices of size M corresponding to random subsample of the whole data sample. Obviously, this is an unbiased estimator of the ELBO, hence $E_S[\hat{\mathcal{L}}_{VI}(\boldsymbol{\eta})] = \mathcal{L}_{VI}(\boldsymbol{\eta})$. A stochastic gradient descent over several subsamples with indices S can be applied for this part, yielding some local extremum of $\mathcal{L}_{VI}(\boldsymbol{\eta})$ (Gal, 2016). Another complicated issue is to obtain $\frac{\partial}{\partial \boldsymbol{\eta}} \int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}) \log p(y_i|\boldsymbol{\theta}, \boldsymbol{x}_i, m) \mathrm{d}\boldsymbol{\theta}$ for computing the gradient. This issue has been attempted to be resolved in the deep Bayesian learning community by the following simple approach. With no loss of generality, consider an integral derivative of the form:

$$I(\eta) = \frac{\partial}{\partial \eta} \int_{\Omega_x} f(x) p_{\eta}(x) \mathrm{d}x, \tag{6.16}$$

where $f(x)$ is a function differentiable almost everywhere and $p_{\eta}(x)$ is a density function with parameter η. Also assume that the integral exists and it is finite in order to switch the order of integration and differentiation. One of the approaches to compute (6.16) is to remember a standard algebraic trick and notice that $\frac{\partial}{\partial \eta} p_{\eta}(x) = p_{\eta}(x) \frac{\partial}{\partial \eta} \log p_{\eta}(x)$, yielding, under the assumption of existence and finity of the integral:

$$\frac{\partial}{\partial \eta} \int_{\Omega_x} f(x) p_{\eta}(x) \mathrm{d}x = \int_{\Omega_x} f(x) \frac{\partial}{\partial \eta} p_{\eta}(x) \mathrm{d}x = \int_{\Omega_x} f(x) p_{\eta}(x) \frac{\partial}{\partial \eta} \log p_{\eta}(x) \mathrm{d}x. \tag{6.17}$$

Obviously, it follows that $I(\eta)$ can be approximated by an unbiased estimator of the form $\hat{I}(\eta) = \frac{1}{M} \sum_{i=1}^{M} f(x_i) \frac{\partial}{\partial \eta} \log p_{\eta}(x_i)$, with $x_i \sim p_{\eta}(x)$. This estimator, however, exhibits large variance and hence some variance reduction techniques like a common random numbers approach (known as a repartmentalization trick in the machine learning literature) should be used (Gal, 2016).

Variationally approximated marginal likelihood $\hat{p}_{\mathrm{VB}}(\mathcal{D}|m)$ for a given model m can be extremely biased in practice. When performing MCMC algorithms based on $\hat{p}_{\mathrm{VB}}(\mathcal{D}|m)$, one has to keep in mind that there are no guarantees of asymptotic unbiasedness of the results. Within the discussed approach this comes as the price for feasibility of inference. More research has to be conducted on how to find a balanced trade off between the bias and computational complexity. On the other hand, one can be not interested in using variationally approximated marginal likelihood and instead use variational inference for Bayesian model selection directly. There are

successful applications of this approach in the context of linear models (Carbonetto et al., 2012). A natural extension of the idea suggested in Carbonetto et al. (2012) to a general variable selection context of form (4.16)-(4.17) can be adapted. Consider, for instance, a model (4.16)-(4.17) with features $F_j(\boldsymbol{x}_i|\boldsymbol{\omega_j})$ from the topology of deep Bayesian regression models, described in Paper III of this thesis. Loosely speaking, the q features in this context correspond to all possible architectures of neural networks of a limited depth, based on all combinations of addressed activation functions and all possible connections between the neurons. Consider, without loss of generality, β_0 fixed and known and define variational approximation as

$$q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \prod_{j \in \{1,...,q\}} q_{\boldsymbol{\eta}_j}(\boldsymbol{\omega}_j, \gamma_j), \tag{6.18}$$

where now the factorized components $q_{\boldsymbol{\eta}_j}(\boldsymbol{\omega}_j, \gamma_j)$ are of the form

$$q_{\boldsymbol{\eta}_j}(\boldsymbol{\omega}_j, \gamma_j) = \begin{cases} \alpha_j N_{p_{\gamma_j}}(\boldsymbol{\mu}_{\gamma_j}, \boldsymbol{\Sigma}^2_{\gamma_j}), \text{ if } \gamma_j = 1; \\ (1 - \alpha_j)\delta_0(\boldsymbol{\omega}_j), \text{ if } \gamma_j = 0. \end{cases} \tag{6.19}$$

Here $\delta_0(\cdot)$ is the delta mass or "spike" in the notation from Section 4.2 at zero and $\boldsymbol{\eta}_j = (\boldsymbol{\mu}_{\gamma_j}, \boldsymbol{\Sigma}^2_{\gamma_j}, \alpha_j)$. Thus, with probability α_j, the parameters of a feature j are multivariate normal with a diagonal covariance matrix structure ("slab"), and otherwise the feature is considered to have no effect on the observations $\mathbf{y}$. Then minimization of the KL divergence between the true posterior and the variational approximation can be performed similarly to the way described above in this section. This Bayesian feature selection approach scales linearly with respect to the number of features, unlike the approaches developed in this thesis (which in the limit search through 2^q configurations). However if q itself is exponential in the number of input explanatory variables (like in the case of deep Bayesian regression models or Bayesian logic regression), calculation of (6.18) becomes infeasible and some novel techniques to approximate it in a pragmatic (yet reasonable) way should be developed. One could think, for instance, of introducing the latent binary indicators already for all of the components (both weights and activations) of a single dense and fully connected ANN with the mean vector of the form (2.15). Recall (2.13), define $\mathcal{G} = \{\sigma_1(\cdot), ..., \sigma_T(\cdot)\}$ to be the set of all allowed activation functions (just as defined in Paper III) and update (2.14) as

$$z_{ij}^{(l+1)} = \sum_{t=1}^{T} \delta_{tj}^{(l)} \sigma_t \left(\tau_{0j}^{l} \beta_{0j}^{(l)} + \sum_{k=1}^{p^{(l)}} \tau_{kj}^{(l)} \beta_{kj}^{(l)} z_{ik}^{(l)} \right), \tag{6.20}$$

where the latent binary indicators $\delta_{tj}^{(l)} \in \{0, 1\}, t \in \{1, ..., T\}, l \in \{1, ..., L-1\}$ such that $\sum_{t=1}^{T} \delta_{tj}^{(l)} = 1$ are used for choosing the corresponding activation function at layer l and $\tau_{kj}^{(l)} \in \{0, 1\}, k \in \{0, ..., p^{(l)}\}, l \in \{1, ..., L-1\}$ are the binary indicators for switching on the corresponding beta coefficients at layer l. To complete Bayesian specification priors for all parameters of (6.20) should be specified: $\phi \sim p_\phi(\phi), \boldsymbol{\delta} \sim p_{\boldsymbol{\delta}}(\boldsymbol{\delta}), \boldsymbol{\tau} \sim p_{\boldsymbol{\tau}}(\boldsymbol{\tau}), \boldsymbol{\beta}|\boldsymbol{\tau} \sim p_{\boldsymbol{\beta}}(\boldsymbol{\beta})$. As discussed in Paper III of this thesis the priors depend drastically upon the application. On the one hand, notation of (6.20) maps to the definition of DBRM models, described in Paper III. On the other hand, if one uses Gaussian priors for the beta coefficients $p(\boldsymbol{\beta}|\boldsymbol{\tau}) = N_{|\boldsymbol{\tau}|}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$

and independent Bernoulli priors for the latent binary indicators, $\tau_{kl} \sim \text{Bernoulli}(\rho_\tau)$ and $\delta_{tl} \sim \text{Bernoulli}(\rho_\delta)$, this idea becomes closely related to variational dropout, actively used in the Bayesian deep learning community (Gal and Ghahramani, 2016; Molchanov et al., 2017; Gal and Ghahramani, 2015; Kingma et al., 2015). Performing variational inference on all parameters of equation (6.20) allows to linearize the variational distribution for feature selection even further, making the approach indeed scalable. Gal and Ghahramani (2016) claim that it is prohibitive to use fully Bayesian inference for such models due to the computational complexity of the latter to support usage of variational Bayes. The question then is whether this pragmatic simplification will cause serious problems for the inference based on the model. This is particularly important in the light of the warning made by Jordan et al. (2013), namely: "Gatherers of large-scale data are often forced to turn to ad hoc procedures that perhaps do provide *algorithmic guarantees* but which may provide *no statistical guarantees* and which in fact may have poor or even *disastrous statistical properties*." And indeed Bayesian deep learning, performed with variational Bayes, has received a fair amount of criticism. Hron et al. (2017), for instance, claim that a popular variational Gaussian dropout technique is not Bayesian at all, since it induces an ill-posed posterior. The latter means that one cannot make inference based on this approach in the Kolmogorov's probability notion (Kolmogorov), however it has to be noted (and it is not mentioned in Hron et al. (2017)) that delicately tuned inference *may* be done for the improper distributions in the Renyi's probability notion (Rényi, 1955), as noticed in Taraldsen (2018); Taraldsen and Lindqvist (2007, 2010); Lindqvist and Taraldsen (2017).

Other remarks

Returning to other directions for continuing the research performed in this thesis, one can admit that the use of graphical processing units (GPU) in the developed approaches will allow to achieve even better scalability. In particular, it could allow to speed up the process of obtaining the marginal likelihood based on variational approximations for individual models. In the machine learning community it is standard to compare all novel approaches with the existing ones based on some benchmark data sets. These datasets typically include MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky and Hinton, 2009), CIFAR100 (Krizhevsky and Hinton, 2009), and IMAGENET (Deng et al., 2009). In order to convince scientists from that community that the developed approaches are reasonable and efficient, some efforts to compare their performance to other machine learning techniques have to be done. Furthermore, once a scalable approach for large sample sizes becomes available, it would be of a particular interest to carry out epigenetic data analysis, like the one appearing in Papers I and III, on the full genome (including several million observations), to get more evidence for finding sophisticated nonlinear patterns for explanation of epigenetic observations.

References

A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola. Scalable inference in latent variable models. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 123–132. ACM, 2012.

M. Aitkin. Paradigms for statistical inference. *University of Melbourne Press*, 2011.

J. Akeret, A. Refregier, A. Amara, S. Seehars, and C. Hasner. Approximate Bayesian computation for forward modeling in cosmology. *Journal of Cosmology and Astroparticle Physics*, 2015(08):043, 2015.

F. Al-Awadhi, M. Hurn, and C. Jennison. Improving the acceptance rate of reversible jump MCMC proposals. *Statistics and Probability Letters*, 69(2):189 – 198, 2004. ISSN 0167-7152. doi: http://dx.doi.org/10.1016/j.spl.2004.06.025. URL `http://www.sciencedirect.com/science/article/pii/S016771520400183X`.

P. Alquier, J. Ridgway, and N. Chopin. On the properties of variational approximations of Gibbs posteriors. *JMLR*, 17(239):1–41, 2016.

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, pages 697–725, 2009.

J. Arifovic and R. Gencay. Using genetic algorithms to select architecture of a feedforward artificial neural network. *Physica A: Statistical mechanics and its applications*, 289(3-4): 574–594, 2001.

H. Attias. A variational Baysian framework for graphical models. In *Advances in neural information processing systems*, pages 209–215, 2000.

K. Banaszek. Maximum-likelihood estimation of photon-number distribution from homodyne statistics. *Physical Review A*, 57(6):5013, 1998.

M. Banterle, C. Grazian, and C. P. Robert. Accelerating Metropolis-Hastings algorithms: Delayed acceptance with prefetching. *arXiv preprint arXiv:1406.2660*, 2014.

M. Banterle, C. Grazian, A. Lee, and C. P. Robert. Accelerating Metropolis-Hastings algorithms by delayed acceptance. *arXiv preprint arXiv:1503.00996*, 2015.

M. M. Barbieri, J. O. Berger, et al. Optimal predictive model selection. *The annals of statistics*, 32(3):870–897, 2004.

R. Bardenet, A. Doucet, and C. Holmes. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *International Conference on Machine Learning*, pages 405–413, 2014.

R. Bardenet, A. Doucet, and C. Holmes. On Markov chain Monte Carlo methods for tall data. *The Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

A. Basu, I. R. Harris, N. L. Hjort, and M. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

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

M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406, 2010.

M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

J. Berger et al. The case for objective Bayesian analysis. *Bayesian analysis*, 1(3):385–402, 2006.

A. Beskos, G. Roberts, A. Stuart, and J. Voss. MCMC methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

R. S. Bivand, V. Gómez-Rubio, and H. Rue. Approximate Bayesian inference for spatial econometrics models. *Spatial Statistics*, 9:146–165, 2014.

L. Bottolo, S. Richardson, et al. Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis*, 5(3):583–618, 2010.

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

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

G. E. Box and G. C. Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretical Approach (Second Edition)*. Springer-Verlag New York, Inc., 2002. ISBN 9780387224565.

P. Carbonetto, M. Stephens, et al. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian analysis*, 7(1):73–108, 2012.

R. Carnap. *Logical Foundations of Probability*. The University of Chicago Press, Chicago, 1950.

R. Carnap. *The Continuum of Inductive Methods*. The University of Chicago Press, Chicago, 1952.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the american statistical association*, 90(432):1313–1321, 1995.

S. Chib, E. Greenberg, and R. Winkelmann. Posterior simulation and Bayes factors in panel count data models. *Journal of Econometrics*, 86(1):33 – 54, 1998.

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

G. Claeskens, N. L. Hjort, et al. Model selection and model averaging. *Cambridge Books*, 2008.

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

J. S. Cramer. *Econometric applications of maximum likelihood methods*. CUP Archive, 1989.

K. Csilléry, M. G. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

R. Davidson, J. G. MacKinnon, et al. Estimation and inference in econometrics. *OUP Catalogue*, 1993.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

A. Einstein. On the method of theoretical physics. *Philosophy of science*, 1(2):163–169, 1934.

T. Elsken, J.-H. Metzen, and F. Hutter. Simple and efficient architecture search for convolutional neural networks. *arXiv preprint arXiv:1711.04528*, 2017.

L. Fahrmeir and S. Lang. Bayesian inference for generalized additive mixed models based on Markov random field priors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 50(2):201–220, 2001.

W. E. Ferson, S. R. Foerster, et al. Finite sample properties of the generalized method of moments in tests of conditional asset pricing models. *Journal of Financial Economics*, 36 (1):29–55, 1994.

D. A. Freedman. *Statistical models: theory and practice*. Cambridge university press, 2009.

M. Frenklach and S. J. Harris. Aerosol dynamics modeling using the method of moments. *Journal of colloid and interface science*, 118(1):252–261, 1987.

A. Fritsch. *A Full Bayesian Version of Logic regression for SNP Data*. PhD thesis, Diploma Thesis, 2006.

A. Fritsch and K. Ickstadt. Comparing Logic Regression Based Methods for Identifying SNP Interactions. *Springer Berlin / Heidelberg, Lecture Notes in Computer Science*, 4414:90–103, 2007.

F. Frommlet, F. Ruhaltinger, P. Twaróg, and M. Bogdan. Modified versions of Bayesian information criterion for genome-wide association studies. *CSDA*, 56:1038–1051, 2012.

Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Insights and applications. In *Deep Learning Workshop, ICML*, volume 1, page 2, 2015.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

A. Gelman. Bayes, Jeffreys, prior distributions and the philosophy of statistics. *Statistical Science*, 24(2):176–178, 2009.

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

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

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

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2): 123–214, 2011.

A. Gola, M. Bogdan, and F. Frommlet. EA-MOSGWA: a tool for identifying associated SNPs in Genome Wide Association Studies. *Theoretical and Applied Informatics*, 25, 2013.

I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

A. Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, pages 2348–2356, 2011.

Y. Guan and M. Stephens. Bayesian variable selection regression for genome-wide association studies, and other large-scale problems. *Ann. Appl. Stat.*, 5:1780–1815, 2011.

A. R. Hall. *Generalized method of moments*. Oxford University Press, 2005.

J. Hannig. On generalized fiducial inference. *Statistica Sinica*, pages 491–544, 2009.

L. P. Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054, 1982.

T. Hastie and R. Tibshirani. *Generalized additive models*. Wiley Online Library, 1990.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

J. Hron, A. G. d. G. Matthews, and Z. Ghahramani. Variational Gaussian Dropout is not Bayesian. *arXiv preprint arXiv:1711.02989*, 2017.

A. Hubin. EMJMCMC2016, 2018. URL `http://aliaksah.github.io/EMJMCMC2016/`.

K. Humphreys and D. Titterington. Approximate Bayesian inference for simple mixtures. In *Proc. Computational Statistics 2000*, pages 331–336, 2000.

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems*, pages 487–493, 1999.

R. Jeffrey. Valuation and acceptance of scientific hypotheses. *Philosophy of Science*, 23(1): 237–246, 1956.

H. Jeffreys. *Theory of Probability*. Oxford University Press, Oxford, 1939.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

M. I. Jordan et al. On statistics, computation and scalability. *Bernoulli*, 19(4):1378–1390, 2013.

J. Keynes. *A Treatise on Probability*. Macmillan, London, 1921.

H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, 2004.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

P. K. Kitanidis. Prediction by the method of moments of transport in a heterogeneous formation. *J. Hydrol*, 102(1-4):453–473, 1988.

A. Kolmogorov. *Foundations of the Theory of Probability: Second English Edition*.

C. Kooperberg and I. Ruczinski. Identifying Interacting SNPs Using Monte Carlo Logic Regression. *Genetic Epidemiology*, 28:157–170, 2005.

C. Kooperberg, J. C. Bis, K. D. Marciante, S. R. Heckbert, T. Lumley, and B. M. Psaty. Logic Regression for Analysis of the Association between Genetic Variation in the Renin-Angiotensin System and Myocardial Infarction or Stroke. *American Journal of Epidemiology*, 165(3):334–343, 2007.

A. Korattikara, Y. Chen, and M. Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *International Conference on Machine Learning*, pages 181–189, 2014.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

P. C. Lambert et al. Fractional polynomials and model averaging. 2007.

N. S. Lari and M. S. Abadeh. A new approach to find optimum architecture of ANN and tuning it's weights using krill-herd algorithm. In *Technology, Communication and Knowledge (ICTCK), 2014 International Congress on*, pages 1–7. IEEE, 2014.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

C. Li, S. Srivastava, and D. B. Dunson. Simple, scalable and accurate posterior interval estimation. *Biometrika*, 104(3):665–680, 2017.

F. Liese and I. Vajda. Convex statistical distances. 2007.

B. H. Lindqvist and G. Taraldsen. On the proper treatment of improper distributions. *Journal of Statistical Planning and Inference*, 2017.

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

S. Liu, G. Mingas, and C.-S. Bouganis. An exact MCMC accelerator under custom precision regimes. In *Field Programmable Technology (FPT), 2015 International Conference on*, pages 120–127. IEEE, 2015.

D. J. MacKay. Ensemble Learning for Hidden Markov Models. *Unpublished manuscript*, 1997.

D. Maclaurin and R. P. Adams. Firefly Monte Carlo: Exact MCMC with Subsets of Data. In *UAI*, pages 543–552, 2014.

G. S. Maddala and K. Lahiri. *Introduction to econometrics*, volume 2. Macmillan New York, 1992.

J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

P. McCullagh and J. Nelder. *Generalized Linear Models. 2nd Edition*. Chapman and Hall, London, 1989.

C. E. McCulloch and J. M. Neuhaus. *Generalized linear mixed models*. Wiley Online Library, 2001.

S. Minsker, S. Srivastava, L. Lin, and D. Dunson. Scalable and robust Bayesian inference via the median posterior. In *International Conference on Machine Learning*, pages 1656–1664, 2014.

D. Molchanov, A. Ashukha, and D. Vetrov. Variational dropout sparsifies deep neural networks. *arXiv preprint arXiv:1701.05369*, 2017.

G. F. Montúfar and J. Morton. When does a mixture of products contain a product of mixtures? *SIAM Journal on Discrete Mathematics*, 29(1):321–347, 2015.

S. Morugin. Ostagram, 2015. URL `https://www.ostagram.me/`.

J. Naudts. Estimators, escort probabilities, and phi-exponential families in statistical physics. *J. Ineq. Pure Appl. Math*, 5(4):102, 2004.

W. Neiswanger, C. Wang, and E. Xing. Asymptotically exact, embarrassingly parallel MCMC. *arXiv preprint arXiv:1311.4780*, 2013.

M. Pagel. Maximum-likelihood models for glottochronology and for reconstructing linguistic phylogenies. *Time depth in historical linguistics*, 1:189–207, 2000.

R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

R. D. Payne and B. K. Mallick. Bayesian Big Data Classification: A Review with Complements. *arXiv preprint arXiv:1411.5653*, 2014.

H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

T. Poggio. Deep learning: mathematics and neuroscience. *A Sponsored Supplement to Science, Brain-Inspired intelligent robotics: The intersection of robotics and neuroscience*, pages 9–12, 2016.

N. G. Polson, V. Sokolov, et al. Deep Learning: A Bayesian Perspective. *Bayesian Analysis*, 12 (4):1275–1304, 2017.

M. Quiroz, R. Kohn, M. Villani, and M.-N. Tran. Speeding up MCMC by efficient data subsampling. *arXiv preprint arXiv:1404.4178*, 2014.

M. Quiroz, M. Villani, and R. Kohn. Exact subsampling MCMC. *arXiv preprint arXiv:1603.08232*, 2016.

M. Quiroz, M.-N. Tran, M. Villani, and R. Kohn. Speeding up MCMC by delayed acceptance and data subsampling. *Journal of Computational and Graphical Statistics*, pages 1–11, 2017.

F. P. Ramsey. *Truth and Probability*. Routledge and Kegan Pau, London, 1931.

H. Reichenbach. *A translation by Ernest R. Hutton and Maria Reichenbach of Wahrscheinlichkeitslehre. Eine Untersuchung uber die logischen und mathematischen Grundlagen der Wahrscheinlichskeitrechnung*. University of California Press, Leiden, 1935.

A. C. Rencher and W. F. Christensen. Chapter 10, multivariate regression–section 10.1, introduction. *Methods of Multivariate Analysis, Wiley Series in Probability and Statistics*, 709: 19, 2012.

A. Rényi. On a new axiomatic theory of probability. *Acta Mathematica Academiae Scientiarum Hungarica*, 6(3-4):285–335, 1955.

A. Rényi. On measures of entropy and information. Technical report, HUNGARIAN ACADEMY OF SCIENCES Budapest Hungary, 1961.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

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

G. O. Roberts and O. Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4(4):337–357, 2002.

I. Ruczinski, C. Kooperberg, and M. LeBlanc. Logic regression. *J. Comput Graphical Statist.*, 12(3):474–511, 2003.

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

C. Sabatti, S. Service, A. Hartikainen, A. Pouta, S. Ripatti, J. Brodsky, C. Jones, N. Zaitlen, T. Varilo, M. Kaakinen, U. Sovio, A. Ruokonen, J. Laitinen, E. Jakkula, L. Coin, C. Hoggart, A. Collins, H. Turunen, S. Gabriel, P. Elliot, M. McCarthy, M. Daly, M. Järvelin, N. Freimer, and L. Peltonen. Genome-wide association analysis of metabolic traits in a birth cohort from a founder population. *Nat Genet*, 41:35–46, 2009.

R. J. Schalkoff. *Artificial neural networks*, volume 1. McGraw-Hill New York, 1997.

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

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

S. L. Scott, A. W. Blocker, F. V. Bonassi, H. A. Chipman, E. I. George, and R. E. McCulloch. Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

H. L. Seal. Studies in the History of Probability and Statistics. XV The historical development of the Gauss linear model. *Biometrika*, 54(1-2):1–24, 1967.

S. G. Self and K.-Y. Liang. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association*, 82(398):605–610, 1987.

L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE transactions on medical imaging*, 1(2):113–122, 1982.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

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

V. Srinivasan and C. H. Mason. Nonlinear least squares estimation of new product diffusion models. *Marketing science*, 5(2):169–178, 1986.

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

B. E. Stranger, A. C. Nica, M. S. Forrest, A. Dimas, C. Bird, C. Beazley, C. E. Ingle, M. Dunning, P. Flicek, S. Montgomery, S. Tavare, P. Deloukas, and E. T. Dermitzakis. Population genomics of human gene expression. *Nat Genet*, 39:1217–1224, 2007.

G. Taraldsen. Deep Neural Learning with Objective Beliefs. 2018.

G. Taraldsen and B. H. Lindqvist. Bayes theorem for improper priors. *Preprint Statistics*, (4), 2007.

G. Taraldsen and B. H. Lindqvist. Improper priors are not improper. *The American Statistician*, 64(2):154–158, 2010.

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

A. Ullah. Entropy, divergence and distance measures with econometric applications. *Journal of Statistical Planning and Inference*, 49(1):137–162, 1996.

F. van Veen. THE NEURAL NETWORK ZOO, 2016. URL http://www.asimovinstitute.org/neural-network-zoo/.

G. Wahba. A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409, 1965.

G. Wahba. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 364–372, 1978.

X. Wang and D. B. Dunson. Parallelizing MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

A. Weyant, C. Schafer, and W. M. Wood-Vasey. Likelihood-free cosmological inference with type Ia supernovae: Approximate Bayesian computation for a complete treatment of uncertainty. *The Astrophysical Journal*, 764(2):116, 2013.

X. Yan and X. Su. *Linear regression analysis: theory and computing*. World Scientific, 2009.

Y. Yang, D. Pati, and A. Bhattacharya. Alpha variational inference with statistical guarantees. *arXiv preprint arXiv:1710.03266*, 2017.

Z. Yang. Paml 4: phylogenetic analysis by maximum likelihood. *Molecular biology and evolution*, 24(8):1586–1591, 2007.

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

S. Zabell. *Symmetry and Its Discontents: Essays on the History of Inductive Probability*. Cambridge University Press, Cambridge, 2005.

B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2017.

Paper I

Computational Statistics and Data Analysis 127 (2018) 281–297

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Mode jumping MCMC for Bayesian variable selection in GLMM

Aliaksandr Hubin [*,1], Geir Storvik

Department of Mathematics, University of Oslo, Norway

HIGHLIGHTS

- Standard MCMC algorithms often get stuck in local extrema.
- Suggested mode jumping MCMC algorithms allow to accurately explore the domain of interest.
- Parallel computing and using mixtures of proposals can further improve mode jumping MCMC algorithms.
- Bayesian model selections and model averaging can be efficiently done by means of the suggested mode jumping MCMC algorithms.

ARTICLE INFO

Article history:
Received 17 October 2017
Received in revised form 12 March 2018
Accepted 25 May 2018
Available online 5 June 2018

Keywords:
Bayesian variable selection
Bayesian model averaging
Generalized linear mixed models
Auxiliary variables MCMC
Combinatorial optimization
High performance computations

ABSTRACT

Generalized linear mixed models (GLMM) are used for inference and prediction in a wide range of different applications providing a powerful scientific tool. An increasing number of sources of data are becoming available, introducing a variety of candidate explanatory variables for these models. Selection of an optimal combination of variables is thus becoming crucial. In a Bayesian setting, the posterior distribution of the models, based on the observed data, can be viewed as a relevant measure for the model evidence. The number of possible models increases exponentially in the number of candidate variables. Moreover, the space of models has numerous local extrema in terms of posterior model probabilities. To resolve these issues a novel MCMC algorithm for the search through the model space via efficient mode jumping for GLMMs is introduced. The algorithm is based on that marginal likelihoods can be efficiently calculated within each model. It is recommended that either exact expressions or precise approximations of marginal likelihoods are applied. The suggested algorithm is applied to simulated data, the famous U.S. crime data, protein activity data and epigenetic data and is compared to several existing approaches.

