Numerically Stable Approximate Bayesian Methods for Generalized Linear Mixed Models and Linear Model Selection

Mark Greenaway

A thesis submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

Statistics

University of Sydney

March 2019

CONTENTS

List of Figures

Acknowledgements

Firstly, I would like to thank the Federal Government for their generous support of my PhD in the form of the Australian Postgraduate Award scholarship. I would also like to thank the University of Sydney and particularly the School of Mathematics and Statistics for providing support through the opportunity tutor and lecturer statistics and mathematics subjects within the School.

Thanks to Professor Samuel Müeller for providing excellent advice and the opportunity to tutor STAT3012. Thanks also to Professor Jean Yang and the Bioinformatics research group at the University of Sydney for the stimulating research meetings, friendliness and company. Especially thanks for the Thai lunches and the pizza! I've learned a lot from all of you. Special thanks to Kevin Wang for your boundless enthusiasm and inspiration, your offbeat sense of humour, and the very insightful discussions we've had regarding `tidyverse` and `ggplot2`.

To my family – Charles, Eleanor, and Ben: Thank you for all your love and support during this sometimes challenging period.

To Charles Gray, who I met at the AMSI Summer School in 2014. Thanks for the supportive chats and the opportunity to stay with you. I look forward to reading your thesis with great interest, and sharing many `tidyverse` adventures with you in the future.

To Sarah Romanes and especially the ever patient Weicheng Yu – Thanks for reading and offering feedback on drafts of this thesis while I was writing up. I'll be sure to return the favour when the time comes. Extra thanks to Weicheng as a fellow denizen of Carslaw 807a for many interesting mathematical and statistical discussions, and always providing a friendly ear when my editing wasn't going well or my programs were crashing.

Last, but by no means least, my most sincere and humble thanks to my supervisor and friend Dr John Ormerod. Thanks for introducing me to the exciting world of computational Bayesian statistics and numerics. For always taking the time to explain difficult theoretical concepts, the endless good recommendations of papers and books to read, and teaching me the fine art of mathematical writing and editing. On a personal level, thanks for your seemingly limitless patience and understanding. I've learned a great deal from you over the past half-decade, and some of it even had to do with statistics!

Abstract

Bayesian models offer great flexibility, but can be computationally demanding to fit. The gold standard for fitting Bayesian models, when posterior distributions are not available analytically, are Monte Carlo Markov Chain methods. However, these can be slow and prone to convergence problems. Approximate methods of fitting Bayesian models allow these models to be fit using deterministic algorithms in substantially less time. Variational Bayes (VB) is a method for approximating the posterior distributions of the model parameters sometimes with only a slight loss of accuracy. In this thesis, we consider two important problems – variable selection for linear models, and zero inflated mixed models.

The first problem we address is variable selection, a task of central importance in modern statistics. Here, Bayesian model selection has the advantage of incorporating the uncertainty of the model selection process itself which propagates to the estimates of the model parameters. Linear regression models with Gaussian priors are ubiquitous in applied statistics due to their ease of fitting and interpretation. We use the popular g-prior Zellner (1986) for model selection of linear models with normal priors where g is a prior hyperparameter. This raises the question of how best to choose g. Liang et al. (2008) show that a fixed choice of g leads to problems, such as Bartlett's Paradox and the Information Paradox. These

paradoxes, and other problems, can be avoided by putting a prior on g. Using several popular priors on g, we derive exact expressions for the model selection Bayes Factors in terms of special functions depending only on the sample size, number of covariates and correlation of the model being considered. We show that these expressions are accurate, fast to evaluate, and numerically stable. An R package `blma` for doing Bayesian linear model averaging using these exact expressions has been released on GitHub.

For data sets with a small number of covariates, it is computationally feasible to perform exact model averaging. As the number of covariates increases the model space becomes too large to explore exhaustively. Recently, Ročková (2017) introduced Particle EM (PEM), a population-based method for efficiently exploring a subset of the model space with high posterior probability. The population-based method allows the method to seek multiple local modes, and captures greater total posterior mass from the model space than choosing a single model would. We extend this method using Particle Variational Approximation and the exact posterior marginal likelihood expressions to derive a computationally efficient algorithm for model selection on data sets with a large number of covariates. We demonstrate the algorithm's performance on a number of data sets for different combinations of g-prior, model selection prior and population size. We also compare our method to the existing methods such as lasso, SCAD, and MCP penalized regression methods, and PEM in terms of model selection performance, and show that our method outperforms these. We also show that total posterior mass increases and mean marginal variable error decreases, as the number of models in the population increases. Our algorithm performs very well relative to

previous algorithms in the literature, completing in 8 seconds on a model selection problem with a sample size of 600 and 7200 covariates.

The second problem we address is zero-inflated models have many applications in areas such as manufacturing and public health, but pose numerical issues when fitting them to data. We apply a variational approximation to zero-inflated Poisson mixed models with Gaussian distributed random effects using a combination of VB and the Gaussian Variational Approximation (GVA). We demonstrate that this approximation is accurate and fast on a number of simulated and benchmark data sets. We also incorporate a novel parameterisation of the covariance of the GVA using the Cholesky factor of the precision matrix, similar to Tan and Nott (2018), and discuss the computational advantages of this parameterisation due to the sparsity of the precision matrix for mixed models and resolve associated numerical difficulties.

CHAPTER 1

Introduction

1.1. Motivation

The advent of digital computers and the internet have lead to an explosion in the volume of data being collected. With technological progress marching on, this trend seems only set to continue and accelerate. In the future, as technology continues to advance more data will be able to be stored and processed, and so this trend of increasing volumes of data is set to continue (Gandomi and Haider, 2015). But this data is only of value if it can be analysed and understood.

This incredible increase in the volume of data has introduced new computational difficulties in processing and modelling such large amounts of data, so-called *Big Data*, which is so large that it is difficult to process on one computer. This data raises new challenges which modern statisticians must be ready to meet. Approaches to modelling data are needed which can handle large volumes of data in a computationally efficient manner while retaining the probabilistic underpinning of classical statistics and statistical machine learning, providing a rigorous underlying theory for inference. This realisation has created an explosion of interest in *Data Science*, incorporating ideas from both statistics and computer science in recent years. Machine learning problems are being tackled with algorithms which use probability models for the data – motivating the development of the

new field of statistical learning which combines many of the best elements of statistics and machine learning (James et al., 2014; MacKay, 2002; Hastie et al., 2001; Murphy, 2012).

1.2. Choosing an inferential paradigm

How one proceeds given the above needs can be addressed through an inferential paradigm. The most common of these are the frequentist and Bayesian statistical paradigms. The difference between frequentist and Bayesian approaches begins with a difference in philosophy. Frequentists define an event's probability as its' relative frequency after a large number of trials. While Bayesians view probability as our reasonable expectation about an event, representing our state of knowledge about the event.

There are many practical reasons to choose Bayesian approaches to modelling data. It is flexible in modelling statistical complications, such as missing or hierarchical data, and complicated models can be built by chaining together multiple levels of simple models. These models can then be fit to data by calculating the posterior probability of the parameters using Bayes' Theorem,

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{1}$$

where $\mathbf{y}$ is a vector of observed data, $\boldsymbol{\theta}$ are the model parameters, $p(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood function, $p(\boldsymbol{\theta})$ is a prior distribution on $\boldsymbol{\theta}$, and $p(\mathbf{y}) = \int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. Here the integral is performed over the range of $\boldsymbol{\theta}$. If a subset of $\boldsymbol{\theta}$ are discrete random variables then the integral over these parameters is replaced with a combinatorial sum over all possible values of these discrete random variables.

There are many models which are difficult to fit under the frequentist paradigm, as the likelihood can be difficult to maximise for complex models. Furthermore, as the Bayesian paradigm treats each of the parameters in a model as uncertain, the full uncertainty associated with all of the parameters can be estimated via the uncertainty in the posterior distribution. This approach avoids many of the pitfalls of statistical inference encountered with the frequentist approach using significance testing and p-values (Cox, 2005).

The ability to build a model one component at a time and have the uncertainty propagate through the model makes Bayesian modelling particularly appropriate for mixed effects and hierarchical models. In particular, uncertainty regarding model selection is taken into account in the context of model selection. Thus for the two classes of problems we consider in this thesis the Bayesian approach is more suitable.

1.2.1. Bayes Factors. In the Bayesian inferential paradigm, two competing hypotheses can be compared using Bayes Factors. The Bayes Factor is the ratio of the marginal likelihoods under the assumption of each of the models being compared

$$\text{BF}(M_1, M_2) = \frac{P(\mathbf{y}|M_1)}{P(\mathbf{y}|M_2)} = \frac{\int P(\boldsymbol{\theta}_1|M_1)P(\mathbf{y}|\boldsymbol{\theta}_1, M_1)d\boldsymbol{\theta}_1}{\int P(\boldsymbol{\theta}_2|M_2)P(\mathbf{y}|\boldsymbol{\theta}_2, M_2)d\boldsymbol{\theta}_2} = \frac{P(M_1|\mathbf{y})P(M_2)}{P(M_2|\mathbf{y})P(M_1)}$$

where $\mathbf{y}$ is the data observed, M_1 and M_2 are the models being compared and $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are the model parameter vectors associated with each model respectively.

1.3. Research problems

In this section, we introduce the major problems that will be addressed in this thesis. The themes of flexible modelling of data using Generalised Linear Mixed Models and model selection of linear models with normal priors will be explored.

1.3.1. Exponential family and the canonical form of linear regression models. The concept of the exponential family of probability distributions was first introduced by Koopman (1935) and Pitman (1936). The canonical form of a regression model from the exponential family is

$$p(\mathbf{y}|\boldsymbol{\theta}) = h(\mathbf{y}) \exp\{\boldsymbol{\theta}^\top T(\mathbf{y}) - b(\boldsymbol{\theta})\} \tag{2}$$

for a parameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and observed data $\mathbf{y}$. The sufficient statistic T and h are functions of the observed data, while the cumulant function $b(\boldsymbol{\theta})$ is a function of the parameter $\boldsymbol{\theta}$. The cumulant function is the logarithm of the normalisation constant.

Many commonly used probability distributions of practical interest, such as the Gaussian, Bernoulli, Poisson, Exponential and Gamma probability distributions, can be expressed as an exponential family by making an appropriate choice of h, T and b functions. The exponential family of distributions have several appealing statistical and computational properties which derive from the convexity of the parameter space Θ for which the exponential family distribution is defined, and the convexity of the cumulant function (Jordan, 2010). The mean of an exponential family distribution can be obtained by calculating the first derivative of the cumulant function and then evaluating at zero, while the variance can be obtained

by calculating the second derivatives of the cumulant function and evaluating at zero.

The exponential family of distributions allow us to extend linear models to more general situations where the response variable is not normally distributed but may be categorical, discrete or continuous and the relationship between the response and the explanatory variables need not be of simple linear form. By choosing the parameterisation $\boldsymbol{\theta} = \mathbf{X}\boldsymbol{\beta}$ where $\mathbf{X}$ is the matrix of observed covariates in $\mathbb{R}^{n\times p}$ and $\boldsymbol{\beta}$ are regression parameters in $\mathbb{R}^p$, for n the sample size and p the number of covariates, a canonical form of generalised linear regression models may be written as

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \mathbf{y}^\top \mathbf{X}\boldsymbol{\beta} - \mathbf{1}^\top b(\mathbf{X}\boldsymbol{\beta}) + \mathbf{1}^\top c(\mathbf{y}) \tag{3}$$

where $c(\boldsymbol{\theta})$ is the log of $h(\mathbf{y})$ from (2). A choice of $b(x) = e^x$ corresponding to the Poisson family of distributions specifies a Poisson linear model appropriate for modelling count data, while a choice of $b(x) = \log(1 + e^x)$ corresponding to the logistic family of distributions specifies a logistic linear model appropriate to modelling binary data.

1.3.2. Generalised Linear Mixed Models. Generalised Linear Mixed Models, an extension of Generalised Linear Models to include both fixed and random effects, are applicable to many complicated modelling situations.

Linear and generalised linear regression models are the standard tools used by applied statisticians to explain the relationship between an outcome variable

and one or more explanatory variables. They provide a general method to analyse quantified relationships between variables within a data set in an easily interpretable way. A standard assumption is that the outcomes are independent, and that the effect of the explanatory variables on the outcome is fixed. But if the outcomes are dependent and this assumption is not met, then linear and generalised linear models can be extended to linear mixed models. These allow us to incorporate dependencies amongst the observations via the assumption of a more complicated covariance structure, including random effects for different subgroups or longitudinal data and other extensions such as splines. This additional flexibility makes their application popular in many fields, such as public health, psychology and agriculture (Kleinman et al., 2004; Lo and Andrews, 2015; Kachman, 2000).

In the frequentist paradigm, model parameters are fixed and uncertainty enters the model through random errors, which have an associated covariance. The data is modelled as a combination of these fixed parameters and random errors. In the Bayesian paradigm, the uncertainty in the parameters and the data is accounted for by the likelihood function.

1.3.2.1. *A Canonical Form for Generalised Linear Mixed Models.* The generalised form for linear models in (3) can easily be extended to include random effects. Following the conventions for Generalised Design of Zhao et al. (2006), we adopt the canonical form for Generalised Linear Mixed Models exponential family with Gaussian random effects take the general form

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{u}) &= \exp\{\mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}) - \mathbf{1}^\top b(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}) + \mathbf{1}^\top c(\mathbf{y})\}, \\ \mathbf{u}|\mathbf{G} &\sim \mathrm{N}(\mathbf{0}, \mathbf{G}), \end{aligned}$$

where the fixed effects are denoted by the vector $\boldsymbol{\beta}$, the random effects are denoted by $\mathbf{u}$ and $\mathbf{G}$ is the covariance matrix of random effects. The covariance structure in $\mathbf{G}$ is usually chosen to capture the dependencies of interest between the observations in the application, such as the dependency between repeated observations on an individual within a longitudinal study, the dependency between observations within a cluster in a hierarchical model or the spatial dependency between observations that are close to one another in a spatial model. The design matrix for the fixed effects is denoted by $\mathbf{X}$ and the design matrix for the random effects are denoted by $\mathbf{Z}$.

Random effects are very flexible in the variety of models they allow us to fit to our data. Through specification of the covariance structures in the matrix $\mathbf{G}$ with the appropriate data in the design matrix $\mathbf{Z}$, complicated dependencies amongst the responses $\mathbf{y}$ can be specified, allowing modelling of longitudinal data, fitting smoothing splines to the data and modelling spatial relationships between responses. This allows us to fit hierarchical models with random intercepts and slopes, capturing levels of variation within groups within the data (Gelman and Hill, 2007).

While mixed models are very useful for gaining insight into a data set, fitting them can be computationally challenging. For all but the simplest situations, fitting these models involves computing high-dimensional integrals which are often analytically and computationally intractable. The standard technique for fitting Bayesian versions of these models is to use Monte Carlo Markov Chains techniques. Thus, an approximation must be used in order to fit these models within a reasonable time frame. Our approach to this problem is outlined in Chapter 4.

1.4. Splines and smoothing

While linear models are statistically convenient to work with and easy to interpret once fitted, the relationship between the response and explanatory variables may not always be linear in practice. Thus a generalisation of linear models to nonlinear situations is needed that still retains the beneficial statistical and interpretive properties of linear models as much as possible. The most general form of the univariate regression problem is $y_i = f(x_i)$ where $f : \mathbb{R} \to \mathbb{R}$ is unknown, and we wish to estimate it. Fully nonparametric regression is a difficult problem to solve, but the problem can be simplified by prespecifying the points at which the function may change curvature, which we refer to as *knots*.

1.4.1. B-Splines. There are many families of basis functions which can be conveniently used for function approximation, including orthogonal polynomials. The B-spline basis (de Boor, 1972) is numerically stable and efficient to computationally evaluate. A B-Spline is a piecewise polynomial function of degree $< n$ in a variable x. It is defined over a domain $\kappa_0 \leq x \leq \kappa_m, m = n$. The points where $x = \kappa_j$ are known as knots or break points. The number of internal knots is equal to the degree of the polynomial if there are no knot multiplicities. The knots must be in ascending order. The number of knots is the minimum for the degree of the B-spline, which has a non-zero value in the range between the first and last knot. Each piece of the function is a polynomial of degree less than n between and including adjacent knots. A B-Spline is continuous at its knots. When all internal knots are distinct its derivatives are also continuous up to the derivative of degree $n - 1$. If internal knots coincide at a given value of x, the continuity of derivative order is reduced by 1 for each additional knot.

For any given set of knots, the B-spline for approximating a given function is a unique linear combination of basis functions recursively defined as

$$B_{i,0}(x) \quad := \begin{cases} 1 & \text{if } \kappa_i \leq x < \kappa_{i+1}; \quad \text{and} \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, K + 2M - 1$ and

$$B_{i,k}(x; \boldsymbol{\kappa}) \quad = \frac{x - \kappa_i}{\kappa_{i+k} - \kappa_i} Q_{i,k-1}(x; \boldsymbol{\kappa}) + \frac{\kappa_{i+k+1} - x}{\kappa_{i+k+1} - \kappa_{i+1}} Q_{i+1,k-1}(x; \boldsymbol{\kappa})$$

for $i = 1, \ldots, K + 2M - m$ with

$$Q_{m,i}(x;\kappa) = \begin{cases} B_{m,i}(x;\kappa), & \kappa_{i+m} > \kappa_i; \quad \text{and} \\ 0, & \text{otherwise.} \end{cases}$$

We define the B-Spline basis in this way so that the definition remains correct in the case where knots are repeated in $\boldsymbol{\kappa}$. We choose piecewise cubic splines as cubics are numerically well behaved while still capturing the curvature of functions we wish to approximate well (Press et al., 2007a). Thus we select the knot sequence $\boldsymbol{\kappa}$ to be

$$a = \kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 < \kappa_5 < \ldots < \kappa_{K+5} = \kappa_{K+6} = \kappa_{K+7} = \kappa_{K+8} = b.$$

There are many ways of choosing knots for applied statistical problems. A typical approach is to choose the internal knots using the sample quantiles of the data set being examined. A common choice is to select $\min(n_U/4, 35)$ internal knots where n_U is the unique number of x_i's.

1.4.2. O'Sullivan Splines. In this section, we follow the discussion of semiparametric regression in Ruppert et al. (2003). Using a mixed models setup to fit spline models protects against overfitting, we construct a $\mathbf{Z}$ matrix with the appropriate B-Spline function evaluations in each of row of the matrix, where each column in the matrix corresponds to one of the knots we have selected.

O'Sullivan introduced a class of penalised splines based on the B-spline basis functions in O'Sullivan (1986) which are a direct generalisation of smoothing splines. Let $B_1, \ldots, B_{K+4}$ be the cubic B-spline basis functions defined by the knots

κ_1 to κ_{K+4}. O'Sullivan splines are splines which are penalised using the penalty matrix $\mathbf{\Omega}$. Let $\mathbf{\Omega}$ be the $(K+4) \times (K+4)$ matrix where the $(k, k') - th$ element is

$$\mathbf{\Omega}_{kk'} = \int_a^b B_k''(x) B_{k'}''(x) dx.$$

Then the O'Sullivan spline estimate of the true function f at the point x is

$$\widehat{f}_O(x; \lambda) = \mathbf{B}_x \widehat{\boldsymbol{\nu}}_O,$$

where $\widehat{\boldsymbol{\nu}}_O = (\mathbf{B}^\top \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^\top \mathbf{y}$, as shown in Ruppert et al. (2003).

The matrix $\mathbf{\Omega}$ is defined in this way to penalise oscillation, which is measured by the second derivative. This penalty differs from the penalty for "penalised B-Splines" or P-splines in that the P-spline penalty matrix is $\mathbf{D}_2^\top \mathbf{D}_2$ where $\mathbf{D}_2$ is the second-order differencing matrix.

1.5. Variable selection

It is often the case in applied statistics that many covariates are available, but it is unknown a priori which covariates explain the response variable of interest. An automatic method of exploring which model among many possible candidate models incorporating these covariates explains the response variable best would relieve the burden of having to fit and compare the performance of many such models manually.

The problem of selecting a statistical model from a set of candidate models given a data set, hence referred to as *model selection*, is one of the most important problems encountered in practice by applied statisticians. It is one of the central tasks of science, and there is a correspondingly large literature on the subject –

Claeskens and Hjort (2008); Nengjun Yi (2013); Johnstone et al. (2009) together give a comprehensive overview.

The problem of model selection for normal linear models is particularly well studied, owing to the popularity and importance of normal linear models in applications. While new types of model are continually being developed, linear models with normal priors remain a popular and essential modelling tool owing to the ease of fitting these models, statistical inference on the parameters and, most importantly, the ease which these models can be interpreted. But for a data set with a moderate or large number of parameters, the question is immediately raised of which covariates we should include in our model. One of the problems that we address in this thesis is *variable selection* on linear models with normal priors.

The bias-variance trade-off is one of the central issues in statistical learning (Murphy, 2012; Bishop, 2006; Hastie et al., 2001). The guise this issue takes in model selection is balancing the quality of the model fit against the complexity of the model, in an attempt to find a compromise between over-fitting and under-fitting, in the hope that the model fit will generalise well beyond the training data we have observed to the general population and that we haven't simply learned the noise in the training set.

There have been many approaches to model selection proposed, including criteria based approaches, approaches based on functions of the residual sum of squares, penalised regression such as the lasso and L_1 regression, and Bayesian modelling approaches. Model selection is a difficult problem in high-dimensional spaces in general because as the dimension of the space increases, the number of possible models increases combinatorially (Schelldorfer et al., 2010). Many model

selection algorithms use heuristics in an attempt to search the model space more efficiently but still find an optimal or near-optimal model within a reasonable period of time. A major motivation for this field of research is the need for a computationally feasible approach to performing model selection on large scale problems where the number of covariates is large.

1.5.1. Frequentist approaches to model selection.

1.5.1.1. *Information Criteria.* Let $\boldsymbol{\gamma}$ be a p-dimensional vector of indicators, where a 1 in the jth position indicates that the jth covariate is included in the model, while a 0 indicates it is excluded. Thus $\boldsymbol{\gamma}$ defines a model with covariates drawn from a p column data matrix $\mathbf{X}$.

In a frequentist context, there are many functions which can be used to judge which model is best, such as Akaike's Information Criteria (AIC) and the Bayesian Information Criteria (BIC). These are functions $f : \boldsymbol{\gamma} \to \mathbb{R}^+$ which allow the models under consideration to be ranked, and the best model chosen from those available. Thus the optimal model selected by an information criteria is $\boldsymbol{\gamma}^* = \min_{\boldsymbol{\gamma}} f(\boldsymbol{\gamma})$. These functions typically attempt to balance log-likelihood against the complexity of the model, achieving a compromise between each.

Information Criteria are frequently used to compare models. Letting $\boldsymbol{\gamma}$ denote the candidate model, Information Criteria take the form "negative twice times the log-likelihood plus a term penalising for complexity of the mode"

$$\text{Information Criteria} = -2 \log p(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}}) + \text{complexity penalty},$$

where $\widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}}$ is the maximum likelihood estimate of the model parameters $\boldsymbol{\theta}$ for the model $\boldsymbol{\gamma}$ and $\log p(\mathbf{y}|\widehat{\boldsymbol{\theta}_{\gamma}})$ is the log-likelihood of that model with that parameter

estimate and the complexity penalty is a function of the sample size n and the number of parameters p of the model. Information criteria attempt to successfully compromise between goodness of fit and model complexity.

The most popular of the Information Criteria is the AIC (Akaike, 1974). AIC calculates an estimate of the information lost when a given model is used to represent the process that generates the data and so is an estimator of the Kullback-Leibler divergence of the true model from the fitted model. The AIC of the model $\boldsymbol{\gamma}$ is defined as

$$\text{AIC}(\boldsymbol{\gamma}) = -2 \log p(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}}) + 2p_{\boldsymbol{\gamma}},$$

where $p_{\boldsymbol{\gamma}}$ is the number of parameters in the model $\boldsymbol{\gamma}$. The model with the lowest AIC is selected as the 'best'.

Of a similar form as the AIC, but derived via a more Bayesian framework is the BIC. The BIC approximates the posterior probability of the candidate model $\boldsymbol{\gamma}$. The BIC is defined as

$$\text{BIC}(\boldsymbol{\gamma}) = -2 \log p(\mathbf{y}|\widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}}) + p_{\boldsymbol{\gamma}} \log(n).$$

This is a more severe penalty for model complexity than in the Akaike's Information Criteria when n is greater than 8. BIC can be shown to be approximately equivalent to model selection using Bayes Factors in certain contexts (Kass and Raftery, 1993).

1.5.1.2. *Penalised regression.* Another approach is to make the process of model selection can be made implicit in the model fitting process itself. The well-known lasso regression method (Tibshirani, 1996) takes this approach. As Breiman (1996) and Efron (2013) showed, while the standard formulation of a linear model is

unbiased, the goodness of fit of these models is numerically unstable. Breiman showed that by introducing a penalty on the size of the regression coefficients, as in ridge regression, this numerical instability can be avoided. This reduces the variances of the coefficient estimates, at the expense of introducing some bias – which is another instance of the bias-variance trade-off.

Penalised regression methods trade introducing some bias in the estimator for reducing the variance and thus fitting a more parsimonious model. The major advantages are that a model with fewer covariates will be correspondingly easier to interpret, and that the variance of the regression co-efficient estimator will be less. In penalised regression, the regression coefficients are subjected to a penalty or constraint. This is typically expressed as the minimisation of the sum of a goodness of fit function such as squared Euclidean distance and a penalty function

$$\widehat{\boldsymbol{\beta}}_{\text{penalised}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \text{penalty}(\boldsymbol{\beta}).$$

From a Bayesian perspective, the penalty can be considered as a prior distribution on the regression coefficients where smaller values of $\boldsymbol{\beta}$ are given more weight than larger ones. Here the penalised estimate of the regression coefficients is the mode of their posterior distribution.

1.5.1.3. *Ridge regression.* Ridge regression is a penalised regression method, introduced in Hoerl and Kennard (1970). The penalty on the regression coefficients is the Euclidean norm of the regression coefficients. This penalty shrinks the estimated coefficients towards zero. The ridge regression coefficients can thus be estimated by solving the constrained optimisation problem

$$\widehat{\boldsymbol{\beta}}_{\text{ridge}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_2 \leq \lambda$$

where λ is a pre-specified free parameter specifying the amount of regularisation. This constrained optimisation problem can be transformed by the method of Lagrange multipliers into the sum of the residual sum of squares and the product of the Lagrange multiplier and the constraint, which acts as a penalty on the Euclidean norm of the regression coefficients.

1.5.1.4. *Lasso regression.* Lasso regression is a penalised regression method developed in Tibshirani (1996), which was directly inspired by ridge regression. The penalty is the l_1 norm of the coefficient vector. The lasso regression coefficients can be estimated by solving the constrained optimisation problem

$$\widehat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \text{ subject to } \|\boldsymbol{\beta}\|_1 \leq \lambda,$$

where λ is a pre-specified free parameter specifying the amount of regularisation. Similarly to the constrained optimisation problem for ridge regression, the constrained optimisation problem can be transformed by the method of Lagrange multipliers into the sum of the residual sum of squares and the product of the Lagrange multiplier and the constraint, which acts as a penalty on the l_1 norm of the regression coefficients. It follows from Minkowski's inequality that the function above is convex, and thus the optimisation problem is convex, and can be solved using standard methods from convex optimisation (Boyd and Vandenberghe, 2010). The constraint on the l_1 norm has the effect of shrinking the coefficients, and setting some of them to zero. This forces the models fit by lasso

regression to be sparse, providing model selection as part of the model-fitting process.