1. Introduction

In this paper we study variable selection in generalized linear mixed models (GLMM) addressed in a Bayesian setting. Being one of the most powerful modeling tools in modern statistical science (Stroup, 2013) these models have proven to be efficient in numerous applications including simple banking scoring problems (Grossi and Bellini, 2006), insurance claims modeling (David, 2015), studies on the course of illness in schizophrenia, linking diet with heart diseases (Skrondal and Rabe-Hesketh, 2003), analyzing sophisticated astrophysical data (de Souza et al., 2015), and inferring on genomics

* Corresponding author.
E-mail address: aliaksah@math.uio.no (A. Hubin).
[1] Ph.D. candidate at the University of Oslo, 0851 Moltke Moes vei 35 Oslo, Norway.

https://doi.org/10.1016/j.csda.2018.05.020
0167-9473/

data (Lobraux and Melodelima, 2015). In many of these applications, the number of candidate explanatory variables (covariates) is large, making variable selection a difficult problem, both conceptually and numerically. In this paper we will focus on efficient Markov chain Monte Carlo (MCMC) algorithms for such variable selection problems. Our focus will be on posterior model probabilities although other model selection criteria can also easily be adopted within the algorithm.

Algorithms for variable selection in the Bayesian settings have been previously addressed, but primarily in the combined space of models *and* parameters. George and McCulloch (1997) describe and compare various hierarchical mixture prior formulations for Bayesian variable selection in normal linear regression models. They outline computational methods including Gray Code sequencing and standard MCMC for posterior evaluation and exploration of the space of models. They also comment on the infeasibility of exhaustive exploration of the space of models for moderately large problems as well as the inability of standard MCMC techniques to escape from local optima efficiently. Al-Awadhi et al. (2004) consider using several MCMC steps within a new model to obtain good proposals within the combined parameter and model domain while Yeh et al. (2012) propose local annealing approaches. Ghosh (2015) also addresses MCMC algorithms to estimate the posterior distribution over models. She observes that estimates of posterior probabilities of individual models based on MCMC output are often not reliable because the number of MCMC samples is typically considerably smaller than the size of the model space. As a consequence she considers the median probability model of Barbieri et al. (2004) instead and shows that this algorithm can, under some conditions, outperform standard MCMC. Yet another approach for Bayesian model selection is addressed by Bottolo et al. (2011), who propose the moves of MCMC between local optima through a permutation based genetic algorithm that has a pool of solutions in a current generation suggested by parallel tempered chains. A similar idea is considered by Frommlet et al. (2012). Multiple try MCMC methods with local optimization have been described by Liu et al. (2000). Song and Liang (2015) address the case when there is by far more explanatory variables than observations. They suggest a split and merge Bayesian model selection algorithm that first splits the set of covariates into a number of subsets, then finds relevant variables from these subsets and in the second stage merges these relevant variables and performs a new selection from the merged set. This algorithm in general cannot guarantee convergence to a global optimum or find the true posterior distribution of the models, however under some strict regularity conditions it does so asymptotically.

For an increasing number of model classes, marginal likelihoods for specific models can be efficiently calculated, either exactly or approximately. This makes the exploration of models much easier. Bové and Held (2011) consider an MCMC algorithm within the model space, but only allow local moves. This might be a severe limitation in cases where multiple sparsely located modes are present in the model space. Bivand et al. (2014) combine approximations of marginal likelihood with Bayesian model averaging within spatial models. Clyde et al. (2011) suggest a Bayesian adaptive sampling (BAS) algorithm as an alternative to MCMC allowing for perfect sampling without replacement.

In the general MCMC literature, various algorithms for exploration of model spaces with multiple sparse modes have been suggested. These approaches can be divided into two groups: methods based on exploration of the tempered target distributions (allowing to flatten or increase multimodality for different temperatures) and methods based on utilization of local gradients. The first group of algorithms was initialized with the parallel tempering approach (Geyer, 1991), which further had numerous modifications (Liang, 2010; Miasojedow et al., 2013; Salakhutdinov, 2009). One of the most prominent extensions is the equi-energy sampling approach (Kou et al., 2006), which utilizes the physical duality between temperature and energy. This approach targets directly the former to flatten or tighten the parameter spaces. Another extension is the multi domain sampling approach (Zhou, 2011), which first uses the target distribution tempering idea to find the set of local modes and then uses local MCMC to explore the regions around them for further global inference. The second group of algorithms uses auxiliary variables combined with gradients of the extended distribution to explore the state space accurately (Neal et al., 2011; Chen et al., 2014; Sengupta et al., 2016 and many others). Both groups of algorithms are mainly developed for exploration of continuous parameter spaces. All of these algorithms can in principle be adapted to discrete space problems. The approach in this article will be to adapt the mode jumping MCMC idea of Tjelmeland and Hegstad (1999) to the variable selection problem, utilizing the existence of marginal likelihoods for models of interest.

Different approaches can be applied for calculation of marginal likelihoods. For linear models with conjugate priors, analytic expressions are available (Clyde et al., 2011). In more general settings, MCMC algorithms combined with e.g. Chib's method (Chib, 1995) can be applied, giving however computationally expensive procedures. See also Friel and Wyse (2012) for alternative MCMC based methods. For Gaussian latent variables, the computational task can be efficiently solved through the integrated nested Laplace approximation (INLA) approach (Rue et al., 2009). Hubin and Storvik (2016) compare INLA with MCMC based methods, showing that INLA based approximations are extremely accurate and require much less computational effort than the MCMC approaches for within-model calculations.

In this paper we introduce a novel MCMC algorithm for search through the model space, the mode jumping MCMC (MJMCMC). The focus will be on Gaussian latent variable models, for which efficient approximations to marginal likelihoods are available. The algorithm is based on the idea of mode jumping within MCMC—resulting in an MCMC algorithm which manages to efficiently explore the model space by means of mode jumping, applicable through large jumps combined with local optimization. Mode jumping MCMC methods within a continuous space setting were first suggested by Tjelmeland and Hegstad (1999). We modify the algorithm to the discrete space of possible models, requiring both new ways of making large jumps and of performing local optimization. We include mixtures of proposal distributions and parallelization to further improve the performance of the algorithm. A valid acceptance probability within the Metropolis–Hastings setting is constructed based on the use of backward kernels.

2. The generalized linear mixed model

We consider the following generalized linear mixed model:

$$Y_i|\mu_i \sim \mathfrak{f}(y|\mu_i), \quad \mu_i = g^{-1}(\eta_i), \tag{1}$$

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j x_{ij} + \delta_i \tag{2}$$

and

$$\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n) \sim N_n(\mathbf{0}, \boldsymbol{\Sigma}_b). \tag{3}$$

Here Y_i is the response variable while $x_{ij}, j = 1, \ldots, p$ are the covariates. We assume $\mathfrak{f}(y|\mu)$ is a density/distribution from the exponential family with corresponding link function $g(\cdot)$. The latent indicators $\gamma_j \in \{0, 1\}, j = 1, \ldots, p$ define if covariate x_{ij} is to be included into the model ($\gamma_j = 1$) or not ($\gamma_j = 0$) while $\beta_j \in \mathbb{R}, j = 0, \ldots, p$ are the corresponding regression coefficients. We are also addressing the unexplained variability of the responses and the correlation structure between them through random effects δ_i with a specified parametric covariance matrix structure defined through $\boldsymbol{\Sigma}_b = \boldsymbol{\Sigma}_b(\boldsymbol{\psi}) \in \mathbb{R}^{n\times n}$, where $\boldsymbol{\psi}$ are parameters describing the correlation structure.

In order to put the model into a Bayesian framework, we assume

$$\gamma_j|q \sim \text{Binom}(1, q), \quad j = 1, \ldots, p \tag{4}$$

and

$$q \sim \text{Beta}(a_q, b_q), \tag{5}$$

where q is the prior probability of including a covariate into the model. For $(\boldsymbol{\beta}, \boldsymbol{\psi})$ different priors are possible, see the applications in Section 4.

Let $\boldsymbol{\gamma} = (\gamma_1, \ldots \gamma_p)$, which uniquely defines a specific model. Assuming the constant term β_0 is always included, there are $L = 2^p$ different models to consider. We want to find a set of the best models with respect to posterior model probabilities $p(\boldsymbol{\gamma}|\mathbf{y})$, where $\mathbf{y} = (y_1, \ldots, y_n)$. We assume that marginal likelihoods $p(\mathbf{y}|\boldsymbol{\gamma})$ are available for a given $\boldsymbol{\gamma}$, and then use MCMC to explore $p(\boldsymbol{\gamma}|\mathbf{y})$. By Bayes formula

$$p(\boldsymbol{\gamma}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}'\in\Omega} p(\mathbf{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')}. \tag{6}$$

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

$$\widetilde{p}(\boldsymbol{\gamma}|\mathbf{y}) = \frac{\sum_{i=1}^{W} \mathbb{1}(\boldsymbol{\gamma}^{(i)} = \boldsymbol{\gamma})}{W} \xrightarrow[W\to\infty]{d} p(\boldsymbol{\gamma}|\mathbf{y}), \tag{7}$$

where $\mathbb{1}(\cdot)$ is the indicator function. An alternative, named the renormalized model (RM) estimates by Clyde et al. (2011), is

$$\widehat{p}(\boldsymbol{\gamma}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}'\in\mathbb{V}} p(\mathbf{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')}\mathbb{1}(\boldsymbol{\gamma} \in \mathbb{V}), \tag{8}$$

where now $\mathbb{V}$ is the set of visited models during the MCMC run. Although both (8) and (7) are asymptotically consistent, (8) will often be the preferable estimator since convergence of the MCMC based approximation (7) is much slower, see Clyde et al. (2011).

We aim at approximating $p(\boldsymbol{\gamma}|\mathbf{y})$ by means of searching for some subspace $\mathbb{V}$ of Ω making the approximation (8) as precise as possible. Models with high values of $p(\boldsymbol{\gamma}|\mathbf{y})$ are important to be addressed. This means that modes and near modal values of marginal likelihoods are particularly important for construction of reasonable $\mathbb{V} \subset \Omega$ and missing them can dramatically influence our estimates. Note that these are aspects just as important if the standard MCMC estimate (7) is to be used. A main difference is that while for using (7) the number of times a specific model is visited is important, for (8) it is enough that a model is visited at least once. In this context the denominator of (8), which we would like to be as high as possible, becomes an extremely relevant measure for the quality of the search in terms of being able to capture whether the algorithm visits all of the modes, whilst the size of $\mathbb{V}$ should be low in order to save computational time.

The posterior marginal inclusion probability $p(\gamma_j = 1|\mathbf{y})$ can be approximated by

$$\widehat{p}(\gamma_j = 1|\mathbf{y}) = \sum_{\boldsymbol{\gamma}'\in\mathbb{V}} \mathbb{1}(\gamma_j' = 1)\widehat{p}(\boldsymbol{\gamma}'|\mathbf{y}), \tag{9}$$

giving a measure for assessing importance of the covariates. Other parameters can be estimated similarly.

Algorithms for estimating $\mathbb{V}$ are described in Section 3. In practice $p(\mathbf{y}|\boldsymbol{\gamma})$ may not be available analytically. We then rely on some precise approximations $\widehat{p}(\mathbf{y}|\boldsymbol{\gamma})$. Such approximations introduce additional errors in (8) and (9), but we assume them to be small enough to be ignored. This is further discussed in Section 3.4.

3. Mode jumping Markov Chain Monte Carlo

MCMC algorithms (Robert and Casella, 2005) have been extremely popular for the exploration of model spaces for model selection, being capable of providing samples from the posterior distribution of the models. In our setting, the most important aspect becomes building a method to explore the model space in a way to efficiently switch between potentially sparsely located modes, whilst avoiding visiting models with a low $p(\boldsymbol{y}|\boldsymbol{\gamma})$ too often.

3.1. Standard Metropolis–Hastings

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

$$r_{mh}(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}^*)q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^*)}{\pi(\boldsymbol{\gamma})q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})}\right\}, \tag{10}$$

and otherwise remains in the old state $\boldsymbol{\gamma}$. This will generate a Markov chain which, given the chain is irreducible and aperiodic, will have π as stationary distribution. Theoretical results related to convergence of MCMC based estimates can be found in e.g. Tierney (1996). Note that the discrete finite space of models make these results easily applicable in our case.

Given that the γ_j's are binary, changes correspond to swaps between the values 0 and 1. One can address various options for generating proposals. A simple proposal is to first select the number of components to change, e.g. $S \sim \text{Unif}\{\zeta, \ldots, \eta\}$, followed by a sample of size S without replacement from $\{1, \ldots, p\}$. This implies that in (10) the proposal probability for switching from $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma}^*$ becomes symmetric, simplifying calculation of the acceptance probability. Other possibilities for proposals are summarized in Table 1, allowing, among others, different probabilities of swapping for the different components. Such probabilities can for instance be associated with marginal inclusion probabilities from a preliminary MCMC run.

3.2. MJMCMC—the mode jumping MCMC

The main problem with the standard Metropolis–Hastings algorithms is the trade-off between possibilities of large jumps (by which we understand proposals with a large neighborhood) and high acceptance probabilities. Large jumps will typically result in proposals with low probabilities. In a continuous setting, Tjelmeland and Hegstad (1999) solved this by introducing local optimization after large jumps, which results in proposals with higher acceptance probabilities. We adapt this approach to the discrete model selection setting by the following algorithm:

Algorithm 1 Mode jumping MCMC

1: Generate a large jump $\boldsymbol{\chi}_0^*$ according to a proposal distribution $q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})$.
2: Perform a local optimization, defined through $\boldsymbol{\chi}_k^* \sim q_o(\boldsymbol{\chi}_k^*|\boldsymbol{\chi}_0^*)$.
3: Perform a small randomization to generate the proposal $\boldsymbol{\gamma}^* \sim q_r(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)$.
4: Generate backwards auxiliary variables $\boldsymbol{\chi}_0 \sim q_l(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)$, $\boldsymbol{\chi}_k \sim q_o(\boldsymbol{\chi}_k|\boldsymbol{\chi}_0)$.
5: Put

$$\boldsymbol{\gamma}' = \begin{cases} \boldsymbol{\gamma}^* & \text{with probability } r_{mh}(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*); \\ \boldsymbol{\gamma} & \text{otherwise,} \end{cases}$$

where

$$r_{mh}^*(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}^*)q_r(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)}{\pi(\boldsymbol{\gamma})q_r(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)}\right\}. \tag{11}$$

Here a large jump corresponds to changing a large number of γ_j's while the local optimization will be some iterative procedure based on, at each iteration, changing a small number of components until a local mode is reached.

The procedure is illustrated in Fig. 1 where the backward sequence $\boldsymbol{\gamma}^* \to \boldsymbol{\chi}_0 \to \boldsymbol{\chi}_k \to \boldsymbol{\gamma}$, needed for calculating the acceptance probability, is included. For this algorithm, three proposals need to be specified; $q_l(\cdot|\cdot)$ specifying the first large jump, $q_o(\cdot|\cdot)$ specifying the local optimizer, and $q_r(\cdot|\cdot)$ specifying the last randomization, all to be described in more detail below.

π-invariance of the MJMCMC procedures is given by the following theorem (based on similar arguments as in Storvik, 2011; Chopin et al., 2013):

Theorem 1. *Assume $\boldsymbol{\gamma} \sim \pi(\cdot)$ and $\boldsymbol{\gamma}'$ is generated according to Algorithm 1. Then $\boldsymbol{\gamma}' \sim \pi(\cdot)$.*

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

Type	Proposal	Label
1	$\frac{\prod_{j\in\{j_1,...,j_S\}}\rho_j}{\binom{p}{S}(\eta-\zeta+1)}$	*Random change with random size of the neighborhood*
2	$\frac{\prod_{j\in\{j_1,...,i_S\}}\rho_j}{\binom{p}{S}}$	*Random change with fixed size of the neighborhood*
3	$\frac{1}{\binom{p}{S}(\eta-\zeta+1)}$	*Swap with random size of the neighborhood*
4	$\binom{p}{S}^{-1}$	*Swap with fixed size of the neighborhood*
5	$\frac{1-\mathbb{1}\left(\sum_j^p\gamma_j=p\right)}{p-\sum_j^p\gamma_j+\mathbb{1}\left(\sum_j^p\gamma_j=p\right)}$	*Uniform addition of a covariate*
6	$\frac{1-\mathbb{1}\left(\sum_j^p\gamma_j=0\right)}{\sum_j^p\gamma_j+\mathbb{1}\left(\sum_j^p\gamma_j=0\right)}$	*Uniform deletion of a covariate*

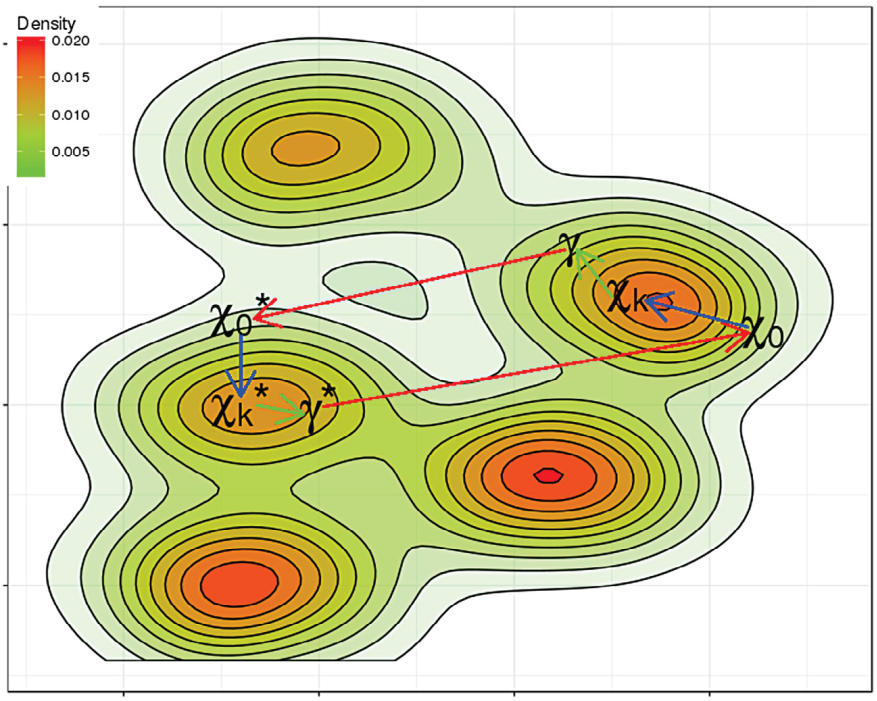

Fig. 1. Graphical illustration of a MJMCMC step with a large jump followed by a locally optimized proposal. The red arrows correspond to the large jumps, the blue arrows correspond to local optimization, the green arrows correspond the randomization steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Proof. Since $\boldsymbol{\gamma} \sim \pi(\cdot)$ and $(\boldsymbol{\chi}_0^*, \boldsymbol{\chi}_k^*) \sim q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})q_o(\boldsymbol{\chi}_k^*|\boldsymbol{\chi}_0^*)$ we have that

$$(\boldsymbol{\gamma}, \boldsymbol{\chi}_0^*, \boldsymbol{\chi}_k^*) \sim \pi(\boldsymbol{\gamma})q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})q_o(\boldsymbol{\chi}_k^*|\boldsymbol{\chi}_0^*) \equiv \bar{\pi}(\boldsymbol{\gamma}, \boldsymbol{\chi}_0^*, \boldsymbol{\chi}_k^*).$$

We may now consider $(\boldsymbol{\gamma}^*, \boldsymbol{\chi}_0, \boldsymbol{\chi}_k)$ as a proposal in the extended space, generated according to the distribution $q_r(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)q_l(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)q_o(\boldsymbol{\chi}_k|\boldsymbol{\chi}_0)$. An ordinary Metropolis–Hastings iteration with respect to $\bar{\pi}(\boldsymbol{\gamma}, \boldsymbol{\chi}_0^*, \boldsymbol{\chi}_k^*)$ is then to accept $(\boldsymbol{\gamma}^*, \boldsymbol{\chi}_0, \boldsymbol{\chi}_k)$ with probability $r_{mh}^* = \min\{1, \alpha_{mh}^*\}$ where

$$\begin{aligned}\alpha_{mh}^* &= \frac{\bar{\pi}(\boldsymbol{\gamma}^*, \boldsymbol{\chi}_0, \boldsymbol{\chi}_k)q_r(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})q_o(\boldsymbol{\chi}_k^*|\boldsymbol{\chi}_0^*)}{\bar{\pi}(\boldsymbol{\gamma}, \boldsymbol{\chi}_0^*, \boldsymbol{\chi}_k^*)q_r(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)q_l(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)q_o(\boldsymbol{\chi}_k|\boldsymbol{\chi}_0)} \\ &= \frac{\pi(\boldsymbol{\gamma}^*)q_l(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)q_o(\boldsymbol{\chi}_k|\boldsymbol{\chi}_0)q_r(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})q_o(\boldsymbol{\chi}_k^*|\boldsymbol{\chi}_0^*)}{\pi(\boldsymbol{\gamma})q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})q_o(\boldsymbol{\chi}_k^*|\boldsymbol{\chi}_0^*)q_r(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)q_l(\boldsymbol{\chi}_0|\boldsymbol{\gamma}^*)q_o(\boldsymbol{\chi}_k|\boldsymbol{\chi}_0)} = \frac{\pi(\boldsymbol{\gamma}^*)q_r(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)}{\pi(\boldsymbol{\gamma})q_r(\boldsymbol{\gamma}^*|\boldsymbol{\chi}_k^*)},\end{aligned}$$

proving the algorithm has $\bar{\pi}(\cdot)$ as invariant distribution. Since this distribution has $\pi(\cdot)$ as marginal distribution it follows that $\boldsymbol{\gamma}' \sim \pi(\cdot)$. □

Table 2
Illustration of a MJMCMC step with a large jump followed by a locally optimized proposal. The red components correspond to components swapped in the large jumps, the blue components to the ones changed in the optimizer, the green components of $\boldsymbol{\gamma}$ to the randomization step. (For interpretation of the references to color in this table legend, the reader is referred to the web version of this article.)

	Forward		**Backward**	
	Model	$\log(p(\mathbf{y}\|\boldsymbol{\gamma}))$	Model	$\log(p(\mathbf{y}\|\boldsymbol{\gamma}))$
Initial mode	$\boldsymbol{\gamma}$ =1010110111	1606.21	$\boldsymbol{\gamma}^*$ =1101100001	1612.27
Large jump	$\boldsymbol{\chi}_0^*$ =1001110001	1541.51	$\boldsymbol{\chi}_0$ =1110100111	1608.55
Optimize	$\boldsymbol{\chi}_k^*$ =1101100000	1616.16	$\boldsymbol{\chi}_k$ =1010100110	1612.00
Randomize	$\boldsymbol{\gamma}^*$ =1101100001	1612.27	$\boldsymbol{\gamma}$ =1010110111	1606.21
Acceptance probability: $\min\{1, 541.11\}$, accept $\boldsymbol{\gamma}' = \boldsymbol{\gamma}^*$ =**1101100001**				

Note that neither the large jump distribution $q_l(\cdot)$ nor the optimization distribution $q_o(\cdot)$ (which can be both deterministic and stochastic) is involved in the acceptance probability. This gives great flexibility in the choice of these distributions.

Large jumps are not performed at each iteration, but rather through a composition of standard Metropolis–Hastings steps with local moves and large jumps. As a rule of thumb, based on suggestions of Tjelmeland and Hegstad (1999) and our own experience, we recommend that in not more than 5% of the iterations large jumps are performed. This is believed to provide a global Markov chain with both good mixing between the modes and accurate exploration of the regions around the modes. This in turn induces good performance of the algorithm in terms of the captured posterior mass for a given number of iterations. However, some tuning might well be required for the particular practical applications.

The mode jumping MCMC steps can be modified to include a mixture of different proposal kernels q_l, q_o, and q_r and parallelized using the multiple try MCMC idea. Technical details are given in Appendix A.

An illustrative example. Assume 10 covariates $x_1, \ldots, x_{10}$ and thus 1024 possible models. We generated $Y \sim N(1 + 10x_1 + 0.89x_8 + 1.43x_5, 1)$ with correlated binary covariates (see supplementary code for details, Appendix A) and 1000 observations. We used a Gaussian linear regression with a Zellner's g-prior (Zellner, 1986) with $g = 1000$. This model has tractable marginal likelihoods described in detail in Section 4. We consider an MJMCMC step with a large jump swapping randomly 4 components of $\boldsymbol{\gamma}$ and a local greedy search, changing only one component at a time, as optimization routine. The last randomization changes each component of $\boldsymbol{\gamma}$ independently with probability equal to 0.1. A typical MJMCMC step with locally optimized proposals is illustrated in Table 2.

Large jumps. A change is defined by the components that are to be swapped. A simple choice is to give all components an equal probability ρ to be swapped and independence between components, in which case

$$q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma}) = \prod_{j=1}^{p} \rho^{I_j}(1-\rho)^{1-I_j} = \rho^S(1-\rho)^{p-S},$$

where I_j is a binary variable equal to 1 if component γ_j is to be swapped and $S = \sum_{j=1}^{p} I_j$ is the number of components to be swapped. An alternative is to first draw the number of components, S, to swap according to a distribution $q_S(\cdot)$ and thereafter choose (uniformly) among the possible changes of size S. Table 1 describes different ways of making large jumps where tuning parameters should be chosen such that the probability of a high value of S is large.

Optimization. In order to increase the quality of proposals and consequently both improve the acceptance ratio and increase the probability of escaping from local optima, the large jump is followed by a local optimization step. Typically, $q_o(\cdot)$ contains many iterations, generating intermediate states $\boldsymbol{\chi}_0^* \to \boldsymbol{\chi}_1^* \to \cdots \to \boldsymbol{\chi}_k^*$ but none of these intermediate states are needed for the final evaluation. Different local learning and optimization routines can be applied for the generation of $\boldsymbol{\chi}_k^*$, both deterministic and stochastic ones, see Appendix A.2 for further details. We will consider several feasible computationally options: local greedy optimization, local simulated annealing (SA) optimization, and local MCMC methods.

Randomization. A last randomization step defined through $q_r(\cdot)$ is needed in order to make the move back from $\boldsymbol{\gamma}^*$ to $\boldsymbol{\gamma}$ feasible. We typically use randomizing kernels with a high mass on a small neighborhood around the mode but with a positive probability for any change. The two possible appropriate kernels from Table 1 are the random change of either random $S \sim \text{Unif}\{1, \ldots, p\}$ or deterministic $S = p$ number of components with reasonably small but positive probabilities $0 < \rho_i \ll 1$. This guarantees that the MJMCMC procedure is irreducible in Ω.

Symmetric large jumps. In order for the acceptance probability to be high, it is crucial that the auxiliary variables in the reverse sequence $\boldsymbol{\chi} = (\boldsymbol{\chi}_0, \boldsymbol{\chi}_k)$ make $\boldsymbol{\gamma}$ plausible ($q_r(\boldsymbol{\gamma}|\boldsymbol{\chi}_k)$ should be large in (11)). This may be difficult to achieve because the backwards large jump has no guarantee to be close to the current state. One way to achieve this is to choose $q_l(\boldsymbol{\chi}_0^*|\boldsymbol{\gamma})$ to be symmetric, increasing the probability of returning close to the initial mode in the reverse large jump. The symmetry is achieved by swapping the same set of $\boldsymbol{\gamma}_j$'s in the large jumps in the forward simulation as in the backwards simulation.

We record the components I that have been swapped. In our current implementation we require that only the components that do not correspond to I can be changed in optimization transition kernels. The following algorithm is a modification of Algorithm 1 taking a symmetric large jump into account.

Algorithm 2 Mode jumping MCMC with symmetric backwards jump

1: Generate a large jump $\boldsymbol{\chi}_0^*$ by first generating a set $I \subset \{1, \ldots, p\} \sim q_I(\cdot)$ defining the components to be swapped.
2: Perform a local optimization, defined through $\boldsymbol{\chi}_k^* \sim q_o(\boldsymbol{\chi}_k^* | \boldsymbol{\chi}_0^*)$.
3: Perform a small randomization to generate the proposal $\boldsymbol{\gamma}^* \sim q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*)$.
4: Define the backwards large jump $\boldsymbol{\chi}_0$ through swapping the components I in $\boldsymbol{\gamma}^*$.
5: Generate $\boldsymbol{\chi}_k \sim q_o(\boldsymbol{\chi}_k | \boldsymbol{\chi}_0)$.
6: Put

$$\boldsymbol{\gamma}' = \begin{cases} \boldsymbol{\gamma}^* & \text{with probability } r_m(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*); \\ \boldsymbol{\gamma} & \text{otherwise,} \end{cases}$$

where

$$r_{mh}^*(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}^*) q_r(\boldsymbol{\gamma} | \boldsymbol{\chi}_k)}{\pi(\boldsymbol{\gamma}) q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*)}\right\}. \tag{12}$$

The following theorem shows that also this algorithm also is π-invariant.

Theorem 2. *Assume $\boldsymbol{\gamma} \sim \pi(\cdot)$ and $\boldsymbol{\gamma}'$ is generated according to Algorithm 2 . Then $\boldsymbol{\gamma}' \sim \pi(\cdot)$.*

Proof. The stochastic auxiliary components are now I, $\boldsymbol{\chi}_k^*$ and $\boldsymbol{\chi}_k$ where $\boldsymbol{\chi}_0^*$ and $\boldsymbol{\chi}_0$ are deterministic functions of $(\boldsymbol{\gamma}, I)$ and $(\boldsymbol{\gamma}^*, I)$, respectively. We have

$$(\boldsymbol{\gamma}, I, \boldsymbol{\chi}_k^*) \sim \pi(\boldsymbol{\gamma}) q_I(I) q_o(\boldsymbol{\chi}_k^* | \boldsymbol{\chi}_0^*) \equiv \bar{\pi}(\boldsymbol{\gamma}, I, \boldsymbol{\chi}_k^*).$$

We may now consider $(\boldsymbol{\gamma}^*, I, \boldsymbol{\chi}_k)$ as a proposal in the extended space, generated according to the distribution $q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*) q_o(\boldsymbol{\chi}_k | \boldsymbol{\chi}_0)$. An ordinary Metropolis–Hastings iteration with respect to $\bar{\pi}(\boldsymbol{\gamma}, I, \boldsymbol{\chi}_k^*)$ is then to accept $(\boldsymbol{\gamma}^*, I, \boldsymbol{\chi}_k)$ with probability $r_{mh}^* = \min\{1, \alpha_{mh}^*\}$ where

$$\begin{aligned} \alpha_{mh}^* &= \frac{\bar{\pi}(\boldsymbol{\gamma}^*, I, \boldsymbol{\chi}_k) q_r(\boldsymbol{\gamma} | \boldsymbol{\chi}_k) q_o(\boldsymbol{\chi}_k^* | \boldsymbol{\chi}_0^*)}{\bar{\pi}(\boldsymbol{\gamma}, I, \boldsymbol{\chi}_k^*) q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*) q_o(\boldsymbol{\chi}_k | \boldsymbol{\chi}_0)} \\ &= \frac{\pi(\boldsymbol{\gamma}^*) q_I(I) q_o(\boldsymbol{\chi}_k | \boldsymbol{\chi}_0) q_r(\boldsymbol{\gamma} | \boldsymbol{\chi}_k) q_o(\boldsymbol{\chi}_k^* | \boldsymbol{\chi}_0^*)}{\pi(\boldsymbol{\gamma}) q_I(I) q_o(\boldsymbol{\chi}_k^* | \boldsymbol{\chi}_0^*) q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*) q_o(\boldsymbol{\chi}_k | \boldsymbol{\chi}_0)} = \frac{\pi(\boldsymbol{\gamma}^*) q_r(\boldsymbol{\gamma} | \boldsymbol{\chi}_k)}{\pi(\boldsymbol{\gamma}) q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*)}, \end{aligned}$$

proving the algorithm has $\bar{\pi}(\cdot)$ as invariant distribution. Since this distribution has $\pi(\cdot)$ as marginal distribution it follows that $\boldsymbol{\gamma}' \sim \pi(\cdot)$. □

3.3. Delayed acceptance

The most computationally demanding parts of the MJMCMC algorithms are the forward and backward optimizations. In many cases, the proposal generated through the forward optimization may lead to a very small value of $\pi(\boldsymbol{\gamma}^*)$ resulting in a low acceptance probability regardless of the way the backwards auxiliary variables are generated. In such cases, one would like to reject directly without the need for performing the backward optimization. Such a scheme can be constructed by the use of the delayed acceptance procedure (Christen and Fox, 2005; Banterle et al., 2015). We then have:

Theorem 3. *Assume $\boldsymbol{\gamma} \sim \pi(\cdot)$ and assume $\boldsymbol{\gamma}^*$ is generated according to either Algorithm 1 or Algorithm 2 . Accept $\boldsymbol{\gamma}^*$ if both*

1. *$\boldsymbol{\gamma}^*$ is preliminary accepted with a probability* $\min\{1, \frac{\pi(\boldsymbol{\gamma}^*)}{\pi(\boldsymbol{\gamma})}\}$
2. *and is finally accepted with a probability* $\min\{1, \frac{q_r(\boldsymbol{\gamma} | \boldsymbol{\chi}_k)}{q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*)}\}$.

Then also $\boldsymbol{\gamma}^* \sim \pi(\cdot)$.

Proof. We have that

$$\alpha_{mh}^*(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*) = \alpha_{mh}^1(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*) \times \alpha_{mh}^2(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*)$$

where

$$\alpha_{mh}^1(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*) = \frac{\pi(\boldsymbol{\gamma}^*)}{\pi(\boldsymbol{\gamma})}, \quad \alpha_{mh}^2(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}_k^*) = \frac{q_r(\boldsymbol{\gamma} | \boldsymbol{\chi}_k)}{q_r(\boldsymbol{\gamma}^* | \boldsymbol{\chi}_k^*)}.$$

Since $\alpha^j_{mh}(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*; \boldsymbol{\chi}_k, \boldsymbol{\chi}^*_k) = [\alpha^j_{mh}(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}; \boldsymbol{\chi}^*_k, \boldsymbol{\chi}_k)]^{-1}$ for $j = 1, 2$, it follows by the general results in Banterle et al. (2015) that we obtain an invariant kernel with respect to $\bar{\pi}$. □

In general the total acceptance rate will be smaller than without delayed acceptance (Banterle et al., 2015 remark 1), but the gain by avoiding a backwards optimization step if not accepted in the preliminary step can compensate on this.

3.4. Calculation of marginal densities

In practice exact calculation of the marginal density can only be performed in simple models such as linear Gaussian ones, so alternatives need to be considered. One approach is to use estimators that are accurate enough to neglect the approximation errors involved. Such approximative approaches have been used in various settings of Bayesian variable selection and Bayesian model averaging. Laplace's method (Tierney and Kadane, 1986) has been widely used, but is based on rather strong assumptions. The harmonic mean estimator (Newton and Raftery, 1994) is an easy to implement MCMC based method but can give high variability in the estimates. Chib's method (Chib, 1995), and its extension (Chib and Jeliazkov, 2001), have gained increasing popularity and can be very accurate provided enough MCMC iterations are performed. Approximate Bayesian Computation (Marin et al., 2012) has also been considered in this context, being much faster than MCMC alternatives, but also giving cruder approximations. Variational methods (Jordan et al., 1999) provide lower bounds for the marginal likelihoods and have been used for model selection in e.g. mixture models (McGrory and Titterington, 2007). Integrated nested Laplace approximation (INLA, Rue et al., 2009) provides accurate estimates of marginal likelihoods within the class of latent Gaussian models. In the context of generalized linear models, BIC type approximations can be used.

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

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

3.5. Parallelization and tuning parameters of the search

With large number of potential explanatory variables it is important to be able to utilize multiple cores and GPUs of either local machines or clusters in parallel. General principles of utilizing multiple cores in local optimization are provided in Eksioglu et al. (2002). At every step of the local optimization within the large jump steps we allow to simultaneously draw several proposals with respect to a certain transition kernel during the optimization procedure and then sequentially calculate the transition probabilities as the proposed models are evaluated by the corresponding CPUs, GPUs or clusters in the order they are returned. In those iterations where no large jumps are performed, we are utilizing multiple cores by means of addressing multiple try MCMC to explore the solutions around the current mode. The parallelization strategies are described in detail in Appendix A.

In practice, tuning parameters of the local optimization routines such as the choice of the neighborhood, generation of proposals within it, the cooling schedule for *simulated annealing* (Michiels et al., 2010) or number of steps in greedy optimization also become crucially important and it yet remains unclear whether we can optimally tune them before or during the search. Mixing of proposals from Table 1 and of optimizers is also possible. Tuning the probabilities of addressing these different options can be beneficial. Such tuning is a sophisticated mathematical problem, which we are not trying to resolve optimally within this paper, however we suggest a simple practical idea for obtaining reasonable solutions. Within the BAS algorithm, an important feature was to utilize the marginal inclusion probabilities of different covariates. We have introduced this in our algorithms as well by allowing insertion of estimates of the ρ_i's in proposals given in Table 1 based on some burn-in period. They then correspond to the marginal inclusion probabilities after burn-in shifted with some small ϵ from 0 and 1 if necessary in order to guarantee irreducibility. Additional literature review on search parameter tuning can be found in Luo (2016).

4. Experiments

In this section we are going to apply the MJMCMC algorithm to different data sets and analyze the results in relation to other algorithms. Linear regression is addressed through the U.S. Crime Data (Raftery et al., 1997) and a protein activity

data (Clyde et al., 1998). Logistic regression is considered in a simulated example based on a data set and through an Arabidopsis epigenetic data set. The Arabidopsis example also includes random effects.

We compare the performance of our approach to competing MCMC methods such as the MCMC model composition algorithm (MC^3, Madigan et al., 1995; Raftery et al., 1997) and the random-swap (RS) algorithm (Clyde et al., 2011) as well as the BAS algorithm (Clyde et al., 2011). Both MC^3 and RS are simple MCMC procedures based on the standard Metropolis–Hastings algorithm with proposals chosen correspondingly as an inversion or a random change of one coordinate in $\boldsymbol{\gamma}$ at a time (Clyde et al., 2011). BAS carries out sampling without repetition from the space of models with respect to the adaptively updated marginal inclusion probabilities. For one of the examples, also a comparison with the ESS++ software (evolutionary stochastic search, Bottolo et al., 2011) is made. For the cases when full enumeration of the model space is possible we additionally compare all of the aforementioned approaches to the benchmark TOP method that consists of the best quantile of models in terms of the posterior probability for the corresponding number of addressed models $\|\mathbb{V}\|$ and cannot by any chance be outperformed in terms of the posterior mass captured.

The different algorithms that are compared are implemented in different programming languages, making it difficult to compare CPU time fairly. We have therefore focused on both the total number of visited models and the number of unique models visited, since this is the main computational burden (marginal likelihood values of visited models can be stored). The number of models visited for MJMCMC includes all of the models visited during global and local moves as well as local combinatorial optimization, hence the comparison on the same number of totally visited and uniquely visited models is fair.

Following Clyde et al. (2011), approximations for model probabilities (8) and marginal inclusion probabilities (9) based on a subspace of models are further referred to as RM (renormalized) approximations, whilst the corresponding MCMC based approximations (7) are referred to as MC approximations. The validation criteria addressed include root mean squared errors and bias of parameters of interest based on multiple replications of each algorithm, similar to Clyde et al. (2011). In addition to marginal inclusion probabilities, we also include a global measure

$$C(\boldsymbol{\gamma}) = \frac{\sum_{\boldsymbol{\gamma}' \in \mathbb{V}} p(\boldsymbol{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')}{\sum_{\boldsymbol{\gamma}' \in \Omega} p(\boldsymbol{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')}, \tag{13}$$

describing the fraction of probability mass contained in the subspace $\mathbb{V}$. This measure allows us to address how well the search works in terms of capturing posterior mass within a given model space. By formula (8) maximization of $C(\boldsymbol{\gamma})$ automatically induces minimization of the bias in terms of posterior marginal model probabilities, which vanishes gradually when $C(\boldsymbol{\gamma}) \to 1$.

Mixtures of different proposals from Table 1 and local optimizers mentioned in Section 3.2 were used in the studied examples in the MJMCMC algorithm. A validation of the gain in using such mixtures is given in Example 4.1, where we address both MJMCMC with mixtures and a simpler version where only one choice of proposal distributions is used (the details are given in the example). The details on the choices and frequencies of different proposals for the other examples are given in Tables B.1–B.5 in Appendix B. The choices are based on some tuning on a simulated data example, reported in Appendix C.1. Further small adaptations were made in some of the examples. Generally speaking, we cannot claim that the choices of the tuning parameters are optimal. It is rather some subjectively rational choice.

4.1. Example 1

Here we address the U.S. Crime data set, first introduced by Vandaele (1978) and stated to be a test bed for evaluation of methods for model selection (Raftery et al., 1997). The data set consists of $n = 47$ observations on 15 covariates and the responses, which are the corresponding crime rates. We will compare performance of the algorithms based on a linear Bayesian regression model using a Zellner's g-prior (Zellner, 1986) with $g = 47$. This implies that the marginal likelihood is of the following form:

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) \propto (1+g)^{(n-p-1)/2}(1+g[1-R^2_{\boldsymbol{\gamma}}])^{-(n-1)/2}, \tag{14}$$

where $R^2_{\boldsymbol{\gamma}}$ is the usual coefficient of determination of a linear regression model. With this scaling, the marginal likelihood of the null model (the model containing no covariates) is 1.0.

This is a sophisticated example with a total of $2^{15} = 32{,}768$ potential models and with several local modes. As a result, all simple MCMC methods easily get stuck and have extremely poor performances in terms of the captured mass and precision of both the marginal posterior inclusion probabilities and the posterior model probabilities. Table 3 shows the RMSE (scaled by 10^2) for the model parameters over 100 repeated runs for each algorithm. The True column contains the true marginal inclusion probabilities (obtained from full enumeration) while the TOP column shows the RMSE results based on the 3276 models with *highest* posterior probabilities (about 10% of the total number of models). The MJMCMC columns show the results based on using mixtures of proposals and optimizers (see Tables B.1–B.5 for details) while the MJMCMC* results are based on one specific choice of proposals with swaps of 4 components at a time for the large jumps (Type 4 in Table 1) and a local greedy optimizer changing two components at a time with a last randomization of type 2 (Table 1). For the standard MCMC steps, a type 4 with two changing components was used.

For this example, both the MC^3 and the RS methods got stuck in some local modes and for the 3276 models only 829/1071 unique models where visited. These algorithms did not reach 3276 unique models within a reasonable time for this example

Table 3
Average root mean squared error (RMSE) over the 100 repeated runs of every algorithm on the Crime data (example 1); the values reported in the table are RMSE $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$. $C(\boldsymbol{\gamma})$ is defined in Eq. (13). Tot is the total number of visited models, while Eff is the number of unique models visited during the iterations of the algorithms (for the TOP column all 2^{15} models were visited but the RMSE are based on the best 3276 models). RM corresponds to using the renormalization procedure Eq. (8) while MC corresponds to using the MC procedure Eq. (7). MJMCMC^2 differs from MJMCMC in the number of unique models visited (Eff) while MJMCMC* corresponds to a run with no mixtures of proposals. The corresponding biases are reported in Appendix C in Table C.2.

Par	True	TOP	MJMCMC		MJMCMC^2		BAS	MC^3		RS		MJMCMC*	
Δ	π_j	–	RM	MC	RM	MC	RM	MC	RM	MC	RM	RM	MC
γ_8	0.16	3.51	6.57	10.68	5.11	10.29	5.21	6.49	3.49	5.87	3.31	6.23	9.06
γ_{13}	0.16	3.34	7.46	10.54	5.60	10.19	6.26	8.62	3.39	8.83	3.05	6.38	10.54
γ_{14}	0.19	3.24	8.30	12.43	6.30	12.33	6.20	6.58	2.55	6.22	2.46	7.15	10.91
γ_{12}	0.22	3.27	6.87	13.61	5.57	13.64	3.10	5.81	6.23	4.93	5.27	5.29	10.93
γ_5	0.23	2.56	6.30	13.45	4.59	13.65	1.84	6.07	13.05	5.13	12.77	5.39	10.90
γ_9	0.23	3.27	9.49	16.21	7.40	16.21	9.27	5.99	2.99	5.70	2.60	7.68	11.06
γ_7	0.29	2.31	4.37	13.63	3.45	12.73	2.28	4.74	9.61	3.46	9.70	3.91	10.10
γ_4	0.30	1.57	6.18	19.22	3.79	17.31	0.99	13.24	21.84	13.53	21.48	4.63	13.22
γ_6	0.33	1.92	8.61	19.71	6.14	19.49	3.11	10.19	7.47	10.99	7.12	5.87	15.43
γ_1	0.34	2.51	11.32	22.68	7.29	20.50	8.43	22.89	25.19	23.63	24.71	7.58	12.97
γ_3	0.39	0.43	3.95	11.13	2.38	6.99	5.02	21.48	30.24	21.39	29.94	2.99	12.66
γ_2	0.57	1.58	5.92	13.21	3.82	9.03	13.78	30.81	37.57	29.27	37.15	5.11	14.04
γ_{11}	0.59	0.58	3.57	13.49	2.37	15.94	4.04	11.88	21.79	11.16	21.31	2.77	12.77
γ_{10}	0.77	3.25	7.62	7.28	5.97	4.78	15.45	21.83	19.18	20.53	19.65	6.41	14.27
γ_{15}	0.82	3.48	9.23	4.45	6.89	5.85	14.50	69.68	76.81	69.19	76.30	6.75	14.76
$C(\boldsymbol{\gamma})$	1.00	0.86	0.58	0.58	0.71	0.71	0.66	0.10	0.10	0.10	0.10	0.60	0.60
Eff	2^{15}	3276	1909	1909	3237	3237	3276	829	829	1071	1071	3264	3264
Tot	2^{15}	3276	3276	3276	5936	5936	3276	3276	3276	3276	3276	4295	4295

(most likely the algorithm could not escape from local extrema), hence such a scenario is not reported. For this example MJMCMC gives a much better performance than the other MCMC methods in terms of both MC and RM based estimations with respect to the posterior mass captured, $C(\boldsymbol{\gamma})$. With a total of 3276 visited, BAS slightly outperforms MJMCMC. However, when running MJMCMC so that the number of *unique* models visited ($\|\mathbb{V}\|$) are comparable with BAS, MJMCMC gives better results (columns marked with MJMCMC^2 in Table 3). The comparison is performed in terms of posterior mass captured, biases and root mean squared errors for both posterior model probabilities and marginal inclusion probabilities (Table 3).

BAS has the property of always visiting new unique models, whilst all MCMC based procedures tend to do revisiting with respect to the corresponding posterior probabilities. When generating a proposal is much cheaper than estimating marginal likelihoods of the model (which is usually the case, also in this example) and we are storing the results for the already visited models, having generated a bit more models by MJMCMC does not seem to be a serious issue. Those unique models that are visited have a higher posterior mass than those suggested by BAS (for the same number of models visited). Furthermore MJMCMC (like BAS) can escape from local modes.

Also the results based on no mixture of proposals (MJMCMC* in the table) are much better than standard MCMC methods, however the results obtained by the MJMCMC algorithm with a mixture of proposals were even better. We have tested this on some other examples too and the use of mixtures was always beneficial and thus recommended. For this reason only the cases with mixtures of proposals are addressed in other experiments.

4.2. Example 2

In this example we are considering a new simulated data set for logistic regression. We generated $p = 20$ covariates as a mixture of binary and continuous variables. The correlation structure is shown in Fig. 2 while the full details of how the data was generated is given in Appendix B.1.

A total of $2^{20} = 1{,}048{,}576$ potential models need to be considered in this case. Additionally, in this example $n = 2000$, which makes estimation of a single model significantly slower than in the previous example. For $\boldsymbol{\gamma}$ we use the binomial prior (4) with $q = 0.057$. We are in this case using the BIC-approximation for the marginal likelihood,

$$\log \widehat{p}(\mathbf{y}|\boldsymbol{\gamma}) = \log \hat{p}(\mathbf{y}|\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}) - \frac{n}{2}\log(|\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}|), \tag{15}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ is the maximum likelihood (or MAP) estimate for the β_j's involved and $|\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}|$ is the number of parameters. This choice was made in order to compare the results with implementations of BAS, RS and MC^3 available in the supplementary to Clyde et al. (2011), where this approximation is considered. In that way, the model search procedures are compared based on the same selection criterion.

Some of the covariates involved have large correlations. This induces both multimodality within the space of models and sparsity of the locations of the modes and creates an interesting example for comparison of different search strategies. As

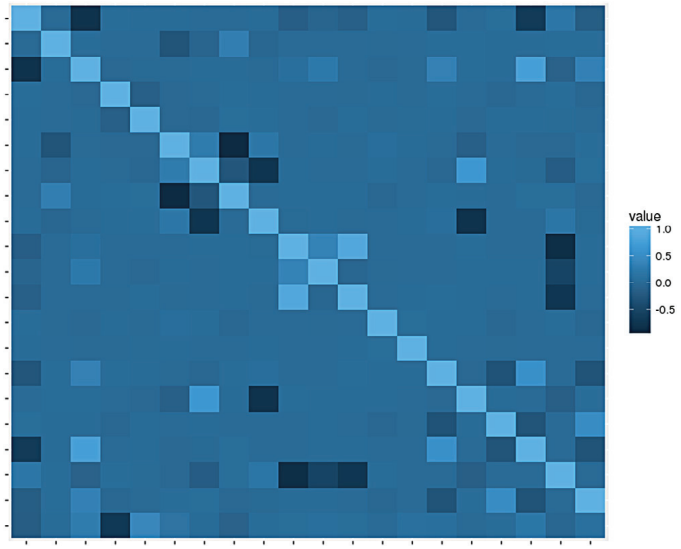

Fig. 2. Correlation structure of the covariates in Example 3.

Table 4
Average root mean squared error (RMSE) from the 100 repeated runs of every algorithm on the simulated logistic regression data (example 2); the values reported in the table are RMSE $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$. See the caption of Table 3 for further details. The corresponding biases are reported in Appendix C in Table C.2.

Par	True	TOP	MJMCMC		MJMCMC^2		BAS	BAS-RS	RS	
Δ	π_j	–	RM	MC	RM	MC	RM	RM	RM	MC
γ_6	0.29	0.00	7.38	15.54	4.54	16.62	6.47	3.67	6.01	2.11
γ_8	0.31	0.00	6.23	15.50	3.96	16.94	5.58	3.02	5.37	2.55
γ_{12}	0.35	0.00	4.86	14.62	2.78	13.66	4.22	2.12	3.91	2.37
γ_{15}	0.35	0.00	4.55	15.24	2.56	15.45	4.66	1.64	3.40	2.56
γ_2	0.36	0.00	4.90	16.52	2.92	17.39	5.42	2.45	3.65	2.61
γ_{20}	0.37	0.00	4.82	14.35	2.66	14.08	3.32	1.80	4.15	2.18
γ_3	0.40	0.00	9.25	20.93	5.65	22.18	9.75	4.82	6.76	2.83
γ_{14}	0.44	0.00	3.14	17.54	1.58	16.24	3.73	1.30	1.33	2.93
γ_{10}	0.44	0.00	4.60	18.73	2.29	17.90	4.87	1.30	1.51	2.42
γ_5	0.46	0.00	3.10	17.17	1.53	16.97	4.06	1.51	1.09	2.85
γ_9	0.61	0.00	3.68	16.29	1.63	13.66	3.89	1.39	2.19	2.35
γ_4	0.88	0.00	5.66	6.70	3.74	6.26	6.60	5.57	7.61	2.15
γ_{11}	0.91	0.00	5.46	6.81	3.95	6.90	4.66	3.14	4.32	1.57
γ_1	0.97	0.00	1.90	1.74	1.35	1.34	2.43	1.96	2.30	1.1
γ_{13}	1.00	0.00	0.00	0.43	0.00	0.32	0.00	0.00	0.00	0.37
γ_7	1.00	0.00	0.00	0.57	0.00	0.41	0.00	0.00	0.00	0.33
γ_{16}	1.00	0.00	0.00	0.41	0.00	0.33	0.00	0.00	0.00	0.23
γ_{17}	1.00	0.00	0.00	0.43	0.00	0.39	0.00	0.00	0.00	0.23
γ_{18}	1.00	0.00	0.00	0.47	0.00	0.35	0.00	0.00	0.00	0.24
γ_{19}	1.00	0.00	0.00	0.52	0.00	0.36	0.00	0.00	0.00	0.41
C($\boldsymbol{\gamma}$)	1.00	1.00	0.72	0.72	0.85	0.85	0.74	0.85	0.68	0.68
Eff	2^{20}	10,000	5148	5148	9988	9988	10,000	10,000	1889	1889
Tot	2^{20}	10,000	9998	9998	19,849	19,849	10,000	10,000	10,000	10,000

one can see in Table 4, MJMCMC outperformed pure BAS by far both in terms of posterior mass captured and in terms of root mean square errors of marginal inclusion probabilities when based on the same number of unique models. MJMCMC outperformed RS as well. The latter got stuck in some local modes and could only reach 1889 unique models for the 10,000 models visited. We could not reach 10,000 unique models for the RS algorithm within a reasonable time for this example either (again most likely the algorithm could not escape from local extrema), hence such a scenario is not reported. Even for almost two times less originally visited models in $\mathbb{V}$, comparing to BAS, MJMCMC gives almost the same results in terms of the posterior mass captured and errors. MJMCMC, for the given number of unique models visited, did not outperform a combination of MCMC and BAS (BAS-RS), which is recommended by Clyde et al. (2011) for larger model spaces; both of them gave approximately identical results.