A disadvantage of lasso regression is that the constraint on the regression coefficients depends on the free tuning parameter which must be selected a priori or through cross-validation. But a much greater issue is that the model selection process intrinsic to lasso regression does not take into account the uncertainty of the model selection process itself, particularly the selection of λ, as Bayesian model selection methods do.

1.5.2. Bayesian approaches to Model Selection. Parallel to the frequentist approaches, model selection can be performed using a Bayesian approach. This can be done, for example, by using Bayes Factors to compare the marginal likelihoods of the candidate models to see which is most probable given the observed data (Kass and Raftery, 1993). Rather than selecting one candidate model, several models can be combined together using Bayesian model averaging (Hoeting et al., 1999; Raftery et al., 1997; Fernández et al., 2001; Papaspiliopoulos and Rossell, 2016).

1.5.2.1. *Variable selection.* A special case of model selection is variable selection, where the focus is on selecting individual covariates, rather than entire models. Variable selection approaches search over the variables in the model space for the best covariates to include in the candidate model. Due to the large number of possible combinations of covariates – typically 2^p where p is the number of covariates, such searches are often stochastic. This approach can either be fully Bayesian or empirically Bayesian (Cui and George, 2008). This search can be driven by posterior probabilities (Casella and Moreno, 2006), or by Gibbs sampling approaches

such in George and McCulloch (1993). These two approaches of model selection and variable selection can be combined (Geweke, 1996). Variable selection can also be accomplished by selecting the median probability model, consisting of those models whose posterior inclusion probability is at least $1/2$ (Barbieri and Berger, 2004).

A challenge to applying this method of model selection is that exact model fitting may be computationally infeasible for models involving even moderate numbers of observations and covariates, and popular alternatives for fitting Bayesian models such as Monte Carlo Markov Chains (MCMC) are still extremely computationally intensive.

1.6. Approximate Bayesian inference

When the prior and model chosen for a Bayesian model is conjugate, the posterior distribution is available in closed form and can be easily calculated. When the prior is non-conjugate, the integral in Equation 1 to calculate the posterior distribution is typically intractable and so numerical methods must be used to calculate it approximately. The gold standard for Bayesian inference is to use MCMC methods such as Metropolis-Hastings or Gibbs sampling. But these methods are computationally intensive, to the point where they are simply impractical in Big Data situations where n or p are large. Moreover, they can be prone to convergence problems. Thus there is a need for approximate Bayesian inference methods which are less computationally intensive while being nearly as accurate for some models.

1.6.1. Variational Bayes. We now introduce Variational Bayes (VB), the popular approximate inference method for Bayesian models. It is used to accelerate Bayesian model fitting by tens or hundreds of times, with sometimes only minor loss in accuracy for some models. This method plays a central role in this thesis, particularly in the third and fourth chapters.

As described previously, Bayesian models may be computationally difficult or intractable to fit. The calculation of the true posterior distribution for the model is often either computationally intractable or no closed form exists for the posterior distribution. We may be able to gain much of the same insight from a given data set by fitting an accurate approximation of the model, allowing us to summarise the data and perform statistical inference. Variational approximation aims to approximate a true, possibly intractable probability distribution $p(x)$ by a simpler, more tractable distribution $q(x)$ of known form.

Variational approximation often takes the form minimising the Kullback-Leibler divergence between the true posterior $p(\boldsymbol{\theta}|\mathbf{y})$ and an approximating distribution $q(\boldsymbol{\theta})$, sometimes called a q-density. For an introduction, see Ormerod and Wand (2010).

The KL divergence between the probability distributions p and q is defined as

$$\mathrm{KL}(q||p) \equiv \int q(\boldsymbol{\theta}) \log \left[\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})} \right] d\boldsymbol{\theta}.$$

Suppose that a class of candidate approximating distributions $q(\boldsymbol{\theta})$ is parameterised by a vector of variational parameters $\boldsymbol{\xi}$ and write $q(\boldsymbol{\theta}) \equiv q(\boldsymbol{\theta}; \boldsymbol{\xi})$. We

attempt to find an optimal approximating distribution $q^*(\boldsymbol{\theta})$ such that

$$q^*(\boldsymbol{\theta}) = \underset{\boldsymbol{\xi} \in \Xi}{\operatorname{argmin}} \ \mathrm{KL}\{q(\boldsymbol{\theta}; \boldsymbol{\xi}) || p(\boldsymbol{\theta}|\mathbf{y})\}.$$

If $\boldsymbol{\theta}$ is partitioned into M partitions $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M$ then a simple form of approximation to adopt is the factored approximation of the form

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{M} q(\boldsymbol{\theta}_i)$$

where each of the density $q(\boldsymbol{\theta}_i)$ is a member of a parametric family of density functions. This form of approximation is computationally convenient, but assumes that the partitions of $\boldsymbol{\theta}$ are completely independent of one another.

The optimal mean field update for each of the parameters $\boldsymbol{\theta}_i$ can be shown to be

$$q^*(\boldsymbol{\theta}_i) \propto \exp\left[\mathbb{E}_q\left\{\log p(\mathbf{y}; \boldsymbol{\theta})\right\}\right].$$

For details of the proof, and a more thorough introduction to the topic of variational approximations, see Ormerod and Wand (2010). It can easily be shown that

$$\log p(\mathbf{y}) = \int q(\boldsymbol{\theta}; \boldsymbol{\xi}) \log\left[\frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta}; \boldsymbol{\xi})}\right] d\boldsymbol{\theta} + \mathrm{KL}(q(\boldsymbol{\theta}; \boldsymbol{\xi}) || p(\boldsymbol{\theta}|\mathbf{y})).$$

As the Kullback-Leibler divergence is strictly positive, the first term on the right hand side is a lower bound on the marginal log-likelihood which we will define by

$$\log \underline{p}(\mathbf{y}; \boldsymbol{\xi}) \equiv \int q(\boldsymbol{\theta}; \boldsymbol{\xi}) \log\left[\frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta}; \boldsymbol{\xi})}\right] d\boldsymbol{\theta}$$

and maximizing $\log \underline{p}(\mathbf{y}; \boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$ is equivalent to minimizing $\mathrm{KL}(q(\boldsymbol{\theta}; \boldsymbol{\xi}) || p(\boldsymbol{\theta}|\mathbf{y}))$. The term $\log \underline{p}(\mathbf{y}; \boldsymbol{\xi})$ is referred to as the variational lower bound.

When the optimal distributions for each $q_i^*(\boldsymbol{\theta}_i)$ are calculated, they yield a set of equations, sometimes called the consistency conditions, which need to be satisfied simultaneously. These yield a series of mean field updates for the parameters of each approximating distribution. By executing the mean field update equations in turn for each parameter in the model, the variational lower bound for the model $\underline{p}(\boldsymbol{\theta}; \mathbf{y})$ is iteratively increased. It can be shown that by calculating $q_i^*(\boldsymbol{\theta}_i)$ for a particular i with the remaining $q_j^*(\boldsymbol{\theta}_j)$, $j \neq i$ fixed, results in a monotonic increase in the variational lower bound, and thus a monotonic decrease in the Kullback-Leibler divergence between $p(\boldsymbol{\theta}|\mathbf{y})$ and $q(\boldsymbol{\theta})$.

The variational lower bound is maximised iteratively. On each iteration, the value of each parameter in the model is calculated as the expectation of the full likelihood relative to the other parameters in the model, which is referred to as the mean field update. This is done for each parameter in the model in turn until the variational lower bound's increase is negligible and convergence is achieved. Note that this approach can be extended to a wide range of models such as semi-parametric models as has been formalized by Rohde and Wand (2015).

This approach works well for classes of models where all of the parameters are conjugate. For more general classes of models, the mean field updates are not analytically tractable and general gradient-based optimisation methods must be used, such as for the Gaussian Variational Approximation (Ormerod and Wand, 2012). These methods are generally difficult to apply in practice, as the problems can involve the optimisation of many parameters over high-dimensional, constrained spaces whose constraints cannot be simply expressed.

Recently, several stochastic Variational Bayes approaches to approximation problems of this type have emerged. Gershman et al. (2012) used a uniform weighted mixture of isotropic Gaussians to approximate complex posterior distributions. The variational lower bound is approximated with first and second-order Taylor series expansions, and then optimised with L-BFGS. In Kingma and Welling (2013), the expectations in the expression for the variational lower bound are approximated using Monte Carlo integration. The variational lower bound is reparameterised in terms of an auxiliary noise variable such as a standard normal, to reduce the variance of the Monte Carlo estimate. Tan and Nott (2018) takes an approach closest to the one we will adopt, using a Gaussian Variational Approximation. By parameterising the covariance matrix of the Gaussian using Cholesky factors of the precision matrix, the covariance matrix is guaranteed to be sparse due to the conditional independence between fixed and random effects of the mixed model. The variational lower bound can be rewritten so that it does not depend on the variational parameters. By making a transformation in terms of a noise variable to standardise the variational parameters, efficient gradient estimators can be derived, then estimated using subsampling of the data set. Sampling from the fixed normal distribution on each iteration rather than a multivariate normal depending on the variational parameters in the current iteration reduces the variance of the estimator. Subsampling of the data set and sampling from the noise variable make the fitting algorithm doubly stochastic.

Other approximate Bayesian inference techniques exist in the literature, such as Laplace approximation (Tierney and Kadane, 1986), integrated nested Laplace approximation (Rue et al., 2009), and Expectation Propagation (Minka, 2013). These

have been applied to the problem of fitting count models (Barber et al., 2016; Kim and Wand, 2017). But Expectation Propagation requires very difficult algebra to complete the derivations required for the updates, and can exhibit convergence problems. Laplace approximation relies on a Gaussian approximation to the log of the posterior found by Taylor expanding around the mode, which performs poorly when the true posterior is not symmetric, as is the case for Poisson regression models.

1.6.2. Gaussian Variational Approximation. In cases where there is a strong dependence between partitions of $\boldsymbol{\theta}$, such as between the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in a hierarchical Gaussian model, a factored approximation may not approximate the true distribution accurately. In this case, an alternate form of approximation may be used with the parameters considered together to take their dependence into account. One such form of approximation is the Gaussian Variational Approximation (Ormerod and Wand, 2012), which assumes that the distribution of the parameters being approximated is multivariate Gaussian. The covariance matrix of the Gaussian allows the approximation to capture the dependence amongst the elements of $\boldsymbol{\theta}$, which increases the accuracy of the variational approximation relative to the factored approximation. This will be the approach used in Chapter 4.

1.6.3. Laplace Method of approximation. Laplace's method of approximation, as described in Butler (2007) or MacKay (2002), is used to approximate integrals of a unimodal function f with negative second derivative at the mode,

indicating that the function is decreasing rapidly away from this point. The essential idea is that if the function is decreasing rapidly away from the mode, the bulk of the area under the function will be within a neighbourhood of the mode. Thus, the integral of the function can be well approximated by an integral over the neighbourhood of the mode. How large that neighbourhood needs to be is estimated using how fast the function is changing at the mode x_m, which is estimated by $|f''(x_m)|$.

Consider an exponential integral of the form

$$\int_a^b e^{Mf(x)}dx$$

where $f(x)$ is twice differentiable and $f''(x_m) < 0$, $M \in \mathbb{R}$ and $a, b \in \mathbb{R} \cup \{-\infty, \infty\}$. Let $f(x)$ have a unique mode at x_m. Then, Taylor expanding about x_m, we have

$$f(x) = f(x_m) + f'(x_m)(x - x_m) + \frac{1}{2}f''(x_m)(x - x_m)^2 + \mathcal{O}\left((x - x_m)^3\right).$$

As f has a global maximum at x_m, the first derivative of f is zero at x_m. Thus, the function $f(x)$ may be approximated by

$$f(x) \approx f(x_m) - \frac{1}{2}|f''(x_m)|(x - x_m)^2$$

for x sufficiently close to x_m, as the second derivative is negative at x_m. This ensures the approximation of the integral

$$\int_a^b e^{Mf(x)}dx \approx e^{Mf(x_m)} \int_a^b e^{-M|f''(x_m)|(x-x_m)^2}dx$$

is accurate. The integral on the right-hand side of the equality is a Gaussian integral, and thus we find that

$$\int_a^b e^{Mf(x)}dx \approx \sqrt{\frac{2\pi}{M|f''(x_m)|}}e^{Mf(x_m)}.$$

Thus, we have approximated our integral by a closed form expression. The error in the approximation is $\mathcal{O}(1/M)$. The approximation can be made more accurate by using a Taylor expansion beyond second order.

1.6.3.1. *Extending to multiple dimensions.* This approach to approximating integrals extends naturally to multiple dimensions. Consider the second order Taylor expansion of $\log f(\boldsymbol{\theta}) : \mathbb{R}^p \to \mathbb{R}$ around the mode $\boldsymbol{\theta}_m \in \mathbb{R}^p$ given by

$$\begin{aligned}\log f(\boldsymbol{\theta}) \quad &\approx f(\boldsymbol{\theta}_m) + (\boldsymbol{\theta} - \boldsymbol{\theta}_m)^\top \nabla \log f(\boldsymbol{\theta}_m) + \tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_m)^\top \mathbf{H}_{\log f}(\boldsymbol{\theta}_m)(\boldsymbol{\theta} - \boldsymbol{\theta}_m) \\ &\quad + \mathcal{O}(\|\boldsymbol{\theta} - \boldsymbol{\theta}_m\|^3).\end{aligned}$$

where $\nabla \log f(\boldsymbol{\theta}_m)$ is the gradient of the log-likelihood at $\boldsymbol{\theta}_m$ and $\mathbf{H}_{\log f}(\boldsymbol{\theta}_m)$ is the Hessian matrix of the log-likelihood at $\boldsymbol{\theta}_m$. Assuming that $\boldsymbol{\theta}_m$ is a stationary point of $\log f$, then $\nabla f(\boldsymbol{\theta}_m) = \mathbf{0}$ and so

$$\log f(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}_m) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_m)^\top \mathbf{H}_{\log f}(\boldsymbol{\theta}_m)(\boldsymbol{\theta} - \boldsymbol{\theta}_m) + \mathcal{O}(\|\boldsymbol{\theta} - \boldsymbol{\theta}_m\|^3)$$

at such a point. The quadratic form in $\boldsymbol{\theta}$ in the approximate expression for the log likelihood above leads to a Gaussian approximation for the likelihood

$$\mathrm{N}(\boldsymbol{\theta}_m, -\mathbf{H}_{\log f}(\boldsymbol{\theta}_m)^{-1}).$$

The approximation is crude but can be quite accurate if the likelihood is symmetric and unimodal, which is often the case when the sample size is large.

1.6.4. Other methods: Expectation propagation. Expectation Propagation is an approximate Bayesian inference method, first proposed in Minka (2001). It relies on minimising the reverse KL divergence $\mathrm{KL}(p||q)$ between the true and approximating distributions p and q. A factorised form of the distribution

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{n} q(\boldsymbol{\theta}_i)$$

is assumed. In general, fully minimising the KL divergence between p and q is intractable, so Expectation Propagation approximates this by minimising the KL divergence of each of the factors individually. It does this by cycling through each of the factors matching the sufficient statistics of each, incorporating the information already in the other factors. The factors are cycled through several times until convergence is achieved.

While promising, unlike with Variational Bayes, there is no guarantee of convergence, and there is still much work to be done before it is as mature as other approximation methods like Variational Bayes and Laplace approximation.

A linear model with normal priors allows exact inference on the regression and model selection parameters in closed form, which might appear to negate the benefits of a variational approximation to the model. However, the performance of our variational approximation should remain similar if the priors are altered to cater for complications such as robustness, while exact Bayesian inference calculations are no longer possible in closed form in these situations.

1.7. Our contributions

In this section, we briefly outline the major contributions in this thesis.

- A popular choice of Bayesian model selection is to use regression models with g-priors. For the Beta Prime prior (Maruyama and George, 2011) we were able to derive closed form expressions for the posterior distributions of most of the parameters of the model in terms of the hypergeometric function.
- An important consideration in model selection is being able to compare models against one another. Calculation of the Bayes Factors for comparing models requires being able to compute the posterior distribution of g. In our second chapter, we derive closed form expressions in terms of special functions for the posterior distributions of g for a number of choices of g prior from the literature: Liang's hyper-g prior, Liang's hyper-g/n prior (Liang et al., 2008), Bayarri's robust g prior (Bayarri et al., 2012) and the Beta-Prime (Maruyama and George, 2011) prior.
- Exact inference for model selection for linear models with normal priors is computationally feasible when the number of covariates is small, with p below 40. But exhaustively exploring the search space is not efficient, and often not computationally feasible for a larger number of covariates. To deal with this situation, in our fourth chapter, we adopt a population- based technique inspired by Ročková's work on population-based EM (Ročková, 2017) to efficiently explore the posterior model space. Instead, approximate methods can be used to search the parts of the model space for which the posterior model likelihood is the highest. In our third chapter, we propose a population-based algorithm, which works by adding or removing

a covariate at a time to each of the fitted models in the population. We implement this algorithm for a number of model selection priors from the literature: the Liang's hyper-g prior, the Liang's hyper-g/n prior (Liang et al., 2008), Bayarri's robust g prior (Bayarri et al., 2012) and the Maruyama and George Beta-Prime prior (Maruyama and George, 2011).

- We are able to implement this algorithm efficiently by using rank-one updates and downdates and the closed forms of the posteriors for the model selection priors that we consider. The population-based approach allows us to estimate the uncertainty in the model selection process.
- Generalised Linear Mixed Models are an appealing way to model data, as they are flexible enough to model a range of data types and situations. But the Bayesian versions of these models typically require computationally demanding MCMC, which can also be prone to convergence problems. Instead, we consider approximate Bayesian inference techniques, which are computationally efficient and deterministic.
- It is desirable to use normal priors for the regression coefficients of these models, as these are easily interpreted. But for Generalised Linear Mixed Models with a non-normal response, these priors are non-conjugate, making VB difficult to apply as the required mean field updates are intractable. We apply Gaussian Variational Bayes – an extension to Variational Bayes, to fit a multivariate normal distribution to the regression coefficients of our models.
- In our fourth chapter, we present a Gaussian Variational Approximation to a zero-inflated Poisson mixed model which can flexibly incorporate both

fixed and random effects. This allows us to use our model fitting algorithm to fit complicated models to the data incorporating random intercepts and slopes and additive models using O'Sullivan-penalised splines. The model is fit by optimising the conditional likelihood of the Gaussian component of the model given the parameters governing zero-inflation and the covariance matrix $\boldsymbol{\Sigma}$.

- We present a new parameterisation for the covariance matrix of the Gaussian based on the Cholesky factorisation of the precision matrix, and detail computation and numerical advantages of this factorisation, owing to its sparsity when the form of the covariance matrix of the Gaussian is known due to knowledge of the random effects in the model.

CHAPTER 2

Calculating Bayes factors for linear models using mixture g-priors

Abstract

In this chapter, we consider the numerical evaluation of Bayes factors for linear models using different mixture g-priors. In particular, we consider hyperpriors for g leading to closed-form expressions for the Bayes factor including the hyper-g and hyper-g/n priors of Liang et al. (2008), the beta-prime prior of Maruyama and George (2011), the robust prior of Bayarri et al. (2012), and the Cake prior of Ormerod et al. (2017). In particular, we describe how each of these Bayes factors, except for Bayes factor under the hyper-g/n prior, can be evaluated in an efficient, accurate and numerically stable manner. We also derive a closed form expression for the Bayes factor under the hyper-g/n for which we develop a convenient numerical approximation. We implement an R package for Bayesian linear model averaging, and discuss some associated computational issues. We illustrate the advantages of our implementation over several existing packages on several small datasets. [1]

[1]This chapter corresponds to the collaborative paper: Greenaway M.J. & Ormerod J.T (2018). Numerical aspects of calculating Bayes factors for linear models using mixture g-priors. Submitted to the Journal of Computational and Graphical Statistics.

2.1. Introduction

There has been a large amount of research in recent years into the appropriate choice of suitable and meaningful priors for linear regression models in the context of Bayesian model selection and averaging. Specification of the prior structure of these models must be made with great care in order for Bayesian model selection and averaging procedures to have good theoretical properties. A key problem in this context occurs when the models have differing dimensions and non-common parameters where inferences are typically highly sensitive to the choice of priors for the non-common parameters due to the Jeffreys-Lindley-Bartlett paradox (Lindley, 1957; Bartlett, 1957; Ormerod et al., 2017). Furthermore, this sensitivity does not necessarily vanish as the sample size grows (Kass and Raftery, 1995; Berger and Pericchi, 2001).

Bayes factors in the context of linear model selection (Zellner and Siow, 1980a,b; Mitchell and Beauchamp, 1988; George and McCulloch, 1993; Fernández et al., 2001; Liang et al., 2008; Maruyama and George, 2011; Bayarri et al., 2012) have received an enormous amount of attention. A landmark paper in this field is Liang et al. (2008). Liang et al. (2008) considers a particular prior structure for the model parameters. In particular they consider a Zellner's g-prior (Zellner and Siow, 1980a; Zellner, 1986) for the regression coefficients where g is a prior hyper-parameter. The parameter g requires special consideration. If g is set to a large constant most of the posterior mass is placed on the null model, a phenomenon sometimes referred to as Bartlett's paradox. Due to this problem they discuss previous approaches which set g to a constant, e.g., setting $g = n$ (Kass and Wasserman, 1995), $g = p^2$ (Foster and George, 1994), and $g = \max(n, p^2)$ (Fernández

et al., 2001). However, Liang et al. (2008) showed that all of these choices lead to what they call the information paradox, where the posterior probability of the true model does not tend to 1 as the sample size grows. Finally, Liang et al. (2008) also consider a local and global empirical Bayes (EB) procedure for selecting g. In these cases Liang et al. (2008) show that these EB procedures are model selection consistent except when the true model is the null model (the model containing the intercept only).

The above problems suggest that a hyperprior should be placed on g. Bayarri et al. (2012) also discuss in some depth desirable properties priors should have in the context of linear model averaging and selection. In this chapter we review the prior structures, specifically the hyperpriors on g, that lead to closed form expressions of Bayes factors for comparing linear models. These include linear models with Zellner g-priors with mixture g priors including the hyper-g prior of Liang et al. (2008), the beta-prime prior of Maruyama and George (2011), and the robust prior of Bayarri et al. (2012), and most recently the Cake prior of Ormerod et al. (2017). We concern ourselves with the efficient, accurate and numerically stable evaluation of Bayes factors, Bayesian model averaging, and Bayesian model selection for linear models under the above choices of prior structures for the model parameters.

Our main contributions in this chapter are as follows.

a) To the above list of hyperpriors on g leading to closed form Bayes factors we add the hyper-g/n prior of Liang et al. (2008) for which we derive a new closed form expression for the Bayes factor in terms of the Appell hypergeometric function.

b) We derive an alternative expression for the Bayes factor when using the robust prior of Bayarri et al. (2012) in terms of the Gaussian hypergeometric function.

c) We describe how the Bayes factors corresponding to the hyper-g prior of Liang et al. (2008) and robust prior of Bayarri et al. (2012) can be calculated in an efficient, accurate and numerically stable manner without the need for special software or approximation.

d) We derive a reasonably accurate approximation for the Appell hypergeometric function which can be calculated in an efficient and numerically stable manner when the number of non-zero coefficients in a particular model is strictly greater than 2.

e) We make available a highly efficient and *numerically stable* `R` package called `blma` available for exact Bayesian linear model averaging using the above prior structures which is available for download from the following web address.

`http://github.com/certifiedwaif/blma`

We demonstrate the advantages of our implementation of exact Bayesian model averaging over some existing `R` packages using several small datasets.

The chapter is organised as follows. Section 2.2 describes Bayesian model averaging and model selection for linear models. Section 2.3 outlines and justifies our chosen model and prior structure for the linear regression model parameters. Section 2.4 derives closed form expressions for various marginal likelihoods using different hyperpriors for g and, wherever possible, describes how these may

be evaluated well numerically. In Section 2.6, we discuss details of our implementation which made our implementation computationally feasible. In Section 2.7 we perform a series of numerical experiments to show the advantages of our approach.

2.2. Bayesian linear model selection and averaging

Suppose $\mathbf{y} = (y_1, \ldots, y_n)^T$ is a response vector of length n, $\mathbf{X}$ is an $n \times p$ matrix of covariates where we anticipate a linear relationship between $\mathbf{y}$ and $\mathbf{X}$, but do not know which of the columns of $\mathbf{X}$ are important to the prediction of $\mathbf{y}$. Bayesian model averaging seeks to improve prediction by averaging over multiple predictions over different choices of combinations of predictors.

We consider the linear model for predicting $\mathbf{y}$ with design matrix $\mathbf{X}$ via

$$\mathbf{y}|\alpha, \boldsymbol{\beta}, \sigma^2 \sim \mathrm{N}_n(\mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}), \tag{4}$$

where α is the model intercept, $\boldsymbol{\beta}$ is a coefficient vector of length p, σ^2 is the residual variance, and $\mathbf{I}$ is the $n \times n$ identity matrix. Without loss of generality, to simplify later calculations, we will standardize $\mathbf{y}$ and $\mathbf{X}$ so that $\overline{y} = 0$, $\|\mathbf{y}\|^2 = \mathbf{y}^T\mathbf{y} = n$, $\mathbf{X}_j^T\mathbf{1} = 0$, and $\|\mathbf{X}_j\|^2 = n$ where $\mathbf{X}_j$ is the jth column of $\mathbf{X}$.

Suppose that we wish to perform Bayesian model selection, model averaging or hypothesis testing where we are interested in comparing how different subsets of predictors (which correspond to different columns of the matrix $\mathbf{X}$) have on the response $\mathbf{y}$. To this end, let $\boldsymbol{\gamma} \in \{0, 1\}^p$ be a binary vector of indicators for the inclusion of the pth column of $\mathbf{X}$ in the model where $\mathbf{X}_{\boldsymbol{\gamma}}$ denotes the design matrix formed by including only the jth column of $\mathbf{X}$ when $\gamma_j = 1$, and excluding it

otherwise. Let $\boldsymbol{\beta}_{-\boldsymbol{\gamma}}$ denote the elements of the regression co-efficients not included in the model $\boldsymbol{\gamma}$.

In order to keep our exposition as general as possible we will assume a prior structure of $p(\alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})$ but, for the time being, we will leave the specific form of $p(\alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}|\boldsymbol{\gamma})$ and $p(\boldsymbol{\gamma})$ unspecified. Let $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ denote the subvector of $\boldsymbol{\beta}$ of length $|\boldsymbol{\gamma}| = \mathbf{1}^T\boldsymbol{\gamma}$ corresponding to the components of $\boldsymbol{\gamma}$ which equal 1. Similarly, let $\boldsymbol{\beta}_{-\boldsymbol{\gamma}}$ denote the subvector of $\boldsymbol{\beta}$ of length $p - |\boldsymbol{\gamma}|$ corresponding to the components of $\boldsymbol{\gamma}$ which equal 0. We adopt a prior on $\boldsymbol{\beta}_{-\boldsymbol{\gamma}}$ of the form

$$p(\boldsymbol{\beta}_{-\boldsymbol{\gamma}}|\boldsymbol{\gamma}) = \prod_{j=1}^{p} \delta(\beta_j; 0)^{1-\gamma_j}, \tag{5}$$

where $\delta(x; a)$ is the Dirac delta function with location a. The prior on $\boldsymbol{\beta}_{-\boldsymbol{\gamma}}$ in (5) is the spike in a spike and slab prior where the prior on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is assumed to be flat (the slab). There are several variants of the spike and slab prior initially used in Mitchell and Beauchamp (1988) and later refined in George and McCulloch (1993). The above structure implies that $p(\boldsymbol{\beta}_{-\boldsymbol{\gamma}}|\mathbf{y})$ is a point mass at $\mathbf{0}$ and leads to algebraic and computational simplifications for components of $\boldsymbol{\beta}$ when corresponding elements of $\boldsymbol{\gamma}$ are zero. Thus, $\gamma_j = 0$ is equivalent to excluding the corresponding predictor $\mathbf{X}_j$ from the model.