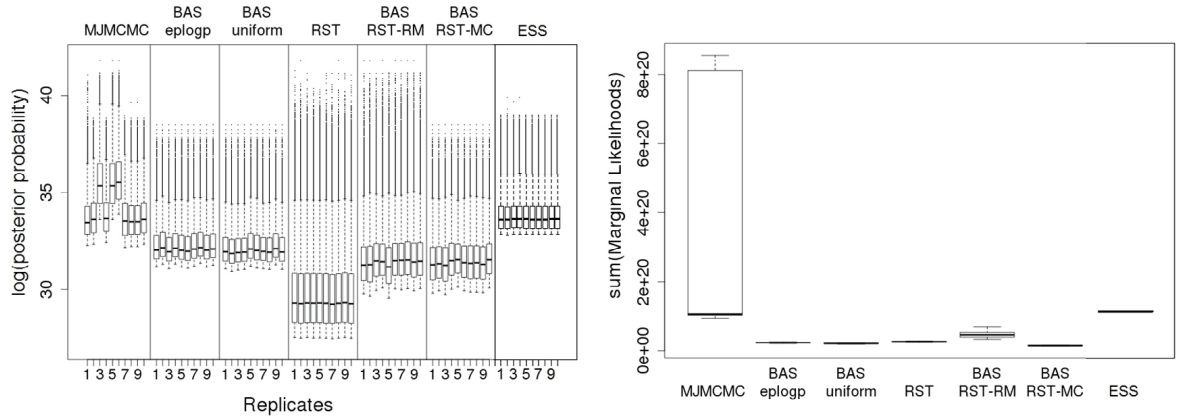

Fig. 3. Comparisons in the protein data of the log posterior probabilities of the top 100,000 models (left) and box-plots of the posterior mass captured (right) obtained by MJMCMC, BAS-eplogp, BAS-uniform, thinned version of Random Swap (RST), BAS with Monte Carlo estimates of inclusion probabilities from the RST samples (BAS-RST-MC), BAS re-normalized estimates of inclusion probabilities (BAS-RST-RM) from the RST samples, and ESS++ (ESS).

4.3. Example 3

This experiment is based on a much larger model space in comparison to all of the other examples. We address the protein activity data (Clyde et al., 1998) and consider all main effects together with the two-way interactions and quadratic terms of the continuous covariates resulting in 88 covariates in total. This corresponds to a model space of cardinality 2^{88}, a number far to high to perform full search through all models. This model space is additionally multimodal, which is the result of having high correlations between numerous of the addressed covariates (17 pairs of covariates have correlations above 0.95).

We analyzed the data set using Bayesian linear regression with the binomial prior (4) with $q = 0.5$ for $\boldsymbol{\gamma}$ and a Zellner's g-prior with $g = 96$ for $\boldsymbol{\beta}$ (the data has $n = 96$ observations). We then compared the performance of MJMCMC, BAS and RS. For this example we have also addressed the ESS++ algorithm (Bottolo et al., 2011).

The reported RS results are based on the RS algorithm run for 88×2^{20} iterations and a thinning rate of $\frac{1}{88}$ (named RST in Clyde et al. (2011)). BAS was run with several choices of initial sampling probabilities such as uniformly distributed within the model space one, eplogp adjusted (Clyde et al., 2011), and those based on RM and MC approximations obtained by the RST algorithm. For the first two initial sampling probabilities BAS was run for 2^{20} iterations. For the two latter (the BAS-RST-RM and BAS-RST-MC) algorithms) first RS was run for 88×2^{19} iterations providing 2^{19} models for estimating initial sampling probabilities and then BAS was run for the other 2^{19} iterations based on RM or MC estimates of the marginal inclusion probabilities. MJMCMC was run until 2^{20} unique models were obtained. ESS++ was run with default search settings until 2^{20} unique models were visited. All of the algorithms were replicated 10 times.

In Fig. 3 box-plots of the best 100,000 models captured by the corresponding replications of the algorithms as well as posterior masses captured by them are displayed. BAS with both uniform and eplogp initial sampling probabilities performed rather poorly in comparison to other methods, whilst BAS combined with RM approximations from RST did slightly better. ESS++ as well as MJMCMC show the most promising results. BAS with RM initial sampling probabilities usually managed to find models with the highest posterior probabilities, however MJMCMC in general captured by far higher posterior mass within the same amount of unique models addressed. Marginal inclusion probabilities obtained by the best run of MJMCMC with respect to mass (denominator of (8) with value 8.56×10^{20} in Fig. 3) are reported in Fig. 4, whilst those obtained by other methods can be found in Clyde et al. (2011). Since MJMCMC obtained the highest posterior mass, we expect that the corresponding RM estimates of the marginal inclusion probabilities are the least biased, moreover they perfectly agree with the MC approximations. Although MJMCMC in all of the obtained replications outperformed most of the competitors in terms of the posterior mass captured, it itself exhibited significant variation between the runs (right panel of Fig. 3). The latter issue can be explained by that we are only allowing visiting 3.39×10^{-19}% of the total model space in the addressed replications, which might be not enough to always converge to the same posterior mass captured. Note however that the variability in the results obtained from different runs of MJMCMC clearly indicates that more iterations are needed, while the other methods may indicate (wrongly) that sufficient iterations have been performed.

4.4. Example 4

In this example we illustrate how MJMCMC works for GLMM models. As illustration, we address genomic and epigenomic data on Arabadopsis. Arabadopsis is a plant model organism with a lot of genomic/epigenomic data easily available (Becker et al., 2011). At each position on the genome, a number of reads are allocated. At locations with a nucleotide of type cytosine (C), reads are either methylated or not. Our focus will be on modeling the amount of methylated reads through different

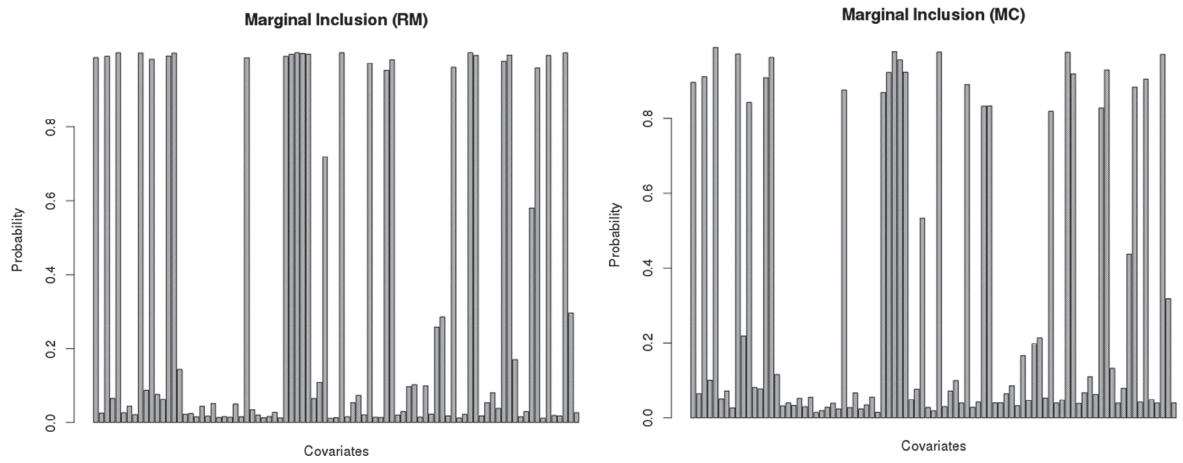

Fig. 4. Comparisons of RM (left) and MC (right) estimates of marginal posterior inclusion probabilities obtained by the best run of MJMCMC with 8.56e+20 posterior mass captured.

covariates including (local) genomic structures, gene classes and expression levels. The studied data was obtained from the NCBI GEO archive (Barrett et al., 2013).

We model the number of methylated reads $Y_i \in \{1, \ldots, R_i\}$ per loci $i = 1, \ldots, n$, where $R_i \in \mathbb{N}$ is the number of reads, through (1)–(3) by a Poisson distribution for the response and $n = 1502$. Since in general the ratio of methylated bases is low, we have preferred the Poisson distribution of the responses to the binomial. The mean η_i is modeled via the log link to the chosen covariates, including an offset defined by R_i per location, and a spatially correlated random effect δ_i which is modeled via an $AR(1)$ process with parameter $\rho \in \mathbb{R}$, namely $\delta_i = \rho\delta_{i-1} + \epsilon_i \in \mathbb{R}$ with $\epsilon_i \sim N(0, \sigma_\epsilon^2)$, $i = 1, \ldots, n$. Thus, we take into account spatial dependence structures of methylation rates along the genome as well as the variance of the observations not explained by the covariates. We use the binomial prior (4) with $q = 0.5$ for $\boldsymbol{\gamma}$ and the Gaussian prior for the regression coefficients:

$$\boldsymbol{\beta}|\boldsymbol{\gamma} \sim N_{p_\gamma}(\boldsymbol{\mu}_{\beta_\gamma}, \boldsymbol{\Sigma}_{\beta_\gamma}).$$

For the parameters within the random effects, we first reparametrize to $\psi_1 = \log \frac{1}{\sigma^2_{\epsilon,t}}(1-\rho^2)$, $\psi_2 = \log \frac{1+\rho}{1-\rho}$ and assume

$$\psi_1 \sim \text{logGamma}(1, 5 \times 10^{-5}) \tag{16}$$

and

$$\psi_2 \sim N(0, 0.15^{-1}). \tag{17}$$

Marginal likelihoods were for this example calculated through the INLA package (www.r-inla.org).

We have addressed $p = 13$ different covariates in addition to the intercept. We have considered a factor with 3 levels corresponding to whether a location belongs to a CGH, CHH or CHG genetic region, where H is either A, C or T and thus generating two covariates X_1 and X_2 corresponding to whether a location is CGH or CHH. A second factor indicates whether a distance to the previous cytosine nucleobase (C) in DNA is 1, 2, 3, 4, 5, from 6 to 20 or greater than 20 inducing the binary covariates $X_3 - X_8$. A third factor corresponds to whether a location belongs to a gene from a particular group of genes of biological interest, these groups are indicated as M_α, M_γ, M_δ or M_0 inducing 3 additional covariates $X_9 - X_{11}$. Finally, we have considered two binary covariates X_{12} and X_{13} represented by expression levels exceeding 3000 and 10,000, respectively. The cardinality of our search space Ω is $2^{13} = 8192$ for this example. The correlation structure between these 13 covariates is represented in Fig. 5.

As seen from Table 5 (TOP column), within just the 385 best unique models (2.35% of the total model space) we were able to capture almost full posterior mass for this problem. The model space, as shown in Fig. 6, has very few sparsely located modes in a quite large model space. In this example we compared MJMCMC and a simple MCMC algorithm, the latter was allowed to only swap one component per iteration (similar to the RS algorithm within the BAS package). This example contains most of the mass in just two closely located models as can be seen in Fig. 6. This is why a simple RS MCMC can capture essentially most of the mass after 10,000 iterations. At the same time there are a few small modes that lie a bit further from the region of the high concentration of mass, which the simple RS MCMC algorithm did not capture. Essentially, RS MCMC stayed within a few modes for most of the time, never being able to travel to the more remote parts of the model space and generated very few (155 on average) unique models. This number is here very low compared to the total number of models visited (10,000). If there were more sparsely located remote modes, the simple RS MCMC algorithm would run into the problems similar to those discussed in the previous examples and miss a significant amount of mass. For

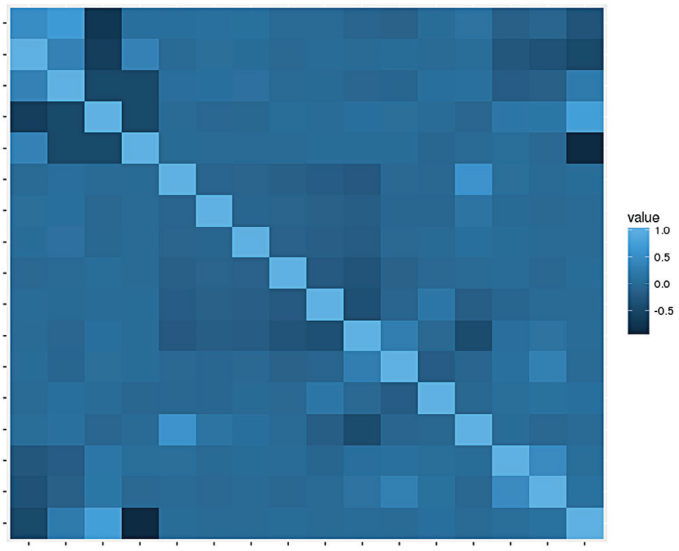

Fig. 5. Correlation structure of the covariates in Example 4.

Table 5
Average root mean squared error (RMSE) from the 100 simulated runs of MJMCMC on the epigenetic data (example 4); the values reported in the table are RMSE $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$.

Par	True	TOP	MJMCMC		RS MCMC	
Δ	π_j	RM	RM	MC	RM	MC
γ_4	0.0035	0.0005	0.0022	2.0416	0.0198	1.9768
γ_6	0.0048	0.0006	0.0051	2.0899	0.0257	1.9352
γ_7	0.0065	0.0006	0.0056	2.3459	0.0353	0.6887
γ_3	0.0076	0.0007	0.0017	3.3660	0.0353	1.2374
γ_8	0.0076	0.0007	0.0079	2.3279	0.0344	1.6163
γ_5	0.0096	0.0007	0.0075	2.3342	0.0455	1.7170
γ_{11}	0.0813	0.0007	0.0200	3.6851	0.1679	2.8022
γ_{12}	0.0851	0.0006	0.0134	2.7179	0.0766	1.9136
γ_9	0.1185	0.0008	0.0184	3.3149	0.1773	3.0463
γ_{10}	0.3042	0.0006	0.0071	9.4926	0.1106	3.7344
γ_{13}	0.9827	0.0002	0.0063	2.5350	0.0638	1.5681
γ_1	1.0000	0.0007	0.0000	4.7091	0.0000	1.2258
γ_2	1.0000	0.0000	0.0000	2.7343	0.0000	0.9971
C($\boldsymbol{\gamma}$)	1.0000	1.0000	0.9998	0.9998	0.9977	0.9977
Eff	8192	385	1758	1758	155	155
Tot	8192	385	3160	3160	10,000	10,000

MJMCMC, we ran the algorithm until 3160 models where visited, resulting in 1758 unique models. MJMCMC was able to capture the mass also from the remote small modes, adding a bit to the captured mass, slightly outperforming the simple RS MCMC algorithm. As can be seen in Table 5, MJMCMC outperformed the simple RS MCMC algorithm in terms of the errors of marginal model probabilities. Marginal inclusion probabilities in terms of RM are also more precise when MJMCMC is used. MC based approximations are also in this case worse than the RM versions, in this case with MJMCMC slightly worse.

According to marginal inclusion probabilities (π_j column in Table 5, obtained from full enumeration), factors of whether the location is CGH or CHH (γ_1 and γ_2) are both extremely significant, as well as the higher cut off for the level of expression (γ_{13}). Additionally, factors for M_α and M_δ groups of genes (γ_9 ad γ_{10}) have non-zero marginal inclusion probabilities and reasonably high significance. In future it would be of interest to obtain additional covariates such as whether a nucleobase belongs to a particular part of the gene like the promoter or a coding region. Furthermore, it is of interest to address factors whether a base is located within a CpG island (regions with a high frequency of C bases) or whether it belongs to a transposon. Moreover, interactions of these covariates may be interesting. Alternative choices of the response distributions (e.g. binomial or negative binomial) and/or other types of random effects ($AR(k)$, $ARMA(l, k)$) might also be of an interest.

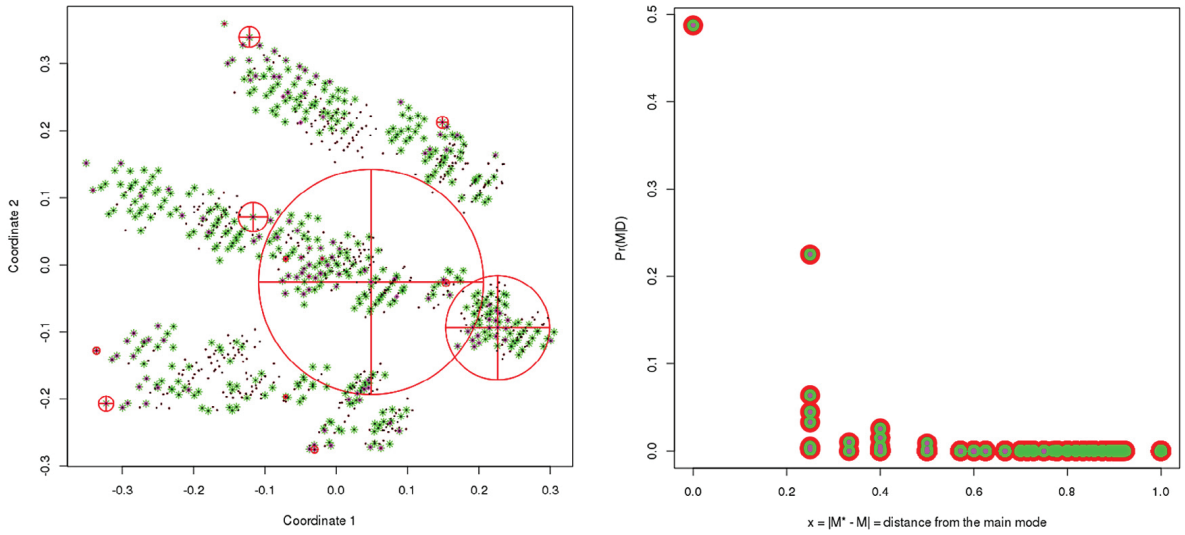

Fig. 6. Left: Multidimensional scale plot (Rohde, 2002) of the best 1024 models in terms of posterior model probability in the space of models (black dots are centers of the models, red circles are proportional to the posterior probabilities of models, green stars—models visited by MJMCMC, purple stars—models visited by MCMC). Right: A plot of posterior probabilities with respect to distance from the global mode (red circles correspond to all the models, the green circles—models visited by MJMCMC, the purple circles—by simple MCMC). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Summary and discussion

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

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

Whilst the renormalized model estimators (8) are Fisher consistent (Clyde et al., 2011), they remain generally speaking biased; although their bias reduces to zero asymptotically (with respect to the number of iterations). Standard MCMC based estimators such as (7), which are both consistent and unbiased, are also available through our procedure; these estimators however tend to have a much higher variance than the aforementioned ones. As one of the further developments it would be of interest to combine knowledge available from both groups of estimators to adjust for bias and variance, which is vital for higher dimensional problems.

Another aspect that requires being discussed is the model selection criterion. Different criteria can sometimes disagree about the results of model selection. In order to avoid confusion, the researcher should be clear about the stated goals. If the goal is prediction rather than inference one should adjust for that and use AIC, WAIC (Watanabe, 2009) or DIC (Spiegelhalter et al., 2002) rather than BIC or posterior model probability as selection criterion in MJMCMC. These choices are possible within the *EMJMCMC* package as well.

Based on several experiments, we claim MJMCMC to be a rather competitive algorithm that is addressing the wide class of Generalized Linear Mixed Models (GLMM). In particular, for this class of models one can incorporate a random effect, which

both can model the variability unexplained by the covariates and can introduce dependence between observations, creating additional modeling flexibility. Estimation of parameters for such models becomes significantly harder in comparison to simple GLM. This creates the necessity to address parallel computing extensively. We have enabled the latter within our package by means of combining methods for calculating marginal likelihoods, such as the INLA methodology, and parallel MJMCMC algorithm.

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

Acknowledgments

The authors gratefully acknowledge the *CELS project at the University of Oslo*, http://www.mn.uio.no/math/english/research/groups/cels/index.html, for giving the opportunity, inspiration and motivation to write this paper. The authors also thank the editor, the associate editor, and the referees for helpful comments and suggestions which significantly improved the manuscript. The authors would also like to acknowledge Paul Grini, Ole Christian Lingjærde and Melinka Butenko for the valuable discussions on the design of Example 4. Furthermore they want to express gratitude to Ole Christian Lingjærde for the final proofreading.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2018.05.020.

References

Al-Awadhi, F., Hurn, M., Jennison, C., 2004. Improving the acceptance rate of reversible jump MCMC proposals. Statist. Probab. Lett. 69 (2), 189–198.
Andrieu, C., Doucet, A., Holenstein, R., 2010. Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Ser. B Stat. Methodol. 72 (3), 269–342.
Andrieu, C., Roberts, G.O., 2009. The pseudo-marginal approach for efficient Monte Carlo computations. Ann. Statist. (2), 697–725.
Banterle, M., Grazian, C., Lee, A., Robert, C.P., 2015. Accelerating Metropolis-Hastings algorithms by delayed acceptance. arXiv preprint arXiv:1503.00996.
Barbieri, M.M., Berger, J.O., et al., 2004. Optimal predictive model selection. Ann. Statist. 32 (3), 870–897.
Barrett, T., Wilhite, S.E., Ledoux, P., Evangelista, C., Kim, I.F., Tomashevsky, M., Marshall, K.A., Phillippy, K.H., Sherman, P.M., Holko, M., et al., 2013. NCBI GEO: archive for functional genomics data sets - update. Nucl. Acids Res. 41 (D1), D991–D995.
Beaumont, M.A., 2003. Estimation of population growth or decline in genetically monitored populations. Genetics 164 (3), 1139–1160.
Becker, C., Hagmann, J., Müller, J., Koenig, D., Stegle, O., Borgwardt, K., Weigel, D., 2011. Spontaneous epigenetic variation in the Arabidopsis thaliana methylome. Nature 480 (7376), 245–249.
Bivand, R., Gómez-Rubio, V., Rue, H., et al., 2015. Spatial data analysis with R-INLA with some extensions. J. Stat. Softw. 63 (i20).
Bivand, R.S., Gómez-Rubio, V., Rue, H., 2014. Approximate Bayesian inference for spatial econometrics models. Spat. Statist. 9, 146–165.
Bottolo, L., Chadeau-Hyam, M., Hastie, D.I., Langley, S.R., Petretto, E., Tiret, L., Tregouet, D., Richardson, S., 2011. ESS++: a C++ objected-oriented algorithm for Bayesian stochastic search model exploration. Bioinformatics 27 (4), 587–588.
Bové, D.S., Held, L., 2011. Bayesian fractional polynomials. Stat. Comput. 21 (3), 309–324.
Chen, T., Fox, E., Guestrin, C., 2014. Stochastic gradient Hamiltonian Monte Carlo. In: International Conference on Machine Learning. pp. 1683–1691.
Chib, S., 1995. Marginal likelihood from the Gibbs output. J. Amer. Statist. Assoc. 90 (432), 1313–1321.
Chib, S., Jeliazkov, I., 2001. Marginal likelihood from the Metropolis–Hastings output. J. Amer. Statist. Assoc. 96 (453), 270–281.
Chopin, N., Jacob, P.E., Papaspiliopoulos, O., 2013. SMC2: an efficient algorithm for sequential analysis of state space models. J. R. Stat. Soc. Ser. B Stat. Methodol. 75 (3), 397–426.
Christen, J.A., Fox, C., 2005. Markov chain Monte Carlo using an approximation. J. Comput. Graph. Statist. 14 (4), 795–810.
Clyde, M., Parmigiani, G., Vidakovic, B., 1998. Multiple shrinkage and subset selection in wavelets. Biometrika 85 (2), 391–401.
Clyde, M.A., Ghosh, J., Littman, M.L., 2011. Bayesian adaptive sampling for variable selection and model averaging. J. Comput. Graph. Statist. 20 (1), 80–101.
David, M., 2015. Auto insurance premium calculation using generalized linear models. Procedia Econ. Finance 20, 147–156.
de Souza, R., Cameron, E., Killedar, M., Hilbe, J., Vilalta, R., Maio, U., Biffi, V., Ciardi, B., Riggs, J., 2015. The overlooked potential of generalized linear models in astronomy, I: Binomial regression. Astron. Comput. 12, 21–32.
Doucet, A., Pitt, M.K., Deligiannidis, G., Kohn, R., 2015. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. Biometrika 102 (2), 295.
Eksioglu, S.D., Pardalos, P.M., Resende, M.G., 2002. Parallel metaheuristics for combinatorial optimization. In: Corra, R., Dutra, I., Fiallos, M., Gomes, F. (Eds.), Models for Parallel and Distributed Computation. In: Applied Optimization, vol. 67, Springer US, pp. 179–206.
Friel, N., Wyse, J., 2012. Estimating the evidence a review. Stat. Neerl. 66 (3), 288–308.
Frommlet, F., Ljubic, I., Arnardttir Helga, B., Bogdan, M., 2012. QTL mapping using a memetic algorithm with modifications of BIC as fitness function. Statist. Appl. Genet. Mol. Biol. 11 (4), 1–26.
George, E.I., McCulloch, R.E., 1997. Approaches for Bayesian variable selection. Statist. Sinica 7 (2), 339–373.
Geyer, C.J., 1991. Markov chain Monte Carlo maximum likelihood.
Ghosh, J., 2015. Bayesian model selection using the median probability model. Wiley Interdiscip. Rev. Comput. Stat. 7 (3), 185–193.
Grossi, L., Bellini, T., 2006. Credit risk management through robust generalized linear models. In: Zani, S., Cerioli, A., Riani, M., Vichi, M. (Eds.), Data Analysis, Classification and the Forward Search. In: Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, pp. 377–386.
Hubin, A., Storvik, G., 2016. Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA). arXiv:1611.01450v1.
Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K., 1999. An introduction to variational methods for graphical models. Mach. Learn. 37 (2), 183–233.
Kou, S., Zhou, Q., Wong, W.H., 2006. Discussion paper equi-energy sampler with applications in statistical inference and statistical mechanics. Ann. Statist. 1581–1619.

Liang, F., 2010. A double Metropolis–Hastings sampler for spatial models with intractable normalizing constants. J. Stat. Comput. Simul. 80 (9), 1007–1022.
Liu, J.S., Liang, F., Wong, W.H., 2000. The multiple-try method and local optimization in metropolis sampling. J. Amer. Statist. Assoc. 95 (449), 121–134.
Lobraux, S., Melodelima, C., 2015. Detection of genomic loci associated with environmental variables using generalized linear mixed models. Genomics 105 (2), 69–75.
Luo, G., 2016. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Netw. Model. Anal. Health Inf. Bioinform. 5 (1), 1–16.
Madigan, D., York, J., Allard, D., 1995. Bayesian graphical models for discrete data. Int. Stat. Rev./Rev. Int. Stat. 215–232.
Marin, J.-M., Pudlo, P., Robert, C.P., Ryder, R.J., 2012. Approximate Bayesian computational methods. Stat. Comput. 22 (6), 1167–1180.
McGrory, C.A., Titterington, D., 2007. Variational approximations in Bayesian model selection for finite mixture distributions. Comput. Statist. Data Anal. 51 (11), 5352–5367.
Miasojedow, B., Moulines, E., Vihola, M., 2013. An adaptive parallel tempering algorithm. J. Comput. Graph. Statist. 22 (3), 649–664.
Michiels, W., Aarts, E., Korst, J., 2010. Theoretical Aspects of Local Search, first ed. Springer Publishing Company, Incorporated.
Neal, R.M., et al., 2011. MCMC using Hamiltonian dynamics. In: Handbook of Markov Chain Monte Carlo, Vol. 2 (11).
Newton, M.A., Raftery, A.E., 1994. Approximate Bayesian inference with the weighted likelihood bootstrap. J. R. Stat. Soc. Ser. B Stat. Methodol. 3–48.
Raftery, A.E., Madigan, D., Hoeting, J.A., 1997. Bayesian model averaging for linear regression models. J. Amer. Statist. Assoc. 92 (437), 179–191.
Robert, C.P., Casella, G., 2005. Monte carlo statistical methods. Springer Texts in Statistics. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
Rohde, D.L.T., 2002. Methods for binary multidimensional scaling. Neural Comput. 14 (5), 1195–1232.
Rue, H., Martino, S., Chopin, N., 2009. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. J. R. Stat. Soc. 71 (2), 319–392.
Salakhutdinov, R.R., 2009. Learning in Markov random fields using tempered transitions. Adv. Neural Inf. Process. Syst. 1598–1606.
Sengupta, B., Friston, K.J., Penny, W.D., 2016. Gradient-based MCMC samplers for dynamic causal modelling. NeuroImage 125, 1107–1118.
Skrondal, A., Rabe-Hesketh, S., 2003. Some applications of generalized linear latent and mixed models in epidemiology: Repeated measures, measurement error and multilevel modeling. Nor. J. Epidemol. 13 (2), 265–278.
Song, Q., Liang, F., 2015. A split-and-merge Bayesian variable selection approach for ultrahigh dimensional regression. J. R. Stat. Soc. Ser. B Stat. Methodol. 77 (5), 947–972.
Spiegelhalter, D.J., Best, N.G., Carlin, B.P., Van Der Linde, A., 2002. Bayesian measures of model complexity and fit. J. R. Stat. Soc. Ser. B Stat. Methodol. 64 (4), 583–639.
Storvik, G., 2011. On the flexibility of metropolis-Hastings acceptance probabilities in auxiliary variable proposal generation. Scand. J. Stat. 38, 342–358.
Stroup, W.W., 2013. Generalized Linear Mixed Models : Modern Concepts, Methods and Applications. CRC Press, Taylor & Francis, Boca Raton.
Tierney, L., 1996. Introduction to general state-space Markov chain theory. In: Markov Chain Monte Carlo in Practice. pp. 59–74.
Tierney, L., Kadane, J.B., 1986. Accurate approximations for posterior moments and marginal densities. J. Amer. Stat. Assoc. 81 (393), 82–86.
Tjelmeland, H., Hegstad, B.K., 1999. Mode jumping proposals in MCMC. Scand. J. Statist. 28, 205–223.
Vandaele, W., 1978. Participation in illegitimate activities: Ehrlich revisited. Deterrence and Incapacitation 1, 270–335.
Watanabe, S., 2009. An introduction to algebraic geometry and statistical learning theory.
Yeh, Y.T., Yang, L., Watson, M., Goodman, N., Hanrahan, P., 2012. Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. ACM Trans. Graph. 31 (4), 56–58.
Zellner, A., 1986. On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In: Bayesian Inference and Decision Techniques: Essays in Honor of Bruno De Finetti, Vol. 6, pp. 233–243.
Zhou, Q., 2011. Multi-domain sampling with applications to structural inference of Bayesian networks. J. Amer. Statist. Assoc. 106 (496), 1317–1330.

A Details of the MJMCMC algorithm

A.1 Multiple try MCMC algorithm

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

$$r_m(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = \min\left\{1, \frac{w(\boldsymbol{\chi}_1^*, \boldsymbol{\gamma}) + \ldots + w(\boldsymbol{\chi}_S^*, \boldsymbol{\gamma})}{w(\boldsymbol{\chi}_1, \boldsymbol{\gamma}^*) + \ldots + w(\boldsymbol{\chi}_S, \boldsymbol{\gamma}^*)}\right\}. \tag{A-1}$$

In the implementation of the algorithm, ordinary MCMC is considered as a special case of multiple try MCMC with $S = 1$. We recommend ordinary or multiple try MCMC steps are used in at least 95% of the iterations with proposals of large jumps for the remaining 5%.

A.2 Choice of proposal distributions

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

- Table 1 lists the current possibilities for drawing indexes to swap in the first large jump. One should choose distributions where a large number of components are swapped.
- An important ingredient of the MJMCMC algorithm is the choice of local optimizer. In the current implementation of the algorithm, several choices are possible; simulated annealing, greedy optimizers based on best neighbor optimization or first improving neighbor (Blum and Roli, 2003) which is another variant of greedy local search accepting the first randomly selected solution better than the current. For each alternative the neighbors are defined through swapping a few of the γ_j's in the current model.
- For the last randomization, again Table 1 lists the possibilities, but in this case a small number of swaps will be preferable.

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

A.3 Parallel computing in local optimizers

General principles of utilizing multiple cores in local optimization are provided in Eksioglu et al. (2002). Given a current state $\boldsymbol{\chi}^*$ in the optimization routine, one can simultaneously draw several proposals $\boldsymbol{\chi}_1, ..., \boldsymbol{\chi}_K$ with respect to a certain transition kernel $s_o(\cdot|\boldsymbol{\gamma})$ and, if necessary, calculate the transition probabilities as the proposed models are evaluated. This step can be performed by parallel CPUs, GPUs or clusters. Consider an optimizer with the acceptance probability function $r_o^t(\boldsymbol{\chi}_j; \boldsymbol{\chi}^*), j \in 1, ..., K$, which either changes over the time (iterations) t or remains unchanged. For the greedy local search $r_o^t(\boldsymbol{\chi}; \boldsymbol{\chi}^*) = \mathbb{1}\{\pi(\boldsymbol{\chi}) \geq \pi(\boldsymbol{\chi}^*)\}, t \in 1, 2, ...$ For the implemented version of the simulated annealing algorithm we consider $r_o^t(\boldsymbol{\chi}; \boldsymbol{\chi}^*) = \min\left\{1, \exp\left(\frac{\log \pi(\boldsymbol{\chi}) - \log \pi(\boldsymbol{\chi}^*)}{T_t}\right)\right\}, i \in 1, ..., N$, where T_t is the SA temperature (Blum and Roli, 2003) parameter at iteration t. The proposed parallelization strategy is given in detail in Algorithm A.1.

A.4 Parallel MJMCMC with a mixture of proposals

Here we described the full version of our algorithm based on a combination of Algorithm 2 and the multiple try idea. The suggested MJMCMC approach allows to both jump between local modes efficiently and to explore the solutions around the modes simultaneously whilst keeping the desired ergodicity of the MJMCMC procedure. This implementation allows for mixtures of both local optimizers and proposals to

Algorithm A.1 Parallel optimization

procedure OPTIMIZE(N)
  $\boldsymbol{\chi}^* \leftarrow \boldsymbol{\chi}_0^*$
  for $i = 1, ..., N$ **do**
    $\boldsymbol{\chi}_{i,1}, ..., \boldsymbol{\chi}_{i,K} \sim s_o(\cdot|\boldsymbol{\chi}^*)$ ▷ make K proposals in parallel
    ▷ and calculate marginal likelihoods
    for $j = 1, ..., K$ **do**
      $r \leftarrow r_o^i(\boldsymbol{\chi}_{i,j}; \boldsymbol{\chi}^*)$ ▷ calculate acceptance probability
      if Unif$[0; 1] \leq r$ **then**
        $\boldsymbol{\chi}^* \leftarrow \boldsymbol{\chi}_{i,j}$ ▷ accept the transition
      end if
    end for
    $\boldsymbol{\chi}_i^* \leftarrow \boldsymbol{\chi}^*$
  end for
  return $\boldsymbol{\chi}_N^*$
end procedure

be addressed within MJMCMC. Both the local optimization and the multiple try steps utilize multiple CPUs and GPUs of a single machine or a cluster of nodes. The pseudo-code of the algorithm is given in Algorithm A.2 below. In this pseudo-code we consider the following notation:

- ϱ - the probability for a large jump;
- $\mathsf{P_o}(\cdot)$ - the distribution for the choice of the local optimizers, a discrete distribution over a finite number of possibilities;
- $\mathsf{P_l}(\cdot)$ - the distribution for the choice of large jump transition kernel, a discrete distribution over the possibilities in Table 1 with high probabilities on a large number of swaps;
- $\mathsf{P_r}(\cdot)$ - the distribution for the choice of the randomizing kernel, a discrete distribution over a finite number of possibilities, also from Table 1, but with a small number of changes;
- $\mathsf{P_g}(\cdot)$ - the distribution for the choice of proposals within the multiple try MCMC, a discrete distribution over the possibilities in Table 1 with a high probability on a small number of swaps.

The essential ingredients of the parallel version of the MJMCMC with a mixture of proposals (Algorithm A.2) are as follows:

- Multiple try MCMC steps are performed for the steps with no mode jumps;
- At the iterations with mode jumps the large jump proposals $q_l \sim \mathsf{P_l}(\boldsymbol{\zeta})$, the optimization proposals $q_o \sim \mathsf{P_o}(\boldsymbol{\zeta})$, and the randomizing kernels $q_r \sim \mathsf{P_r}(\boldsymbol{\zeta})$ are chosen randomly;
- At the iterations with no mode jumps the proposal is chosen randomly as $q_g \sim \mathsf{P_g}(\boldsymbol{\zeta})$;
- The optimization steps are parallelized as described in A.3.
- The multiple-try steps are parallelized.

B Supplementary materials for the experiments

Table B.1 describes some of the tuning parameters used for the different examples. Here, MTMCMC refers to the multiple try MCMC steps. The remaining tuning parameters, describing the mixture distributions P_0, P_l and P_r are specified in tables B.2 (example 1), B.3 (example 2), B.4 (example 3) and B.5 (example 4).

Algorithm A.2 Mode jumping MCMC

procedure MJMCMC($Numit$)
  $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma}_0$ ▷ define the initial state
  for $t = 1, ..., Numit$ **do**
    if Unif$[0; 1] \leq \varrho$ **then** ▷ large jump with local optimization
      $q_l \sim \mathsf{P_l}(\cdot)$ ▷ choose large jump kernel
      $q_o \sim \mathsf{P_o}(\cdot)$ ▷ choose local optimizer
      $q_r \sim \mathsf{P_r}(\cdot)$ ▷ choose randomization kernel
      $I \sim q_l(\cdot|\boldsymbol{\gamma})$ ▷ Indices for large jump
      $\boldsymbol{\chi}_0^* \leftarrow \text{SWAP}(\boldsymbol{\gamma}, I)$ ▷ large jump
      $\boldsymbol{\chi}_\mathsf{k}^* \sim q_o(\cdot|\boldsymbol{\chi}_0^*)$ ▷ local optimization
      $\boldsymbol{\gamma}^* \sim q_r(\cdot|\boldsymbol{\chi}_\mathsf{k}^*)$ ▷ randomization around the mode
      $\boldsymbol{\chi}_0 \leftarrow \text{SWAP}(\boldsymbol{\gamma}^*, I)$ ▷ reverse large jump
      $\boldsymbol{\chi}_\mathsf{k} \sim q_o(\cdot|\boldsymbol{\chi}_0)$ ▷ local optimization
      $r \leftarrow r_m(\boldsymbol{\chi}, \boldsymbol{\gamma}; \boldsymbol{\chi}^*, \boldsymbol{\gamma}^*)$ ▷ from (12)
    else ▷ ordinary proposal
      $q_g \sim \mathsf{P_g}(\cdot)$ ▷ choose multiple try proposal kernel
      $\boldsymbol{\gamma}^* \sim q_g(\cdot|\boldsymbol{\gamma})$ ▷ proposed solution
      $r \leftarrow r_m(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ ▷ from (A-1)
    end if
    if Unif$[0; 1] \leq r$ **then**
      $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma}^*$ ▷ accept the move
    end if
  end for
end procedure

Example	CPU	SA				Greedy			MT	
No	Num	S_t	Δt	t_0	t_f	S	LS	FI	Size	Steps
1	4	4	3	10	14×10^{-5}	15	F	T	4	15
2	2	5	3	10	14×10^{-5}	20	F	T	2	20
3	10	18	3	10	14×10^{-5}	88	F	T	10	88
4	1	3	3	10	14×10^{-5}	13	F	T	2	13
S.1	4	4	3	10	14×10^{-5}	15	F	T	4	15

Table B.1: Tuning parameters for local optimization within MJMCMC in the examples (Example No); CPU (Num) - the number of CPUs utilized within the examples; S_t - number of iterations per temperature in SA algorithm; Δt - cooling factor of the cooling schedule of SA algorithm; t_0 - initial temperature of SA algorithm; t_f - final temperature of SA algorithm; S - number of iterations in Greedy algorithm (per run); LS - if local stop is allowed in Greedy algorithm; FI - if the first improving neighbor strategy is applied in Greedy algorithm; Size - number of proposals per step in the multiple try steps; Steps - number of multiple try iterations within the local optimizer.

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

Table B.2: Other tuning parameters of MJMCMC for all proposal types (q_g, g_l, q_o, and q_r) in example 1; Optimizer - to which optimizer the proposal belongs (if not relevant "-"); Frequency - the frequency at which the proposal is addressed (ϱ for q_g and $1 - \varrho$ for q_l) and the frequency within the set of local optimizers ($\mathsf{P_o}$ for local optimizers); Type X - the frequency of proposal of type X Table 1; S - maximal allowed size of the neighborhood for the corresponding proposal; ρ_i - probability of change of component i of the current solution (if applicable to the proposal), where $\widehat{p}(\gamma_j|\boldsymbol{y}) = \widehat{p}(\gamma_j = 1|\boldsymbol{y})$ are the approximations of marginal inclusion probabilities. Notice that for MJMCMC* reported in the example only proposals of type 4 are used.

Proposal	Optimizer	Frequency	Type 1	Type 4	Type 3	Type 5	Type 6	Type 2
q_g	-	$\varrho = 0.9820$	0.1179	0.3357	0.2779	0.0200	0.2459	0.0021
S	-	-	$\{1, 1\}$	1	$\{1, 1\}$	1	1	20
ρ_j	-	-	$\widehat{p}(\gamma_j \vert \boldsymbol{y})$	-	-	-	-	$\widehat{p}(\gamma_j \vert \boldsymbol{y})$
q_l	-	0.0180	0	1	0	0	0	0
S	-	-	-	5	–	–	–	–
ρ_i	-	-	-	-	-	-	-	-
q_o	SA	0.5042	0.0636	0.3249	0.1571	0.2288	0.2246	0.0009
q_o	GREEDY	0.2183	0.0160	0.3085	0.1779	0.2474	0.2493	0.0007
q_o	MTMCMC	0.2774	0.2879	0.3016	0.1582	0.1107	0.1401	0.0013
S	-	-	$\{1, 1\}$	1	$\{1, 1\}$	1	1	20
ρ_j	-	-	$\widehat{p}(\gamma_j \vert \boldsymbol{y})$	-	-	-	-	$\widehat{p}(\gamma_j \vert \boldsymbol{y})$
q_r	-	-	0	0	0	0	0	1
S	-	-	-	-	–	–	–	20
ρ_j	-	-	-	-	-	-	-	0.0010

Table B.3: Other tuning parameters of MJMCMC for all proposal types (q_g, g_l, q_o, and q_r) in example 2; see Tables 1 and B.2 for details.

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

Table B.4: Other tuning parameters of MJMCMC for all proposal types (q_g, g_l, q_o, and q_r) in example 3; see Table 1 and B.2 for details.

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

Table B.5: Other tuning parameters of MTMCMC for all proposal types (q_g, g_l, q_o, and q_r) in example 4; see Table B.2 and 1 for details.

B.1 Details on example 2

In the addressed data set the true regression parameters were chosen to be $\beta_0 = 99$ for the intercept, and for the slope coefficients

$$\boldsymbol{\beta} = (-4, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1.2, 0, 37.1, 0, 0, 50, -0.00005, 10, 3, 0).$$

What concerns the covariates, X_1 and X_3 are factors from a group with 3 levels, X_4 and X_6 are from another group with 3 levels but additionally correlated with X_1 and X_3, X_7 and X_8 are two exponentially distributed variables with rate 0.3 jointly made dependent through copulas, X_9, X_{10} and X_{11} are all uniformly distributed with range from -1 to 10 and also jointly dependent through copulas, X_{12}, X_{13}, X_{14} and X_{15} are multivariate normal with a zero mean, standard deviation of 0.2 and some covariance structure, X_{16} represents some seasonality incorporated by the sinus transformation of the radiant representation of some angle equal to the corresponding ordering numbers of observations, X_{17} is the quadratic trend associated to the squared value of positions of observations, $X_{19} = (-4 + 5X_1 + 6X_3)X_{15}$ and $X_{20} = (-4+5X_1+6X_3)X_{11}$, finally to avoid over specification 2 layers from the mentioned above groups

of factors were replaced with some auxiliary covariates $X_2 = (X_{10}+X_{14})\times X_9$ and $X_5 = (X_{11}+X_{15})\times X_{12}$. The linear predictor is drawn as $\eta \sim N(\beta' X, 0.5)$, whilst the observations Y are independent Bernoulli variables with the probability of success modeled by a logit transformation of the linear predictor, namely $Y \sim Bernoulli\left(p = \frac{\exp(\eta)}{1+\exp(\eta)}\right)$.

C Further results

In tables C.1 (example 1), C.2 (example 2) and C.3 (example 4) the estimated biases, corresponding to the RMSE estimates given in tables 3, 4 and 5, are reported. In addition, an extra simulation experiment on linear regression based on simulated data is reported in C.1.

Par	True	TOP	MJMCMC		MJMCMC^2		BAS	MC^3		RS		MJMCMC*	
Δ	π_j	-	RM	MC	RM	MC	RM	MC	RM	MC	RM	RM	MC
γ_8	0.16	-3.51	-6.54	-10.28	-5.09	-9.64	-5.19	5.37	-3.20	4.96	-3.06	6.23	9.06
γ_{13}	0.16	-3.34	-7.44	-10.12	-5.57	-9.94	-6.25	7.46	2.86	8.06	2.65	6.38	10.54
γ_{14}	0.19	-3.24	-8.27	-11.69	-6.28	-11.93	-6.19	5.27	-1.86	5.37	-2.03	7.15	10.91
γ_{12}	0.22	-3.27	-6.82	-12.91	-5.54	-13.15	-3.08	3.00	-5.82	3.76	-5.06	5.29	10.93
γ_5	0.23	-2.56	-6.21	-12.71	-4.55	-13.35	-1.80	-4.79	-12.98	-4.28	-12.72	5.39	10.90
γ_9	0.23	-3.27	-9.45	-15.67	-7.35	-16.11	-9.26	4.53	-2.45	4.33	-2.10	7.68	11.06
γ_7	0.29	-2.31	-4.15	-12.04	-3.41	-12.36	-2.24	-0.47	-9.41	-1.00	-9.56	3.91	10.10
γ_4	0.30	-1.57	-5.82	-18.74	-3.67	-17.10	0.85	-12.67	-21.79	-13.24	-21.45	4.63	13.22
γ_6	0.33	-1.92	-8.49	-19.07	-6.09	-18.84	-3.06	8.99	7.16	10.09	6.81	5.87	15.43
γ_1	0.34	-2.51	-11.25	-21.94	-7.25	-20.29	-8.42	22.36	25.10	23.32	24.63	7.58	12.97
γ_3	0.39	-0.43	3.51	-7.20	2.09	-4.43	4.98	-21.11	-30.20	-21.13	-29.92	2.99	12.66
γ_2	0.57	1.58	5.66	-8.73	3.71	-7.51	13.73	-30.41	-37.52	-29.05	-37.12	5.11	14.04
γ_{11}	0.59	0.58	2.86	11.75	2.13	15.32	-3.95	10.67	21.68	10.29	21.23	2.77	12.77
γ_{10}	0.77	3.25	7.50	-2.57	5.91	2.33	15.42	-21.22	-19.06	-20.01	-19.55	6.41	14.27
γ_{15}	0.82	3.48	9.17	0.22	6.85	3.65	14.50	-69.61	-76.81	-69.14	-76.30	6.75	14.76
C(γ)	1.00	0.86	0.58	0.58	0.71	0.71	0.66	0.10	0.10	0.10	0.10	0.60	0.60
Eff	2^{15}	3276	1909	1909	3237	3237	3276	829	829	1071	1071	3264	3264
Tot	2^{15}	3276	3276	3276	5936	5936	3276	3276	3276	3276	3276	4295	4295

Table C.1: Bias for the 100 simulated runs of every algorithm on the Crime data (example 1); the values reported in the table are Bias $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$. See the caption of Table 3 for further details.

Par	True	TOP	MJMCMC				BAS	BAS-RS	RS	
Δ	π_j	-	RM	MC	RM	MC	RM	RM	RM	MC
γ_6	0.29	0.00	-7.23	-14.89	-4.48	-16.40	-6.46	-3.59	-5.96	0.23
γ_8	0.31	0.00	-5.97	-13.94	-3.89	-16.57	-5.57	-2.85	-5.28	-0.35
γ_{12}	0.35	0.00	-4.07	-8.12	-2.56	-11.65	-4.20	-1.82	-3.80	0.06
γ_{15}	0.35	0.00	-3.66	-8.85	-2.21	-12.04	-4.58	-1.35	-3.25	-0.28
γ_2	0.36	0.00	-4.60	-14.71	-2.81	-16.80	-5.39	-2.19	-3.51	0.04
γ_{20}	0.37	0.00	-4.16	-8.38	-2.46	-12.03	-3.30	-1.75	-4.07	-0.12
γ_3	0.40	0.00	-8.99	-19.22	-5.58	-21.72	-9.73	-4.63	-6.69	0.23
γ_{14}	0.44	0.00	1.08	7.12	0.51	7.63	3.68	-0.62	-0.99	0.22
γ_{10}	0.44	0.00	-2.68	-7.62	-1.68	-11.89	-4.79	-0.29	-1.19	0.13
γ_5	0.46	0.00	-1.74	-10.78	-0.88	-12.29	-3.93	0.57	0.55	-0.23
γ_9	0.61	0.00	0.32	-2.29	0.00	-1.24	3.78	0.22	1.99	-0.11
γ_4	0.88	0.00	5.61	6.20	3.71	6.13	6.60	5.54	7.58	-0.45
γ_{11}	0.91	0.00	5.36	6.47	3.87	6.84	4.64	3.01	4.29	-0.28
γ_1	0.97	0.00	1.86	0.98	1.32	1.17	2.43	1.94	2.28	-0.31
γ_{13}	1.00	0.00	0.00	-0.33	0.00	-0.29	0.00	0.00	0.00	-0.3
γ_7	1.00	0.00	0.00	-0.41	0.00	-0.36	0.00	0.00	0.00	-0.27
γ_{16}	1.00	0.00	0.00	-0.33	0.00	-0.31	0.00	0.00	0.00	-0.17
γ_{17}	1.00	0.00	0.00	-0.38	0.00	-0.35	0.00	0.00	0.00	-0.17
γ_{18}	1.00	0.00	0.00	-0.37	0.00	-0.32	0.00	0.00	0.00	-0.19
γ_{19}	1.00	0.00	0.00	-0.40	0.00	-0.32	0.00	0.00	0.00	-0.34
C($\boldsymbol{\gamma}$)	1.00	1.00	0.72	0.72	0.85	0.85	0.74	0.85	0.68	0.68
Eff	2^{20}	10000	5148	5148	9988	9988	10000	10000	1889	1889
Tot	2^{20}	10000	9998	9998	19849	19849	10000	10000	10000	10000

Table C.2: Bias for the 100 simulated runs of every algorithm on the simulated data of experiment 2; the values reported in the table are Bias $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$.
See the caption of Table 3 for further details.

C.1 Example S.1

In this experiment we compared MJMCMC to BAS and competing MCMC methods (MC^3, RS) using simulated data following the same linear Gaussian regression model as Clyde et al. (2011) with $p = 15$ and $n = 100$. All columns of the design matrix except for the ninth were generated from independent standard normal random variables and then centered. The ninth column was constructed so that its correlation with the second column was approximately 0.99. The regression parameters were chosen as $\beta_0 = 2$, $\boldsymbol{\beta} = (-0.48, 8.72, -1.76, -1.87, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0)$ while the variance used was $\sigma^2 = 1$.

When performing inference, Zellner's g-prior with $g = T$ was used for the regression parameters within each model. The marginal likelihood of a model could then be calculated through (14). To complete the prior specification, we used (4) with $q = 0.5$. This lead to a rather simple example with two main modes in the model space. Simple approaches were expected to work well in this case. The exact posterior model probabilities could be obtained by enumeration of the model space in this case, making comparison with the truth possible.

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

Table C.4, showing the root mean squared errors for different quantities, demonstrate that MJMCMC is outperforming simpler MCMC methods in terms of RM approximations of marginal posterior inclusion probabilities and the total captured mass. However, the MC approximations seem to be slightly poorer for this example. Whenever both MC and RM approximations are available one should address the latter

Par	True	TOP	MJMCMC		RS	
Δ	π_j	RM	RM	MC	RM	MC
γ_4	0.0035	-0.0005	-0.0019	1.7361	-0.0189	1.6397
γ_6	0.0048	-0.0006	-0.0041	1.8155	-0.0241	1.5437
γ_7	0.0065	-0.0006	-0.0045	1.9763	-0.0338	0.2191
γ_3	0.0076	-0.0007	-0.0014	2.9714	-0.0339	0.5167
γ_8	0.0076	-0.0007	-0.0066	1.8370	-0.0326	1.1101
γ_5	0.0096	-0.0007	-0.0055	1.5439	-0.0430	1.1780
γ_{11}	0.0813	-0.0007	-0.0131	-0.7623	-0.1060	1.0394
γ_{12}	0.0851	-0.0006	-0.0042	-0.4290	-0.0637	0.3118
γ_9	0.1185	-0.0008	-0.0121	-1.3414	-0.1277	-0.4439
γ_{10}	0.3042	-0.0006	-0.0036	-8.4912	-0.0501	2.6866
γ_{13}	0.9827	-0.0002	0.0051	-1.6177	0.0607	-1.0082
γ_1	1.0000	0.0007	0.0000	-4.4528	0.0000	-1.0018
γ_2	1.0000	0.0000	0.0000	-2.3865	0.0000	-0.7782
C($\boldsymbol{\gamma}$)	1.0000	1.0000	0.9998	0.9998	0.9977	0.9977
Eff	8192	385	1758	1758	155	155
Tot	8192	385	3160	3160	10000	10000

Table C.3: Bias of the mean squared error (BIAS) from the 100 simulated runs of MJMCMC on the epigenetic data (example 4); the values reported in the table are BIAS $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$. See the caption of Table 3 for further details.

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

Table C.4: Average root mean squared error (RMSE) from the 100 repeated runs of every algorithm on the simulated data (example S.1); the values reported in the table are RMSE $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$. See the caption of Table 3 for further details. The corresponding biases are reported in the appendix C in Table C.2. The corresponding biases are reported in Table C.6.

since they always have less noise. Comparing MJMCMC results to RM approximations provided by BAS (MC are not available for this method, MJMCMC performed slightly worse when we had 3276 proposals (but 1906 unique models visited). However MJMCMC became equivalent to BAS when we considered 6046 proposals with 3212 unique models visited in MJMCMC (corresponding to similar computational time as BAS). In this example we were not facing a really multiple mode issue having just two modes. All MCMC based methods tended to revisit the same states from time to time and for such a simple example one can hardly ever beat BAS, which never revisits the same solutions and simultaneously draws the models to be estimated in a clever adaptive way with respect to the current marginal posterior inclusion probabilities of individual covariates.

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

Table C.5: Other tuning parameters of MJMCMC for all proposal types (q_g, g_l, q_o, and q_r) in example S.1; see Tables 1 and B.2 for details.

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

Table C.6: Bias for the 100 simulated runs of every algorithm on the simulated data of experiment S.1; the values reported in the table are Bias $\times 10^2$ for $p(\gamma_j = 1|\boldsymbol{y})$. See the caption of Table 3 for further details.

References

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

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.

Eksioglu, S. D., Pardalos, P. M., and Resende, M. G. (2002). Parallel metaheuristics for combinatorial optimization. In Corrêa, R., Dutra, I., Fiallos, M., and Gomes, F., editors, *Models for Parallel and Distributed Computation*, volume 67 of *Applied Optimization*, pages 179–206. Springer US.

Liu, J. S., Liang, F., and Wong, W. H. (2000). The Multiple-Try Method and Local Optimization in Metropolis Sampling. *Journal of the American Statistical Association*, 95(449):121–134.

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

Paper II

A novel algorithmic approach to Bayesian Logic Regression*

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 remained less well known than other approaches to epistatic association mapping. Here we will adopt 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 mapping data for Recombinant Inbred Lines in *Arabidopsis thaliana* and from a backcross population in *Drosophila* where we identify several interesting epistatic effects.