Exact Bayesian model averaging revolves around the posterior probability of a model $\boldsymbol{\gamma}$ using Bayes theorem

$$p(\boldsymbol{\gamma}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}'} p(\mathbf{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')} = \frac{p(\boldsymbol{\gamma})\text{BF}(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}'} p(\boldsymbol{\gamma}')\text{BF}(\boldsymbol{\gamma}')} \quad \text{where} \quad p(\mathbf{y}|\boldsymbol{\gamma}) = \int p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\gamma})\, d\boldsymbol{\theta},$$

letting $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}, \sigma^2)$, using $\sum_{\boldsymbol{\gamma}}$ to denote a combinatorial sum over all 2^p possible values of $\boldsymbol{\gamma}$, and $\text{BF}(\boldsymbol{\gamma}) = p(\mathbf{y}|\boldsymbol{\gamma})/p(\mathbf{y}|\mathbf{0})$ is the null based Bayes factor for model $\boldsymbol{\gamma}$. Note that the Bayes factor is a statistic commonly used in Bayesian hypothesis testing (Kass and Raftery, 1995; Ormerod et al., 2017). Prediction is based on the posterior distributions of α and $\boldsymbol{\beta}$ where $p(\boldsymbol{\beta}|\mathbf{y}) = \sum_{\boldsymbol{\gamma}} p(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\gamma}) \cdot p(\boldsymbol{\gamma}|\mathbf{y})$ (with similar expressions for α and σ^2). The posterior expectation of $\boldsymbol{\gamma}$ is given by $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y}) = \sum_{\boldsymbol{\gamma}} \boldsymbol{\gamma} \cdot p(\boldsymbol{\gamma}|\mathbf{y})$.

If one is required to select a single model, say $\boldsymbol{\gamma}^*$, two common choices are the highest posterior model (HPM) which uses $\boldsymbol{\gamma}^* = \boldsymbol{\gamma}_{\text{HPM}} = \text{argmax}_{\boldsymbol{\gamma}}\{\, p(\mathbf{y}|\boldsymbol{\gamma}) \,\}$, or the median posterior model (MPM) where $\boldsymbol{\gamma}^*$ is obtained by rounding each element of $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y})$ to the nearest integer. The MPM has predictive optimality properties (Barbieri and Berger, 2004). If the MPM is used for model selection the quantity $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y})$ is sometimes referred to as the posterior (variable) inclusion probability (PIP) vector.

Ignoring for the moment the problems associated with specifying $p(\alpha, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma})$, all of the above quantities are conceptually straightforward. In practice the computation of the quantities $p(\boldsymbol{\gamma}|\mathbf{y})$, $p(\boldsymbol{\beta}|\mathbf{y})$ and $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y})$ are only feasible for small values of p (say around $p = 30$). For large values of p we need to pursue alternatives to exact inference.

2.3. Prior specification for linear model parameters

We will specify the prior $p(\alpha, \boldsymbol{\beta}, \sigma^2|\boldsymbol{\gamma})$ as follows

$$p(\alpha) \propto 1, \qquad \boldsymbol{\beta}_{\boldsymbol{\gamma}}|\sigma^2, g, \boldsymbol{\gamma} \sim \text{N}_p(\mathbf{0}, g\sigma^2(\mathbf{X}_{\boldsymbol{\gamma}}^T\mathbf{X}_{\boldsymbol{\gamma}})^{-1}), \quad \text{and} \quad p(\sigma^2) \propto (\sigma^2)^{-1}, \tag{6}$$

where we have introduced a new prior hyperparameter g. For the time being we will defer specification of $p(g)$ and $p(\boldsymbol{\gamma})$. We will now justify each element of the above prior structure.

The priors on α and σ^2 are improper Jeffreys priors and have been justified in Berger et al. (1998). In the context of Bayesian model selection, model averaging or hypothesis testing α and σ^2 appear in all models so that when comparing models the proportionality constants in the corresponding Bayes factors cancel. It can be shown that the parameter posteriors are proper provided $n \geq 2$ (see Bayarri et al., 2012).

The prior on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is Zellner's g-prior (see for example, Zellner, 1986) with prior hyperparameter g. This family of priors for a Gaussian regression model where the prior covariance matrix of $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is taken to be a multiple of g with the Fisher information matrix for $\boldsymbol{\beta}$. This places the most prior mass for $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ on the section of the parameter space where the data is least informative, and makes the marginal likelihood of the model scale-invariant. Furthermore, this choice of prior removes a log-determinant of $\mathbf{X}_{\boldsymbol{\gamma}}^T\mathbf{X}_{\boldsymbol{\gamma}}$ term from the expression for the marginal likelihood, which is an additional computational burden to calculate. The prior on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ combined with the prior on $\boldsymbol{\beta}_{-\boldsymbol{\gamma}}$ in (6) constitutes one variant of the spike and slab prior for $\boldsymbol{\beta}$.

An alternative choice of prior on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ was proposed by Maruyama and George (2011). Let $p_{\boldsymbol{\gamma}} = |\boldsymbol{\gamma}|$, the number of non-zero elements in $\boldsymbol{\gamma}$. We will now describe their prior on $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ for the case where $p_{\boldsymbol{\gamma}} < n - 1$. Let $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ be an eigenvalue decomposition of $\mathbf{X}_{\boldsymbol{\gamma}}^T\mathbf{X}_{\boldsymbol{\gamma}}$ where $\mathbf{U}$ is an orthonormal $p_{\boldsymbol{\gamma}} \times p_{\boldsymbol{\gamma}}$ matrix, and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_{p_{\boldsymbol{\gamma}}})$ is a diagonal matrix of eigenvalues with $\lambda_1 \geq \cdots \geq \lambda_{p_{\boldsymbol{\gamma}}} > 0$.

Then Maruyama and George (2011) propose a prior for $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ of the form

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}}|\sigma^2, g \sim \mathrm{N}(\mathbf{0}, \sigma^2(\mathbf{U}\mathbf{W}\mathbf{U}^\top)^{-1}), \tag{7}$$

where $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_{p_{\boldsymbol{\gamma}}})$ with $w_j = \lambda_j/[\nu_j(1+g)-1]$ for some prior hyperparameters $\nu_q < \ldots < \nu_1$. Maruyama and George (2011) suggest as a default choice for the ν_j's to use $\nu_j = \lambda_j/\lambda_{p_{\boldsymbol{\gamma}}}$, for $1 \leq j \leq p_{\boldsymbol{\gamma}}$. This choice down-weights the prior on the rotated parameter space of $(\mathbf{U}\boldsymbol{\beta})_j$ when the corresponding eigenvalue λ_j is large, which leads to prior variances on the regression coefficients that are approximately the same size. Note that when $\nu_1 = \ldots = \nu_{p_{\boldsymbol{\gamma}}} = 1$ the prior (7) reduces to the prior for $\boldsymbol{\beta}$ in (6).

The choice between (7) and the prior for $\boldsymbol{\beta}$ in (6) represents a trade-off over computational efficiency and desirable statistical properties. We choose (6) because it avoids the computational burden of calculating an eigenvalue or a singular value decomposition of a $p_{\boldsymbol{\gamma}} \times p_{\boldsymbol{\gamma}}$ matrix for every model considered, which typically can be computed in $O(p_{\boldsymbol{\gamma}}^3)$ floating point operations. It also means that we can exploit efficient matrix updates to traverse the entire model space in a computationally efficient manner allowing this to be done feasibly when p is less than around 30 on a standard 2017 laptop (see Section 2.6 for details).

The marginal likelihood for the model (4) and under prior structure (6). Integrating out α, $\boldsymbol{\beta}$, and σ^2 from $p(\mathbf{y}, \alpha, \boldsymbol{\beta}, \sigma^2|g, \boldsymbol{\gamma})$ we obtain

$$p(\mathbf{y}|g, \boldsymbol{\gamma}) = K(n)(1+g)^{(n-p_{\boldsymbol{\gamma}}-1)/2}(1+g(1-R^2_{\boldsymbol{\gamma}}))^{-(n-1)/2}, \tag{8}$$

where $K(n) = [\Gamma((n-1)/2)]/[\sqrt{n}(n\pi)^{(n-1)/2}]$, and $R^2_{\boldsymbol{\gamma}} = \mathbf{y}^T\mathbf{X}^T_{\boldsymbol{\gamma}}(\mathbf{X}^T_{\boldsymbol{\gamma}}\mathbf{X}_{\boldsymbol{\gamma}})^{-1}\mathbf{X}^T_{\boldsymbol{\gamma}}\mathbf{y}/n$ is the usual R-squared statistic for model $\boldsymbol{\gamma}$. This is the same expression as Liang

et al. (2008) Equation (5) after simplification. Note that when $\boldsymbol{\gamma} = \mathbf{0}$, i.e., the null model, then $p_{\boldsymbol{\gamma}} = 0$, and $R^2_{\boldsymbol{\gamma}} = 0$ leading to the simplification $p(\mathbf{y}|g, \mathbf{0}) = K(n)$ for all g. Hence, $p(\mathbf{y}|\mathbf{0}) = K(n)$ provided the hyperprior for g is a proper density. We will now discuss the specification of g.

2.4. Hyperpriors on g

Here we outline some of the choices of hyperpriors for g used in the literature, their properties, and where possible how to implement these in an efficient, accurate, and numerically stable manner. We cover the hyper-g and hyper-g/n priors of Liang et al. (2008), the beta-prime prior of Maruyama and George (2011), the robust prior of Bayarri et al. (2012), and the Cake prior of Ormerod et al. (2017). We also considered the prior structure implied by Zellner and Siow (1980a), but were unable to make meaningful progress on existing methodology for this case.

We show that many of the hyperpriors on g result in Bayes factors which can be expressed in terms of the Gaussian hypergeometric function denoted ${}_2F_1(\cdot, \cdot; \cdot; \cdot)$ (see for example Chapter 15 of Abramowitz and Stegun, 1972). The Gaussian hypergeometric function is notoriously prone to overflow and numerical instability (Pearson et al., 2017). When such numerical issues arise Liang et al. (2008) derive a Laplace approximation to ${}_2F_1$ implemented in the `R` package `BAS`. Key to achieving accuracy, efficiency and numerical stability for several different mixture g-priors is the following result.

Result 1: *For $x \in (0, 1)$, $c > 1$, and $b + 1 > c$, $a > 0$ we have*

$$ {}_2F_1(a + b, 1; a + 1; x) = \frac{a}{x(1 - x)} \frac{\text{pbeta}(x, a, b)}{\text{dbeta}(x, a, b)}, \tag{9} $$

where pbeta$(x; a, b)$ *and* dbeta$(x; a, b)$ *are the cdf and pdf of the beta distribution respectively.*

Proof: Using identity 2.5.23 of Abramowitz and Stegun (1972) the cdf of the beta distribution can be written as

$$\text{pbeta}(x; a, b) = \frac{x^a}{a\text{Beta}(a, b)} \cdot {}_2F_1(a, 1-b; a+1; x)$$

where Beta(a, b) is the beta function. Using the Euler transformation ${}_2F_1(a, b; c, x) = (1-x)^{c-a-b}{}_2F_1(c-a, c-b; c, x)$, and the fact that ${}_2F_1(a, b; c, x) = {}_2F_1(b, a; c, x)$, we obtain

$$\text{pbeta}(x; a, b) = \frac{x^a(1-x)^b}{a\text{Beta}(a, b)} \cdot {}_2F_1(a+b, 1; a+1; x).$$

Lastly, after rearranging we obtain Result 1.

□

Numerical overflow can be avoided since standard libraries exist for evaluating pbeta(x, a, b) and dbeta(x, a, b) on the log scale. Recently, Nadarajah (2015) stated an equivalent result originally derived in Prudnikov et al. (1986).

2.4.1. The hyper-g prior. Initially, Liang et al. (2008) suggest the hyper g-prior where

$$p_g(g) = \frac{a-2}{2}(1+g)^{-a/2}, \tag{10}$$

for $a > 2$ and $g > 0$. Combining (8) with (10), we have

$$\begin{aligned} p_g(\mathbf{y}|\boldsymbol{\gamma}) &= K(n)\frac{a-2}{2} \\ &\quad \times \int_0^\infty (1+g)^{-a/2}\,(1+g)^{(n-p_{\boldsymbol{\gamma}}-1)/2}\left[1+g(1-R^2_{\boldsymbol{\gamma}})\right]^{-(n-1)/2} dg. \end{aligned} \tag{11}$$

After applying 3.197(5) of Gradshteyn and Ryzhik (2007), i.e.,

$$\int_0^\infty x^{\lambda-1}(1+x)^{\nu}(1+\alpha x)^{\mu}dx = \text{Beta}(\lambda, -\mu-\nu-\lambda)_2F_1(-\mu, \lambda; -\mu-\nu; 1-\alpha), \tag{12}$$

(which holds provided $-(\mu+\nu) > \lambda > 0$), again using ${}_2F_1(a, b; c, x) = {}_2F_1(b, a; c, x)$, and using the mappings

$$\lambda \leftrightarrow 1, \qquad \nu \leftrightarrow \frac{n-p_{\boldsymbol{\gamma}}-1}{2}, \qquad \alpha \leftrightarrow 1-R^2_{\boldsymbol{\gamma}}, \qquad \text{and} \qquad \nu \leftrightarrow -\frac{n-1}{2}$$

leads to

$$\text{BF}_g(\boldsymbol{\gamma}) = \frac{p_g(\mathbf{y}|\boldsymbol{\gamma})}{p_g(\mathbf{y}|\mathbf{0})} = \left(\frac{a-2}{p_{\boldsymbol{\gamma}}+a-2}\right) \cdot {}_2F_1\left(\frac{n-1}{2}, 1; \frac{p_{\boldsymbol{\gamma}}+a}{2}; R^2_{\boldsymbol{\gamma}}\right). \tag{13}$$

Using Result 1 the Bayes factor under the hyper-g prior can be written as

$$\text{BF}_g(\boldsymbol{\gamma}) = \frac{a-2}{2R^2_{\boldsymbol{\gamma}}(1-R^2_{\boldsymbol{\gamma}})} \frac{\text{pbeta}\left(R^2_{\boldsymbol{\gamma}}, \frac{p_{\boldsymbol{\gamma}}+a-2}{2}, \frac{n-p_{\boldsymbol{\gamma}}-a+1}{2}\right)}{\text{dbeta}\left(R^2_{\boldsymbol{\gamma}}, \frac{p_{\boldsymbol{\gamma}}+a-2}{2}, \frac{n-p_{\boldsymbol{\gamma}}-a+1}{2}\right)}. \tag{14}$$

Unfortunately, Liang et al. (2008) also showed that (13) is not model selection consistent when the true model is the null model (the model only containing the intercept) and so alternative hyperpriors for g should be used.

2.4.2. The hyper-g/n prior. Given the problems with the hyper-g prior, Liang et al. (2008) proposed a modified variant of the hyper-g prior which uses

$$p_{g/n}(g) = \frac{a-2}{2n}\left(1+\frac{g}{n}\right)^{-a/2}, \tag{15}$$

which they call the hyper-g/n prior where again $a > 2$ and $g > 0$. They show that this prior leads to model selection consistency. Combining (8) with (15), and using

the transform $g = u/(1-x)$, the quantity $p(\mathbf{y}|\boldsymbol{\gamma})$ can be expressed as the integral

(16)
$$p_{g/n}(\mathbf{y}|\boldsymbol{\gamma}) = K(n)\frac{a-2}{2n} \times \int_0^1 (1-u)^{p/2+a/2-2}\left(1-u\left(1-\tfrac{1}{n}\right)\right)^{-a/2}\left(1-uR^2_{\boldsymbol{\gamma}}\right)^{-(n-1)/2} du.$$

Employing Equation 3.211 of Gradshteyn and Ryzhik (2007), i.e.,

$$\int_0^1 x^{\lambda-1}(1-x)^{\mu-1}(1-ux)^{-\delta}(1-vx)^{-\sigma}dx = \text{Beta}(\mu,\lambda)F_1(\lambda,\delta,\sigma,\lambda+\mu;u,v)$$

provided $\lambda > 0$ and $\mu > 0$ where F_1 is the Appell hypergeometric function in two variables (Weisstein, 2009), using the mappings

$$\lambda \leftrightarrow 1, \quad \mu \leftrightarrow \frac{p+a-2}{2}, \quad u \leftrightarrow 1-\frac{1}{n} \quad \delta \leftrightarrow \frac{a}{2}, \quad v \leftrightarrow R^2_{\boldsymbol{\gamma}} \quad \text{and} \quad \sigma \leftrightarrow \frac{n-1}{2}$$

and using properties of the Beta and Gamma functions leads to

(17)
$$\text{BF}_{g/n}(\boldsymbol{\gamma}) = \frac{a-2}{n(p_{\boldsymbol{\gamma}}+a-2)}F_1\left(1,\frac{a}{2},\frac{n-1}{2};\frac{p_{\boldsymbol{\gamma}}+a}{2};1-\frac{1}{n},R^2_{\boldsymbol{\gamma}}\right),$$

which is to our knowledge a new expression for the Bayes factor under the hyper g/n-prior.

Unfortunately, the expression (17) is extremely difficult to evaluate numerically since the second last argument of the above F_1 is asymptotically close to the branch point with the last argument at 1. Liang et al. (2008) again suggest Laplace approximation for this choice of prior. We now derive an alternative approximation. Using the fact that

$$F_1(1,b_1,b_2,c;1,y) = (c-1)\int_0^1 (1-t)^{c-b_1-2}(1-yt)^{-b_2}\,dt = (c-1)\frac{{}_2F_1(1,b_2;c-b_1;y)}{c-b_1-1}$$

and the approximation $F_1(1, b_1, b_2, c; 1-1/n, y) \approx F_1(1, b_1, b_2, c; 1, y)$ (which should be reasonable for large n), for $p_{\boldsymbol{\gamma}} > 2$ we obtain

$$\text{BF}_{g/n}(\boldsymbol{\gamma}) \approx \frac{a-2}{2nR^2_{\boldsymbol{\gamma}}(1-R^2_{\boldsymbol{\gamma}})} \frac{\text{pbeta}\left(R^2_{\boldsymbol{\gamma}}, \frac{p_{\boldsymbol{\gamma}}-2}{2}, \frac{n-p_{\boldsymbol{\gamma}}+1}{2}\right)}{\text{dbeta}\left(R^2_{\boldsymbol{\gamma}}, \frac{p_{\boldsymbol{\gamma}}-2}{2}, \frac{n-p_{\boldsymbol{\gamma}}+1}{2}\right)}. \tag{18}$$

For the cases where $p \in \{1, 2\}$ we will use numerical quadrature. When $p = 0$, we also have that $R^2_{\boldsymbol{\gamma}} = 0$ so $\text{BF}_{g/n}(\boldsymbol{\gamma}) = 1$. Figure 1 illustrates the differences between "exact" values of the $\text{BF}_{g/n}$ (obtained using numerical quadrature) as a function of n, $p_{\boldsymbol{\gamma}}$, and $R^2_{\boldsymbol{\gamma}}$. From this figure we see that the approximation has a good relative error except for $R^2_{\boldsymbol{\gamma}}$ values close to 1 when the approximation overestimates the true value of the log Bayes factor. We found numerical quadrature to be more reliable than using (17) evaluated using the `appell()` function in the package `Appell`.

2.4.3. Robust prior. Next we will consider the robust hyperprior for g as proposed by Bayarri et al. (2012) designed to have several nice theoretical properties outlined there. Using the default parameter choices the hyperprior for g used by Bayarri et al. (2012) corresponds to:

$$p_{rob}(g) = \tfrac{1}{2} r^{1/2} (1+g)^{-3/2}, \tag{19}$$

for $g > L$ where $L = r - 1$ and $r = (1+n)/(1+p_{\boldsymbol{\gamma}})$. Combining (8) with (19) leads to an expression for $p(\mathbf{y}|\boldsymbol{\gamma})$ of the form

$$p_{rob}(\mathbf{y}|\boldsymbol{\gamma}) = K(n) \tfrac{1}{2} r^{1/2} \int_L^\infty (1+g)^{(n-p_{\boldsymbol{\gamma}})/2-2} (1 + g\widehat{\sigma}^2_{\boldsymbol{\gamma}})^{-(n-1)/2} dg, \tag{20}$$

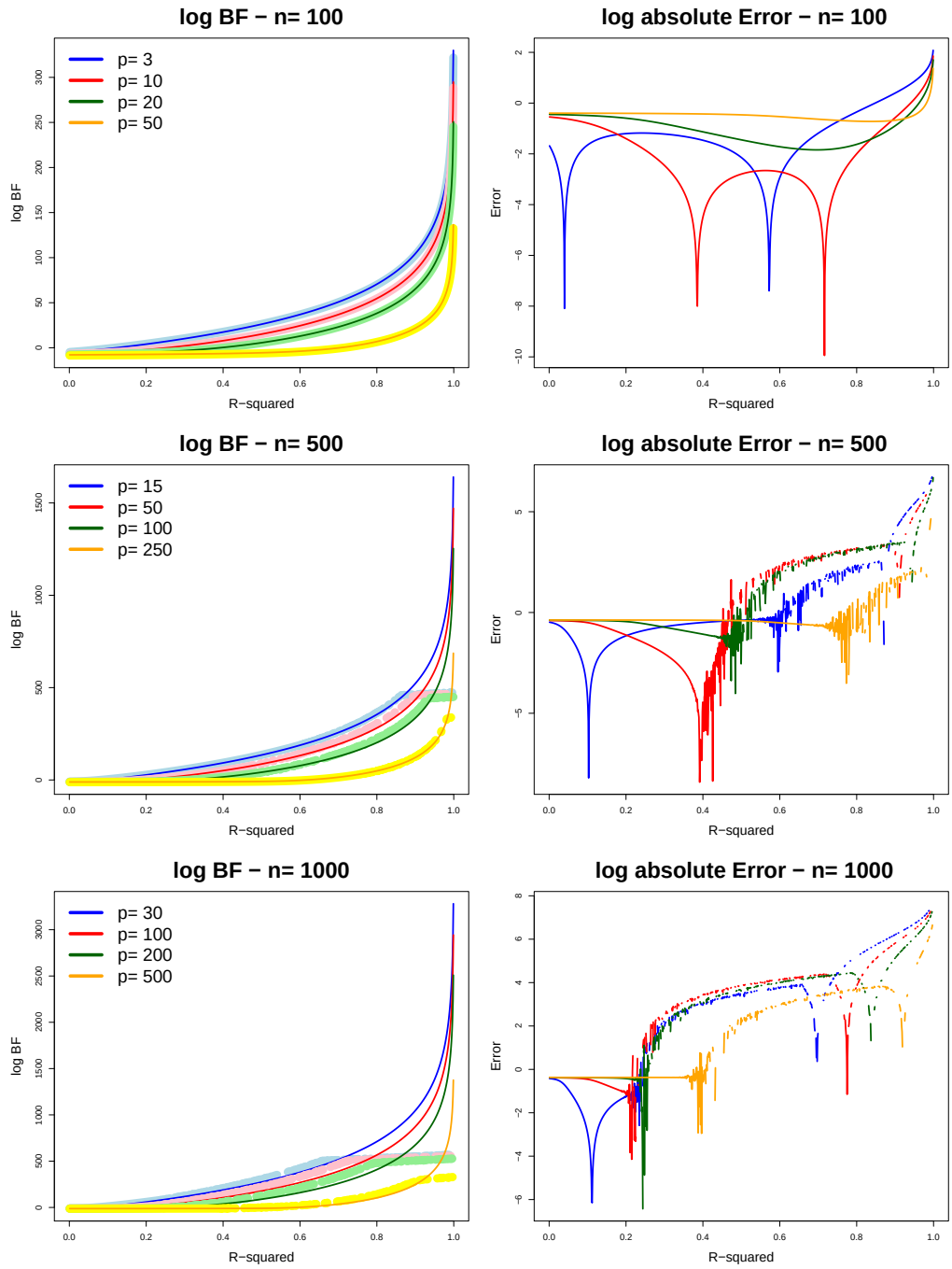

FIGURE 1. On the left side panels are plotted the values of log of $\text{BF}_{g/n}$ (light versions of the colours) and their corresponding approximation (dark version of the colours) as a function of n, p over a the range $R^2 \in (0, 0.999)$. Right side panels display the log of the absolute value of the exact values of log of $\text{BF}_{g/n}$ minus the corresponding approximations. Smaller values indicate better approximations, larger values indicate worse approximations.

where $\widehat{\sigma}^2_\gamma = 1 - R^2_\gamma$ is the MLE for σ^2 for model (4) when $\mathbf{X}$ is replaced with $\mathbf{X}_\gamma$ under the standardization described in Section 2. Using the substitution $x = r/(g-L)$ and some minor algebraic manipulation leads to

$$\begin{aligned}\text{BF}_{rob}(\boldsymbol{\gamma}) &= \tfrac{1}{2} r^{-p_\gamma/2} (\widehat{\sigma}^2_\gamma)^{-(n-1)/2} \\ &\quad \times \int_0^\infty x^{(p_\gamma-1)/2} (1+x)^{(n-p_\gamma-4)/2} \left(1 + \frac{(1+\widehat{\sigma}^2_\gamma L)x}{(1+L)\widehat{\sigma}^2_\gamma}\right)^{-(n-1)/2} dx.\end{aligned}$$

Using Equation 3.197(5) of Gradshteyn and Ryzhik (2007), i.e. (12), with the mappings

$$\lambda \leftrightarrow \frac{p_\gamma + 1}{2}, \quad \nu \leftrightarrow \frac{n - p_\gamma - 4}{2}, \quad \alpha \leftrightarrow \frac{(1+\widehat{\sigma}^2_\gamma L)}{(1+L)\widehat{\sigma}^2_\gamma}, \quad \text{and} \quad \mu \leftrightarrow -\frac{n-1}{2},$$

the conditions required by (12) are satisfied provided $\alpha \in (-1, 1)$ (which is a relatively restrictive condition). This leads to

$$\text{BF}_{rob}(\boldsymbol{\gamma}) = \left(\tfrac{n+1}{p_\gamma+1}\right)^{-p_\gamma/2} \tfrac{(\widehat{\sigma}^2_\gamma)^{-(n-1)/2}}{p_\gamma+1} {}_2F_1\left(\tfrac{n-1}{2}, \tfrac{p_\gamma+1}{2}; \tfrac{p_\gamma+3}{2}; \tfrac{(1-1/\widehat{\sigma}^2_\gamma)(p_\gamma+1)}{1+n}\right), \tag{21}$$

which is the same expression as Equation 26 of Bayarri et al. (2012) modulo notation.

The expression (21) is difficult to deal with numerically for two reasons. When $\widehat{\sigma}^2_\gamma$ becomes small the last argument of ${}_2F_1$ function can become less than -1 which falls outside the unit interval. The `BayesVarSel` package which implements this choice of prior deals with these problems using numerical quadrature.

Instead suppose we begin with the substitution $x = g - L$ which after minor algebraic manipulation leads to

$$\text{BF}_{rob}(\boldsymbol{\gamma}) = \tfrac{1}{2} r^{1/2} \left(\widehat{\sigma}^2_\gamma\right)^{-(n-1)/2} \int_0^\infty (r+x)^{(n-p_\gamma-4)/2} \left(\tfrac{1+\widehat{\sigma}^2_\gamma L}{\widehat{\sigma}^2_\gamma} + x\right)^{-(n-1)/2} dx.$$

Employing Equation 3.197(1) of Gradshteyn and Ryzhik (2007), i.e.,

$$\int_0^\infty x^{\nu-1}(\beta+x)^{-\mu}(x+\gamma)^{-\varrho}dx = \beta^{-\mu}\gamma^{\nu-\varrho}\text{Beta}(\nu,\mu-\nu+\varrho){}_2F_1(\mu,\nu;\mu+\varrho;1-\gamma/\beta),$$

(which holds provided $\nu > 0$, $\mu > \nu - \varrho$), with the mappings

$$\nu \leftrightarrow 1, \quad \beta \leftrightarrow \frac{1+\widehat{\sigma}^2_{\boldsymbol{\gamma}}L}{\widehat{\sigma}^2_{\boldsymbol{\gamma}}}, \quad \mu \leftrightarrow (n-1)/2 \quad \gamma \leftrightarrow r \quad \text{and} \quad \varrho \leftrightarrow -(n-p_{\boldsymbol{\gamma}}-4)/2,$$

The conditions of the integral result easily hold. Hence, after some algebraic manipulation and applying Result 1, and letting $\widetilde{R}^2_{\boldsymbol{\gamma}} = R^2_{\boldsymbol{\gamma}}/(1+L\widehat{\sigma}^2_{\boldsymbol{\gamma}})$ we obtain

$$\text{BF}_{rob}(\boldsymbol{\gamma}) = \left(\frac{1+n}{1+p_{\boldsymbol{\gamma}}}\right)^{(n-p_{\boldsymbol{\gamma}}-1)/2} \frac{\left(1+L\widehat{\sigma}^2_{\boldsymbol{\gamma}}\right)^{-(n-1)/2}}{2\widetilde{R}^2_{\boldsymbol{\gamma}}(1-\widetilde{R}^2_{\boldsymbol{\gamma}})} \frac{\text{pbeta}\left(\widetilde{R}^2_{\boldsymbol{\gamma}}, \frac{p_{\boldsymbol{\gamma}}+1}{2}, \frac{n-p_{\boldsymbol{\gamma}}-2}{2}\right)}{\text{dbeta}\left(\widetilde{R}^2_{\boldsymbol{\gamma}}, \frac{p_{\boldsymbol{\gamma}}+1}{2}, \frac{n-p_{\boldsymbol{\gamma}}-2}{2}\right)}. \tag{22}$$

This expression is numerically far easier to evaluate efficiently and accurately in a numerically stable manner. Due to simplifications we have $0 \leq \widehat{\sigma}^2_{\boldsymbol{\gamma}} < 1$, we also have $L > 0$ so that the last argument of the ${}_2F_1$ above is bounded in the unit interval.

2.4.4. Beta-prime prior. Next we will consider the prior

$$p_{bp}(g) = \frac{g^b(1+g)^{-(a+b+2)}}{\text{Beta}(a+1,b+1)}, \tag{23}$$

proposed by Maruyama and George (2011) where $g > 0$, $a > -1$ and $b > -1$. This is a Pearson Type VI or beta-prime distribution. More specifically,

$$g \sim \text{Beta-prime}(b+1, a+1)$$

using the usual parametrization of the beta-prime distribution (Johnson et al., 1995). Then combining (8) with (23) the quantity $p(\mathbf{y}|\boldsymbol{\gamma})$ can be expressed as the integral

$$p_{bp}(\mathbf{y}|\boldsymbol{\gamma}) = \frac{K(n)}{\text{Beta}(a+1, b+1)} \int_0^\infty g^b (1+g)^{(n-p_{\boldsymbol{\gamma}}-1)/2-(a+b+2)} (1+g(1-R^2_{\boldsymbol{\gamma}}))^{-(n-1)/2} dg.$$

If we choose $b = (n - p_{\boldsymbol{\gamma}} - 5)/2 - a$, then the exponent of the $(1 + g)$ term in the equation above is zero. Using Equation 3.194 (iii) of Gradshteyn and Ryzhik (2007), i.e.,

$$\int_0^\infty \frac{x^{\mu-1}}{(1+\beta x)^\nu} dx = \beta^{-\mu} \text{Beta}(\mu, \nu - \mu),$$

provided $\mu, \nu > 0$ and $\nu > \mu$, we obtain

$$\text{BF}_{bp}(\mathbf{y}|\boldsymbol{\gamma}) = \frac{\text{Beta}(p/2 + a + 1, b + 1)}{\text{Beta}(a + 1, b + 1)} (1 - R^2_{\boldsymbol{\gamma}})^{-(b+1)} \tag{24}$$

which is a simplification of the Bayes factor proposed by Maruyama and George (2011).

Note that (24) is proportional to a special case of the prior structure considered by Maruyama and George (2011) who refer to this as a model selection criterion (after Zellner's g prior). This choice of b also ensures that $g = O(n)$ so that $\text{tr}\{\text{Var}(\boldsymbol{\beta}|g, \sigma^2)\} = O(1)$, preventing Bartlett's paradox. Note that in comparison to the priors we have previously discussed, this choice of prior yields a marginal likelihood that can be expressed entirely with gamma functions, which are well-behaved numerically. Maruyama and George (2011) showed the prior (23) leads to model selection consistency. For derivation of the above properties and further discussion see Maruyama and George (2011).

2.4.5. BIC via Cake priors. Ormerod et al. (2017) developed the Cake prior, which allows arbitrarily diffuse priors while avoiding Bartlett's paradox. Cake priors can be thought of as a Jefferys prior in the limit as the prior becomes increasingly diffuse and enjoy nice theoretical properties including model selection consistency. Ormerod et al. (2017) departs from the prior structure (6) and instead uses

$$\alpha|\sigma^2, g \sim \mathrm{N}(0, g\sigma^2), \quad \boldsymbol{\beta}_{\boldsymbol{\gamma}}|\sigma^2, g \sim N\left(\mathbf{0}, g\sigma^2\left(\tfrac{1}{n}\mathbf{X}_{\boldsymbol{\gamma}}^T\mathbf{X}_{\boldsymbol{\gamma}}\right)^{-1}\right) \quad \text{and} \quad p(g|\boldsymbol{\gamma}_j) = \delta(g; h^{1/(1+p_{\boldsymbol{\gamma}})}) \tag{25}$$

where h is a common prior hyperparameter for all models. After marginalizing out α, $\boldsymbol{\beta}$, σ^2 and g the null based Bayes factor for model $\boldsymbol{\gamma}$ is of the form

$$\log \mathrm{BF}(\boldsymbol{\gamma}; h) = -\tfrac{n}{2}\log\left(1 - \tfrac{h^{1/(1+p_{\boldsymbol{\gamma}})}}{1+h^{1/(1+p_{\boldsymbol{\gamma}})}}R^2_{\boldsymbol{\gamma}}\right) - \tfrac{p_{\boldsymbol{\gamma}}}{2}\log\left(n + h^{-1/(1+p_{\boldsymbol{\gamma}})}\right).$$

Taking $h \to \infty$ we obtain a null based Bayes factor of

$$\mathrm{BF}(\boldsymbol{\gamma}) = \exp\left[-\tfrac{n}{2}\log\left(1 - R^2_{\boldsymbol{\gamma}}\right) - \tfrac{p_{\boldsymbol{\gamma}}}{2}\log(n)\right] = \exp\left[-\tfrac{1}{2}\mathrm{BIC}(\boldsymbol{\gamma})\right] \tag{26}$$

where $\mathrm{BIC}(\boldsymbol{\gamma}) = n\log\left(1 - R^2_{\boldsymbol{\gamma}}\right) + p_{\boldsymbol{\gamma}}\log(n)$.

2.5. Prior on the model space/size

The last ingredient to a fully Bayesian model specification is the prior on $\boldsymbol{\gamma}$, sometimes referred to as a prior on the model space, or model size. For this problems where $n < p$ a uniform prior of the form

$$p(\boldsymbol{\gamma}) = 2^{-p} \qquad \text{for all } \boldsymbol{\gamma} \in \{0,1\}^p, \tag{27}$$

often works well. This is equivalent to $p(\gamma_j) = 1/2$, $1 \leq j \leq p$ Scott and Berger (2010) state that this model prior provides no multiplicity control in a multiple testing setting, and uses an a priori model size of $p/2$ with a standard deviation of $\sqrt{p}/2$ leading to an a priori large fraction of covariates being included when p is large. The beta-binomial prior on the model space uses a prior on $\boldsymbol{\gamma}$ implied by the hierarchy

$$p(\boldsymbol{\gamma}) = \prod_{j=1}^{p} \rho^{\gamma_j}(1-\rho)^{1-\gamma_j} \tag{28}$$

where ρ is the prior probability a variable is included in the mode, and a and b are fixed prior hyperparameters. After marginalizing out ρ we have

$$p(\boldsymbol{\gamma}) = \frac{\text{Beta}(a + p_{\boldsymbol{\gamma}}, b + p - p_{\boldsymbol{\gamma}})}{\text{Beta}(a, b)}, \tag{29}$$

which is a beta-binomial distribution on the model size. Note $a = b = 1$ corresponds to a uniform prior on the prior variable inclusion probability (and can also be viewed as a "flat" prior), but is quite different to placing a uniform prior on the set of all models. The theory developed by Castillo et al. (2015) suggests $a = 1$ and $b = p^u$ for some constant $u > 1$ in the asymptotic regime where $p > n$ with p growing slightly slower than exponentially.

2.6. Implementation

Key to the feasibility of the model selection and averaging is an efficient implementation of these procedures. We employ two main strategies to achieve computational efficiency (i) efficient software implementation using highly optimized software libraries; and (ii) efficient calculation of R-squared values for all models

based on using a Gray code and appropriate matrix algebraic simplifications. For ease of use we implemented an `R` package called `blma`. The internals of `blma` are implemented in `C++` and use the `R` packages `Rcpp` and `RcppEigen` to enhance computational performance. The library `OpenMP` was used to exploit parallel computation.

There are two main special functions used in the paper – the Gaussian hypergeometric function, and the Appell hypergeometric function of two variables. During the implementation process we tried several packages which implemented the Gaussian hypergeometric function. We found that the `R` package `gsl` (Hankin, 2006) was the most accurate, numerically stable implementation amongst the packages we tried. The `R` package `Appell` implements the Appell hypergeometric function (Bové et al., 2013). We also developed our own numerical quadrature routine to evaluate the Appell hypergeometric function to check our results.

2.6.1. Gray code. The Gray code was originally developed by Frank Gray in 1947 (Press et al., 2007b, Section 22.3) to aid in detecting errors in analog to digital conversions in communications systems. It is a sequence of binary numbers whose key feature is that one and only one binary digit is different between binary numbers in the sequence. Gray codes can be constructed using a sequence of "reflect" and "prefix" steps. Let $\mathbf{\Gamma}_1 = (0,1)^T \in \{0,1\}^{2\times 1}$ be the first Gray code matrix and let $\mathbf{\Gamma}_k$ be the kth Gray code matrix. Then we can obtain the $(k+1)$th Gray code matrix given $\mathbf{\Gamma}_k$ via

$$\mathbf{\Gamma}_{k+1} = \begin{bmatrix} \mathbf{0} & \mathbf{\Gamma}_k \\ \mathbf{1} & \text{reflect}(\mathbf{\Gamma}_k) \end{bmatrix}$$

where reflect($\mathbf{\Gamma}_k$) is the matrix obtained by reversing the order of rows of $\mathbf{\Gamma}_k$, and the $\mathbf{0}$ and $\mathbf{1}$ are vectors of zeros and ones of length 2^k respectively. In `C` and `C++` these Gray codes can be efficiently constructed using bit-shift operations on binary strings in such a way that $\mathbf{\Gamma}_k$ matrices are never computed and stored explicitly.

Gray codes allow the enumeration of the entire model space in an order which only adds or removes one covariate from the previous model at a time. We can then use standard matrix inverse results to perform rank one updates and downdates in the calculation of the R^2, $(\mathbf{X}^T\mathbf{X})^{-1}$ and $\widehat{\boldsymbol{\beta}}$ values for each model in the model space.

2.6.2. Model updates and downdates. Both updates and downdates depend on the fact that the inverse of a real symmetric matrix can be written as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}^{-1}\mathbf{B}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{B}\mathbf{C}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{30}$$

$$= \begin{bmatrix} \widetilde{\mathbf{A}} & -\widetilde{\mathbf{A}}\mathbf{B}\mathbf{C}^{-1} \\ -\mathbf{C}^{-1}\mathbf{B}^T\widetilde{\mathbf{A}} & \mathbf{C}^{-1} + \mathbf{C}^{-1}\mathbf{B}^T\widetilde{\mathbf{A}}\mathbf{B}\mathbf{C}^{-1} \end{bmatrix} \tag{31}$$

where $\widetilde{\mathbf{A}} = \left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T\right)^{-1}$ provided all inverses in (30) and (31) exist. For both the update and downdate formula we assume that the quantities $\mathbf{X}^T\mathbf{y}$, $\mathbf{X}^T\mathbf{X}$ have been precalculated, and that the $(\mathbf{X}_{\boldsymbol{\gamma}_i}^T\mathbf{X}_{\boldsymbol{\gamma}_i})^{-1}$, $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_i}$ and $R^2_{\boldsymbol{\gamma}_i}$ values have been computed from the previous step. These update/downdate operations are equivalent to the "sweep" operator described in Goodnight (1979).

We want to update the model inverse matrix, coefficient vector and R^2 values for the model $\boldsymbol{\gamma}_{i+1}$ where $\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}$ is the matrix given by $\mathbf{X}_{\boldsymbol{\gamma}_i}$ with a column $\mathbf{z}$ inserted

into the appropriate position. For clarity of exposition we will assume that the column $\mathbf{z}$ is located in the last column of $\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}$, i.e., $\mathbf{X}_{\boldsymbol{\gamma}_{i+1}} = [\mathbf{X}_{\boldsymbol{\gamma}_i}, \mathbf{z}]$. This can be achieved, if necessary, by appropriate permuting columns of various matrices.

The updates for the model inverse matrix, coefficient estimates, and R^2 values can be obtained by following the steps below.

a) Calculate $\widehat{\mathbf{z}} = (\mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{X}_{\boldsymbol{\gamma}_i})^{-1} \mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{z}$, $\kappa = 1/(n - \mathbf{z}^T \widehat{\mathbf{z}})$, and $s = \mathbf{y}^T(\mathbf{z} - \widehat{\mathbf{z}})$.

b) The model inverse matrix can be updated via

$$(\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}^T \mathbf{X}_{\boldsymbol{\gamma}_{i+1}})^{-1} = \begin{bmatrix} (\mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{X}_{\boldsymbol{\gamma}_i})^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \kappa \begin{bmatrix} \widehat{\mathbf{z}} \\ -1 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{z}} \\ -1 \end{bmatrix}^T.$$

c) The coefficient estimators $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_i} = (\mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{X}_{\boldsymbol{\gamma}_i})^{-1} \mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{y}$, and

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_{i+1}} = (\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}^T \mathbf{X}_{\boldsymbol{\gamma}_{i+1}})^{-1} \mathbf{X}_{\boldsymbol{\gamma}_{i+1}}^T \mathbf{y}.$$

Then using the block inverse formula we have the relation

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_{i+1}} = \begin{bmatrix} \widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_i} \\ 0 \end{bmatrix} - \kappa s \begin{bmatrix} \widehat{\mathbf{z}} \\ -1 \end{bmatrix}.$$

d) The R^2 value let $R^2_{\boldsymbol{\gamma}_i} = \frac{1}{n} \mathbf{y}^T \mathbf{X}_{\boldsymbol{\gamma}_i} (\mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{X}_{\boldsymbol{\gamma}_i})^{-1} \mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{y}$. Then using the block inverse formula we have

$$R^2_{\boldsymbol{\gamma}_{i+1}} = R^2_{\boldsymbol{\gamma}_i} + \frac{\kappa s^2}{n}.$$

Presuming relevant summary quantities have been precomputed the above updates costs $O(p^2_{\boldsymbol{\gamma}_i} + n)$ time.

Suppose want to downdate the model summary quantities for the model $\boldsymbol{\gamma}_{i+1}$ where $\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}$ is the matrix given by $\mathbf{X}_{\boldsymbol{\gamma}_i}$ with a column $\mathbf{z}$ removed from the appropriate position. Similarly as for updates for clarity of exposition we will assume that $\mathbf{z}$ will be removed from the last column of $\mathbf{X}_{\boldsymbol{\gamma}_i}$, i.e., we assume that $\mathbf{X}_{\boldsymbol{\gamma}_i} = [\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}, \mathbf{z}]$. Again, this can be achieved by permuting the columns of various matrices. Then the downdates for model summary values are given by the following steps.

a) Suppose we partition the matrix $(\mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{X}_{\boldsymbol{\gamma}_i})^{-1}$ so that

$$(\mathbf{X}_{\boldsymbol{\gamma}_i}^T \mathbf{X}_{\boldsymbol{\gamma}_i})^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix}.$$

Calculate the model inverse matrix by $(\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}^T \mathbf{X}_{\boldsymbol{\gamma}_{i+1}})^{-1} = \mathbf{A} - c^{-1}\mathbf{b}\mathbf{b}^T$.

b) Calculate $\widehat{\mathbf{z}} = (\mathbf{X}_{\boldsymbol{\gamma}_{i+1}}^T \mathbf{X}_{\boldsymbol{\gamma}_{i+1}})^{-1} \mathbf{X}_{\boldsymbol{\gamma}_{i+1}}^T \mathbf{z}$, $\kappa = 1/(n - \mathbf{z}^T \widehat{\mathbf{z}})$, and $s = \mathbf{y}^T(\mathbf{z} - \widehat{\mathbf{z}})$.

c) The coefficient estimates downdate can be obtained via

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_{i+1}} = \left[\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_i}\right]_{-|\boldsymbol{\gamma}_i|} + \kappa s \widehat{\mathbf{z}},$$

where $[\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_i}]_{-|\boldsymbol{\gamma}_i|}$ removes the last column from $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}_i}$.

d) The R^2 downdate can be obtained via

$$R^2_{\boldsymbol{\gamma}_{i+1}} = R^2_{\boldsymbol{\gamma}_i} - \frac{\kappa s^2}{n}.$$

Again, presuming relevant summary quantities have been precomputed the updates for all of the above quantities costs $O(p^2_{\boldsymbol{\gamma}_i} + n)$ time.

2.7. Numerical results

We will now compare the different Bayes factors under different hyperpriors on g that we have explored. Firstly we will look at these Bayes factors by comparing them directly. We will then compare the results based on exact Bayesian linear model averaging on some available datasets.

2.7.1. Numerical comparison of g hyperpriors. Note that each of the Bayes factors is a function of three quantities R^2, $p_{\boldsymbol{\gamma}}$ and n. Figure 2 illustrates various log Bayes factors over a grid of $p_{\boldsymbol{\gamma}}$ values from 1 to 20 and $R^2 \in \{0.1, 0.5, 0.9\}$ and $n \in \{100, 500, 1000\}$. In the context of Bayesian hypothesis testing values above the y-axis value 0 indicate that the alternative model is preferred, while lines below 0 indicate the null model is preferred. Note that Cake priors (BIC) have the strongest penalty for larger $p_{\boldsymbol{\gamma}}$, followed by the beta-prime prior (ZE), the robust prior, hyper-g/n prior and lastly the hyper-g prior. Increasing n and/or R^2 leads to all of the different Bayes factors becoming increasingly close to one another. We also see that the `appell()` function becomes unstable as n and/or R^2 becomes large. For the Bayes factor corresponding to the hyper-g/n prior our approximation tracks very closely to the methods using the `appell()` function and our numerical quadrature approach.

2.7.2. Settings for `R` packages. We will now compare three different popular `R` implementations of Bayesian model averaging on several small datasets. We compare the `R` packages `BAS` (Clyde, 2017), `BayesVarSelect` (García-Donato and Forte, 2016), and `BMS` (Zeugner and Feldkircher, 2015). For each method we

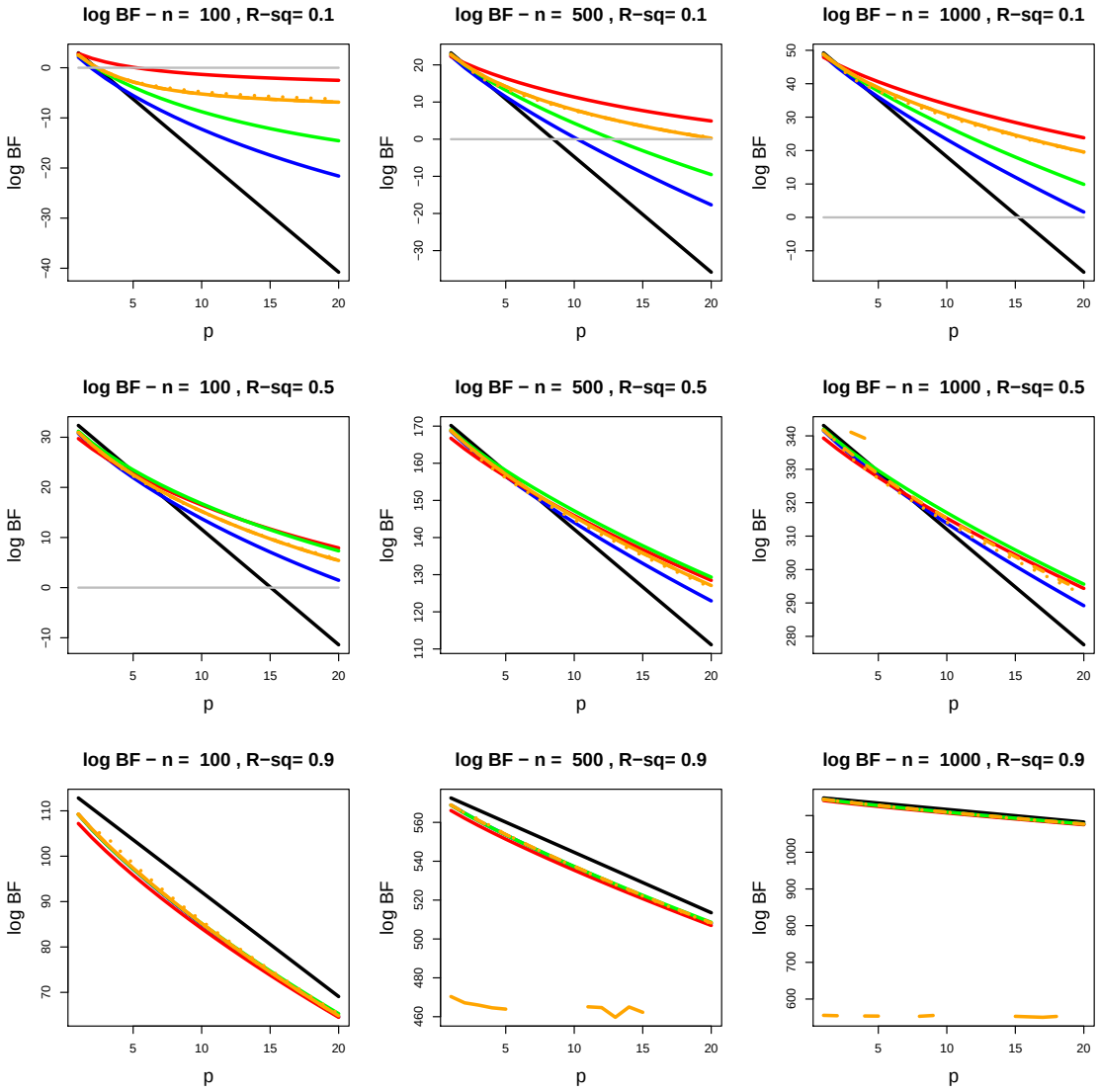

FIGURE 2. Cake prior or BIC (black), beta-prime prior (blue), hyper-g prior (red), robust prior (green), hyper-g/n (`appell` - solid orange), hyper-g/n (quadrature - dashed orange), and hyper-g/n (approximation - dotted orange). The grey line corresponds to the Bayes factor equal to 1. Above the grey line the alternative model is preferred, below the grey line the null model is preferred.

assumed a uniform prior on the model space, i.e. $p(\boldsymbol{\gamma}) \propto 2^{-p}$. We used the setting implied by the following commands for each of these methods.

- `BAS`: We used the command

  ```
  bas.lm(y~X,prior=prior.val,modelprior=uniform())
  ```

  where `prior.val` takes the value `"hyper-g"`, `"hyper-g-laplace"` or `"hyper-g-n"`. These correspond to a direct implementation of (13), a Laplace approximation of (11), and the Laplace approximation of (16) respectively. The value $a = 3$ is implicitly used.

- `BayesVarSelect`: We used the command

  ```
  Bvs(formula="y~.",data=data.frame(y=y,X=X),
      prior.betas=prior.val,prior.models="Constant",
      time.test=FALSE,n.keep=50000)
  ```

  where `prior.val` takes the value `"Liangetal"` or `"Robust"`. These correspond to a direct implementation of (13) with $a = 3$, and a hybrid approach which uses (21) directly and numerical quadrature based on (20) if this fails respectively. Again, the value $a = 3$ is implicitly used.

- `BMS`: We used the command

  ```
  bms(cbind(y,X),nmodel=50000,mcmc="enumerate",
      g="hyper=3",mprior="uniform")
  ```

  which uses a direct implementation of (13) for the hyper-g prior with $a = 3$.

The syntax for `blma` is relatively straightforward:

```
blma(vy, mX, prior, mprior, cores = 1L)
```

where the arguments of `blma` are explained below.

- `vy` – a vector of length n of responses (this vector does not need to be standardized).
- `mX` – a design matrix with n rows and p columns (the columns of `mX` do not need to be standardized).
- `prior` – the choice of mixture g-prior used to perform Bayesian model averaging. The choices available include:
  - `"BIC"` – the Bayesian information criterion obtained by using the Cake prior of Ormerod et al. (2017).
  - `"ZE"` – special case of the prior structure in Maruyama and George (2011).
  - `"liang_g1"` – the mixture g-prior of Liang et al. (2008) with prior hyperparameter $a = 3$ evaluated directly using (13) where the Gaussian hypergeometric function is evaluated using the `gsl` library. Note: this option can lead to numerical problems and is only meant to be used for comparative purposes.
  - `"liang_g2"` – the mixture g-prior of Liang et al. (2008) with prior hyperparameter $a = 3$ evaluated directly using (14).
  - `"liang_g_n_appell"` – the mixture g/n-prior of Liang et al. (2008) with prior hyperparameter $a = 3$ evaluated using the `appell R` package.
  - `"liang_g_approx"` – the mixture g/n-prior of Liang et al. (2008) with prior hyperparameter $a = 3$ using the approximation (18) for $p_{\boldsymbol{\gamma}} > 2$ and numerical quadrature (see below) ofr $p_{\boldsymbol{\gamma}} \in \{1, 2\}$.