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 based on Cockerham's coding (Wang and Zeng, 2009) 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

*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.

†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

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 becomes important which implicitly takes into account model uncertainty.

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 unfeasible 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: Jeffrey's prior which corresponds to computing marginal likelihoods with the Laplace approximation as in BIC-like model selection criteria and the robust g-priors as a state of the art choice for priors of regression coefficients in variable selection problems. For the robust g-prior the marginal likelihood is efficiently computed using ILA, 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 (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 considers various traits from backcross data of *Drosophila Simulans* and *Drosophila Mauritana* (Zeng et al., 2000).

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 $X_1, X_2, \ldots, X_m$. Our primary example will be genetic association studies where, depending on the context, each binary covariate, $X_j,\ j \in \{1, 2, \ldots, m\},$ 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 adopt 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

$$Y \sim \mathfrak{f}(y \mid \mu(\boldsymbol{X}); \phi) \tag{2.1}$$

$$h(\mu(\boldsymbol{X})) = \alpha + \sum_{j=1}^{q} \gamma_j \beta_j L_j, \tag{2.2}$$

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 $\beta_j, j \in \{1, ..., q\}$ 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 X via the logic expressions L_j according to (2.2). 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 models which include no more than k_{max} trees and each tree has at most C_{max} leaves. Consequently the total number of considered trees q will be finite. 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. We start with defining the prior for M by

$$p(M) \propto \mathbb{I}\left(|M| \leq k_{max}\right) \prod_{j=1}^{q} \rho(\gamma_j). \tag{2.3}$$

Here $|M| = \sum_{j=1}^{q} \gamma_j$ is the number of logical trees included in the model and k_{max} being 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

$$\rho(\gamma_j) = a^{\gamma_j c(L_j)} \tag{2.4}$$

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 $|M|$. 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 which will be estimated below. 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 similarly as discussed for example by Bogdan et al. (2008b) in the context of modifications of the BIC.

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

$$N(s) = \binom{m}{s} 2^{2s-2}. \tag{2.5}$$

Finally for a model of size $k = |M|$ the full model prior is of the form

$$P(M) \propto \mathbb{I}\left(k \leq k_{max}\right) \prod_{r=1}^{k} \frac{\mathbb{I}\left(s_{j_r} \leq C_{max}\right)}{\binom{m}{s_{j_r}} 2^{2s_{j_r}-2}} \ , \tag{2.6}$$

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 (2.1) 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 adopt the commonly used improper prior (Li and Clyde, 2018; Bayarri et al., 2012)

$$\pi(\phi) = \phi^{-1} \ . \tag{2.7}$$

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 $j \in \{j_1, ..., j_{|M|}\}$ correspond to the non-zero coefficients of model M, we will consider two different types of priors, simple Jeffrey's priors and robust g-priors. Jeffrey's prior (Chen et al., 2008) assumes for the parameters of the model an improper prior distribution of the form

$$\pi_\alpha(\alpha)\pi_\beta(\boldsymbol{\beta}) = |J_n(\alpha, \boldsymbol{\beta})|^{\frac{1}{2}} \ , \tag{2.8}$$

where $J_n(\alpha, \boldsymbol{\beta})$ is the observed information. To obtain model posterior probabilities according to equation (2.12) 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 Jeffrey's prior is that this integration becomes rather simple due to its relationship with the Laplace approximation (Claeskens and Hjort, 2008). In case of the Gaussian model choosing Jeffrey's prior (2.8) for the coefficients and the simple prior (2.7) for the variance

term yields that the Laplace approximation becomes exact (Claeskens and Hjort, 2008) and gives a marginal likelihood of the simple form

$$P(Y \mid M) \propto P(Y \mid M, \hat{\theta})\, n^{\frac{|M|}{2}} \ , \tag{2.9}$$

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 Jeffrey's prior becomes approximately (2.9) 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 perfectly well, within the Bayesian literature using Jeffrey's prior for model selection has been widely criticized for not being consistent once the true model coincides with the null model (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 which recommend 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 Jeffrey's prior. However, we want to point out that the choice of priors for the regression coefficients is not the real focus of this paper.

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

$$P(\boldsymbol{\beta} \mid g) \sim N_{|M|}\left(\mathbf{0}, g \cdot \phi \mathcal{J}_n(\boldsymbol{\beta})^{-1}\right) \ . \tag{2.10}$$

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

$$P\left(\frac{1}{1+g}\right) \sim tCCH\left(\frac{a}{2}, \frac{b}{2}, r, \frac{s}{2}, v, \kappa\right). \tag{2.11}$$

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

$$P(M \mid Y) = \frac{P(Y \mid M)P(M)}{\sum_{M' \in \Omega} P(Y \mid M')P(M')} \ , \tag{2.12}$$

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 (2.12) will be so small that they can be neglected. Considering a subset $\Omega^* \subseteq \Omega$ containing the most important models we can therefore approximate (2.12) by

$$P(M \mid Y) \approx \tilde{P}(M \mid Y) = \frac{P(Y \mid M)P(M)}{\sum_{M' \in \Omega^*} P(Y \mid M')P(M')} \ . \tag{2.13}$$

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 of 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 to be included in a model (also referred to as the marginal inclusion probability) by

$$\tilde{P}(L \mid Y) = \sum_{M \in \Omega^* : L \in T(M)} \tilde{P}(M \mid Y). \tag{2.14}$$

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. More generally one can approximate the posterior probability of some parameter Δ via model averaging as

$$\tilde{P}(\Delta \mid Y) = \sum_{M \in \Omega^*} P(\Delta \mid M, Y)\tilde{P}(M \mid Y) \ , \tag{2.15}$$

where Δ might be for example the predictor of unobserved data based on a specific set of covariates.

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 $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_q)$ 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.

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 (Hubin and Storvik, 2018).

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 the number q of 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. The resulting algorithm is similar to feature engineering (Xu et al., 2012) and allows to consider combinations of covariates that can be adapted throughout the search.

To be more specific, we consider different populations $\mathcal{S}_1, \mathcal{S}_2, ...$ 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 $\mathcal{S}_t$. 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 (to be published in a forthcoming paper). However, utilization of the approximation (2.13) 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 to fulfill detailed balance which is typically required for MCMC when model posterior probabilities are estimated by the relative frequency of how often a model has been visited.

The algorithm is initialized by first running MJMCMC for a given number of iterations N_{init} on the set of all binary covariates $X_1, ..., X_m$ 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 trees with largest marginal inclusion probability. In our current implementation we select the

d_1 leaves which have posterior probabilities 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 2.1 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 $S_t, t \in \{1, ..., T_{max}\}$. 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 $L_{j_i}^c$ 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 $L_{j_1}^c$ 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 $\mathcal{S}_{T_{max}}$.

Algorithm 1 GMJMCMC

1: Run the MJMCMC algorithm for N_{init} iterations on $X_1, ..., X_m$ and define $\mathcal{S}_0$ as the set of d_1 variables among them with the largest estimated marginal inclusion probabilities.
2: 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 $\mathcal{S}_1$.
3: Run the MJMCMC algorithm within search space $\mathcal{S}_1$.
4: **for** $t = 2, ..., T_{max}$ **do**
5: Delete trees within $\mathcal{S}_{t-1} \backslash \mathcal{S}_0$ which have estimated inclusion probabilities less than ρ_{min}.
6: Add new trees which are generated by crossover, mutation or reduction operators until the having again a set of size d, which becomes $\mathcal{S}_t$.
7: Run the MJMCMC algorithm within search space $\mathcal{S}_t$.
8: **end for**

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

Theorem 2.1. *Assume Ω^* is the set of models visited through the GMJMCMC algorithm where $d - d_1 \geq k_{max}$. Then the model estimates based on* (2.13) *will converge to the true model probabilities as the number of iterations $T_{\max}$ converges to ∞.*

Proof. Note that the approximation (2.13) 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 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 $\{(\mathcal{S}_t, M_{S_t}, \boldsymbol{X})\}$ 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 $\{(\mathcal{S}_t, M_{S_t}, \boldsymbol{X})\}$ 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. At the same time 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 2.1 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 like those from Chib, or Chib and Jaliazkov could be incorporated relatively easily (Hubin and Storvik, 2016), 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, GPUs or clusters. If one is mainly interested in model probabilities, then equation (2.13) 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:

$$\tilde{P}(\Delta \mid Y) = \sum_{b=1}^{B} w_b \tilde{P}_b(\Delta \mid Y) \; . \tag{2.16}$$

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 (2.15) from run b of GMJMCMC.

Due to the irreducibility of the GMJMCMC procedure it holds that $\lim_{k\to\infty} \tilde{P}(\Delta \mid Y) = P(\Delta \mid Y)$ where k is the number of iterations. 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 (2.15) 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 (2.16), 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 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 with binary responses and the second part three scenarios with quantitative responses. For each scenario we generated $N = 100$ datasets according to a regression model described by equations (2.1) and (2.2) 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 $j \in \{1, \ldots, 50\}$ in the first two scenarios and as $X_j \sim \text{Bernoulli}(0.5)$ for $j \in \{1, \ldots, 50\}$ in the last four scenarios. All computations were performed on the Abel cluster[1].

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

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

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

[1]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.

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 even more demanding with a model including three-way and four-way interactions. Effect sizes were accordingly increased to give sufficient power to detect these higher order trees.

For the binary response scenarios GMJMCMC was compared with FBLR and MCLR, where GMJMCMC was run with Jeffrey's prior as well as with the robust g-prior. Additionally we ran the algorithm with Jeffrey's prior and calculated posteriors for the visited models with respect to both Jeffrey's and robust g-prior. For all three algorithms we predefined $C_{max} = 2$ leaves per tree for Scenario 1 and 2 and $C_{max} = 5$ for Scenario 3. 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.

To evaluate the performance of the different algorithms we estimated the following metrics:

Individual power - the power to detect a particular true tree (a 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.

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 Jeffrey's prior gave slightly better results than using the robust g-prior. In the first two scenarios GMJMCMC with Jeffrey's prior worked almost perfectly. In the few instances where it did not detect the true tree it reported instead the two corresponding main effects. GMJMCMC with the robust g-prior had a few more instances where pairs of singletons were reported instead of the correct two-way interaction. 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.

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 Jeffrey's prior (Jef.) or the robust g-prior (R.g.). 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.

	FBLR	MCLR	GMJMCMC	
Scenario 1			**Jef.**	**R. g**
$L_1 = X_1^c \wedge X_4$	0.30	≤ 0.67	0.97	0.98
$L_2 = X_5 \wedge X_9$	0.42	≤ 0.61	1.00	0.95
$L_3 = X_{11} \wedge X_8$	0.33	≤ 0.59	0.91	0.77
Overall Power	0.35	≤ 0.62	0.96	0.90
FP	3.88	≥ 2.70	0.25	0.63
FDR	0.77	≥ 0.06	0.06	0.15
WL	0	0	0	0
Scenario 2				
$L_1 = X_1^c \wedge X_4$	0.32	≤ 0.66	0.97	0.97
$L_2 = X_5 \wedge X_9$	0.40	≤ 0.67	0.99	0.96
$L_3 = X_{11} \wedge X_8$	0.37	≤ 0.60	0.86	0.76
Overall Power	0.36	≤ 0.64	0.94	0.90
FP	3.83	≥ 2.58	0.38	0.66
FDR	0.75	≥ 0.06	0.09	0.16
WL	1	1	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

The third scenario is more complex than the previous ones but nevertheless GMJMCMC with Jeffrey's 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.

Finally, when the search was performed using Jeffrey's prior but the posteriors were obtained using the robust g-priors, then the posterior estimates were almost identical to those using only Jeffrey's prior throughout and there was no difference in terms of detected trees. This indicates that the choice of priors for the regression coefficients is of some importance for the quality of the search through the model space.

Continuous responses

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

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

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 contain 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 could only study the performance of GMJMCMC since the other approaches cannot handle continuous responses (MCLR has an implementation but that does not work properly). For these scenarios the settings of GMJMCMC were adapted to the increasing complexity of the model. We used $k_{max} = 10, 10$ 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 Jeffrey's 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 Jeffrey's 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

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 brackets for Scenario 6 are explained in the text.

Scenario 4	**Jeffrey's**	**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	**Jeffrey's**	**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	**Jeffrey's**	**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

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 Jeffrey's 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 Jeffrey's 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 four the results became slightly more ambiguous with power estimated to 0.32 and 0.21, respectively, for Jeffrey's 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 Jeffrey's 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 Jeffrey's prior.

Sensitivity analysis

We perform sensitivity analysis for the power to detect the four-way interaction L_4 based on $\tilde{P}(L_4|Y) > 0.5$ in Scenario 5. Specifically we consider the following three questions. How is the power effected by

1. a change in the corresponding coefficient β_4?
2. a change in the sample size n?
3. a change in the population size d?

In all three scenarios the parameters were increased uniformly in 10 steps within a given range and k_{max} was set to 20. The results presented in Figures 1-2 are based on 10 runs for each parameter value, both for Jeffrey's prior and for the robust g-prior.

The left plot of Figure 1 illustrates the dependence of power to detect L_4 on the corresponding coefficient β_4 varying between 1 and 10. For both priors the power curves sharply increase when β_4 changes from 4 to 6. This characteristic of the power curve depends on the number of leaves of the tree to be detected. Our model prior is designed to penalize more complex trees more severely in order to control FDR. For interaction terms of lower order the rise of the power curve would therefore occur already for smaller values of the corresponding regression coefficient. The fluctuations observed in the power curves in Figure 1 are due to the fairly small number of simulation runs per value.

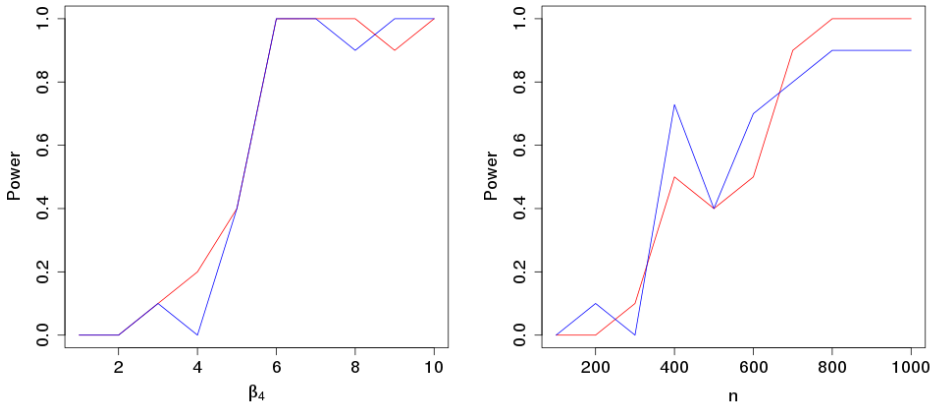

Figure 1: Dependence of power to detect L_4 on the regression coefficient β_4 (left) and the sample size n (right) both for Jeffrey's prior (red) and the robust g-prior (blue).

The right plot of Figure 1 presents power curves for the detection of L_4 depending on the sample size n. Once again due to the small number of simulation runs there is some fluctuation but one can see for both priors clearly that the power grows gradually when n varies between 100 and 1000. In spite of the low resolution it is fairly clear that for an effect of $\beta_4 = 7$ one needs at least a sample size of $n = 400$ to have some power to detect this four-way interaction. One can expect that for trees of lower complexity effects of the same size can be detected already with smaller sample sizes. This is again explained by the nature of our model prior, which parsimoniously penalizes more complex trees in order to control FDR.

Figure 2 is concerned with the influence of the population size d from the GMJMCMC algorithm on the power to detect L_4. Here d ranges from 15 to 150 and $n = 1000$. As one can see for both priors power 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 2.1, 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 2.1 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 a data generating model is larger than $d - d_1$.

3.2 Real data analysis

Our simulation results indicate that there is no large difference in the performance of GMJMCMC between using Jeffrey's prior or the robust g-prior. On the other hand the clear computational advantage of Jeffrey's prior seems to justify to omit the robust g-prior for analyzing real data. Hence in this section GMJMCMC always refers to GMJMCMC when using Jeffrey's

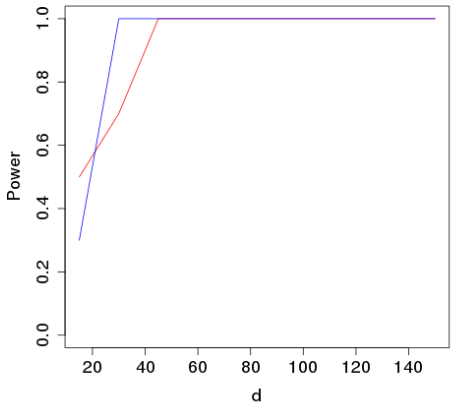

Figure 2: Dependence of power to detect L_4 on the population size d in GMJMCMC both for Jeffrey's prior (red) and the robust g-prior (blue) for $n = 1000$.

prior. We will analyze two data sets for QTL mapping which are publicly available. In both cases we used $k_{max} = 15$ and $d = 25$ which allows for way more complex models than we would expect to see.

Arabidopsis

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 [2].

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-QTL package (`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 Balasubramanian et al. (2009) reported 4 potential QTL's, the strongest one on chromosome 4 in the regions

[2]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`

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). Only trees for which $\tilde{P}(L \mid Y) > 0.05$ are reported.

Phenotype	Chr	Marker expression	$\tilde{P}(L \mid Y)$
Blue Light	4	X44606688	0.767
Blue Light	5	X44607250	0.335
Blue Light	2	X21607656	0.309
Blue Light	4∧2	X44606688∧X44606810	0.203
Red Light	2	MSAT2.36	0.441
Red Light	2	PHYB	0.353
Red Light	2∧1	PHYB^c∧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

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 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.

Drosophila

As a second real data example we considered the *Drosophila* back cross data from Zeng et al. (2000) [3]. There are five quantitative traits available for each species (abbreviated as `pc1`, `adjpc1`, `area`, `areat` and `tibia`) which quantify the size and shape of the posterior lobe

[3] Data downloaded from `ftp://statgen.ncsu.edu/pub/qtlcart/data/zengetal99`. There one can also find a linkage map in centiMorgan for the markers on three different chromosomes

Table 4: Results for *Drosophila Simulans* are presented for the trait `pc1` from Zeng et al. (2000). Posterior probabilities for additive and epistatic effects detected with GMJMCMC (column $\tilde{P}(L \mid Y)$) are compared with the findings reported by Bogdan et al. (2008a) using mBIC as a selection criterion (column mBIC). Posterior probabilities are only reported for trees with $\tilde{P}(L \mid Y) > 0.3$ are reported.

Marker	Chr	Marker name	$\tilde{P}(L \mid Y)$	mBIC
m2	X	w	1.000	x
m4	X	v	1.000	x
m7	2	gl	0.960	x
m9	2	cg	1.000	
m10	2	gpdh		x
m14	2	mhc	1.000	x
m18	2	sli	0.414	x
m22	2	zip	0.838	x
m23	2	lsp	0.998	x
m26	3	dbi	1.000	x
m29	3	fz	1.000	x
m32	3	rdg		x
m33	3	ht	1.000	
m35	3	ninaE		x
m37	3	mst	1.000	x
m40	3	hb	0.942	
m41	3	rox		x
m44	3	jan	1.000	x
m12, m34	2, 3	glt$\wedge$ant		x
m11, m35	2, 3	ninaE $\wedge$ ninaC	0.998	

of the male genital arch. The original publication (Zeng et al., 2000) only includes results on the first measure `pc1`, which was later analyzed for epistatic effects using a model selection approach based on the Cockerham coding (Bogdan et al., 2008a).

Compared with the Arabidopsis example this backcross data set has a much larger sample size combined with a smaller number of genetic markers, which both helps to increase the power to detect QTL. Genotype data from 45 markers is available for 471 samples from *Drosophila Simulans* and 491 samples from *Drosophila Mauritana*. Six markers are located on chromosome X, 16 markers on chromosome 2 and 23 markers on chromosome 3. Imputation of the few missing genotypes was performed by a simple maximum likelihood approach based on flanking markers. More details on the experiments and the measured traits can be found in Zeng et al. (2000).

Table 4 reports trees with posterior probabilities larger than 0.3 for the trait `pc1` of *Drosophila Simulans* and compares with the model obtained with mBIC - based forward selection by Bogdan et al. (2008a). The logic regression approach detected most of the main effects also previously reported, which in itself is quite interesting because as we allowed for higher order interactions we looked at a much larger model space and used therefore implicitly larger penalties than mBIC. In two locations GMJMCMC preferred a neighboring marker (*cg* instead

Table 5: Results for *Drosophila Mauritana* are presented for the trait `pc1` from Zeng et al. (2000). Posterior probabilities for additive and epistatic effects detected with GMJMCMC (column $\tilde{P}(L \mid Y)$) are compared with the findings reported by Bogdan et al. (2008a) using mBIC as a selection criterion (column mBIC). Posterior probabilities are only reported for trees with $\tilde{P}(L \mid Y) > 0.3$ are reported.

Marker	Chr	Marker name	$\tilde{P}(L \mid Y)$	mBIC
m1	X	ewg		x
m4	X	v	0.994	x
m9	2	cg	1.000	x
m11	2	ninaC	0.382	x
m15	2	ddc	1.000	x
m18	2	sli	0.523	x
m22	2	zip	1.000	x
m24	3	ve	0.966	
m25	3	acr		x
m26	3	dbi	0.995	
m28	3	cyc	0.398	x
m29	3	fz	0.834	
m34	3	ant	1.000	x
m37	3	mst		x
m39	3	tub	0.999	
m40	3	hb		x
m41	3	rox	0.420	
m44	3	jan	1.000	x
m1, m2	X, X	w$\vee$ewg	0.855	
m2, m36	X, 3	w$\vee$fas		x
m29, m40	3, 3	fz$\vee$hb		x

of *gpdh* on chromosome 2 and *hb* instead of *rox* on chromosome 3. In one region on chromosome 3 mBIC selected 2 markers (*rdg*, *ninaE*) whereas GMJMCMC selected only one marker in the middle. These kind of discrepancies are quite natural due to marker correlations in back cross data (Bogdan et al., 2008a). Just like with the mBIC approach we detected a two-way interaction between chromosome 2 and chromosome 3, where on both locations the two methods chose neighboring markers, respectively. Otherwise the epistatic effect detected with both methods is identical.

Table 5 contains the corresponding results for Drosophila Mauritana. As before GMJMCMC detects most of the additive effects that were reported by mBIC, though it sometimes chooses flanking markers (*ve* and *dbi* instead of *acr*, *tub* instead of *hb*). Interestingly the marker *ewg* on the X-chromosome is not reported as a main effect but rather as a two-way interaction together with *v* also on the X-chromosome, which also shows up as an additive effect. On the other hand the two-way interactions obtained with mBIC are not confirmed. Instead of the interaction between *fz* and *hb* GMJMCMC reports additional main effects on *fz* and *rox* (the neighbor of *hb*). For the interaction between *w* and *fas* there are no substitutes detected.

The results for the other four traits (*adjpc1, area, areat* and *tibia*) are provided in Section

C of the web supplement. In case of *Drosophila Simulans* we detect three two-way interactions for *adjpc1*. For *Drosophila mauritiana* further two-way interactions are found; two for *adjpc1*, three for *area*, and two more for *areat*. We did not find higher order interactions for any of these traits and based on the experience from our simulation study we might conclude that there are actually at least no strong higher epistatic effects.

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 Jeffrey's prior with the robust g-prior. Jeffrey's 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 Jeffrey's 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. However, when the search was performed using Jeffrey's prior but the posteriors were calculated with both Jeffrey's and the robust g-prior, then the results were almost identical between both priors.

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 numer 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. In the future it will be interesting to generalize the mode jumping ideas from Hubin and Storvik (2018) to the Markov chain of search spaces, making it converge to the right limiting distribution in the joint space of models, parameters and search spaces, whilst remaining the property of not getting stuck in local modes.

One important question in the context of logic regression is concerned with how to define true positive and false positive detections in simulations. We adopted 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.

Our simulation study demonstrated the potential of the GMJMCMC algorithm to find true logical expressions with high power and low false discovery rate, whilst in the real data examples GMJMCMC could find interesting epistatic effects in QTL analysis. However, the current

implementation has a slight tendency to prefer a set of several simple trees over a single complicated tree. Specifically it does not properly take into account that a complex tree can be represented in several equivalent ways which leaves space for further improvements. In the future we would also like to extend GMJMCMC to more general non-linear regression settings.

The R package implementing both MJMCMC and GMJMCMC is freely available on GitHub at `http://aliaksah.github.io/EMJMCMC2016/`, where one can also find examples of further logic regression applications.

Supplementary Material

Supplementary materials (https://github.com/aliaksah/EMJMCMC2016/tree/master/examples).

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). 3, 19, 20

Barber, R. F., Drton, M., and Tan, K. M. (2016). *Laplace Approximation in High-Dimensional Bayesian Regression*, 15–36. Cham: Springer International Publishing. 6

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. 5, 6

Bogdan, M., Frommlet, F., Biecek, P., Cheng, R., Ghosh, J. K., and Doerge, R. W. (2008a). "Extending the Modified Bayesian Information Criterion (mBIC) to Dense Markers and Multiple Interval Mapping." *Biometrics*, 64(4): 1162–1169. 21, 22

Bogdan, M., Ghosh, J. K., and Tokdar, S. T. (2008b). "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. 5

Chen, M.-H., Ibrahim, J. G., and Kim, S. (2008). "Properties and Implementation of Jeffreys's Prior in Binomial Regression Models." *Journal of the American Statistical Association*, 103(484): 1659–1664.
URL `http://www.jstor.org/stable/27640213` 5

Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. 5, 6

Fritsch, A. (2006). "A Full Bayesian Version of Logic regression for SNP Data." Ph.D. thesis, Diploma Thesis. 1, 2, 12, 23

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. 1

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. 7

Hubin, A. and Storvik, G. (2016). "Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA)." ArXiv:1611.01450v1. 11

— (2018). "Mode jumping MCMC for Bayesian variable selection in GLMM ." *Computational Statistics and Data Analysis*, –.
URL `https://www.sciencedirect.com/science/article/pii/S016794731830135X` 2, 8, 23, 24

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. 1

Keles, S., van der Laan, M., and Vulpe, C. (2004). "Regulatory motif finding by logic regression." *Bioinformatics*, 20: 2799–2811. 1

Kooperberg, C. and Ruczinski, I. (2005). "Identifying Interacting SNPs Using Monte Carlo Logic Regression." *Genetic Epidemiology*, 28: 157–170. 1, 2, 3, 23

Li, Y. and Clyde, M. A. (2018). "Mixtures of g-priors in generalized linear models." *Journal of the American Statistical Association*, (just-accepted). 2, 5, 6

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. 1, 3

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models. 2nd Edition*. Chapman and Hall, London. 3

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). "Logic regression." *J. Comput Graphical Statist.*, 12(3): 474–511. 1, 2, 3

— (2004). "Exploring Interactions in High-Dimensional Genomic Data: An Overview of Logic Regression, with Applications." *Journal of Multivariate Analysis*, 90: 178–195. 1

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

Schwender, H. and Ickstadt, K. (2008). "Identification of SNP interactions using logic regression." *Biostatistics*, 9: 187–198. 1

Schwender, H. and Ruczinski, I. (2010). "Logic Regression and Its Extensions." *Advances in Genetics*, 72: 25–45. 1

Tierney, L. and Kadane, J. B. (1986). "Accurate Approximations for Posterior Moments and Marginal Densities." *JASA*, 81(393): 82–86. 6

Wang, T. and Zeng, Z.-B. (2009). "Contribution of genetic effects to genetic variance components with epistasis and linkage disequilibrium." *BMC Genetics*, 10(1): 52. 1

Wang, Y. H. (1993). "On the number of successes in independent trials." *Statistica Sinica*, 295–312. 4

Xu, Y., Hong, K., Tsujii, J., and Chang, E. I.-C. (2012). "Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries." *Journal of the American Medical Informatics Association*, 19(5): 824. 8

Zeng, Z. B., Liu, J., Stam, L. F., Kao, C. H., Mercer, J. M., and Laurie, C. C. (2000). "Genetic architecture of a morphological shape difference between two Drosophila species." *Genetics*, 154: 299–310. 3, 20, 21, 22

Supplementary Material for: A novel algorithmic approach to Bayesian Logic Regression

A. HUBIN, G. STORVIK, F. FROMMLET

A GMJMCMC Algortithm

A.1 Tuning parameters

In all the simulations and in real data analysis we used the default tuning parameters of the implementation of MJMCMC downloaded from `http://aliaksah.github.io/EMJMCMC2016/`. The values which were used in the different simulation scenarios and for real data analysis for the parameters not related to MJMCMC but rather to the genetic algorithm part are presented in Table A.1.