  - `"liang_g_n_quad"` – the mixture g/n-prior of Liang et al. (2008) with prior hyperparameter $a = 3$ evaluated using a composite trapezoid rule.
  - `"robust_bayarri1"` – the robust prior of Bayarri et al. (2012) using default prior hyperparameter choices evaluated directly using (21) with the `gsl` library.
  - `"robust_bayarri2"` – the robust prior of Bayarri et al. (2012) using default prior hyperparameter choices evaluated directly using (22).
- `mprior` – the prior to be imposed on the model space. The choices available include:
  - `"uniform"` – corresponds to the prior $p(\boldsymbol{\gamma}) = 2^{-p}$ where p is the number of columns of $\mathbf{X}$, .i.e., a uniform prior on the model space.
  - `"beta-binomial"` – corresponds to a prior of the form

$$p(\boldsymbol{\gamma}) = \prod_{j=1}^{p} \rho^{\gamma_j}(1-\rho)^{1-\gamma_j} \qquad \text{and} \qquad \rho \sim \text{Beta}(a, b),$$

    where ρ is the prior probability a variable is included in the mode, and a and b are fixed prior hyperparameters. After marginalizing out ρ we have

$$p(\boldsymbol{\gamma}) = \frac{\text{Beta}(a + |\boldsymbol{\gamma}|, b + p - |\boldsymbol{\gamma}|)}{\text{Beta}(a, b)},$$

    which is a beta-binomial distribution. Note $a = b = 1$ corresponds to a uniform prior on the prior variable inclusion probability. The values of a and b should be set to be the first and second elements of the `modelpriorvec` argument respectively (see below).

  - `"bernoulli"` – corresponds to a prior of the form

$$p(\boldsymbol{\gamma}) = \prod_{j=1}^{p} \rho_j^{\gamma_j} (1-\rho_j)^{1-\gamma_j}$$

    where the $\rho_j \in (0,1)$. The ρ_j values are specified by `modelpriorvec` (see below). Using $\rho_j = 1/2$, $1 \leq j \leq p$ corresponds to `mprior=="uniform"`.

- `modelpriorvec` – A vector of additional parameters. If `mprior=="uniform"` this argument is ignored. If `mprior=="beta-binomial"` this should be a postive vector of length 2 corresponding to the shape parameters of a Beta distribution (the values a and b above). If `mprior=="bernoulli"` this should be a vector of length p with values on the interval $(0,1)$.
- `cores` – the number of computer cores to use.

The object returned is a list containing:

- `vR2` – the vector R-square values for each model;
- `vp_gamma` – the vector of number of covariates for each model;
- `vlogp` – the vector of logs of the marginal likelihoods of each model; and
- `vinclusion_prob` – the vector of posterior inclusion probabilities for each of the covariates.

Note that we do not return the fitted values of $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ which should only be calculated for a subset of models. We also do not return $\boldsymbol{\Gamma}$, the Gray code matrix which we provide a separate function to calculate. We made the decisions not to return these quantities to reduce the memory overhead.

A short example fitting the `USCrime` data described in Section 2.7.3 below.

```
library(blma); library(MASS)
dat <- UScrime
dat[,-c(2,ncol(UScrime))] <- log(dat[,-c(2,ncol(UScrime))])
vy <- dat$y
mX <- data.matrix(cbind(dat[1:15]))
colnames(mX) <- c("log(AGE)","S","log(ED)","log(Ex0)",
    "log(Ex1)","log(LF)","log(M)","log(N)","log(NW)",
    "log(U1)","log(U2)","log(W)","log(X)","log(prison)",
    "log(time)")
blma_result <- blma(vy, mX, prior="ZE")
```

Results for the above example are summarised as part of the result within Section 2.7.3.

2.7.3. Bayesian linear model averaging on data. We considered several small datasets to illustrate our methodology. These datasets can be found in the `R` packages `MASS` (Venables and Ripley, 2002) and `Ecdat` (Croissant, 2016). Table 1 summarizing the sizes, sources, and response variable used for each dataset used. We chose `USCrime` data because it is used in most papers in the area and is small enough so that näive implementations using special functions will not lead to numerical issues. The `Kakadu` dataset is chosen to be large enough to begin to strain the resources of a typical 2018 laptop so that relative differences in speeds between different packages becomes apparent. Finally, the `Kakadu` dataset is chosen to lead numerical instability in the direct evaluation of Bayes factors for some of the priors on g considered in this paper.

Dataset	n	p	Response	`R` package
UScrime	47	15	y	`MASS`
VietNamI	27765	11	lnhhexp	`Ecdat`
Kakadu	1827	22	income	`Ecdat`

TABLE 1. A summary of the datasets used in the paper and their respective `R` packages.

For each of the datasets some minimal preprocessing was used. We first used the `R` command `na.omit()` to remove samples containing missing predictors. For `USCrime` all variables except the predictor `S` were log-transformed. For all datasets the `R` command `model.matrix()` was used to construct the design matrix using all variables except for the response as predictors.

Tables 2, 3, and 4 summarise the times and variable inclusion probabilities, i.e., $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y})$, for all of the mixture g-prior structures we have considered here under a uniform prior on the model space. All times are based on running `R` code on a dedicated server with 48 cores, each running at 2.70GHz, with a total of 512GB of RAM. The `BVS` package in the table refers to the `BayesVarSelect` `R` package where we have used a this acronym to save space in the tables.

For Table 2 we see that all of the "exact" methods agree with one another to the first 2 decimal places. We note that the Laplace approximation is quite accurate and appears superior to the method "(18)" for the mixture g/n-prior. However, for both of these approximation methods the discrepancies to their exact counterparts is roughly the same size, or perhaps even less, than the differences between each of the choices of mixture g-priors. In terms of speed, `BAS` and `BMLA` are the fastest packages and roughly comparable in speed. Both `BMS` and `BayesVarSelect` are not as fast. For the mixture g-prior we suspect that the package `BAS` relies

on Laplace's method for models where direct evaluation of (13) becomes numerically problematic, which would explain differences between the `BAS` and `blma` packages for the `Kakadu` dataset.

Note that in Table 4 that many of the variables have posterior probabilities either close to 0 or close to 1. This is anticipated since $n = 27765$ is relatively large (and p is small) leading to a single model dominates the model averaging procedure.

2.8. Conclusion

We have reviewed the prior structures that lead to closed form expressions for Bayes factors for linear models. We have described ways that each of these priors with the exception of the hyper-g/n prior can be evaluated in a numerically stable manner and have implemented a package `blma` for performing full exact Bayesian model averaging using this methodology. Our package is competitive with `BAS` and `BMS` in terms of computational speed, is numerically more stable and accurate, and offers some different priors structures not offered in `BAS`. Our package is much faster than `BayesVarSelect` and is also numerically more stable and accurate.

Package	blma	blma	BAS	BAS	BVS	BMS	blma	blma	BAS	blma	blma	blma	BVS	blma	blma
Prior	BIC	ZE	g	g	g	g	g	g	g/n	g/n	g/n	g/n	Robust	Robust	Robust
Method	(26)	(24)	(13)	Laplace	(13)	(13)	(13)	(14)	Laplace	`appell`	quad.	(18)	(21)	(21)	(22)
1	70.87	65.51	65.93	65.99	64.74	65.93	65.93	65.93	65.14	65.10	65.10	65.72	64.74	NaN	64.74
2	19.06	22.88	25.52	25.54	24.51	25.52	25.52	25.52	22.93	22.91	22.91	22.47	24.51	NaN	24.51
3	92.07	86.91	86.23	86.28	85.59	86.23	86.23	86.23	86.54	86.51	86.51	87.24	85.59	NaN	85.59
4	72.53	69.65	69.20	69.22	69.02	69.20	69.20	69.20	69.52	69.51	69.51	69.89	69.02	NaN	69.02
5	37.01	42.36	44.61	44.61	44.08	44.61	44.61	44.61	42.53	42.52	42.52	41.88	44.08	NaN	44.08
6	15.82	20.18	23.06	23.08	22.04	23.06	23.06	23.06	20.27	20.26	20.26	19.73	22.04	NaN	22.04
7	27.06	32.43	34.55	34.55	34.08	34.55	34.55	34.55	32.59	32.59	32.59	32.00	34.08	NaN	34.08
8	60.64	56.91	57.34	57.39	56.47	57.34	57.34	57.34	56.66	56.63	56.63	57.07	56.47	NaN	56.47
9	36.92	35.81	37.66	37.71	36.35	37.66	37.66	37.66	35.64	35.61	35.61	35.71	36.35	NaN	36.35
10	21.92	24.35	27.06	27.10	25.78	27.06	27.06	27.06	24.31	24.29	24.29	24.00	25.78	NaN	25.78
11	55.84	50.19	51.25	51.32	49.66	51.25	51.25	51.25	49.79	49.75	49.75	50.38	49.66	NaN	49.66
12	17.39	21.57	24.46	24.48	23.40	24.46	24.46	24.46	21.65	21.63	21.63	21.12	23.40	NaN	23.40
13	99.92	99.69	99.50	99.51	99.54	99.50	99.50	99.50	99.66	99.66	99.66	99.72	99.54	NaN	99.54
14	90.27	84.92	83.87	83.92	83.45	83.87	83.87	83.87	84.57	84.55	84.55	85.32	83.45	NaN	83.45
15	17.63	22.55	25.49	25.51	24.52	25.49	25.49	25.49	22.67	22.65	22.65	22.05	24.52	NaN	24.52
Time (s)	0.11	0.10	1.07	0.51	1358.61	44.73	0.12	0.10	0.30	12.59	40.36	0.25	618.59	31.81	0.11

TABLE 2. Variable inclusion probabilities (as a percentage) and computational times (in seconds) for the `UScrime` dataset. Each line in the table corresponds to a different variable in the `UScrime` dataset. The first to third line indicates the package, mixture g-prior and evaluation method used respectively. Bracketed terms refer to equations in the paper. NaN entries indicate numerical issues for the prior/implementation pair. The acronym BVS refers to the `BayesVarSelect` package.

Package	blma	blma	BAS	BAS	BVS	BMS	blma	blma	BAS	blma	blma	blma	BVS	blma	blma
Prior	BIC	ZE	g	g	g	g	g	g	g/n	g/n	g/n	g/n	Robust	Robust	Robust
Method	(26)	(24)	(13)	Laplace	(13)	(13)	(13)	(14)	Laplace	`appell`	quad.	approx.	(21)	(21)	(22)
1	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
2	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
3	1.21	3.16	8.65	8.65	NaN	NaN	NaN	8.64	7.17	NaN	7.16	7.65	NaN	4.77	4.77
4	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
5	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
6	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
7	0.62	1.72	5.33	5.33	NaN	NaN	NaN	5.32	4.30	NaN	4.29	4.63	NaN	2.70	2.70
8	96.07	98.32	99.35	99.35	NaN	NaN	NaN	99.35	99.21	NaN	99.20	99.26	NaN	98.86	98.86
9	3.28	8.16	20.69	20.69	NaN	NaN	NaN	20.66	17.46	NaN	17.42	18.52	NaN	12.02	12.02
10	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
11	100.00	100.00	100.00	100.00	NaN	NaN	NaN	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
Time (s)	0.03	0.02	0.88	0.33		5.89	0.02	0.02	0.08	84.73	2.69	0.10	*	2.18	0.01

TABLE 3. Variable inclusion probabilities (as a percentage) and computational times (in seconds) for the `VietNamI` dataset. Each line in the table corresponds to a different variable in the `VietNamI` dataset. The first to third line indicates the package, mixture g-prior and evaluation method used respectively. Bracketed terms refer to equations in the paper. NaN entries indicate numerical issues for the prior/implementation pair. The acronym BVS refers to the `BayesVarSelect` package. * denotes that the method did not complete.

Package	blma	blma	BAS	BAS	BVS	BMS	blma	blma	BAS	blma	blma	blma	BVS	blma	blma
Prior	BIC	ZE	g	g	g	g	g	g	g/n	g/n	g/n	g/n	Robust	Robust	Robust
Method	(26)	(24)	(13)	Laplace	(13)	(13)	(13)	(14)	Laplace	`appell`	quad.	approx.	(21)	(21)	(22)
1	11.96	20.36	34.62	34.64	NaN	34.69	34.69	34.69	31.98	NaN	32.04	32.96	NaN	26.46	26.46
2	43.60	47.24	50.36	50.34	NaN	50.34	50.34	50.34	49.79	NaN	49.78	49.98	NaN	48.73	48.73
3	3.00	7.49	16.97	16.99	NaN	17.10	17.10	17.10	15.02	NaN	15.13	15.80	NaN	11.20	11.20
4	37.14	42.02	46.85	46.88	NaN	46.87	46.87	46.87	46.02	NaN	46.01	46.31	NaN	44.28	44.28
5	81.87	86.11	90.49	90.50	NaN	90.41	90.41	90.41	89.92	NaN	89.85	90.07	NaN	88.59	88.59
6	16.83	26.67	41.70	41.69	NaN	41.83	41.83	41.83	38.98	NaN	39.10	40.05	NaN	33.27	33.27
7	3.22	8.89	21.41	21.43	NaN	21.53	21.53	21.53	18.86	NaN	18.95	19.83	NaN	13.82	13.82
8	4.30	11.09	23.57	23.59	NaN	23.66	23.66	23.66	21.20	NaN	21.26	22.09	NaN	16.34	16.34
9	2.62	7.19	16.97	16.98	NaN	17.09	17.09	17.09	14.97	NaN	15.07	15.76	NaN	11.04	11.04
10	52.53	77.78	90.97	90.99	NaN	90.81	90.81	90.81	89.59	NaN	89.44	89.98	NaN	85.92	85.92
11	92.51	93.73	94.75	94.79	NaN	94.58	94.58	94.58	94.68	NaN	94.49	94.53	NaN	94.35	94.35
12	99.82	99.94	99.97	99.97	NaN	99.97	99.97	99.97	99.97	NaN	99.97	99.97	NaN	99.96	99.96
13	2.45	6.60	15.70	15.72	NaN	15.84	15.84	15.84	13.81	NaN	13.92	14.57	NaN	10.13	10.13
14	8.10	19.91	38.61	38.63	NaN	38.66	38.66	38.66	35.36	NaN	35.39	36.55	NaN	28.37	28.37
15	8.17	18.51	35.17	35.19	NaN	35.24	35.24	35.24	32.19	NaN	32.24	33.29	NaN	25.87	25.87
16	62.99	75.30	83.41	83.42	NaN	83.30	83.30	83.30	82.39	NaN	82.29	82.68	NaN	79.98	79.98
17	3.27	8.53	19.41	19.43	NaN	19.54	19.54	19.54	17.20	NaN	17.31	18.07	NaN	12.85	12.85
18	54.75	74.93	86.65	86.65	NaN	86.55	86.55	86.55	85.31	NaN	85.22	85.74	NaN	81.95	81.95
19	100.00	100.00	100.00	100.00	NaN	100.00	100.00	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
20	26.63	44.11	62.58	62.60	NaN	62.56	62.56	62.56	59.88	NaN	59.83	60.83	NaN	53.58	53.58
21	100.00	100.00	100.00	100.00	NaN	100.00	100.00	100.00	100.00	NaN	100.00	100.00	NaN	100.00	100.00
22	4.95	13.22	29.03	29.05	NaN	29.12	29.12	29.12	26.04	NaN	26.09	27.14	NaN	19.87	19.87
Time(s)	15.43	16.18	14.85	9.53		1735.66	34.925	17.55	10.82	25008.93	5425.11	18.06	4606.92	4275.55	21.03

TABLE 4. Variable inclusion probabilities (as a percentage) and computational times (in seconds) for the `Kakadu` dataset. Each line in the table corresponds to a different variable in the `Kakadu` dataset. The first to third line indicates the package, mixture g-prior and evaluation method used respectively. Bracketed terms refer to equations in the paper. NaN entries indicate numerical issues for the prior/implementation pair. The acronym BVS refers to the `BayesVarSelect` package. Note that the `BayesVarSelect` method ran out of RAM for this example.

CHAPTER 3

Particle Variational Approximation

Abstract

Bayesian model averaging has several desirable properties, but it is computationally expensive unless the number of models to be averaged over is small. Typically the number of models to be averaged grows exponentially in the number of covariates and some form of approximation is required. In this paper we explore a novel particle based collapsed variational approximation for Bayesian model averaging. The resulting objective function can be optimized in a highly parallel manner. We explore several different prior specifications which lead to Bayes factors with closed forms. We show empirically that our approach is fast and effective for moderately large problems on several simulated and publicly available data sets, particularly when parallel computing resources are available. An R package is available implementing our approach.

3.1. Introduction

Bayesian model selection is a powerful set of techniques for model selection. These techniques are especially useful in problems of high-dimension, such as bioinformatics problems where the model space is complex and the optimal model is difficult for statisticians to manually specify. However, Bayesian model selection is computationally expensive, and prone to getting stuck in local minima if the posterior distribution is multimodal. This issue is particularly acute if the spike-and-slab prior, popular for Bayesian model selection, is used. We seek to address both problems by proposing a population non-parametric Variational Bayes approximation algorithm – a population-based optimisation strategy. Maintaining a population of models allows the posterior distribution to be explored more thoroughly, finding multiple maxima. The variational approximation's lower bound includes an entropy term which ensures diversity in the population by penalising similarity, having the particles in the population repel each other. This ensures the high probability regions of the posterior distribution is thoroughly explored, which better reflects model selection uncertainty. In this chapter, we focus on the important case of model selection for normal linear models with priors as described in Section 2.3.

Mitchell and Beauchamp (1988) initially proposed the spike-and-slab prior distribution on regression coefficients not currently included in the model – which places a mixture of a point mass 'spike' at 0 and a diffuse uniform distribution 'slab' elsewhere. The approach was further developed by Madigan and Raftery (1994) to incorporate an alternative Bayesian approach that takes full account of the true model uncertainty by averaging over a small subset of models, and an

efficient search algorithm for finding these models. George and McCulloch (1997) investigated computational methods for posterior evaluation and exploration in this setting, and using Gray Code sequencing and Markov Chain Monte Carlo to explore the model space in moderate and large-sized problems respectively. More recently, Ishwaran and Rao (2005) developed a rescaled spike-and-slab model which improves effective variable selection in terms of risk misclassification by using selective shrinkage.

Existing approaches to the problem of model selection focus upon finding a single best model as quickly as possible, using the least computational effort (You and Ormerod, 2014; Ročková and George, 2014). Exploring the model space using only one model at a time will provide a misleading view of the uncertainty in the posterior, as it is typically highly multimodal.

Many computational schemes for Bayesian model selection exist, using Monte Carlo Markov Chains techniques to approximate the posterior distribution of γ. However, these schemes are both computationally intensive and can become trapped in local maxima of the posterior distribution if the distribution is high-dimensional and multi-modal, as is the case with popular choices of prior for Bayesian model selection problems, such as spike-and-slab priors. The difficulty of becoming trapped in local maxima can be partially mitigated by using population-based Monte Carlo Markov Chains (MCMC) schemes such as Jasra et al. (2007), Bottolo and Richardson (2010), Hans et al (2007), Liang and Wong (2000). However, this increases the computational cost of sampling from the posterior distributions still further, especially in high-dimensional problems.

Ročková (2017) introduced the notion of Particle Expectation Maximisation (EM) . Rather than searching for a single optimal model, Particle EM instead maintains a population of models (particles). This allows the algorithm to explore more of the posterior model space, gaining a better estimate of the uncertainty in the model selection process than an algorithm involving only a single model. It also allows the particles to "interact", searching for the essential posterior modes together. In Particle EM, this is done by incorporating an "entropy term" in the variational lower bound, which ensures diversity amongst the models in the population, preventing all particles from simply seeking the global posterior modes. The algorithm is deterministic.

We build upon this work by proposing a fixed-form parametric Variational Bayes approximation of $\boldsymbol{\gamma}$. We adopt a prior structure incorporating the Cake prior introduced in Ormerod et al. (2017) for variable selection, which avoids the Lindley and Bartlett's paradoxes. The difficulties in implementing practical Bayesian model selection schemes have been noted in Chipman et al. (2014). As our marginal likelihood expression is a function only of n, $p_{\boldsymbol{\gamma}}$ and $R^2_{\boldsymbol{\gamma}}$, our model selection algorithm can be executed efficiently using rank-one updates and downdates to compute $R^2_{\boldsymbol{\gamma}^*}$ for each of the models $\boldsymbol{\gamma}^*$ that we consider. To ensure uniqueness of the K models in the population, before a new candidate model with a covariate added or removed is considered, the population of existing models is checked to see if it already exists in the population. If so, the addition or removal of the covariate is skipped and the next candidate model considered.

Our variational approximation of the posterior model probability is a weighted sum of the indicators of the covariates of the model, where the weights are determined by the relative contribution of each covariate to the model fit in all particles in the population, balanced against the diversity of the particles.

$$q(\boldsymbol{\gamma}) = \sum_{k=1}^{K} w_k \mathrm{I}(\boldsymbol{\gamma}_k)$$

Our main contributions are:

a) Our algorithm searches over the binary strings $\boldsymbol{\gamma}$ directly, as the estimates of $\boldsymbol{\beta}$ are available in closed form once $\boldsymbol{\gamma}$ is known.

b) We make use of a population–based optimisation scheme to search the model space. We take advantage of the population of solutions by incorporating a penalty for lack of entropy, which ensures diversity in the population of solutions.

c) Our model can incorporate different hyperpriors on g and $\boldsymbol{\gamma}$. Using a hyperprior on g avoids Lindley's paradox and Bartlett's paradox, the model selection paradoxes which arise when a fixed choice of g is made.

3.2. Bayesian linear model averaging

The Bayes factors introduced in the previous chapter play a key role in Bayesian linear model averaging. Via Bayes theorem the posterior probability of a model is given by

$$p(\boldsymbol{\gamma}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}'} p(\mathbf{y}|\boldsymbol{\gamma}')p(\boldsymbol{\gamma}')} = \frac{p(\boldsymbol{\gamma})\text{BF}(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}'} p(\boldsymbol{\gamma}')\text{BF}(\boldsymbol{\gamma}')}$$

where $\sum_{\boldsymbol{\gamma}}$ denotes a combinatorial sum over all 2^p possible values of $\boldsymbol{\gamma}$, and $p(\boldsymbol{\gamma})$ is the chosen prior on $\boldsymbol{\gamma}$. Numerical overflow can be avoided by dividing through the numerator and denominator of $p(\boldsymbol{\gamma}|\mathbf{y})$ by the largest product $p(\boldsymbol{\gamma})\text{BF}(\boldsymbol{\gamma})$ and performing calculations on the log scale. The posterior expectation of $\boldsymbol{\gamma}$ is given by $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y}) = \sum_{\boldsymbol{\gamma}} \boldsymbol{\gamma} \cdot p(\boldsymbol{\gamma}|\mathbf{y})$. The median posterior model is obtained by rounding $\mathbb{E}(\boldsymbol{\gamma}|\mathbf{y})$ to the nearest integer and has desirable optimality properties (Barbieri and Berger, 2004).

3.3. Particle based variational approximation

We will now present a population based variational collapsed Bayes approximation (PVA) approach to model selection which is more appropriate to use when p is larger than, say, around 30. This approach is closely related to the PEM method of Ročková (2017), where the main difference being that here we work in a fully Bayesian framework and we consider a wider range of prior structure specifications.

The marginal likelihood for $\mathbf{y}$ is given by

$$\begin{aligned} p(\mathbf{y}) &= \sum_{\boldsymbol{\gamma}} \left[\int p(\mathbf{y}, \alpha, \boldsymbol{\beta}, \sigma^2, g|\boldsymbol{\gamma}) p(\alpha, \boldsymbol{\beta}, \sigma^2, g|\boldsymbol{\gamma}) \, d\alpha \, d\boldsymbol{\beta} \, d\sigma^2 \, dg \right] p(\boldsymbol{\gamma}) \\ &= \sum_{\boldsymbol{\gamma}} p(\mathbf{y}, \boldsymbol{\gamma}), \end{aligned}$$

which is generic to the prior distribution specification for $p(\alpha, \boldsymbol{\beta}, \sigma^2, g|\boldsymbol{\gamma})$ and $p(\boldsymbol{\gamma})$. Our approach is to collapse over α, $\boldsymbol{\beta}$, σ^2, and g, and then use a variational approximation to the posterior probability of the model $\boldsymbol{\gamma}$. This can be done using any combination of the prior specifications described in Section 3.2. This is conceptually equivalent to the collapsed variational approximation technique developed by Teh et al. (2006) who used the concept of collapsing over a subset of variables in the context of Latent Dirichlet Allocation models.

We specify the q-density for $\boldsymbol{\gamma}$ parametrically by

$$q(\boldsymbol{\gamma}) = \sum_{k=1}^{K} w_k I(\boldsymbol{\gamma} = \boldsymbol{\gamma}_k) \tag{32}$$

where $0 < w_k \leq 1$, $\sum_{k=1}^{K} w_k = 1$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K]$ is a population of models (with individual $\boldsymbol{\gamma}_k$ referred to as particles), and $I(\,\cdot\,)$ is the indicator function. Here $\mathbf{w}$ and $\boldsymbol{\Gamma}$ are variational parameters of the probability mass function $q(\boldsymbol{\gamma})$.

Using $q(\boldsymbol{\gamma})$ we derive the following variational lower bound on $\log p(\mathbf{y})$ via

$$\begin{aligned} \log p(\mathbf{y}) &= \log \left[\sum_{\boldsymbol{\gamma}} q(\boldsymbol{\gamma}) \left\{ \frac{p(\mathbf{y}, \boldsymbol{\gamma})}{q(\boldsymbol{\gamma})} \right\} \right] \\ &\geq \sum_{\boldsymbol{\gamma}} q(\boldsymbol{\gamma}) \log p(\mathbf{y}, \boldsymbol{\gamma}) - \sum_{\boldsymbol{\gamma}} q(\boldsymbol{\gamma}) \log q(\boldsymbol{\gamma}) \\ &\equiv \log \underline{p}(\mathbf{y}; \mathbf{w}, \boldsymbol{\Gamma}) \end{aligned} \tag{33}$$

where going from the first to the second line of (33) is obtained using Jensen's inequality. Maximizing the right hand of (33) tightens the bound improving the quality of the approximation $\underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma})$ to $p(\mathbf{y})$. It can be shown that the difference between $\log p(\mathbf{y})$ and $\log \underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma})$ is the Kullback-Leibler divergence between $p(\boldsymbol{\gamma}|\mathbf{y})$ and $q(\boldsymbol{\gamma})$.

The second term of (33) is related to the entropy of q. Following Ročková (2017) the variational lower bound for $\log p(\mathbf{y})$ is given by

$$\log \underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma}) = \sum_{k=1}^{K} w_k \log p(\mathbf{y}, \boldsymbol{\gamma}_k) - w_k \log w_k \tag{34}$$

which has been simplified under the assumption that population of particles $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K$ contains only unique particles.

Since $\log \underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma})$ is a lower bound we can maximize this bound with respect to $\mathbf{w}$ and $\mathbf{\Gamma}$ to make the bound as tight as possible. The main body of the algorithm to optimize $\log \underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma})$ is a two–stage process. This process is similar to that of a tabu search Glover (1986).

In the first stage, we iterate through the population of bitstrings, using a greedy search strategy in an attempt to alter each bit in the model bitstring to increase the log likelihood. If the log-likelihood for the new bitstring is no higher than the previous bitstring, then the alteration is rejected and the next alteration tried. The alterations are also rejected if the new bitstring already exists within the population, ensuring that the constraint that all models in the population are unique is maintained.

In the second stage, we re–calculate the weights for each individual in the population, based on the likelihood of that model relative to the data $p(\mathbf{y}; \boldsymbol{\beta}_{\boldsymbol{\gamma}})$ and

use this to re–calculate the probability–based weights w_i for each bitstring in the population. This is then used to re–calculate the lower bound

$$\log \underline{p}(\mathbf{y}; \mathbf{w}, \Gamma) = \sum_{k=1}^{K} \mathbf{w}_k \log p(\mathbf{y}; \boldsymbol{\gamma}_k) - \mathbf{w}_k \log \mathbf{w}_k$$

which is the sum of the weighted log-likelihood of the population and the entropy of the probability weights. These two stages repeat until the lower bound converges.

Note that for fixed $\mathbf{w}$ each of the $\boldsymbol{\gamma}_k$'s can be optimized independently since (34) is an additive function of $\boldsymbol{\gamma}_k$'s. Hence, the first stage optimizes $\log \underline{p}(\mathbf{y}; \mathbf{w}, \boldsymbol{\Gamma})$ with respect to $\boldsymbol{\Gamma}$ in a greedy search over each of the $\boldsymbol{\gamma}_k$'s. To be more concrete, let $\boldsymbol{\gamma}_{jk}^{(i)} = (\gamma_{1k}, \ldots, \gamma_{j-1,k}, i, \gamma_{j+1,k}, \ldots, \gamma_{pk})^T$. We optimize $\mathbf{w}$ and $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K$ by executing the algorithm given in Algorithm 1 below. Let $\mathbf{p}$ denote the vector of posterior probabilities for each of the models in the population, while H is the entropy of the entire population. Thus $\log \underline{p}(\mathbf{y}; \mathbf{w}, \Gamma)$ balances the weighted posterior probabilities of the particles in the population against the diversity within that population.

Since only one component is modified during each iteration of the inner loop of the algorithm, model updates and downdates can be efficiently used to implement the Algorithm 1 (for details see Chapter 2).

Convergence is declared for a particular particle when no element of the particle is updated over $j = 1, \ldots, p$. Convergence of the algorithm is declared when all particles have been converged. Note that optimization over each of the $\boldsymbol{\gamma}_k$'s can

Algorithm 1 The PVA algorithm

while $\log \underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma})$ is still different from the previous iteration **do**
  for $k = 1, \ldots, K$ **do**
    for $j = 1, \ldots, p$ **do**
      if $p(\mathbf{y}, \boldsymbol{\gamma}_{jk}^{(1)}) > p(\mathbf{y}, \boldsymbol{\gamma}_{jk}^{(0)})$ **then**
        $\boldsymbol{\gamma}_{jk} = 1$
      else
        $\boldsymbol{\gamma}_{jk} = 0$
      end if
    end for
    $w_k = p(\mathbf{y}, \boldsymbol{\gamma}_k) / \sum_{j=1}^{K} p(\mathbf{y}, \boldsymbol{\gamma}_j)$
  end for
  Calculate $\log \underline{p}(\mathbf{y}; \mathbf{w}, \mathbf{\Gamma}) = \sum_{k=1}^{K} \mathbf{w}_k \log p(\mathbf{y}; \boldsymbol{\gamma}_k) - \mathbf{w}_k \log \mathbf{w}_k$
end while

be performed independently and as such implemented in an embarrassingly parallel manner. There is no need to re-optimize $\mathbf{\Gamma}$ for different $\mathbf{w}$ since the optimal values of the $\boldsymbol{\gamma}_k$'s are independent of $\mathbf{w}$.

Once the matrix $\mathbf{\Gamma}$ is fitted, duplicate particles can be discarded. Let $\boldsymbol{\gamma}_1^*, \ldots, \boldsymbol{\gamma}_{K^*}^*$ denote the selected set of K^* unique particles. Then the optimal value of the w_k's satisfy

$$w_k = \frac{p(\mathbf{y}, \boldsymbol{\gamma}_k^*)}{\sum_{j=1}^{K^*} p(\mathbf{y}, \boldsymbol{\gamma}_j^*)} = \frac{p(\boldsymbol{\gamma}_k^*)\text{BF}(\boldsymbol{\gamma}_k^*)}{\sum_{j=1}^{K^*} p(\boldsymbol{\gamma}_j^*)\text{BF}(\boldsymbol{\gamma}_j^*)}, \qquad 1 \leq k \leq K^*. \tag{35}$$

The approximate posterior inclusion probabilities, which we will denote $\boldsymbol{\omega}$ can be calculated using $\boldsymbol{\omega} = \sum_{k=1}^{K^*} w_k \boldsymbol{\gamma}_k^*$. The median posterior model can be obtained by rounding the elements of $\boldsymbol{\omega}$.

The main requirement of the above strategy is that closed form expressions for $\text{BF}(\boldsymbol{\gamma}_k)$, $1 \leq k \leq K$ are needed, or at least approximated in some way. Different

specifications of the prior distributions lead to different approximations of exact Bayesian model averaging.

We implement the above algorithm in `C++` which we developed into an `R` package we call `BLMA`. The internals of `BLMA` are implemented in `C++` and use the `R` packages `Rcpp` and `RcppEigen` to enhance computational performance. The library `OpenMP` was used to exploit parallel computation.

3.4. Numerical results

We will now assess the performance of PVA. In Section 3.4.1 we compare PVA against exact Bayesian model averaging for four small examples with $p < 30$ via the `R` package `blma` using the implementation outlined in Greenaway & Ormerod (2018). All of the following results were obtained in the `R` version 3.4.2 (R Core Team, 2017) and all figures were developed using the `R` package `ggplot2`. In sections 3.4.3 – 3.4.5 we will consider examples with $p > 30$ where it is infeasible to perform Bayesian model averaging exactly. Most simulations were run on a 64 bit Windows 10 Intel i7-7600MX central processing unit at 2.8GHz with 2 hyperthreaded cores and 32GB of random access memory. Multicore comparisons were run on a dedicated server using E5-2697v2 processors with 24 hyperthreaded cores and 512GB of RAM.

3.4.1. Comparing PVA against exact results. We considered several small data sets to illustrate our methodology for situations where we could compare PVA against a gold standard. These data sets can be found in the `R` packages `MASS` (Venables and Ripley, 2002), `ISLR` James et al. (2014) and `Ecdat` (Croissant, 2016). Table 1 summarizes the sizes, sources, and response variable for each data

set used. For each of the data sets some minimal preprocessing was used. We first used the `R` command `na.omit()` to remove samples containing missing predictors. For `USCrime` all variables except the predictor `S` were log-transformed. For all data sets the `R` command `model.matrix()` was used to construct the design matrix using all variables except for the response as predictors. The routines in 'blma' standardise the response and covariate matrix.

Dataset	n	p	Response	`R` package
UScrime	47	15	`y`	`MASS`
College	777	17	`Grad.Rate`	`ISLR`
Hitters	263	19	`Salary`	`ISLR`
Kakadu	1827	22	`income`	`Ecdat`

TABLE 1. A summary of the data sets used in the paper and their respective `R` packages.

To measure the quality of approximation of PVA to BMA we will use two metrics. The total posterior mass (TPM), and the mean marginal variable error (MMVE). These are given by

$$\text{TPM} = \sum_{k=1}^{K^*} p(\boldsymbol{\gamma}_k^*|\mathbf{y}) \qquad \text{and} \qquad \text{MMVE} = \frac{1}{p}\sum_{j=1}^{p} |\omega_j - \mathbb{E}(\gamma_j|\mathbf{y})|.$$

Note that the quantities $p(\boldsymbol{\gamma}_k^*|\mathbf{y})$ and $\mathbb{E}(\gamma_j|\mathbf{y})$ are available as outputs of the function `blma()` from the `R` package `blma`. The average values of TPM and MMVE over 100 random initial values of $\boldsymbol{\Gamma}$ for each of the data sets where independently $\gamma_{kj} \sim$ Bernoulli$(1/10)$, $1 \leq j \leq p$, $1 \leq k \leq K$ over a grid of K values from $K = 25$ to $K = 500$ are summarised in Figure 5. From this figure we see that both TPM and MMVE increase and decrease with K respectively. For each of the data set at least 50% of the total posterior mass is captured with less than $K = 200$ particles.

The mean absolute error in posterior inclusion probability with this value of K is roughly 0.05 which indicates that the median posterior model is reasonably well approximated.

3.4.2. Competing method settings. For data sets with $p > 45$ it is not feasible to perform exact BMA. For these examples we instead compare the model selection performance of PVA against the Lasso, SCAD and MCP penalized regression methods as implemented by the `R` package `ncvreg` (Breheny and Huang, 2011), PEM using the `R` package `PEM` (obtained via personal communication with Veronika Ročková), and Bayesian model averaging via MCMC using the `R` package `BAS`. The setting are implied by the `R` commands below.

- **Penalized regression via `ncvreg` package.** We used the following command.

  ```
  ncvreg(mX,vy,penalty=penalty)
  ```

  where `penalty` is `"MCP"`, `"SCAD"` or `"lasso"` corresponding to the penalties of the same name as described in Breheny and Huang (2011). For these methods we make use of the extended Bayesian information criteria (EBIC) (Chen et al., 2008) to choose the tuning parameter λ. The EBIC minimizes

  $$\text{EBIC}(\lambda) = n \log(\text{RSS}_\lambda/n) + d_\lambda \left[\log(n) + 2\log(p)\right],$$

  where RSS_λ is the estimated residual sum of squares $\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_\lambda\|^2$, $\widehat{\boldsymbol{\beta}}_\lambda$ is the estimated value of $\boldsymbol{\beta}$ for a particular value of λ and d_λ is the number of non-zero elements of $\widehat{\boldsymbol{\beta}}_\lambda$. This differs from the regular BIC by an addition

of a $2d_\lambda \log(p)$ term. Wang and George (2007) showed that this criterion performs well in several contexts.

- **Particle EM via the** `PEM` **package.** We used the following command.

```
PEM(vy,mX,v0,v1,type="betabinomial",
penalty="entropy",epsilon=1.0E-5,theta=0.5,
a=1,b=p,alpha=1,current=t(mGamma),weights="FALSE")
```

  where $\gamma_{kj} \sim$ Bernoulli(ρ), $1 \leq j \leq p$, $1 \leq k \leq K$ with $K = 200$ (noting that the initial population matrix in PVA is the transpose of the initial population matrix used by PEM). The choices used for `rho`, `v0` and `v1` are different for each data set and are described in each of the sections below.

- **MCMC via the** `BAS` **package**: We used the following command.

```
bas.lm(vy~mX, prior="g-prior", modelprior=uniform(),
    initprobs="uniform", MCMC.iterations=1e7)
```

  The estimated median posterior model is used for the purposes of model selection.

- **MCMC via the** `BMS` **package**: We used the following command.

```
res.bms <- bms(cbind(vy,mX), burn = 1000, iter = 1000000,
    nmodel = 10000, mcmc = "bd", g = "BRIC", mprior = "random",
    mprior.size = NA, user.int = TRUE, start.value = start.value,
    g.stats = TRUE, logfile = TRUE, logstep = 100000,
    force.full.ols = FALSE, fixed.reg=numeric(0))
```

  where `start.value` is the value of $\boldsymbol{\gamma}$ given by PVA.

We used simulated data in each of the sections below in such a way that the true data generating model was known. We used the F_1-score (see Rijsbergen, 1979) to assess the quality of model selection for each of the above models, which is defined to be the harmonic mean between precision and recall given by

$$\mathrm{F}_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

with TP, FP and FN being the number of true positives, false positives and false negatives respectively. Note that F_1 is a value between 0 and 1 and higher values are preferred. We use this measure to avoid prefererring either of the two boundary models, that is, selecting none or all of the variables.

3.4.3. Simulated high-dimensional example. We first present an example where $n > p$ and p is relatively small ($p = 12$), to allow for the full enumeration of the model space. Later, we show an example for the important $p > n$ case. We compare our results against the Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), MCP (Zhang, 2010), `BMS` (Zeugner and Feldkircher, 2015) and `VARBVS` (Carbonetto and Stephens, 2011) algorithms.

Our first numerical experiment is designed to show that our algorithm successfully finds the posterior models of high probability, overcoming the difficulties of optimising over the multi-modal spike-and-slab posterior. This example is taken from (Ročková, 2017). We consider a random sample of $n = 50$ observations on $p = 12$ predictors. $\mathbf{X}_i \sim N_p(\mathbf{0}, \mathbf{\Sigma})$ for $i = 1, \ldots, n$ where $\mathbf{\Sigma} = \mathrm{bdiag}(\mathbf{\Sigma}_1, \mathbf{\Sigma}_1, \mathbf{\Sigma}_1, \mathbf{\Sigma}_1)$ with $\mathbf{\Sigma}_1 = (\sigma_{ij})_{i,j=1}^{3,3}$ where $\sigma_{ij} = 0.9$ for $i \neq j$ and $\sigma_{ii} = 1$. The true model is $\boldsymbol{\beta}_0 = (1.3, 0, 0, 1.3, 0, 0, 1.3, 0, 0, 1.3, 0, 0)^\top$. The responses are then generated from $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \mathbf{I}_n)$.

A comparison of the performance of PVA for the hyper-g, robust Bayarri, Beta-prime and Cake priors on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on $\boldsymbol{\gamma}$ using F_1 score is presented in the top panel of Figure 1. A comparison of the performance of the MCP, SCAD, lasso, PVA, BMS, BAS and PEM methods using F_1 score is given in the bottom panel of Figure 1.

From the top panel of 1 we see that beta-binomial(1,1) and beta-binomial(1,p) priors on $\boldsymbol{\gamma}$ lead to better performances. From th bottom panel we see that PVA with beta-binomial(1,1) or beta-binomial(1,p) priors on $\boldsymbol{\gamma}$ performs about better than MCP, SCAD, lasso, and PEM methods, and about the same as BAS. PVA with a uniform prior performs similarly to BMS.

3.4.3.1. *Exploration of the posterior model space.* If the covariates in a model selection problem are highly collinear then the posterior distribution will be highly multi-modal when a spike-and-slab prior structure is used. This can make seeking the optimal model very challenging, due to the many local optima. In this section, we present a series of numerical experiments which demonstrate the capability of our algorithm to successfully find the models with high posterior probability in such situations.

Our population of bit strings $\boldsymbol{\Gamma}^{(0)} = (\boldsymbol{\gamma}_1^{(0)}, \ldots, \boldsymbol{\gamma}_K^{(0)})$ with $K = 20$ particles was randomly initialised from a sequence of independent Bernoulli trials with probability of success $1/2$. Figure 2 shows all 4096 posterior model probabilities for a data set simulated from a regression model with 12 covariates, ordered by the model's bit strings, represented by blue dots. Superimposed over this are the models found by PVA, represented by red dots. We can clearly see a few peaks

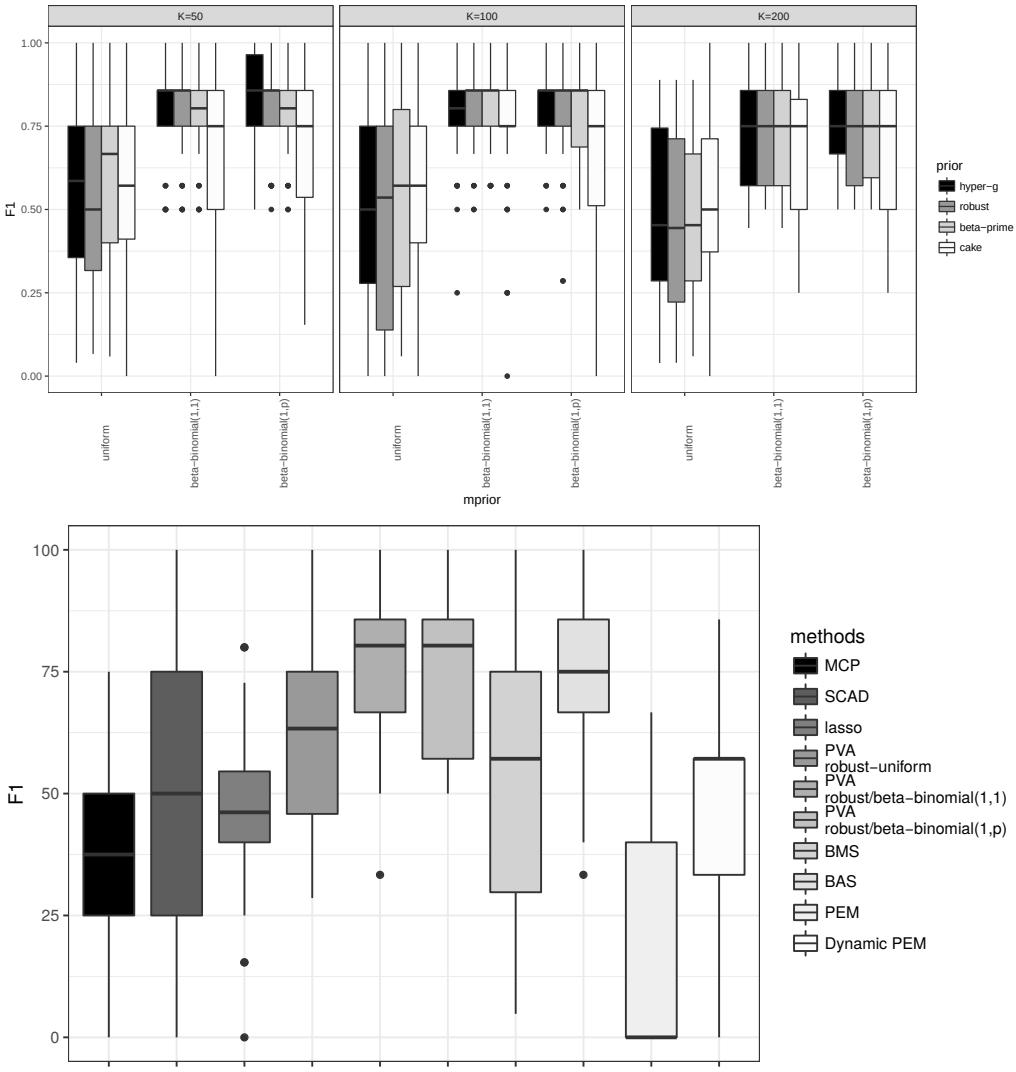

FIGURE 1. Top panel: Comparison of the performance of the PVA method on the high-dimensional data set with different g and $\boldsymbol{\gamma}$ priors using F_1 score. The hyper-g, robust Bayarri, Beta-prime and Cake priors on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on $\boldsymbol{\gamma}$ are used. Bottom panel: Comparison of the performance of the MCP, SCAD, lasso, PVA, BMS, BAS and PEM methods on the high-dimensional data set using F_1 score. For PVA, the robust Bayarri prior on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on $\boldsymbol{\gamma}$ are used.

in the full posterior distribution. Our experiment aims to show that most of these posterior peaks are successfully identified by our algorithm.

As Figure 2 shows, in the plots of the log posterior probabilities of the models, the particles can be seen clustering at the highest probability models first, then spreading through the medium and low probability models. From these plots we can see that once K is high enough, there is a good variety of high, medium and low posterior probability models in the population of particles. The coverage of the posterior probability distribution by the population of particles is high, as the

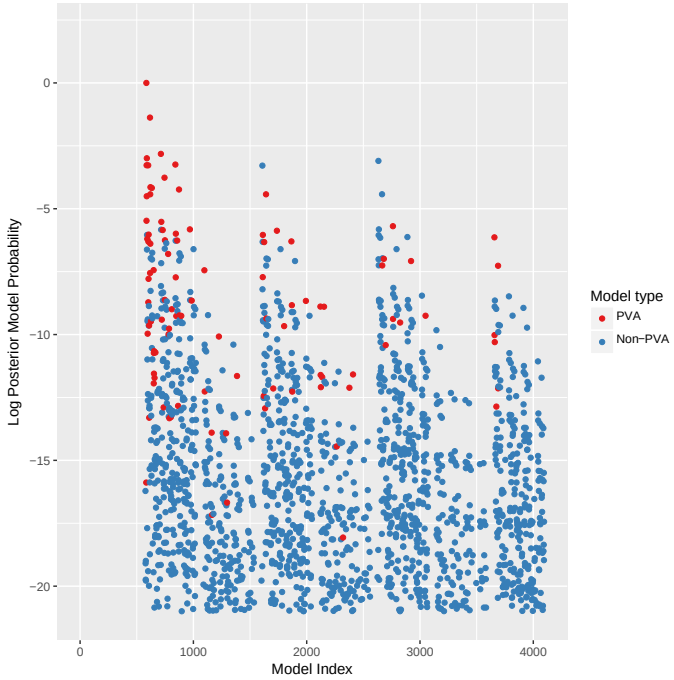

FIGURE 2. Posterior model probabilities when $p = 12$. Red points denote models visited by the PVA algorithm, while blue points are models that were not visited. Note that the PVA algorithm visits the highest posterior probability points first

particles tend to cluster towards the higher posterior probability models as PVA's greedy search algorithm proceeds.

3.4.4. Communities and crime data set. We use the `Communities & Crime` data set obtained from the UCI Machine Learning Repository

`http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`

The data collected was part of a study by Redmond and Baveja (2002) combining socio-economic data from the 1990 United States Census, law enforcement data from the 1990 United States Law Enforcement Management and Administrative Statistics survey, and crime data from the 1995 Federal Bureau of Investigation's Uniform Crime Reports.

The raw data consists of 2215 samples of 147 variables the first 5 of which we regard as non-predictive, the next 124 are regarded as potential covariates while the last 18 variables are regarded as potential response variables. Roughly 15% of the data is missing. We proceed with a complete case analysis of the data. We first remove any potential covariates which contained missing values leaving 101 covariates. We also remove the variables `rentLowQ` and `medGrossRent` since these variables appeared to be nearly linear combinations of the remaining variables (the matrix $\mathbf{X}$ had two singular values approximately 10^{-9} when these variables were included). We use the `nonViolPerPop` variable as the response. We then remove any remaining samples where the response is missing. The remaining data set consist of 2118 samples and 99 covariates. Finally, the response and covariates are standardized to have mean 0 and standard deviation 1. Empirical correlations between variables range from 3.3×10^{-5} to 0.999.

Here we center the $\mathbf{X}$ matrix and simulate new data from $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ and the $\boldsymbol{\varepsilon}_i$ are independently drawn with $\boldsymbol{\varepsilon}_i \sim N(0, \sigma^2)$ where $\sigma^2 = \|\mathbf{y}_{\text{raw}} - \mathbf{X}\boldsymbol{\beta}_0\|/n$ and $\mathbf{y}_{\text{raw}}$ denotes the original response vector. The data was fit using a linear model. Variables with p-values below 0.05 were considered in the model, i.e, the corresponding value of $\boldsymbol{\gamma}$ was set to 1. The MLE coefficients divided by 4 where taken to be the true values of the coefficient in order to make the problem more difficult, i.e., we set $\boldsymbol{\beta}_0 = \widehat{\boldsymbol{\beta}}/4$.

A comparison of the performance of PVA for the hyper-g, robust Bayarri, Beta-prime and Cake priors on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on $\boldsymbol{\gamma}$ using F_1 score is presented in the top panel of Figure 3. A comparison of the performance of the MCP, SCAD, lasso, PVA, BMS, BAS and PEM methods using F_1 score is given in the bottom panel of Figure 3.

From the top panel of 3 we see that all model priors lead to similar performances. From th bottom panel we see that PVA performs similarly to BAS and BMS and better than penalized regression and PEM methods.

3.4.5. Quantitative trait loci data set. For our final $p > n$ simulation example we will use the design matrix based on an experiment on a backcross population of $n = 600$ individuals for a single large chromosome of 1800 cM. This giant chromosome was covered by 121 evenly spaced markers from Xu (2007). Nine of the markers overlapped with QTL of the main effects and 13 out of the $\binom{121}{2} = 7260$ possible marker pairs had interaction effects. The $\mathbf{X}$ matrix combines the main effects and interaction effects to make a 600×7381 matrix. The values of the true coefficients are listed in Table 1 of Xu (2007) ranging from 0.77 to 4.77 in absolute

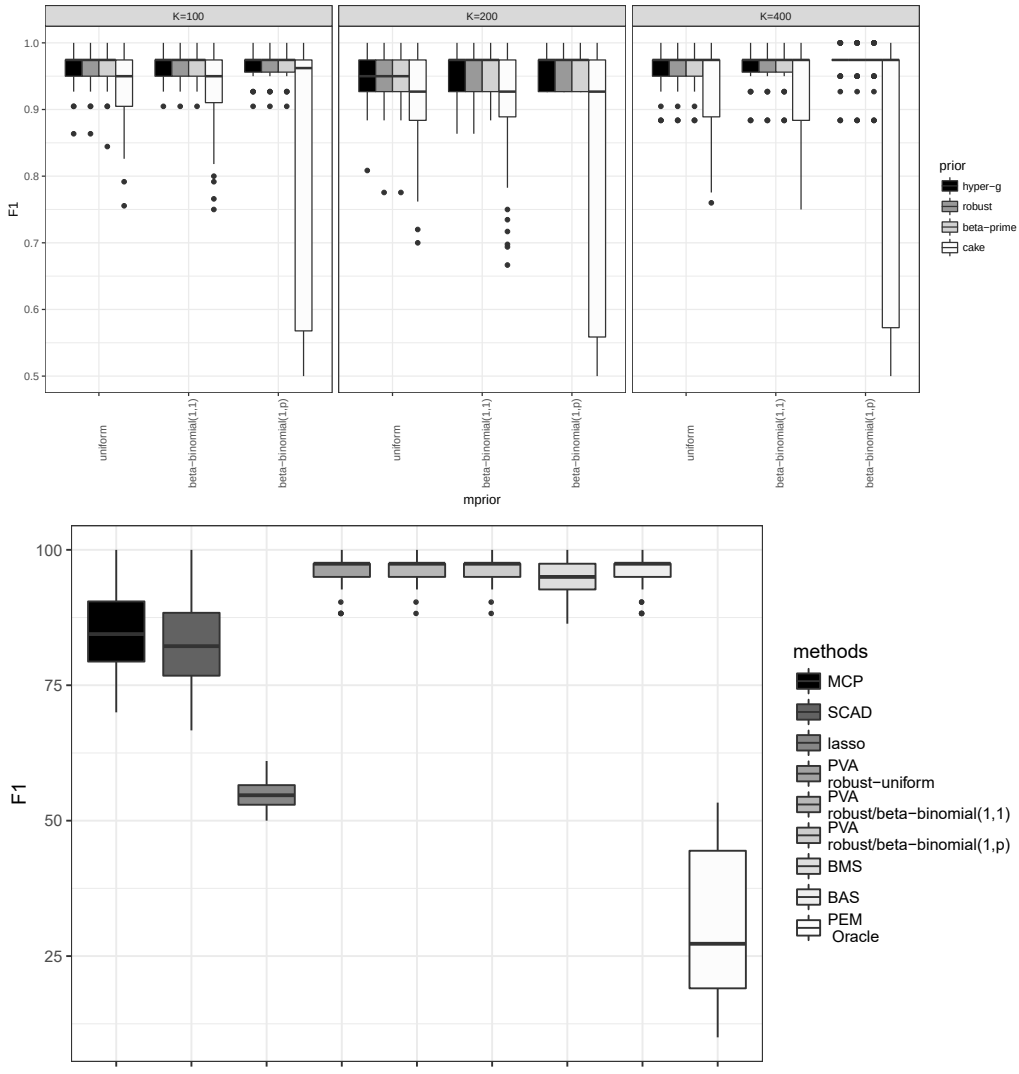

FIGURE 3. Top panel: Comparison of the performance of the PVA method on the Communities and Crime data set with different g and $\boldsymbol{\gamma}$ priors using F_1 score. The hyper-g, robust Bayarri, Beta-prime and Cake priors on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on $\boldsymbol{\gamma}$ are used. Bottom panel: Comparison of the performance of the MCP, SCAD, lasso, PVA, BMS, BAS and PEM methods on the Communities and Crime data set using F_1 score. For PVA, the robust Bayarri prior on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on $\boldsymbol{\gamma}$ are used.

magnitude and correlations range from 0 to 0.8 where most of the higher correlation occurs along the off-diagonal values of the correlation matrix of the covariates. Here we center the $\mathbf{X}$ matrix and simulate new data from $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ and the $\boldsymbol{\varepsilon}_i$ are independently drawn with $\boldsymbol{\varepsilon}_i \sim N(0, 20)$. Similar simulation studies were conducted in Xu (2007) and Kärkkäinen and Sillanpää (2012). This process was repeated 50 times. For this simulation setting PVA has the best model selection accuracy, smallest MSEs and smallest parameter biases of all the methods compared. The Lasso, SCAD, MCP, EMVS, BAS and BMS methods took 1.5, 1.5, 1.8, 1229, 2011, 5327 seconds respectively. Our implementation is quite fast, e.g., we fit a $n = 600$ and $p \approx 7200$ problem with $K = 100$ particles in 1-2 minutes on a standard 2018 laptop using a single core. On a dedicated server with 48 cores the same problem can be fit in around 8 seconds on 20 cores, and as little as around 5 seconds when all cores are used.