Table A.1. Tuning parameters of GMJMCMC in the different examples (Ex.), where simple digits refer to the simulation scenario, RD1 refers to the Arabidopsis data analysis and RD2 to the Drosophila data analysis; Threads (Th.) - the number of CPUs utilized within the examples; N_{init} - the number of steps of MJMCMC during initialization; N_{expl} - the number of steps of MJMCMC between changes of population; M_{fin} - the number of unique models visited by MJMCMC for the final population; T_{max} - index of the final population; ρ_{min} - threshold for the trees to be deleted; P_{and} - probability of an *and* operator in crossovers and mutations; P_{not} - probability of using logical *not* in crossovers and mutations; P_c - probability of crossover to propose replacement trees; P_{init} - probability for a tree to be included into the initial solution for a new MJMCMC run in any iteration $t \geq 1$; ρ_{del} - probability of deletion in the reduction operator; C_{max} - maximal tree size allowed; k_{max} - maximal number of trees allowed in a model;d - size of population of genetic algorithm (number of trees searched by MJMCMC in each iteration).

Ex.	Th.	N_{init}	N_{expl}	M_{fin}	T_{max}	ρ_{min}	P_{and}	P_{not}	P_{init}	P_c	ρ_{del}	C_{max}	k_{max}	d
1	32	300	300	10000	16	0.2	1.0	0.2	0.5	0.9	0.5	2	10	15
2	32	300	300	10000	16	0.2	1.0	0.2	0.5	0.9	0.5	2	10	15
3	32	300	300	15000	33	0.2	0.9	0.1	0.5	0.9	0.5	5	10	15
4	32	300	300	10000	33	0.2	0.9	0.1	0.5	0.9	0.5	5	10	15
5	32	300	300	10000	33	0.2	0.9	0.1	0.5	0.9	0.5	5	10	20
6	32	250	250	20000	40	0.2	0.7	0.1	0.5	0.9	0.5	5	20	40
RD1	64	250	250	35000	40	0.2	0.7	0.1	0.5	0.9	0.5	5	15	25
RD2	64	250	250	15000	40	0.2	0.7	0.1	0.5	0.9	0.5	5	15	25

A.2 Theorem for parallel version of GMJMCMC

The following Theorem generalizes Theorem 1 from the manuscript to the parallelized version of GMJMCMC. Apart from letting the number of iterations go to infinity it is also possible to have only a finite number of iterations within each run but let the number of parallel runs go to infinity.

Theorem A.1. *Assume that we are running GMJMCMC in B parallel chains as describes in Section 2.3 of the manuscript. When the number of iterations within each chain b converges to infinity, the posterior estimates $\tilde{P}(\Delta \mid Y)$ of (16) from the manuscript will converge to $P(\Delta \mid Y)$.*

Assuming the search space $\mathcal{S}_1$ is selected randomly within the total set of possible search spaces and for a finite number of iterations within each chain b, the posterior estimates (16) will converge to $P(\Delta \mid Y)$ when $B \to \infty$.

Proof. When the number of iterations within each chain b converges to infinity, each $\tilde{P}_b(\Delta \mid Y)$ will converge to $P(\Delta \mid Y)$ according to Theorem 1 of the manuscript. Further, each $w_b \to 1/B$, proving the first part of the result.

When the initial search space $\mathcal{S}_1$ is selected randomly, any possible tree can be included. According to the construction of the initial model for the first MJMCMC run any model will have positive probability of being selected, giving the result directly. □

Remark Selecting the search space S_1 randomly among all possible models is in principle not easy due to the difficulty of specifying the complete model space. However, running the GMJMCMC algorithm with no data can be performed extremely fast, making it possible to select the initial population randomly.

B Details of Simulation Results

In this section we present further information on the simulation results of our six scenarios.

B.1 Binary Response

In case of GMJMCMC and FBLR a tree was counted as detected if its corresponding posterior probability was larger than 0.5. The power to detect a true tree is estimated by the percentage of simulation runs in which it was detected. The overall power is then defined as the average power over all individual true trees. A detected tree was counted as true positive if it was logically equivalent to a tree from the data generating model or to its logical complement, otherwise it was counted as false positive. FP denotes the average, over simulation runs, number of false positive detections and FDR was estimated as the average (over simulation runs) proportion of false discoveries, where this proportion was defined to be zero if there were no detections at all. WL is the number of binary covariates (leaves) which were not part of the data generating model but part of at least one detected tree.

Unfortunately, the output delivered by MCLR does not allow to compute the performance measures in the same way. Whenever MCLR detects a tree of size s then all subtrees are also reported as being detected. Furthermore MCLR reports for each detected tree only the set of leaves $v(L)$ and not the exact logical expression L itself. Thus it becomes impossible to define true positives by comparing the reported trees directly with the trees from the data generating model. Instead we considered for MCLR a reported tree L as a true positive whenever $v(L)$ coincided with the set of leaves of a true tree. This definition only gives an upper bound for the achieved power and is strongly biased in favor of MCLR. For the same reason, any reported tree that was a subtree of a true tree was not considered to be a false positive, resulting in only lower bounds of FP and FDR which are again strongly biased in favor of MCLR.

Table B.1 gives details about the frequencies of trees detected by the different methods. The first three lines give for each scenario the frequency with which the three true trees L_j in each scenario were detected. All further detected trees are per definition false positives.

However, we considered different classes of false positives. The first class of false positives are trees which are comprised exclusively of leaves from a true tree L_j, typically subtrees or trees with a different logic expression. Based on the output of MCLR it is not possible to determine the frequencies of this kind of false positive detections as we will discuss in the next paragraph. In case of FBLR and GMJMCMC Table B.1 provides the frequency of this kind of trees in the rows labeled $v(L_j)$. For Scenario 1 and 2 we actually provide more detailed information. Here all true trees are of size 2 and almost all detected trees of the class $v(L_j)$ consisted of single leaves (the only exception was two instances of the expression $X_8^c \wedge X_{11}$ in Scenario 1 for FBLR). We therefore explicitly present the number of detections of the first leave and the second leave of L_j. The $v(M)$ rows give the number of trees combining leaves from different true trees. Finally the rows WL(s) are concerned with the number of trees which include s leaves which were not in the data generating model at all.

In case of MCLR the same sort of classification is not possible due to the fact that MCLR does not report the exact logical tree L that it detects but only the corresponding set of leaves $v(L)$. Furthermore MCLR automatically reports the set of leaves for all subtrees of any detected tree which makes an assessment on how often these subtrees were actually detected by MCLR impossible. As a consequence we simply discarded reported subtrees when computing summary statistics, with one exception. In case of Scenario 3 MCLR reports 40 supertrees (trees for which a tree of interest is a subtree) of L_1, which we classified as false positives themselves but which in principle play an important role for the determination of the power to detect L_1. We ignored the fact that for any detected supertree of L_1 MCLR automatically also reports L_1 itself as detected and we pretended that in all these cases MCLR would actually have detected L_1 itself. Another peculiarity of MCLR is that it allows to search for trees of size 4, but that it does not report if it detected any such trees. In case of the four-way interaction L_3 from Scenario 3 there were 19 simulation runs where MCLR reported all four subtrees of size 3 from L_3 and we counted those instances as true positives, although MCLR did not really report the correct four-way interaction. For Scenario 1 and 2 none of these problems with supertrees occurred for MCLR because we restricted the search to trees of maximal size 2 in accordance with the data generating model.

The first two scenarios include only two-way interactions and we observe that GMJMCMC with Jeffrey's prior worked almost perfectly well. In the few instances where it did not detect the correct tree it reported instead the two corresponding main effects, resulting in a total of 25 and 38 false positive trees for the two scenarios (corresponding to an average of 0.25 and 0.38 false positives within each simulation, see Table 1 of the main manuscript). The robust g-prior resulted in a few more false positives which were also all just single leaves instead of the two-way interactions. FBLR chose in almost two thirds of the simulation runs two main effects instead of the correct interactions. The majority of the remaining false positives combined leaves from different true trees but there was also for each scenario one expression with a wrongly detected leave, respectively. In contrast MCLR reported in approximately two thirds of the cases trees with the correct leaves resulting in larger power than for FBLR. On the other hand MCLR reported a much larger number of trees which combined leaves from different true trees than FBLR. MCLR reported only one tree with a wrong leave in Scenario 2 and no such tree in Scenario 1. In summary we conclude that all three methods were doing extremely well in detecting the correct leaves in these simple scenarios but GMJMCMC was better than FBLR and MCLR in identifying the exact logical expressions.

The conclusion above is even more pronounced in the third scenario, which is more complex than the previous scenarios but still allows GMJMCMC with Jeffrey's prior to perform almost perfectly. It detected both the two-way interaction L_1 and the four-way interaction L_3 with a power of 100%, and had only some minor difficulties to detect the three-way interaction L_2. From the 15 false positive detections the majority consisted of subtrees of L_2 reported in those simulation runs where L_2 itself was not detected.

Table B.1. Number of true and false positive trees for the three simulation scenarios with a binary response. A detailed description of the different classes of false positives ($v(L_j), v(M)$, WL(s)) is given in the text above. The columns Jef. and Rob. g correspond to GMJMCMC with Jeffrey's prior and with the Robust g-prior, respectively.

	FBLR	MCLR	Jef.	Rob. g
S.1				
L_1	30	67	97	98
L_2	42	61	100	95
L_3	33	59	91	77
$v(L_1)$	68+69	*	3 + 3	2 + 2
$v(L_2)$	54+53	*	1 + 0	5 + 5
$v(L_3)$	60+59(+2)	*	9 + 9	25 + 24
$v(M)$	22	270	0	0
WL(1)	1	0	0	0
S.2				
L_1	32	66	97	97
L_2	40	67	99	96
L_3	37	60	86	76
$v(L_1)$	64 + 66	*	3 + 3	3 + 3
$v(L_2)$	56 + 60	*	1 + 1	4 + 4
$v(L_3)$	56 + 56	*	15 + 15	26 + 26
$v(M)$	24	256	0	0
WL(1)	1	1	0	0
S.3				
L_1	93	93 (40SupT)	100	100
L_2	4	67	91	56
L_3	0	19 (SubT)	100	56
$v(L_1)$	20	*	0	0
$v(L_2)$	162	*	8	81
$v(L_3)$	233	*	1	87
$v(M)$	167	195	5	6
WL(1)	34	54	1	0
WL(2)	16	9	0	0
WL(3)	8	0	0	0

Five trees were combinations of leaves from different true trees and there was only one tree including a leave which was not part of the data generating model. GMJMCMC with the robust g-prior had substantially lower power to detect L_2 and L_3 but instead reported many corresponding subtrees. There were six reported logic expressions which mixed leaves from L_2 and L_3. In comparison, both MCLR and FBLR performed much worse and only managed to detect L_1 with fairly large power. FBLR completely failed to detect the higher order terms L_2 and L_3 whereas MCLR had at least some power to detect the three-way interaction L_2. Both approaches reported way more false positive trees than GMJMCMC.

For FBLR we can discuss the structure of false positive detections in more detail. A large number of false positive expression were comprised of leaves from single true trees, 20 for $v(L_1)$, 162 for $v(L_2)$ and 233 for $v(L_3)$. These expressions were either subtrees of true trees or trees with misspecified logical operators and can be seen as substitutes for the true trees. Furthermore there were 167 logical expressions which combined leaves from different trees. Additionally, FBLR reported 34 trees with one wrongly detected leave, 16 trees with two wrongly detected leaves and even 8 trees of size three for which all leaves were

not part of the data generating model. Thus apart from having problems with determining the exact form of the logical expressions in this scenario FBLR produced also a large number of false positive trees which have nothing to do with the correct model at all.

The performance of MCLR was only a little bit better. With respect to the results presented in Table 1 of the main manuscript it is now even more important than for the first two scenarios to emphasize that we are dealing with upper bounds of the power and lower bounds of the number of false positives. MCLR automatically reports all subtrees of any detected tree which makes an assessment how often these tree were actually detected by MCLR impossible. As a consequence we simply discarded reported subtrees from further statistical analysis, with one exception. In case of Scenario 3 MCLR reports some supertrees of L_1, which we classified as false positives themselves but which in principle played an important role for the determination of the power of L_1. We ignored the fact that for any detected supertree of L_1 MCLR automatically also reports L_1 itself as detected and pretend that in all these cases MCLR would actually have detected L_1 itself. Another peculiarity of MCLR is that it allows to search for trees of size 4, but that it does not report if it detected any such trees. In case of the four-way interaction L_3 there were 19 simulation runs where MCLR reported all four subtrees of size 3 from L_4 and we counted those instances as true positives, although MCLR did not really report the correct four-way interaction. For Scenario 1 and 2 none of these problems with supertrees occurred for MCLR because we restricted the search to trees of maximal size 2 in accordance with the data generating trees.

Not counting any subtrees of reported trees as false positives gives MCLR a huge advantage, nevertheless it reported almost 20 times more false positive expressions than GMJMCMC. Among those were 40 supertrees of L_1, which all contributed to the power of L_1 although it is not guaranteed that in all corresponding simulation runs L_1 itself was actually detected. There were 195 false positive trees which combined leaves from different true trees. It was more problematic that there were 54 trees with one wrongly detected leave and 9 trees with two wrongly detected leaves. While there were not as many trees which were completely wrong as for FBLR there were still a considerable number of leaves reported by MCLR which were not part of the data generating model.

B.2 Continuous Response

Table B.2 gives detailed results about the frequencies of detected trees similarly to Table B.1 but now for the three linear regression scenarios. At the beginning we have again for each scenario the number of true positives with L_j referring to the trees of the data generating model. As described above we split the detections of false positives again in the classes $v(L_j)$ which refers to logic expressions consisting only of leaves from L_j, $v(M)$ which refers to logic expressions consisting of leaves from the data generating model but mixing leaves from different trees, and $WL(1)$ corresponding to trees including one wrong leave. For the last expression of Scenario 6 it holds that $L_8 = X_{11} \wedge X_{13} + X_{19} \wedge X_{50} - X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$ and it turned out that in many simulation runs GMJMCMC was detecting the three expression from the alternative version. In the main manuscript we considered these findings potentially as true positives and in Table B.2 we explicitly report the frequency of detection for each of these trees.

There is not much to be said about Scenario 4 apart from the fact that the only false positive detection $L_2 \vee L_3$ was very close to the expression $L_2 + L_3$ of the data generating model. In Scenario 5 the results using Jeffrey's prior and the robust g-prior are very similar. For those trees which were detected in all simulation runs (L_1 and L_2 for Jeffrey's, L_1 and L_3 for the robust g-prior) no false positive subtrees were reported. The majority of false positives for both priors is comprised of subtrees and there are only a very small number of detections which combine leaves from two different true trees (1 for Jeffrey's and 2 for the robust g-prior). Finally GMJMCMC with Jeffrey's prior reported two trees of size 4 and size 5, respectively, each of which included the wrongly detected leave X_{43}, whereas GMJMCMC with the robust g-prior reported five trees which included wrong leaves.

For the most complex Scenario 6 once again Jeffrey's prior and the robus g-prior perform quite similar. For the first 6 data generating trees both priors have very large power. For L_7 the power is much lower

Table B.2. Detailed results for the three simulation scenarios for linear regression. A detailed description of the different classes of false positives ($v(L_j), v(M)$, WL(1)) is given in Section B.1. The columns Jef. and Rob. g correspond to GMJMCMC with Jeffrey's prior and with the Robust g-prior, respectively. In Scenario 4 there was only one false positive detection which is listed explicitly. In Scenario 6 frequencies of the three trees which in combination give L_8 are also listed explicitly.

Scenario 4	Jef.	Rob.g
L_1	100	100
L_2	99	100
L_3	97	98
$L_2 \vee L_3$	1	
Scenario 5		
L_1	100	100
L_2	100	99
L_3	96	100
L_4	89	90
$v(L_2)$	0	2
$v(L_3)$	12	0
$v(L_4)$	22	19
$v(M)$	1	2
WL(1)	2	5

Scenario 6	Jef.	Rob.g
L_1	95	99
L_2	98	99
L_3	98	99
L_4	96	95
L_5	100	100
L_6	95	96
L_7	32	45
L_8	21	16
$X_{11} \wedge X_{13}$	76	78
$X_{19} \wedge X_{50}$	75	81
$X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$	72	69
$v(L_3)$	6	2
$v(L_4)$	12	15
$v(L_5)$	0	2
$v(L_6)$	18	13
$v(L_7)$	84	70
$v(L_8)$	24	49
$v(M)$	58	38
WL(1)	3	7

and both priors report a large number of subtrees which are counted as false positives. For L_8 the alternative representation of the logic expression has been discussed in the main manuscript. Only 58 false positive trees for Jeffrey's prior and 38 for the robust g-prior combined leaves from different true trees. The number of trees including one wrongly detected leave was 3 and 7, respectively, which once more illustrates that GMJMCMC is very good at controlling the type I error when it comes to including leaves which have nothing to do with the data generating model.

C Real Data Analysis on Drosophila

The results for the other four traits (`adjpc1, area, areat` and `tibia`) which were neither analyzed by Zeng et al. (2000) nor by Bogdan et al. (2008) are provided in the following two tables. In case

Table C.1. Posterior probabilities for additive and epistatic effects detected with GMJMCMC for four additional traits: *Drosophila simulans*.

Population	Phenotype	Chr	Marker name	$\widetilde{P}(L\|Y) > 0.5$
Simulans	adjpc1	3	rox	1.000
Simulans	adjpc1	3	dbi	1.000
Simulans	adjpc1	2	gpdh	1.000
Simulans	adjpc1	X	v	1.000
Simulans	adjpc1	2	plu	1.000
Simulans	adjpc1	3	mst	0.999
Simulans	adjpc1	2∧3	(gl)∧(fz)	0.998
Simulans	adjpc1	3	efi	0.985
Simulans	adjpc1	X	w	0.984
Simulans	adjpc1	3	fz	0.983
Simulans	adjpc1	3∧3	(lsp)∧(ht)	0.982
Simulans	adjpc1	2∧3	(duc)∧(fas)	0.978
Simulans	area	3	fz	1.000
Simulans	area	2	mhc	1.000
Simulans	area	3	jan	1.000
Simulans	area	X	w	1.000
Simulans	area	3	dbi	1.000
Simulans	area	X	v	0.999
Simulans	area	3	rox	0.998
Simulans	area	3	ninaE	0.996
Simulans	area	3	ve	0.990
Simulans	area	2	ninaC	0.970
Simulans	area	2	zip	0.952
Simulans	area	3	ht	0.864
Simulans	area	2	cg	0.806
Simulans	areat	3	jan	1.000
Simulans	areat	2	mhc	1.000
Simulans	areat	X	w	1.000
Simulans	areat	3	tub	1.000
Simulans	areat	3	rox	1.000
Simulans	areat	3	ninaE	1.000
Simulans	areat	3	fz	1.000
Simulans	areat	X	v	1.000
Simulans	areat	3	dbi	1.000
Simulans	areat	2	ninaC	1.000
Simulans	areat	2	zip	1.000
Simulans	areat	3	ve	1.000
Simulans	areat	3	ht	0.952
Simulans	areat	2	cg	0.925
Simulans	tibia	X	run	0.747

of *Drosophila Simulans* we detected three two-way interactions for *adjpc1*. For *Drosophila Mauritiana* further two-way interactions were found; two for `adjpc1`, three for `area`, and two more for `areat`. We did not find higher order interactions for any of these traits.

Table C.2. Posterior probabilities for additive and epistatic effects detected with GMJMCMC for four additional traits: *Drosophila Mauritiana*.

Population	Phenotype	Chr	Marker name	$\tilde{P}(L\|Y) > 0.5$
Mauritiana	adjpc1	2	cg	1.000
Mauritiana	adjpc1	3	ant	1.000
Mauritiana	adjpc1	3	jan	1.000
Mauritiana	adjpc1	3	acr	1.000
Mauritiana	adjpc1	3	eip	1.000
Mauritiana	adjpc1	3∨3	(cyc)∨(hb)	1.000
Mauritiana	adjpc1	2	gl	1.000
Mauritiana	adjpc1	2	sli	1.000
Mauritiana	adjpc1	X	ewg	1.000
Mauritiana	adjpc1	3∨X	(mst)∨(v)	0.999
Mauritiana	area	3	ant	1.000
Mauritiana	area	3	jan	1.000
Mauritiana	area	2	cg	1.000
Mauritiana	area	2	zip	0.999
Mauritiana	area	3	acr	0.990
Mauritiana	area	3	fz	0.985
Mauritiana	area	3∨X	(ve)∨(w)	0.984
Mauritiana	area	X	ewg	0.958
Mauritiana	area	3∨X	(tub)∨(v)	0.890
Mauritiana	area	3	rox	0.873
Mauritiana	area	3∨3	(cyc)∨(tub)	0.862
Mauritiana	area	2	sli	0.714
Mauritiana	area	2	ninaC	0.613
Mauritiana	area	2	mhc	0.535
Mauritiana	areat	3	ant	1.000
Mauritiana	areat	3	efi	1.000
Mauritiana	areat	2	zip	1.000
Mauritiana	areat	2	cg	1.000
Mauritiana	areat	X	ewg	1.000
Mauritiana	areat	3	rox	1.000
Mauritiana	areat	X∨3	(v)∨(mst)	1.000
Mauritiana	areat	3	fz	0.996
Mauritiana	areat	3∧2	(1-(fz))∧(ninaC)	0.974
Mauritiana	areat	3	acr	0.973
Mauritiana	areat	3	cyc	0.685
Mauritiana	tibia	X	v	0.999
Mauritiana	tibia	3	hb	0.999
Mauritiana	tibia	2	mhc	0.997
Mauritiana	tibia	2	plu	0.625

References

Bogdan, M., Frommlet, F., Biecek, P., Cheng, R., Ghosh, J. K., and Doerge, R. W. (2008). "Extending the Modified Bayesian Information Criterion (mBIC) to Dense Markers and Multiple Interval Mapping." *Biometrics*, 64(4): 1162–1169.

Zeng, Z. B., Liu, J., Stam, L. F., Kao, C. H., Mercer, J. M., and Laurie, C. C. (2000). "Genetic architecture of a morphological shape difference between two Drosophila species." *Genetics*, 154: 299–310.

Paper V

Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA)

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

Abstract

The marginal likelihood is a well established model selection criterion in Bayesian statistics. It also allows to efficiently calculate the marginal posterior model probabilities that can be used for Bayesian model averaging of quantities of interest. For many complex models, including latent modeling approaches, marginal likelihoods are however difficult to compute. One recent promising approach for approximating the marginal likelihood is Integrated Nested Laplace Approximation (INLA), designed for models with latent Gaussian structures. In this study we compare the approximations obtained with INLA to some alternative approaches on a number of examples of different complexity. In particular we address a simple linear latent model, a Bayesian linear regression model, logistic Bayesian regression models with probit and logit links, and a Poisson longitudinal generalized linear mixed model.

Keywords: Integrated nested Laplace approximation; Marginal likelihood; Model Evidence; Bayes Factor; Markov chain Monte Carlo; Numerical Integration; Linear models; Generalized linear models; Generalized linear mixed models; Bayesian model selection; Bayesian model averaging.

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

1 Introduction

Marginal likelihoods have been commonly accepted to be an extremely important quantity within Bayesian statistics. For data $\mathbf{y}$ and model $\mathcal{M}$, which includes some unknown parameters θ, the marginal likelihood is given by

$$p(\mathbf{y}|\mathcal{M}) = \int_{\Omega_\theta} p(\mathbf{y}|\mathcal{M}, \theta)p(\theta|\mathcal{M})d\theta \tag{1}$$

where $p(\theta|\mathcal{M})$ is the prior for θ under model $\mathcal{M}$ while $p(\mathbf{y}|\mathcal{M}, \theta)$ is the likelihood function conditional on θ. Consider first the problem of comparing models $\mathcal{M}_i$ and $\mathcal{M}_j$ through the ratio between their posterior probabilities:

$$\frac{p(\mathcal{M}_i|\mathbf{y})}{p(\mathcal{M}_j|\mathbf{y})} = \frac{p(\mathbf{y}|\mathcal{M}_i)}{p(\mathbf{y}|\mathcal{M}_j)} \times \frac{p(\mathcal{M}_i)}{p(\mathcal{M}_j)}. \tag{2}$$

The first term of the right hand side is the Bayes Factor (Kass and Raftery, 1995). In this way one usually performs Bayesian model selection with respect to the posterior marginal model model probabilities without the need to calculate them explicitly. However if we are interested in Bayesian model averaging and marginalizing some quantity Δ over the given set of models $\Omega_\mathcal{M}$, we are calculating the posterior marginal distribution, which in our notation becomes:

$$p(\Delta|\mathbf{y}) = \sum_{\mathcal{M}\in\Omega_\mathcal{M}} p(\Delta|\mathcal{M}, \mathbf{y})p(\mathcal{M}|\mathbf{y}). \tag{3}$$

Here $p(\mathcal{M}|\mathbf{y})$ is the posterior marginal model probability for model $\mathcal{M}$ that can be calculated with respect to Bayes theorem as:

$$p(\mathcal{M}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M})p(\mathcal{M})}{\sum_{\boldsymbol{\mathcal{M}}'\in\Omega_\mathcal{M}} p(\mathbf{y}|\mathcal{M}')p(\mathcal{M}')}, \tag{4}$$

Thus one requires marginal likelihoods $p(\mathbf{y}|\mathcal{M})$ in (2), (3) and (4). Metropolis-Hastings algorithms searching through models within a Monte Carlo setting (e.g. Hubin and Storvik, 2016) requires acceptance ratios of the form

$$r_m(\mathcal{M}, \mathcal{M}^*) = \min\left\{1, \frac{p(\mathbf{y}|\mathcal{M}^*)p(\mathcal{M}^*)q(\mathcal{M}|\mathcal{M}^*)}{p(\mathbf{y}|\mathcal{M})p(\mathcal{M})q(\mathcal{M}^*|\mathcal{M})}\right\} \tag{5}$$

also involving the marginal likelihoods. All these examples show the fundamental importance of being able to calculate marginal likelihoods $p(\mathbf{y}|\mathcal{M})$ in Bayesian statistics.

Unfortunately for most of the models that include both unknown parameters θ and some latent variables $\boldsymbol{\eta}$ analytical calculation of $p(\mathbf{y}|\mathcal{M})$ is impossible. In such situations one must use approximative methods that hopefully are accurate enough to neglect the

approximation errors involved. Different approximative approaches have been mentioned in various settings of Bayesian variable selection and Bayesian model averaging. Laplace's method (Tierney and Kadane, 1986) has been widely used, but it is based on rather strong assumptions. The Harmonic mean estimator (Newton and Raftery, 1994) is an easy to implement MCMC based method, but it can give high variability in the estimates. Chib's method (Chib, 1995), and its extension (Chib and Jeliazkov, 2001), are also MCMC based approaches that have gained increasing popularity. They can be very accurate provided enough MCMC iterations are performed, but need to be adopted to each application and the specific algorithm used. Approximate Bayesian Computation (ABC, Marin et al., 2012) has also been considered in this context, being much faster than MCMC alternatives, but also giving cruder approximations. Variational methods (Jordan et al., 1999) provide lower bounds for the marginal likelihoods and have been used for model selection in e.g. mixture models (McGrory and Titterington, 2007). Integrated nested Laplace approximation (INLA, Rue et al., 2009) provides estimates of marginal likelihoods within the class of latent Gaussian models and has become extremely popular. The reason for it is that Bayesian inference within INLA is extremely fast and remains at the same time reasonably precise.

Friel and Wyse (2012) perform comparison of some of the mentioned approaches including Laplace approximations, harmonic mean approximations, Chib's method and other. However to our awareness there were no studies comparing the approximations of the marginal likelihood obtained by INLA with other popular methods mentioned in this paragraph. Hence the main goal of this article is to explore the precision of INLA in comparison with the mentioned above alternatives. INLA approximates marginal likelihoods by

$$p(\mathbf{y}|\mathcal{M}) \approx \int_{\Omega_\theta} \left. \frac{p(\mathbf{y}, \theta, \boldsymbol{\eta}|\mathcal{M})}{\tilde{\pi}_G(\boldsymbol{\eta}|\mathbf{y}, \theta, \mathcal{M})} \right|_{\boldsymbol{\eta}=\boldsymbol{\eta}^*(\theta|\mathcal{M})} d\theta, \tag{6}$$

where $\boldsymbol{\eta}^*(\theta|\mathcal{M})$ is some chosen value of $\boldsymbol{\eta}$, typically the posterior mode, while $\tilde{\pi}_G(\boldsymbol{\eta}|\mathbf{y}, \theta, \mathcal{M})$ is a Gaussian approximation to $\pi(\boldsymbol{\eta}|\mathbf{y}, \theta, \mathcal{M})$. Within the INLA framework both random effects and regression parameters are treated as latent variables, making the dimension of the hyperparmeters θ typically low. The integration of θ over the support Ω_θ can be performed by an empirical Bayes (EB) approximation or using numerical integration based on a central composite design (CCD) or a grid (see Rue et al., 2009, for details).

In the following sections we will evaluate the performance of INLA through a number of examples of different complexities, beginning with a simple linear latent model and ending up with a Poisson longitudinal generalized linear mixed model.

2 INLA versus truth and the harmonic mean

To begin with we address an extremely simple example suggested by Neal (2008), in which we consider the following model $\mathcal{M}$:

$$\begin{aligned} Y|\eta, \mathcal{M} \sim & N(\eta, \sigma_1^2); \\ \eta|\mathcal{M} \sim & N(0, \sigma_0^2). \end{aligned} \tag{7}$$

Then obviously the marginal likelihood is available analytically as

$$Y|\mathcal{M} \sim N(0, \sigma_0^2 + \sigma_1^2),$$

and we have a benchmark to compare approximations to. The harmonic mean estimator (Raftery et al., 2006) is given by

$$p(y|\mathcal{M}) \approx \frac{n}{\sum_{i=1}^{n} \frac{1}{p(y|\eta_i, \mathcal{M})}}$$

where $\eta_i \sim p(\eta|y, \mathcal{M})$. This estimator is consistent, however often requires too many iterations to converge. We performed the experiments with $\sigma_1 = 1$ and σ_0 being either 1000, 10 or 0.1. The harmonic mean is obtained based on $n = 10^7$ simulations and 5 runs of the harmonic mean procedure are performed for each scenario. For INLA we used the default tuning parameters from the package (in this simple example different settings all give equivalent results). As one can see from Table 1, INLA gives extremely precise results

σ_0	σ_1	D	Exact	INLA	H.mean				
1000	1	2	-7.8267	-7.8267	-2.4442	-2.4302	-2.5365	-2.4154	-2.4365
10	1	2	-3.2463	-3.2463	-2.3130	-2.3248	-2.5177	-2.4193	-2.3960
0.1	1	2	-2.9041	-2.9041	-2.9041	-2.9041	-2.9042	-2.9041	-2.9042

Table 1: Comparison of INLA, harmonic mean and exact marginal likelihood

even for a huge variance of the latent variable, whilst the harmonic mean can often become extremely crude even for 10^7 iterations. Due to the bad performance of the harmonic mean (see also Neal, 2008) this method will not be considered further.

3 INLA versus Chib's method in Gaussian Bayesian regression

In the second example we address INLA versus Chib's method (Chib, 1995) for the US crime data (Vandaele, 2007) based on the following model $\mathcal{M}$:

$$\begin{aligned} Y_t|\mu_t, \mathcal{M} &\overset{iid}{\sim} N(\mu_t, \sigma^2); \\ \mu_t|\mathcal{M} &= \beta_0 + \sum_{i=1}^{p_{\mathcal{M}}} \beta_i x_{ti}^{\mathcal{M}}; \\ \frac{1}{\sigma^2}|\mathcal{M} &\sim \text{Gamma}(\alpha_\sigma, \beta_\sigma); \\ \beta_i|\mathcal{M} &\sim N(\mu_\beta, \sigma_\beta^2), \end{aligned} \tag{8}$$

where $t = 1, ..., 47$ and $i = 0, ..., p_{\mathcal{M}}$. We also addressed two different models, $\mathcal{M}_1$ and $\mathcal{M}_2$, induced by different sets of the explanatory variables with cardinalities $p_{\mathcal{M}_1} = 8$ and $p_{\mathcal{M}_2} = 11$ respectively.

In model (8) the hyperparameters were specified to $\mu_\beta = 0, \alpha_\sigma = 1$ and $\beta_\sigma = 1$. Different precisions σ_β^{-2} in the range $[0, 100]$ were tried out in order to explore the properties of different methods with respect to prior settings. Figure 1 shows the estimated marginal log-likelihoods for Chib's method (x-axis) and INLA (y-axis) for model $\mathcal{M}_1$ (left) and $\mathcal{M}_2$ (right). Essentially, the two methods give equal marginal likelihoods in each scenario. Table 2 shows more details for a few chosen values of the standard deviation σ_β. The means

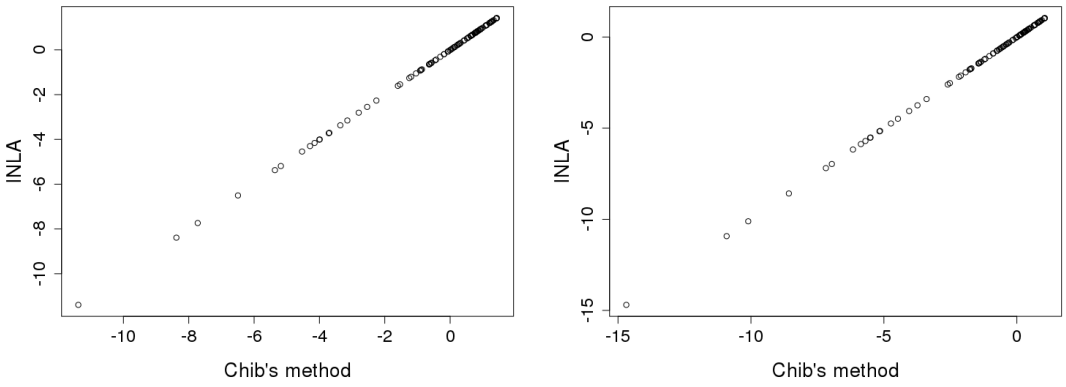

Figure 1: Chib's-INLA plots of estimated marginal log-likelihoods obtained by Chib's method (x-axis) and INLA (y-axis) for 100 different values of σ_β^{-2}. The left plot corresponds to model $\mathcal{M}_1$ while the right plot corresponds to model $\mathcal{M}_2$

of the 5 replications of Chib's method all agree with INLA up to the second decimal.

Going a bit more into details, Figure 2 shows the performance of Chib's method as a function of the number of iterations. The red circles in this graph represent 10 runs of

$\mathcal{M}$	μ_β	σ_β	INLA	Chib's method				
$\mathcal{M}_1$	0	1000	-73.2173	-73.2091	-73.2098	-73.2090	-73.2088	-73.2094
$\mathcal{M}_1$	0	10	-31.7814	-31.7727	-31.7732	-31.7732	-31.7725	-31.7733
$\mathcal{M}_1$	0	0.1	1.4288	1.4379	1.4380	1.4383	1.4378	1.4376
$\mathcal{M}_2$	0	1000	-96.6449	-96.6372	-96.6368	-96.6370	-96.6373	-96.6370
$\mathcal{M}_2$	0	10	-41.4064	-41.3989	-41.3987	-41.3991	-41.3995	-41.3996
$\mathcal{M}_2$	0	0.1	1.0536	1.0625	1.0629	1.0628	1.0626	1.0625

Table 2: Comparison of INLA and Chib's method for marginal log likelihood.

Chib's method for several choices of the number of iterations of the algorithm changing from 200 to 102400. The horizontal solid line shows the INLA estimate with the default settings. In this case, we used a precision of the regression parameters equal to $\sigma_\beta^{-2} = 0.2$, while in order to obtain some difference between Chib's method and INLA we changed the mean to $\mu_\beta = 1$. We only considered model $\mathcal{M}_1$ in this case. Although the differences are still small, this illustrates that INLA can be a bit off the true value. The reason for this deviance is the default choice of values for the tuning parameters in INLA. After tuning the step of numerical integration δ_z defining the grid as well as the convergence criterion of the differences of the log densities π_z (Rue et al., 2009) one can make the difference between INLA and Chib's method arbitrary small for this example. This can be clearly seen in Figure 2, where we depict the default INLA results (dark blue line) and the tuned INLA results (purple line). From Figure 2 one can also see that it might take quite a while

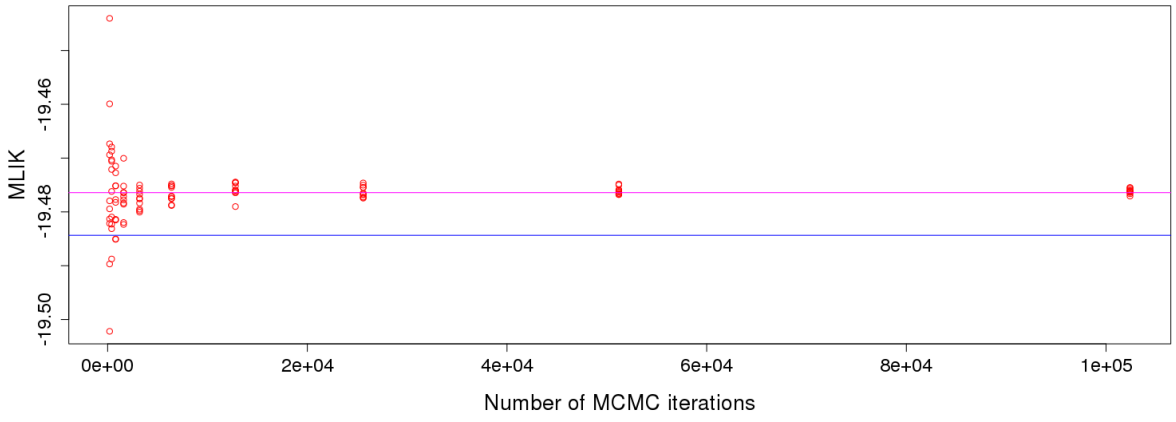

Figure 2: Variability of Chib's method as a function of number of MCMC iterations for the simple linear Gaussian example. Horizontal lines correspond to INLA estimates based on default settings (dark blue) and adjusted settings (purple).

for Chib's method to converge, whilst INLA gives stable results for the fixed values of the

tuning parameters. The total computational time for INLA corresponds to about 50 000 iterations with Chib's method for this model. Whilst 819200 iterations of Chib's method would require at least 15 times more time than INLA on the same machine [1].

The main conclusion that can be drawn from this example is that INLA approximations of marginal likelihoods can indeed be trusted for the studied model, giving a yet another evidence in the support of INLA methodology in general.

4 INLA versus Chib's method for logistic Bayesian regression with a probit link

In the third example we will continue comparing INLA with the Chib's method (Chib, 1995) for approximating the marginal likelihood in logistic regression with a probit link model $\mathcal{M}$. The data set addressed is the simulated Bernoulli data introduced by Hubin and Storvik (2016). The model is given by

$$\begin{aligned} Y_t|p_t, \mathcal{M} &\overset{iid}{\sim} \text{Bernoulli}(\Phi(\eta_t)); \\ \eta_t|\mathcal{M} &= \beta_0 + \sum_{i=1}^{p_\mathcal{M}} \beta_i x_{ti}^\mathcal{M}; \\ \beta_i|\mathcal{M} &\sim N(\mu_\beta, \sigma_\beta^2), \end{aligned} \tag{9}$$

where $t = 1, ..., 2000$ and $i = 0, ..., p_\mathcal{M}$. We addressed two different sets of explanatory variables with different cardinalities of 11 for model $\mathcal{M}_1$ and 13 for model $\mathcal{M}_2$. We used

$\mathcal{M}$	μ_β	σ_β	INLA	Chib's method				
$\mathcal{M}_1$	0	1000	-688.3192	-688.2463	-688.3260	-688.3117	-688.2613	-688.2990
$\mathcal{M}_1$	0	10	-633.0902	-633.1584	-633.0612	-633.0335	-633.1094	-633.0780
$\mathcal{M}_1$	0	0.1	-669.7590	-669.7646	-669.7666	-669.7610	-669.7465	-669.7528
$\mathcal{M}_2$	0	1000	-704.2266	-704.2154	-704.2138	-704.1463	-704.2526	-704.2303
$\mathcal{M}_2$	0	10	-639.8051	-639.7932	-639.8349	-639.8022	-639.7675	-639.8278
$\mathcal{M}_2$	0	0.1	-649.7803	-649.7360	-649.7604	-649.7893	-649.7532	-649.7806

Table 3: Comparison of INLA and Chib's method for logistic Bayesian regression with a probit link

$\mu_\beta = 0$ while the precisions for the regression parameters were varied between 0 and 10 in Figure 3 and chosen as $10^{-6}, 10^{-2}$ and 10^2 in Table 3. Figure 3 shows that INLA and Chib's method give reasonably similar results for both models. The total time for running

[1]Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz with 16 GB RAM was used for all of the computations

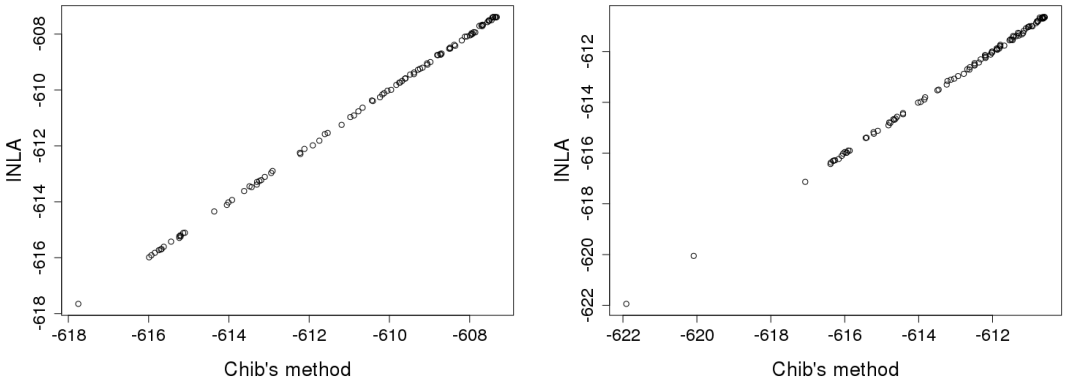

Figure 3: Comparisons of marginal likelihood estimates obtained by Chib's method (x-axis) and INLA (y-axis) for 100 different values of the precision parameter σ_β^{-2} under 2 models with different number of covariates.

INLA within these examples is at most 2 seconds, corresponding to approximately 12000 MCMC iterations in Chib's method. 100 000 MCMC iterations that were used to produce the obtained results in Table 3 required at least 25 seconds per replication on the same machine.

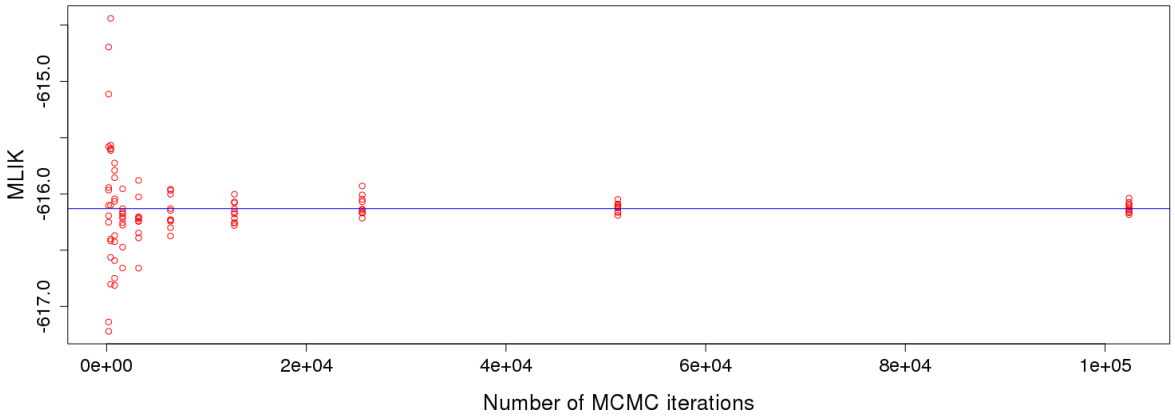

Figure 4: Variability of Chib's method as a function of number of MCMC iterations for logistic Bayesian regression with a probit link. The horizontal line corresponds to the INLA estimate based on default settings.

5 INLA versus other methods for logistic Bayesian regression with a logit link

In the fourth example we will continue comparing marginal likelihoods obtained by INLA with such methods as Laplace approximations, Chib and Jeliazkov's method, Laplace MAP approximations, harmonic mean method, power posteriors, annealed importance sampling and nested sampling. The model $\mathcal{M}$ is the Bayesian logistic regression model addressed by Friel and Wyse (2012), which is given by

$$\begin{aligned} Y_t|p_t, \mathcal{M} &\overset{iid}{\sim} \text{Bernoulli}(\text{logit}^{-1}(\eta_t)); \\ \eta_t|\mathcal{M} &= \beta_0 + \sum_{i=1}^{p_\mathcal{M}} \beta_i x_{ti}^{\mathcal{M}}; \\ \beta_i|\mathcal{M} &\sim N(\mu_\beta, \sigma_\beta^2), \end{aligned} \tag{10}$$

where $t = 1, ..., 532$ and $i = 0, ..., p_\mathcal{M}$. The data set addressed is the Pima Indians data, which consist of some diabetes records for 532 Pima Indian women of different ages. For $\mathcal{M}_1$ we have addressed such predictors as the number of pregnancies, plasma glucose concentration, body mass index and diabetes pedigree function and for $\mathcal{M}_2$ we additionally consider the age covariate. All of the covariates for both of the models have been standardized before the analysis. Then the analysis was performed for $\sigma_\beta^2 = 100$ and $\sigma_\beta^2 = 1$ correspondingly. The prior value of μ_β for both of the cases was chosen to be equal to 1. Table 4 contains the results obtained by all of the methods. Notice that all of the calculations apart from the INLA based ones are reported in Friel and Wyse (2012). Friel and Wyse (2012) claim that the relevant measures were taken to make the implementation of each method as fair as possible. In their runs each Monte Carlo method used the equivalent of 200 000 samples. In particular, the power posteriors used 20 000 samples at each of the 10 steps. The annealed importance sampling a cooling scheme with 100 temperatures and 2 000 samples generated per temperature. Nested sampling was allowed to use 2 000 samples and was terminated when the contribution to the current value of marginal likelihood was smaller than 10^{-8} times the current value. Notice that the default tuning parameters were applied for the INLA calculations. Except for the Harmonic mean, all methods gave comparable results. The INLA method only needed a computational time comparable to Laplace approximations, which is much faster than the competing approaches (Friel and Wyse, 2012). Reasonably good performance of the ordinary Laplace approximation in this case can be explained by having no latent variables in the model.

Method	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_1$	$\mathcal{M}_2$
INLA	-257.25	-259.89	-247.32	-247.59
Laplace approximation	-257.26	-259.89	-247.33	-247.59
Chib and Jeliazkov's method	-257.23	-259.84	-247.31	-247.58
Laplace approximation MAP	-257.28	-259.90	-247.33	-247.62
Harmonic mean estimator	-279.47	-284.78	-259.84	-260.55
Power posteriors	-257.98	-260.59	-247.57	-247.84
Annealed importance sampling	-257.87	-260.43	-247.30	-247.59
Nested sampling	-258.82	-261.38	-246.82	-246.97
σ_β^2 value	100	100	1	1

Table 4: Comparison of INLA and other method for a logistic Bayesian regression with a logit link.

6 INLA versus Chib and Jeliazkov's method for computation of marginal likelihoods in a Poisson with a mixed effect model

As models become more sophisticated we have less methodologies that can be used for approximating the marginal likelihood. In the context of generalized linear mixed models two alternatives will be considered, the INLA approach (Rue et al., 2009) and the Chib and Jeliazkov's approach (Chib and Jeliazkov, 2001).

This model is concerned with seizure counts Y_{jt} for 59 epileptics measured first over an 8-week baseline period $t = 0$ and then over 4 subsequent 2-week periods $t = 1, ..., 4$. After the baseline period each patient is randomly assigned to either receive a specific drug or a placebo. Following previous analyses of these data, we removed observation 49, considered to be an outlier because of the unusual seizure counts. We assume the data to be Poisson distributed and model both fixed and random effects. The model $\mathcal{M}$, originally defined in Diggle et al. (1994), is given by

$$\begin{aligned}
Y_{jt}|\lambda_{jt}, \mathcal{M} &\sim \text{Poisson}(\exp(\eta_{jt}));\\
\eta_{jt}|\mathcal{M} &= \log(\tau_{jt}) + \beta_0 + \beta_1 x_{jt1} + \beta_1 x_{jt2} + \beta_3 x_{jt1} x_{jt2} + b_{j0} + b_{j1} x_{jt1};\\
\beta_i|\mathcal{M} &\sim N(0, 100), \qquad i = 0, ..., 3;\\
\mathbf{b_j}|\mathbf{D}, \mathcal{M} &\sim N_2(0, \mathbf{D});\\
\mathbf{D}^{-1}|\mathcal{M} &\sim \text{Wishart}_2(4, I_2),
\end{aligned} \tag{11}$$

for $j = 1, ..., 58, t = 1, ..., 4$. Here $x_{jt1} \in \{0, 1\}$ is an indicator variable of period (0 if baseline and 1 otherwise), $x_{jt2} \in \{0, 1\}$ is an indicator for treatment status, τ_{it} is the offset

that is equal to 8 in the baseline period and 2 otherwise, and $\mathbf{b_j}$ are latent random effects. In Chib and Jeliazkov (2001) an estimate of the marginal log-likelihood was reported to be -915.49, while also an alternative estimate equal to -915.23 based on a kernel density approach by Chib et al. (1998) was given. INLA gave a value of -915.61 in this case, again demonstrating its accuracy. The computational time for the INLA computation was in this case on average 1.85 seconds.

7 Conclusions

The marginal likelihood is a fundamental quantity in the Bayesian statistics, which is extensively adapted for Bayesian model selection and averaging in various settings. In this study we have compared the INLA methodology to some other approaches for approximate calculation of the marginal likelihood. In all of the addressed examples disregarding their complexity INLA gave reliable estimates. In all cases, default settings of the INLA procedure gave reasonable accurate results. If extremely high accuracy is needed, we recommend that before performing Bayesian model selection and averaging in a particular model space $\Omega_{\mathcal{M}}$ based on marginal likelihoods produced by INLA, the produced estimates should be carefully studied and the tuning parameters adjusted, if required. Experimenting with different settings will also give an indication on whether more accuracy is needed.

SUPPLEMENTARY MATERIAL

Data and code: Data (simulated and real) and *R* scripts for calculating marginal likelihoods under various scenarios are available online at `https://goo.gl/0Wsqgp`.

ACKNOWLEDGMENTS

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

References

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

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

S. Chib, E. Greenberg, and R. Winkelmann. Posterior simulation and Bayes factors in panel count data models. *Journal of Econometrics*, 86(1):33 – 54, 1998.

P. Diggle, K. Liang, and S. Zeger. Analysis of longitudinal data, 1994.

N. Friel and J. Wyse. Estimating the evidence – a review. *Statistica Neerlandica*, 66(3):288–308, 2012. ISSN 1467-9574.

A. Hubin and G. Storvik. Efficient mode jumping MCMC for Bayesian variable selection in GLMM. *arXiv preprint arXiv:1604.06398*, 2016.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430): 773–795, 1995.

J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

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

R. Neal. The Harmonic Mean of the Likelihood: Worst Monte Carlo Method Ever, 2008.

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

A. E. Raftery, M. A. Newton, J. M. Satagopan, and P. N. Krivitsky. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. 2006.

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

L. Tierney and J. B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

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

Postface

Model based Bayesian statistics and deep learning have been historically treated as fairly incompatible. The former was mainly developed by rigorous mathematicians, whilst the latter was pushed forward by computer scientists. The theory of Bayesian statistics, in which one tries to infer the data generating processes, is based on strict model assumptions. This results in interpretable models which explain the data reasonably well and allow to make inference on the data generating process. In contrast, deep learning is a very complex and powerful prediction driven tool, which often uses tones of heuristics and ad hoc solutions, but lacks interpretability and theoretical justifications. At the same time, due to its big success in the trendy applications (like image analysis or natural language processing) the deep learning approach has greatly affected the way statistical learning is addressed in practical applications today. However, due to the fact that deep learning lacks a rigorous mathematical formalism, deep learning's success is often explained by a term *"black magic"* and various allegories based on biology or whatsoever. I sincerely hope that in this thesis I was able to build some bridges between the two paradigms in the attempt to provide ideas behind potentially interpretable Bayesian deep learning techniques, which can recover either complex neural networks as approximation to the phenomena or simple closed formed interpretable models, depending on the data addressed. There always are some technical tricks behind the magic conjurers perform in their shows. Revealing this magic is disappointing for some people from the ordinary audience, but inspiring for professionals to work even more complex and exciting tricks out. I hope to have at least partially revealed the *"black magic"* of deep regression models. Yet some of the approaches suggested in this thesis remain computationally complex and I would most gratefully expect the experienced computer science engineers to suggest novel approaches for speeding them up. Anyway, I believe that even in the current form the developed approaches may become more and more scalable with the development of the technology.

It is my most sincere hope that the work presented in this thesis will contribute to the foundations of an emerging field of study, putting modern deep learning and Bayesian statistics together and joining the efforts of numerous young and experienced statisticians and machine learning scientists to dive further into exciting world of Bayesian deep learning. Such that machine learning and statistics in general will greatly collaborate instead of opposing to one another.