A comparison of the performance of PVA for the hyper-g, robust Bayarri, Beta-prime and Cake priors on g and the uniform, beta-binomial$(1, 1)$ and beta-binomial$(1, p)$ priors on $\boldsymbol{\gamma}$ using F_1 score is presented in the top panel of Figure 4. A comparison of the performance of the MCP, SCAD, lasso, PVA, BMS, BAS and PEM methods using F_1 score is given in the bottom panel of Figure 4.

From the top panel of 4 we see that beta-binomial$(1, 1)$ and beta-binomial$(1, p)$ lead to similar performances. From th bottom panel we see that PVA with beta-binomial$(1, 1)$ and beta-binomial$(1, p)$ performs similarly to BAS and better than penalized regression, PEM and BMS methods.

3.4.6. Comparison of PVA against other model selection methods on simulated data sets. The method used to assess the quality of the variable selection

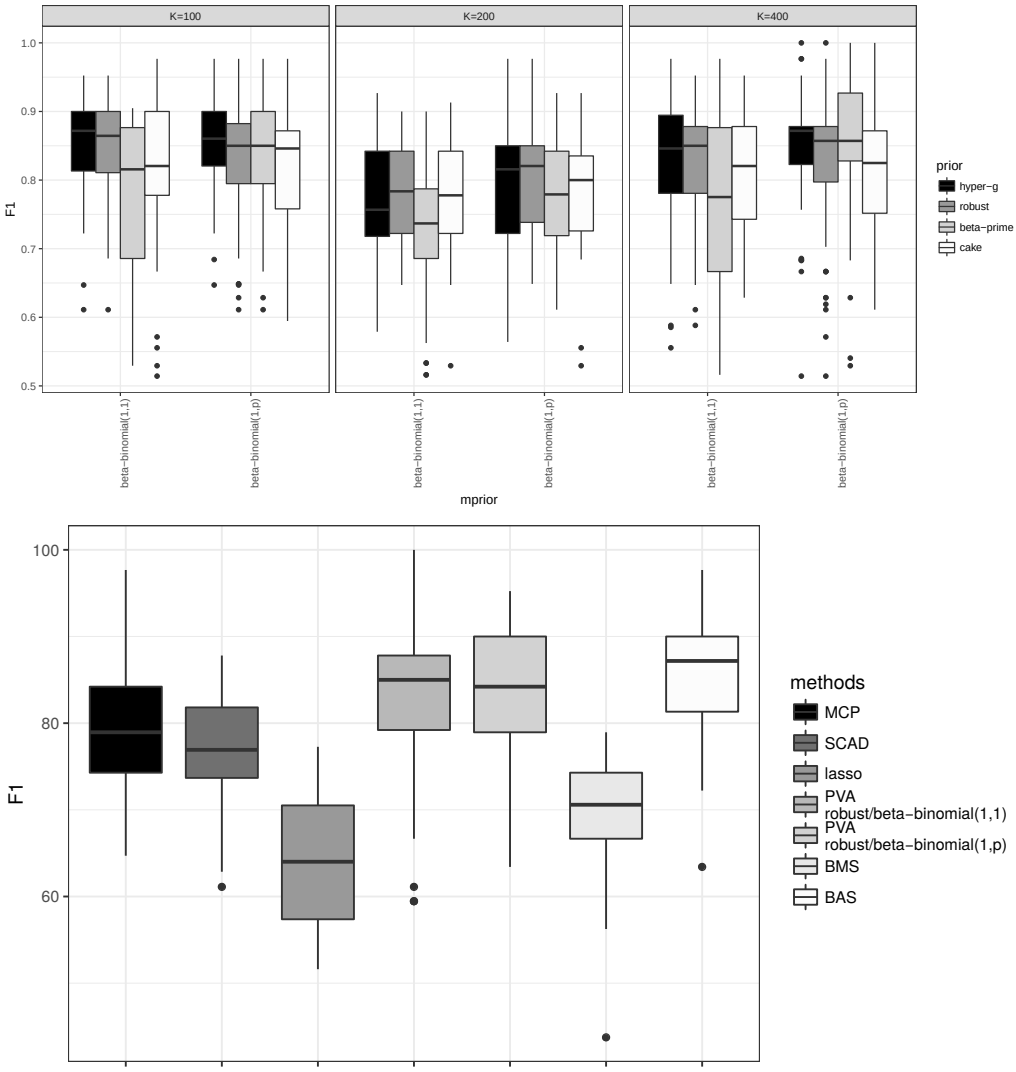

FIGURE 4. Top panel: Comparison of the performance of the PVA method on the QTL data set with different g and γ priors using F_1 score. The hyper-g, robust Bayarri, Beta-prime and Cake priors on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on γ are used. Bottom panel: Comparison of the performance of the MCP, SCAD, lasso, PVA, BMS, BAS and PEM methods on the QTL data set using F_1 score. For PVA, the robust Bayarri prior on g and the uniform, beta-binomial(1, 1) and beta-binomial(1, p) priors on γ are used.

was to generate data from a known true model $\boldsymbol{\gamma}$, and then compare this against the model $\widehat{\boldsymbol{\gamma}}$ found by each of the model selection methods that we compared. We then calculated the F_1 score for $\widehat{\boldsymbol{\gamma}}$.

The experiments were repeated with the Cake prior, Maruyama's Beta-prime prior, Liang's hyper-g prior and Bayarri's robust prior. The results of the algorithm were found to be insensitive to the choice of prior. For each combination of population size, data set, and prior the experiment was repeated 50 times.

3.5. Variable inclusion for small data sets

We compared variable selection using PVA against exact variable selection on five small data sets, Hitters, Bodyfat, Wage, College and US Crime. The variable inclusion probabilities were estimated by taking the sum of the columns of the population of models selected $\boldsymbol{\Gamma}$ weighted by marginal likelihood of each model. The exact variable inclusion probabilities were calculated by summing the columns of the matrix of all possible models $\boldsymbol{\Gamma}$ weighted by the marginal likelihood of each model. The mean relative error of the variable inclusion probabilities estimated by PVA was calculated, and the results of these comparisons are presented in Table 3.5. The number of particles in the population K affected the variable inclusion probability in the variables selected by PVA, while the marginal probability $p(\boldsymbol{\gamma}|\mathbf{y})$ used to weight models in Γ seemed to have only a very minor impact. When the robust Bayarri prior is chosen to rank models in PVA, the marginal probability $p(\boldsymbol{\gamma}|\mathbf{y})$ changes a lot as opposed to ranking models with other priors. Variables with low posterior probability are truncated to 0, as PVA

seeks higher posterior probability models, ignoring the lower posterior probability models.

Dataset	Prior	$<= 0.5$			> 0.5		
		$K = 20$	$K = 50$	$K = 100$	$K = 20$	$K = 50$	$K = 100$
Bodyfat	BIC	0.63	0.48	0.37	0.07	0.01	0.02
	Liang's hyper-g prior	0.66	0.52	0.42	0.07	0.01	0.02
	Bayarri's robust prior	0.65	0.52	0.40	0.07	0.01	0.02
	beta-prime prior	0.65	0.51	0.39	0.06	0.01	0.02
College	BIC	0.70	0.58	0.49	0.03	0.02	0.03
	Liang's hyper-g prior	0.90	0.78	0.64	0.06	0.06	0.06
	Bayarri's robust prior	0.88	0.78	0.63	0.06	0.06	0.06
	beta-prime prior	0.82	0.66	0.57	0.03	0.06	0.06
Hitters	BIC	0.74	0.64	0.50	0.12	0.07	0.06
	Liang's hyper-g prior	NaN	0.81	0.83	NaN	0.17	0.07
	Bayarri's robust prior	0.84	0.81	0.75	0.29	0.17	0.07
	beta-prime prior	0.79	0.72	0.67	0.27	0.13	0.05
USCrime	BIC	0.82	0.70	0.64	0.47	0.15	0.12
	Liang's hyper-g prior	0.76	0.71	0.64	0.45	0.16	0.12
	Bayarri's robust prior	0.79	0.70	0.61	0.35	0.14	0.08
	beta-prime prior	0.76	0.70	0.64	0.45	0.16	0.13
Wage	BIC	0.67	0.49	0.35	0.00	0.00	0.00
	Liang's hyper-g prior	0.69	0.47	0.33	0.00	0.00	0.00
	Bayarri's robust prior	0.69	0.47	0.32	0.00	0.00	0.00
	beta-prime prior	0.69	0.47	0.32	0.00	0.00	0.00

TABLE 2. Relative error of the variable inclusion probability estimated by PVA to the exact variable inclusion probability, partitioned by exact probability under or equal to 0.5 and over 0.5

The same general trends were observed in all small data sets. Total posterior probability is higher for the beta- binomial model prior than for the uniform model prior, while the variable inclusion error is lower. This same general trend is seen regardless of g-prior. Total posterior probability increases with increased population size K, while variable inclusion error decreases. Although the PVA algorithm is deterministic, variation in the results amongst the trials is seen due to the random initialisation of $\mathbf{\Gamma}$.

3.6. Conclusion

We have proposed a deterministic Bayesian model selection algorithm which is computationally efficient and simple. Like Particle EM (Ročková, 2017), our algorithm maintains a population of solutions and ensures diversity of that population to explore the uncertainty of the selected model. This gives far more information about the model selection process than simply choosing one best model. However, whereas Particle EM uses a spike-and-slab prior for the regression coefficients, our approach uses a g-prior, which avoids the Bartlett's Paradox and Information Paradox. Importantly, both approaches can be implemented using rank-one updates and downdates and the model selection posterior probabilities are available in closed form, which allows the algorithm to be implemented in a computationally efficient manner.

While previously model selection algorithms using the Maruyama, Liang-g and robust Bayarri priors have typically been implemented using MCMC, our algorithm allows the advantages of these priors while using a deterministic algorithm. The PVA algorithm presented in this chapter is implemented in the `blma` package in the `pva()` function.

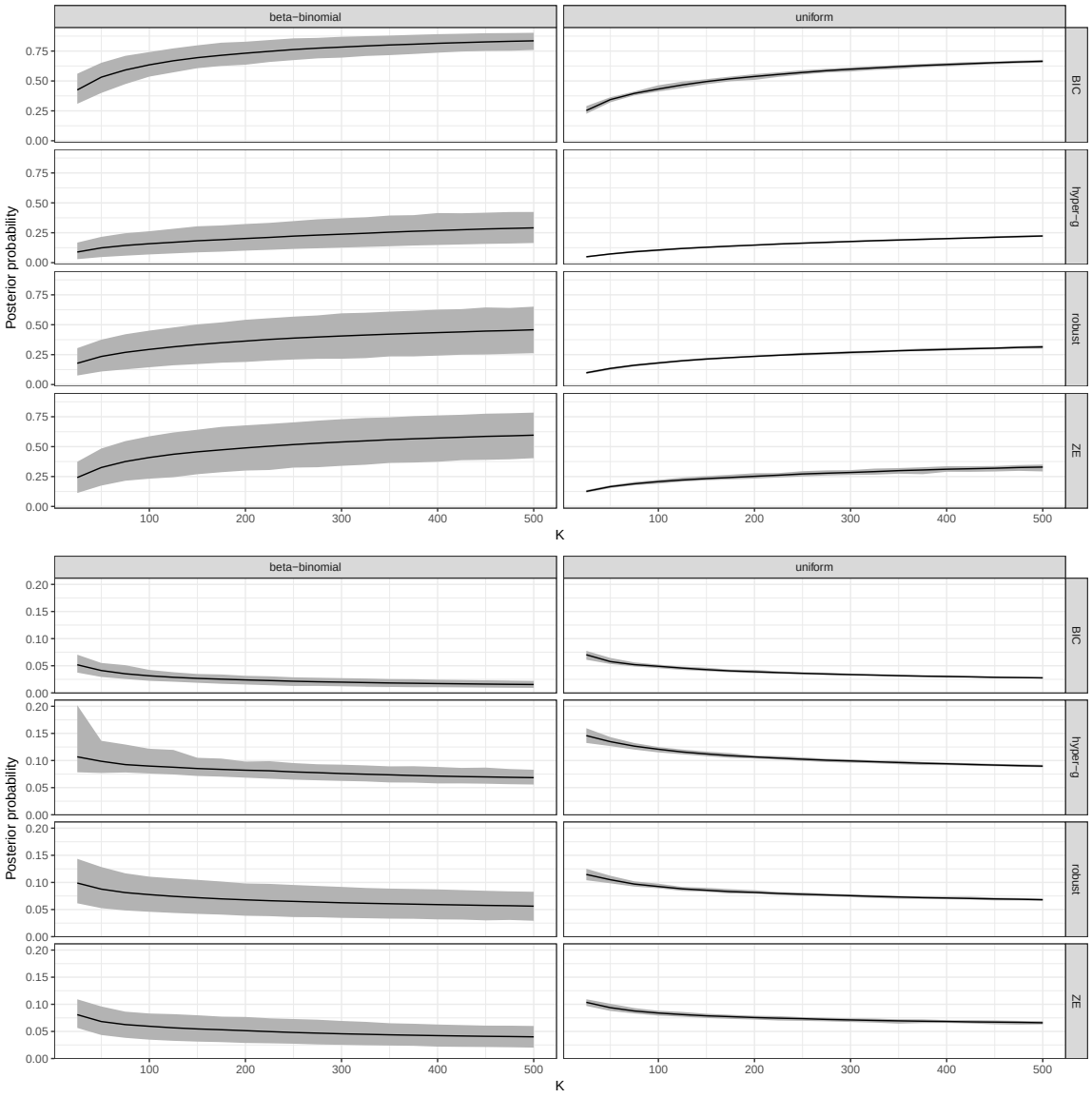

FIGURE 5. PVA was run on the Kakadu data set. The total posterior model probability and error in posterior variable inclusion probability were calculated using the exact posterior model and variable inclusion probability from every possible sub-model. These were calculated for a range of population sizes from 25 to 500, in 25 model increments. As the population increases, the total posterior model probability increases while the error in posterior variable inclusion probability decreases. The labels at the top of each panel refer to model prior used, while the labels to the right of each row refer to the choice of g-prior.

CHAPTER 4

Gaussian Variational Approximation of Zero-inflated Mixed Models

Abstract

In this chapter we consider variational inference for zero-inflated Poisson (ZIP) regression models using a latent variable representation. The model is extended to include random effects which allow simple incorporation of spline and other modelling structures. Several variational approximations to the resulting set of models are presented, including a novel approach based on the inverse covariance matrix rather than the covariance matrix of the approximate posterior density for the random effects. This parameterisation improves upon the computational cost and numerical stability of previous methods. We demonstrate these approximations on simulated and real data sets.

4.1. Introduction

Count data with a large number of zeros arises in many areas of application, such as data arising from physical activity studies, insurance claims, hospital visits or defects in manufacturing processes. Zero inflation is a frequent cause of overdispersion in Poisson data, and not accounting for the extra zeros may lead to biased parameter estimates. These models have been used for many applications, including defects in manufacturing in Lambert (1992), horticulture in Hall (2000), length of stay data from hospital admissions (Yau et al., 2003), psychology pharmaceutical studies (Min and Agresti, 2005), traffic accidents on roadways (Shankar et al., 1997) and longitudinal studies (Lee et al., 2006).

The strength of this approach derives from modelling the zero and non-zero count data separately as a mixture of distributions for the zero and non-zero components, allowing analysis of both the proportion of zeros in the data set and the conditions for the transition from zero observations to non-zero observations. When combined with a multivariate mixed model regression framework, an extremely rich class of models can be fit allowing a broad range of applications to be addressed. Often the transition from zero to non-zero has a direct interpretation in the area of application, and is interesting in its' own right.

Bayesian estimation methods for zero-inflated models were developed in Ghosh et al. (2006) using Monte Carlo Markov Chain (MCMC) implemented with WinBUGS, and in Vatsa and Wilson (2014) using a Variational Bayes solution to the inverse zero-inflated Poisson regression problem. While simple forms of these models are easy to fit with standard maximum likelihood techniques, more general models incorporating random effects, splines and missing data typically have

no closed form solutions and hence present a greater computational challenge to fit.

Fitting these models is typically done with MCMC techniques, but these techniques can be computationally intensive and prone to convergence problems. Other fitting methods such as the Variational Bayes approach above can be inflexible, not allowing complicated models incorporating random effects, splines and missing data.

We build upon a latent variable representation of these models to allow a tractable Semiparametric Mean Field Variational Bayes approximation to be derived. Semiparametric Mean Field Variational Bayes is an approximate Bayesian inference method as detailed in Ormerod and Wand (2010) and Rohde and Wand (2015), which allows us to fit close approximations to these models using a deterministic algorithm which converges much more quickly.

We allow a flexible regression modelling approach incorporating both fixed and random effects by using a Gaussian Variational Approximation (GVA) as defined in Ormerod and Wand (2012) on the regression parameters to allow a non-conjugate Gaussian prior to be used. This makes the resulting Gaussian posterior distribution of the regression parameters easier to interpret. Posterior inference on the other parameters are performed with Mean Field Variational Bayes.

The focus of this chapter is on developing methods of fitting flexible ZIP regression models accurately, and showing the advantages of our methods to previously presented methods. We also investigate stability problems that can arise when using naive versions of these methods, and the modifications to the fitting methods we devised to mitigate these problems. In Section 4.2 we define

our model and provide a framework for our approach incorporating regression modelling and random effects. In Section 4.3 we focus on several approaches to fitting the Gaussian component of our model. In Section 4.4, we present new parameterisations for use in these algorithms which offers substantial advantages in accuracy, numerical stability and computational speed. In Section 4.5 we perform numerical experiments on simulated data which demonstrate these advantages. In Section 4.6 we show an application of our pure Poisson model fitting method to a hierarchical model studying the effect of ethnicity on the rate of police stops, and an application of our zero-inflated Poisson model fitting method to a multi- level longitudinal study of pest control in apartments. Finally, in Section 5 we conclude with a discussion of the results. An appendix contains details of the derivation of the variational lower bound for our model.

4.2. Zero-inflated models

In this section we present a Bayesian zero-inflated Poisson model for count data with extra zeros. After introducing the latent variable representation of Bayesian zero-inflated models, we first extend this to a model incorporating fixed effects regression modelling, and extend the model again to a more flexible mixed model approach incorporating both fixed and random effects.

4.2.1. Modelling zero-inflated Poisson data. We consider a sample of counts y_i, $1 \leq i \leq n$, where there are an excessive number of zeros for a Poisson model, but the sample is otherwise well-modelled by a Poisson distribution. A popular approach using latent variables views the data as the product of two data-generating processes, a Bernoulli process that determines whether the data is definitely zero, and a second process where data is generated from a Poisson distribution which may be zero.

$$P(Y_j = y_i) = \begin{cases} \rho + (1-\rho)e^{-\lambda}, & y_i = 0 \\ (1-\rho)\frac{\lambda^{y_i}e^{-\lambda}}{y_i!}, & y_i \geq 1. \end{cases} \tag{36}$$

This model yields the probability distribution specified in Equation (36). Note that this allows zeros to be generated from the model in one of two ways – either from the Bernoulli process generating a zero or from the Bernoulli process generating a Poisson sample which is then zero. A latent variable representation of this parameterisation introduces the latent variables r_i which equal 1 when $y_i > 0$ and 0 otherwise.

$$\begin{aligned} P(Y_i = y_i | r_i) = & \quad \frac{\exp(-\lambda r_i)(\lambda r_i)^{y_i}}{y_i!} \quad \text{and} \\ r_i \sim & \quad \text{Bernoulli}(1-\rho). \end{aligned} \tag{37}$$

This leads to the specification for the probability distribution used in (37) where Bernoulli(π) denotes the probability distribution $\pi^k(1-\pi)^{1-k}$, with $k \in \{0, 1\}$ and $\pi \in [0, 1]$.

Let p be the number of fixed effects, m be the number of groups in the random effects and b be the block size for each of those groups. We use $\mathbf{1}_p$ and $\mathbf{0}_p$ to denote

the $p \times 1$ column vectors with all entries equal to 1 and 0, respectively. Let $\mathbf{y}$ be the $n \times 1$ vector of counts. The Euclidean norm of a column vector $\mathbf{v}$, defined to be $\sqrt{\mathbf{v}^\top \mathbf{v}}$, is denoted by $\|\mathbf{v}\|$. For a $p \times 1$ vector $\mathbf{a}$, we let $\text{diag}(\mathbf{a})$ denote the $p \times p$ matrix with the elements of $\mathbf{a}$ along its' diagonal.

The function expit(x) denotes the function $1/(1+\exp(-x))$ which is the inverse of the logit function.

We can extend the model naturally to a multiple covariate regression model by using a log link function on the response variable and replacing the parameter λ in the model above with $\mathbf{x}_i^\top \boldsymbol{\beta}$ to specify the mean, where $\mathbf{x}_i, \boldsymbol{\beta} \in \mathbb{R}^p$, with $\mathbf{x}_i$ the vector of observed predictors and $\boldsymbol{\beta}$ the vector of regression coefficients. Letting $\mathbf{r} = (r_1, \ldots, r_n)$, the model becomes

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{r}, \boldsymbol{\beta}) &= \mathbf{y}^\top \mathbf{R}(\mathbf{X}\boldsymbol{\beta}) - \mathbf{r}^\top \exp(\mathbf{X}\boldsymbol{\beta}) - \mathbf{1}_n^\top \log \Gamma(\mathbf{y} + \mathbf{1}_n), \quad \text{and} \\ r_i|\rho &\sim \text{Bernoulli}(1-\rho), \quad 1 \le i \le n, \end{aligned}$$

where $\mathbf{X}$ is the $n \times p$ matrix whose ith row equals $\mathbf{x}_i$ and $\mathbf{R} = \text{diag}(\mathbf{r})$.

4.2.2. Extending to mixed models, incorporating random effects. To be able to construct multivariate models with as much generality as possible, we wish to specify the full model as a General Design Bayesian Generalized Linear Mixed Model, as in Zhao et al. (2006). This allows for a very rich class of models, which can incorporate features such as random intercepts and slopes, within-subject correlation and smoothing splines, as in Wand and Ormerod (2008), into our models.

The zero-inflated model regression model introduced above can be extended to a flexible mixed model by incorporating the latent variable $\mathbf{r}$ which controls the

mixture of the zero and non-zero components from the zero-inflated model above into a Poisson mixed model likelihood.

When the indicator $r_{ij} = 0$, the likelihood is 1 for $y_{ij} = 0$ and 0 for all $y_{ij} > 0$, and when the indicator $r_{ij} = 1$, the likelihood is a Poisson mixed model regression likelihood for y_{ij}. r_{ij} is a Bernoulli indicator with probability ρ, allowing a proportion of zero- inflation in the observed data to be specified. The jth predictor/response pair for the ith group is denoted by $(x_{ij}, y_{ij}), 1 \leq j \leq n_i, 1 \leq i \leq m$, where $x_{ij} \in \mathbb{R}$, and the y_{ij} are non-negative integers. For each $1 \leq i \leq m$, define the $n_i \times 1$ vectors $y_{ij} = [y_{i1}, \ldots, y_{in_i}]^\top$ as the response vector. Vectors $y_1, \ldots, y_m$ are assumed to be independent of each other. We develop a zero-inflated regression model incorporating both fixed effects $\boldsymbol{\beta}$ and random effects u. The log-likelihood for one observation is then

$$\begin{aligned}
\log p(y_{ij}|r_{ij}, \boldsymbol{\beta}, \mathbf{u}) &= y_{ij} r_{ij} (\mathbf{x}_i^\top \boldsymbol{\beta} + \mathbf{z}_{ij}^\top \mathbf{u}) - r_{ij} \exp(\mathbf{x}_{ij}^\top \boldsymbol{\beta} + \mathbf{z}_{ij}^\top \mathbf{u}) - \log \Gamma(y_{ij} + 1), \\
r_{ij}|\rho &\sim \text{Bernoulli}(\rho), 1 \leq i \leq m, 1 \leq j \leq n, \qquad \text{and} \\
\rho &\sim \text{Beta}(\alpha, \beta).
\end{aligned}$$

We now extend this to multiple dimensional random effects. Let $\mathbf{C} = [\mathbf{X}, \mathbf{Z}]$ and $\boldsymbol{\nu} = \left[\boldsymbol{\beta}^\top, \mathbf{u}^\top\right]^\top$. The multivariate model with multiple observations is then

$$\begin{aligned}
\log p(\mathbf{y}|\mathbf{r}, \boldsymbol{\beta}, \mathbf{u}) &= \mathbf{y}^\top \mathbf{R}(\mathbf{C}\boldsymbol{\nu}) - \mathbf{r}^\top \exp(\mathbf{C}\boldsymbol{\nu}) - \mathbf{1}_n^\top \log \Gamma(\mathbf{y} + \mathbf{1}_n), \quad \text{and} \\
r_i &\sim \text{Bernoulli}(\rho), 1 \leq i \leq n,
\end{aligned}$$

with priors

$$\log p(\boldsymbol{\Sigma}_{\mathbf{uu}}) = \text{Inverse Wishart}(\boldsymbol{\Psi}, v),$$

$$\rho \sim \text{Beta}(\alpha, \beta),$$

$$\boldsymbol{\beta}|\sigma^2_{\boldsymbol{\beta}} \sim \text{N}_p(\mathbf{0}, \sigma^2_{\boldsymbol{\beta}}\mathbf{I}), \qquad \text{and}$$

$$\mathbf{u}|\mathbf{G} \sim \text{N}_{mb}(\mathbf{0}, \mathbf{G}),$$

where $\mathbf{X}$ is $n \times p$, $\mathbf{Z}$ is $n \times mb$ and $\boldsymbol{\Sigma}_{\mathbf{uu}}$ is $mb \times mb$ and $\boldsymbol{\Psi}$ is $b \times b$. The covariance of $\mathbf{G} = \text{Cov}(\mathbf{u}) \equiv \text{blockdiag}_{1 \leq i \leq m}(\boldsymbol{\Sigma}_{\mathbf{uu}}) \equiv \mathbf{I}_m \otimes \boldsymbol{\Sigma}_{\mathbf{uu}}$. Inverse Wishart$(\boldsymbol{\Psi}, v)$ denotes the probability distribution

$$\frac{|\boldsymbol{\Psi}|^{\frac{v}{2}}}{2^{\frac{vp}{2}}\Gamma_p\left(\frac{v}{2}\right)}|\mathbf{X}|^{-\frac{v+p+1}{2}} \exp\left[-\tfrac{1}{2}\text{tr}(\boldsymbol{\Psi}\mathbf{X}^{-1})\right]$$

where $\Gamma_p(x)$ denotes the multivariate gamma function and $\mathbf{tr}$ is the trace function.

The covariance matrix of random effects $\boldsymbol{\Sigma}$ will depend on the mixed model being fit. In the random intercept case, $\boldsymbol{\Sigma} = \sigma^2_u\mathbf{I}$ while in the random slopes case

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma^2_{\mathbf{u}_1} & \rho_{\mathbf{u}_1\mathbf{u}_2}\sigma_{\mathbf{u}_1}\sigma_{\mathbf{u}_2} \\ \rho_{\mathbf{u}_1\mathbf{u}_2}\sigma_{\mathbf{u}_1}\sigma_{\mathbf{u}_2} & \sigma^2_{\mathbf{u}_2} \end{pmatrix}$$

where $\sigma^2_{\mathbf{u}_1}$ is the variance of the random intercepts, $\sigma^2_{\mathbf{u}_2}$ is the variance of the random slopes and $\rho_{\mathbf{u}_1\mathbf{u}_2}$ is the correlation between the random intercepts and random slopes.

In the spline case, we use the cubic spline basis

$$\left\{1, x, x^2, x^3, (x-\kappa_1)^3_+, \ldots, (x-\kappa_K)^3_+\right\},$$

where K is the number of knots. Here $\mathbf{\Sigma}$ is a $K+2$ banded matrix. Banded matrices are highly sparse, and matrix operations can be performed on them in $\mathcal{O}(K)$ time. The matrix $\mathbf{\Sigma}$ is symmetric. For $K=3$, the contents of $\mathbf{\Sigma}$ are

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma^2_{\text{intercept}} & \rho_{\text{intercept}x} & \rho_{\text{intercept}x^2} & \rho_{\text{intercept}x^3} & 0 & 0 & 0 \\ \rho_{\text{intercept}x} & \sigma^2_x & \rho_{xx^2} & \rho_{xx^3} & \rho_{x(x-\kappa_1)^3_+} & 0 & 0 \\ \rho_{\text{intercept}x^2} & \rho_{xx^2} & \sigma^2_{x^2} & \rho_{x^2x^3} & \rho_{x^2(x-\kappa_1)^3_+} & \rho_{x^2(x-\kappa_2)^3_+} & 0 \\ \rho_{\text{intercept}x^3} & \rho_{xx^3} & \rho_{x^2x^3} & \sigma^2_{x^3} & \rho_{x^3(x-\kappa_1)^3_+} & \rho_{x^3(x-\kappa_2)^3_+} & \rho_{x^3(x-\kappa_3)^3_+} \\ 0 & \rho_{x(x-\kappa_1)^3_+} & \rho_{x^2(x-\kappa_1)^3_+} & \rho_{x^3(x-\kappa_1)^3_+} & \sigma^2_{(x-\kappa_1)^3_+} & \rho_{(x-\kappa_1)^3_+(x-\kappa_2)^3_+} & \rho_{(x-\kappa_1)^3_+(x-\kappa_3)^3_+} \\ 0 & 0 & \rho_{x^2(x-\kappa_2)^3_+} & \rho_{x^3(x-\kappa_2)^3_+} & \rho_{(x-\kappa_1)^3_+(x-\kappa_2)^3_+} & \sigma^2_{(x-\kappa_2)^3_+} & \rho_{(x-\kappa_2)^3_+(x-\kappa_3)^3_+} \\ 0 & 0 & 0 & \rho_{x^3(x-\kappa_3)^3_+} & \rho_{(x-\kappa_1)^3_+(x-\kappa_3)^3_+} & \rho_{(x-\kappa_2)^3_+(x-\kappa_3)^3_+} & \sigma^2_{(x-\kappa_3)^3_+} \end{pmatrix}.$$

4.2.3. Variational Bayes approximation to the zero-inflated Poisson model.

We choose a factored variational approximation for the model of the form

$$q(\boldsymbol{\nu}, \sigma^2_{\mathbf{u}}, \mathbf{r}_0, \rho) = q(\boldsymbol{\nu}) q(\mathbf{\Sigma}_{\mathbf{uu}}) q(\mathbf{r}_0) q(\rho),$$

where we define $\mathbf{r}_0 = \{r_i : y_i = 0\}$, and

$$\begin{aligned} q(\boldsymbol{\nu}) &= \text{N}(\boldsymbol{\mu}, \mathbf{\Lambda}), \\ q(\sigma^2_{\mathbf{u}}) &= \text{Inverse Wishart}\left(\mathbf{\Psi} + \sum_{i=1}^{m} (\boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top + \mathbf{\Lambda}_{\mathbf{u}_i \mathbf{u}_i}), v + m\right), \text{ and} \\ q(r_i) &= \text{Bernoulli}(p_i), \end{aligned}$$

with

$$p_i = \begin{cases} \text{expit}\left[\psi(\alpha_{q(\rho)}) - \psi(\beta_{q(\rho)}) - \exp\left(c_i^\top \boldsymbol{\mu} + \frac{1}{2} c_i^\top \mathbf{\Lambda} c_i\right)\right], & \text{when } \mathbf{y}_i = 0 \text{ ; and} \\ 1, & \text{otherwise.} \end{cases}$$

The approximation for $\mathbf{r}$ is given by

$$
\begin{aligned}
q(\mathbf{r}) \quad &\propto \exp\left[\mathbb{E}_{-q(\mathbf{r})}\mathbf{y}^\top\mathbf{R}(\mathbf{C}\boldsymbol{\mu}) - \mathbf{r}^\top \exp\left(\mathbf{C}\boldsymbol{\nu}\right) - \tfrac{1}{2}\boldsymbol{\nu}^\top\boldsymbol{\Sigma}_{\mathbf{uu}}\boldsymbol{\nu}\right] \\
&= \exp\left\{\mathbf{y}^\top\mathbf{R}\mathbf{C}\boldsymbol{\mu} - \mathbf{r}^\top \exp\left[\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\right]\right\}.
\end{aligned} \tag{38}
$$

We observe that this expression is close in form to the likelihood of a Poisson regression model with random effects. Poisson regression models are non-conjugate with normal priors, and hence the mean field updates for the regression parameters do not have closed form expressions. But by assuming a multivariate normal distribution for the regression coefficients parameterised by $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, the model can still be fit using a Gaussian Variational Approximation for $\boldsymbol{\beta}$ and $\mathbf{u}$ jointly. Techniques for efficiently fitting these models are described in Ormerod and Wand (2012); Challis and Barber (2013); Opper and Archambeau (2009). GVA has also been shown to have good asymptotic properties in Hall et al. (2011). The model can be fit using Algorithm 2 below. The derivation of the variational lower bound is given in Appendix 4.A.

4.3. Optimising the approximation over the regression coefficients

The most computationally and numerically difficult part of Algorithm 2 above is optimising the mean and covariance of the Gaussian approximation to the regression coefficients $[\boldsymbol{\beta}, \mathbf{u}]^\top$. In this section, we compare the accuracy, stability and speed of four different algorithms for fitting the Gaussian component of our model, $q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ in Algorithm 2. We compare these approaches for accuracy, computational complexity and stability. The measure of accuracy we use to assess our approximations is

Algorithm 2 Iterative scheme for obtaining the parameters in the optimal densities $q^*(\boldsymbol{\mu}, \boldsymbol{\Lambda})$, $q^*(\boldsymbol{\Sigma}_{\mathbf{uu}})$ and $q^*(\rho)$

Require: $\alpha_{q(\rho)} \leftarrow \alpha_\rho + \mathbf{1}^\top \mathbf{p}, p_{q(\boldsymbol{\Sigma}_{\mathbf{uu}})} \leftarrow p + 1$

while the increase in $\log \underline{p}(\mathbf{y}; q)$ is significant **do**

Optimise $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ using $\mathbf{y}, \mathbf{C}, \mathbf{p}$ and $\boldsymbol{\Sigma}_{\mathbf{uu}}$

$\beta_{q(\rho)} \leftarrow \beta_\rho + n - \mathbf{1}^\top \mathbf{p}$

$\boldsymbol{\eta} \leftarrow -\exp\left[\mathbf{C}\boldsymbol{\mu} + \frac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\right] + (\psi(\alpha_{q(\rho)}) - \psi(\beta_{q(\rho)}))\mathbf{1}_n$

$\mathbf{p}_{q(\mathbf{r}_0)} \leftarrow \text{expit}(\boldsymbol{\eta})$

$\boldsymbol{\Psi}_{q(\boldsymbol{\Sigma}_{\mathbf{uu}})} \leftarrow \boldsymbol{\Psi} + \sum_{i=1}^m (\boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top + \boldsymbol{\Lambda}_{\mathbf{u}_i \mathbf{u}_i})$

$\boldsymbol{\Sigma}_{\mathbf{uu}} \leftarrow [\boldsymbol{\Psi}_{q(\boldsymbol{\Sigma}_{\mathbf{uu}})}/(v - d - 1)]^{-1}$

end while

$$\text{Accuracy} = 1 - \tfrac{1}{2} \int \|f(\boldsymbol{\theta}) - g(\boldsymbol{\theta})\| d\boldsymbol{\theta}$$

where $f(\boldsymbol{\theta})$ is the true distribution we're approximating and $g(\boldsymbol{\theta})$ is the approximating distribution.

Our first attempts at implementation of some of these algorithms were prone to numerical stability problems when initialised from some starting points. We also discuss the modifications we made to these algorithms to enhance their numerical stability.

4.3.1. Laplace-Variational Approximation. The Laplace-Variational Approximation method is based on Laplace's method of approximating integrals, as introduced in Section 1.6.3. The variational lower bound is approximated by a Gaussian centred at its mode.

This yields the following approximation to the variational lower bound

$$\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y}) \approx \mathbf{y}^\top \mathbf{P}\mathbf{C}\boldsymbol{\mu} - \mathbf{p}^\top \exp\left(\mathbf{C}\boldsymbol{\mu}\right) - \tfrac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}. \tag{39}$$

This expression can be iteratively optimised with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ using the Newton-Raphson method, with the derivatives for $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ given in Appendix 4.B.1. The steps of the algorithm are shown in Algorithm 3.

Upon implementing the algorithm and performing numerical experiments, we observed numerical issues which had to be dealt with in order for the algorithm to successfully complete. We implemented checks for error conditions, and steps to recover from the error conditions should they arise.

If during an iteration of the Laplace-Variational approximation algorithm the inversion of $\boldsymbol{\Lambda}$ fails, or the diagonal elements of $\boldsymbol{\Lambda}$ become negative when $\boldsymbol{\Lambda}$ must be positive-definite, then $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ were reverted to the previous iteration's $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ values and the algorithm was terminated.

If after calculating the gradient of the Gaussian Variational lower bound with respect to $\boldsymbol{\mu}$, any of its' elements were driven to NaN or ∞ due to numeric overflow followed by matrix inversion during the computation, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ were reverted to the previous iteration's $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ values and the algorithm was terminated.

4.3.2. Gaussian Variational Approximation. The full variational likelihood for a Generalised Linear Mixed model is computationally difficult to calculate, requiring the evaluation of a high dimensional integral. However, Ormerod and Wand (2012) devised an accurate approximation to the full variational likelihood, the Gaussian Variational Lower Bound, which only requires the evaluation of a substantially simpler univariate integral.

Algorithm 3 Laplace scheme for optimising $\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y})$

Require: $\mathbf{C}, \boldsymbol{\Sigma}, \mathbf{p}, \mathbf{y}$ set as in Algorithm 2.

$\boldsymbol{\mu} \leftarrow \mathbf{0}$

$\mathbf{H} \leftarrow [-\mathbf{C}^\top \text{diag}(\mathbf{p} e^{(\mathbf{C}\boldsymbol{\mu})})\mathbf{C} - \boldsymbol{\Sigma}^{-1}]^{-1}$

while the increase in $\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y})$ is significant **do**

$\boldsymbol{\Lambda} \leftarrow \left[\mathbf{P}\mathbf{C}^\top \text{diag}(\exp(\mathbf{C}\boldsymbol{\mu}))\mathbf{C} + \boldsymbol{\Sigma}^{-1}\right]^{-1}$

If $\boldsymbol{\Lambda}$ cannot be inverted, or any diagonal element of $\boldsymbol{\Lambda}$ is negative, revert to previous $\boldsymbol{\Lambda}$ and break

$\mathbf{H} \leftarrow [-\mathbf{C}^\top \text{diag}(\mathbf{p} e^{(\mathbf{C}\boldsymbol{\mu})})\mathbf{C} - \boldsymbol{\Sigma}^{-1}]^{-1}$

If any element of $\mathbf{H}$ is NaN or ∞, break

$\mathbf{g} \leftarrow \mathbf{C}^\top[\mathbf{r}^\top \mathbf{y} - \mathbf{r}^\top \exp(\mathbf{C}^\top \boldsymbol{\mu})] - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$

$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \boldsymbol{\Lambda} g$

end while

To optimise the Gaussian component of the lower bound in each iteration of Algorithm 2, optimal $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ values must be found while keeping the other variational parameters fixed. The variational lower bound is not necessarily unimodal if $\mathbf{p}$ and $\boldsymbol{\Sigma}$ are free to vary, leading to potential problem of optimising to a local rather than the global maximum. However, for fixed $\mathbf{p}$ and $\boldsymbol{\Sigma}$, the variational lower bound is log-concave with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, and so standard optimisation methods such as L-BFGS-B as described in, for example, Liu and Nocedal (1989) and Nocedal and Wright (2006), work well. This leads to an extremely accurate approximation of the posterior probability estimated by MCMC at the expense of some additional computational effort. Care must be taken in the parameterisation of $\boldsymbol{\Lambda}$, as it is both of high dimension $(p + mb)^2$ and constrained to be semi-

positive definite. We present and compare two approaches to parameterising the covariance matrix $\boldsymbol{\Lambda}$ below.

4.3.2.1. *Covariance parameterisation* $\boldsymbol{\Lambda} = \mathbf{R}^\top\mathbf{R}$. This parameterization has been used elsewhere, e.g., Pinheiro and Bates (2000). We fit the Gaussian component of our approximation in Algorithm 2 by maximising the variational lower bound is

$$\begin{aligned}\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y}) &= \mathbf{y}^\top\mathbf{P}\mathbf{C}\boldsymbol{\mu} - \mathbf{p}^\top \exp\{\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\} \\ &\quad -\tfrac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \tfrac{1}{2}\text{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}) + \tfrac{1}{2}\log|\boldsymbol{\Lambda}| + \tfrac{1}{2}\log|\boldsymbol{\Sigma}^{-1}| + \tfrac{p}{2},\end{aligned} \tag{40}$$

with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, keeping $\mathbf{p}$, $\boldsymbol{\Sigma}$ and ρ fixed.

The first variant of the GVA algorithm that we present optimises the Gaussian variational lower bound of the log likelihood with respect to $\boldsymbol{\mu}$ and the Cholesky decomposition $\mathbf{R}$ of $\boldsymbol{\Lambda}$, that is, $\boldsymbol{\Lambda} = \mathbf{R}^\top\mathbf{R}$. This ensures that $\boldsymbol{\Lambda}$ remains positive definite, and reduces the number of parameters we have to optimise over in order to optimise $\boldsymbol{\Lambda}$ to the $(p+1)p/2$, as $\mathbf{R}$ is lower triangular. We refer to this as the covariance parameterisation. The resulting lower bound is

$$\begin{aligned}\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y}) &= \mathbf{y}^\top\mathbf{P}\mathbf{C}\boldsymbol{\mu} - \mathbf{p}^\top \exp\{\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\} \\ &\quad -\tfrac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \tfrac{1}{2}\text{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}) + \log|\mathbf{R}| + \tfrac{1}{2}\log|\boldsymbol{\Sigma}^{-1}| + \tfrac{p}{2},\end{aligned} \tag{41}$$

which can be optimised with L-BFGS-B using the derivatives in Appendix 4.B.2.

4.3.2.2. *Precision parameterisation* $\boldsymbol{\Lambda} = (\mathbf{R}^\top\mathbf{R})^{-1}$. The second variant of the GVA algorithm is similar to the first, but instead of optimising the Gaussian variational lower bound with respect to $\boldsymbol{\mu}$ and the Cholesky factor $\mathbf{R}$ of $\boldsymbol{\Lambda}$, we instead optimise the Cholesky factor of the inverse of $\boldsymbol{\Lambda}$, i.e., $\boldsymbol{\Lambda} = (\mathbf{R}\mathbf{R}^\top)^{-1}$.

The Gaussian variational lower bound in this parameterisation is

$$\begin{aligned}\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y}) &= \mathbf{y}\mathbf{P}\mathbf{C}\boldsymbol{\mu} - \mathbf{p}^\top \exp\{\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\mathrm{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\} \\ &\quad -\tfrac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \tfrac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}) + \tfrac{1}{2}\log|\boldsymbol{\Sigma}^{-1}| + \tfrac{p}{2} - \log|\mathbf{R}|.\end{aligned} \tag{42}$$

The derivative with respect to $\boldsymbol{\mu}$ is the same as that in the first variant of the algorithm, but as the parameterisation of $\boldsymbol{\Lambda}$ has changed, the derivative with respect to $\boldsymbol{\Lambda}$ is

$$\begin{aligned}\frac{\partial \log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y})}{\partial \boldsymbol{\Lambda}} &= (\boldsymbol{\Lambda}^{-1} + \mathbf{H})(-\boldsymbol{\Lambda}\mathbf{R}\boldsymbol{\Lambda}) \\ &= -(\mathbf{I} + \mathbf{H}\boldsymbol{\Lambda})\mathbf{R}\boldsymbol{\Lambda} \\ &= -(\mathbf{R}\boldsymbol{\Lambda} + \mathbf{H}\boldsymbol{\Lambda}\mathbf{R}\boldsymbol{\Lambda}),\end{aligned} \tag{43}$$

where $\mathbf{H} = (\mathbf{P}\mathbf{C})^\top \mathrm{diag}(\exp(\mathbf{C}\boldsymbol{\mu} + \frac{1}{2}\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top))\mathbf{P}\mathbf{C} - \boldsymbol{\Sigma}^{-1}$.

4.3.2.3. *GVA fixed point.* This variant of the algorithm uses Newton-Raphson-like fixed point updates on the Gaussian variational lower bound. We optimise the same variational lower bound as in the covariance parameterisation above, using the derivatives below. The steps are detailed in Algorithm 4 where the derivatives are as presented in Appendix 4.B.3. As this algorithm involves a simple Newton-Raphson style update step, it is computationally simple to implement, but potentially unstable as there is no adaptation of step size, as in L-BFGS-B.

For efficiency, the inversion of $\boldsymbol{\Lambda}$ within the algorithm was implemented using the block inverse formula, where the matrix was partitioned

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{\boldsymbol{\beta}\boldsymbol{\beta}} & \boldsymbol{\Lambda}_{\boldsymbol{\beta}\mathbf{u}} \\ \boldsymbol{\Lambda}_{\boldsymbol{\beta}\mathbf{u}}^\top & \boldsymbol{\Lambda}_{\mathbf{u}\mathbf{u}} \end{pmatrix}$$

with $\boldsymbol{\Lambda}_{\boldsymbol{\beta\beta}}$ the $p \times p$ approximation of the fixed effects covariance, $\boldsymbol{\Lambda}_{\boldsymbol{\beta}\mathbf{u}}$ the $p \times mb$ approximation of the covariances between the fixed and random effects and $\boldsymbol{\Lambda}_{\mathbf{uu}}$ the $mb \times mb$ approximation of the random effects covariance.

Sometimes in the course of executing the algorithm, we observed numerical issues which had to be dealt with in order for the algorithm to successfully complete. If the block $\boldsymbol{\Lambda}_{\mathbf{uu}}$ could not be inverted on an iteration, we reverted to $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ from the previous iteration. If after updating $\boldsymbol{\mu}$ any element of the vector was NaN, we reverted to the $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ from the previous iteration. This greatly improved the numerical stability of the algorithm.

Algorithm 4 The GVA Newton-Raphson fixed point iterative scheme for obtaining the optimal $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ given $\mathbf{y}$, $\mathbf{C}$ and $\mathbf{p}$.

Require: $g = \mathbf{PC}(\mathbf{y} - \mathbf{C}^\top \exp(\mathbf{C}\boldsymbol{\mu} + \frac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top))) - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$.

while the increase in $\log \underline{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \mathbf{y})$ is significant **do**

$\mathbf{g} \leftarrow \mathbf{C}^\top \mathbf{p}\{\mathbf{y} - [\exp(\mathbf{C}\boldsymbol{\mu} + \frac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top))]\} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$

$\mathbf{H} \leftarrow -\mathbf{C}^\top \text{diag}(\mathbf{p}^\top \exp(\mathbf{C}\boldsymbol{\mu} + \frac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top))) - \boldsymbol{\Sigma}^{-1}$

$\boldsymbol{\Lambda} \leftarrow (-\mathbf{H})^{-1}$ using block inversion on $\mathbf{H}$

If the inversion fails, revert to previous $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ and exit the loop

$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{g}$

If any element of $\boldsymbol{\mu}$ is ∞ or NaN, revert to previous $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ and exit the loop

end while

4.4. Parameterisations for Gaussian Variational Approximation

4.4.1. Covariance parameterisation of $\mathbf{\Lambda}$. To ensure symmetry of $\mathbf{\Lambda}$, we parameterise the covariance matrix in terms of $\mathbf{\Lambda}$'s Cholesky factor $\mathbf{R}$. We optimise over the space $(\boldsymbol{\mu}, \overline{\mathbf{R}})$, where $\boldsymbol{\mu} \in \mathbb{R}^{p+m}b$ and $\overline{\mathbf{R}}$ is a lower-triangular $(p + mb) \times (p + mb)$ matrix. Then

$$\mathbf{R}_{ij} = \begin{cases} \exp(\overline{\mathbf{R}}_{ij}), & i = j; \\ \overline{\mathbf{R}}_{ij}, & i > j; \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

We exponentiate the diagonal to ensure positive-definiteness of $\mathbf{R}$. We parameterise $\mathbf{\Lambda}$ as $\mathbf{\Lambda} = \mathbf{R}\mathbf{R}^\top$ so that is is guaranteed to be symmetric, and the number of parameters is reduced from p^2 to $p(p-1)/2$, some of which are constrained.

This parameterisation can lead to numeric overflow when the diagonals of $\overline{\mathbf{R}}$ become moderately large, which can lead to singular matrices when attempting to invert $\mathbf{\Lambda}$. We addressed this issue by defining a new parameterisation using the piecewise function below, which is exponential for arguments less than a threshold t, and quadratic for arguments greater than or equal to t

$$f(r_{ij}) = \begin{cases} e^{r_{ij}}, & r_{ij} < t; \quad \text{and} \\ ar_{ij}^2 + br_{ij} + c, & r_{ij} \geq t; \end{cases} \tag{44}$$

and then choosing the coefficients a, b and c such that the function, first and second derivatives would agree at $r_{ij} = t$. This ensured that the function did not grow

too quickly as the parameters varied, mitigating the issue of numerical overflow for this parameterisation.

To find the coefficients a, b and c for the above function, we solved the system of equations presented below formed by repeatedly differentiating the quadratic at $r_{ij} = t$ and equating it with e^t we have

$$\begin{aligned} e^t &= at^2 + bt + c \\ e^t &= 2at + b \\ e^t &= 2a \end{aligned} \tag{45}$$

to obtain $a = e^t/2$, $b = (1-t)e^t$ and $c = [1 - t^2/2 - (1-t)t]e^t$.

We also addressed the problem of numeric overflow by working with the Cholesky factorisation of $\mathbf{\Lambda}^{-1}$ rather than $\mathbf{\Lambda}$, allowing us to solve a system of equations rather than invert and multiply by a matrix, which is also faster and more numerically stable. We used knowledge of the regression model we are fitting to specify a sparse matrix structure, greatly reducing the dimension of the problem and thus improving both computational speed and numeric accuracy.

Recall that any symmetric matrix $\mathbf{\Lambda}$ can be written as a product of its Cholesky factors, $\mathbf{\Lambda} = \mathbf{R}\mathbf{R}^\top$ where $\mathbf{R}$ is lower triangular. The matrix $\mathbf{R}$ is unique if $\mathbf{R}_{ii} \geq 0$

and

$$\begin{pmatrix} \mathbf{R}_{11} & 0 & 0 \\ \mathbf{R}_{21} & \mathbf{R}_{22} & 0 \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{pmatrix} \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{21} & \mathbf{R}_{31} \\ 0 & \mathbf{R}_{22} & \mathbf{R}_{32} \\ 0 & 0 & \mathbf{R}_{33} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{R}_{11}^2 & & \text{symmetric} \\ \mathbf{R}_{21}\mathbf{R}_{11} & \mathbf{R}_{21}^2 + \mathbf{R}_{22}^2 & \\ \mathbf{R}_{31}\mathbf{R}_{11} & \mathbf{R}_{31}\mathbf{R}_{21} + \mathbf{R}_{32}\mathbf{R}_{22} & \mathbf{R}_{31}^2 + \mathbf{R}_{32}^2 + \mathbf{R}_{33}^2 \end{pmatrix}.$$

We exploit this structure, by interchanging the fixed and random effects in the design matrix $\mathbf{C} = [\mathbf{X}, \mathbf{Z}]$ to $\mathbf{C} = [\mathbf{Z}, \mathbf{X}]$, and re-ordering the dimensions of $\boldsymbol{\mu}, \boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}$ in the same manner, using the independence between the blocks relating to the random effects in $\mathbf{Z}$ to induce sparsity in the Cholesky factor $\mathbf{R}$ of $\boldsymbol{\Lambda}^{-1}$, as can be seen in Figures 1 and 2. Thus the Gaussian $q(\boldsymbol{\nu}) \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ can be optimised over a space of dimension $\frac{1}{2}p(p+1) + pq + \frac{1}{2}q(q+1)$ rather than dimension $\frac{1}{2}(p+mq)(p+mq+1)$ as in the dense parameterisation. This leads to substantial performance gains when m is large, as is typically the case in problems of practical importance such as longitudinal or clinical trials with many subjects or the application presented in Section 4.6.

By re-ordering the fixed and random effects in $\boldsymbol{\Lambda}$, we end up with a covariance structure which is sparse in the first diagonal block.

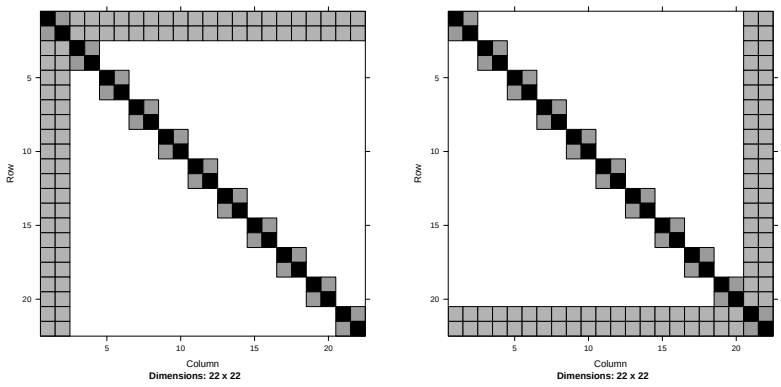

FIGURE 1. Inverse Covariance matrix of approximate posterior for $\boldsymbol{\nu}$ – Fixed effects before random effects and random before fixed effects.

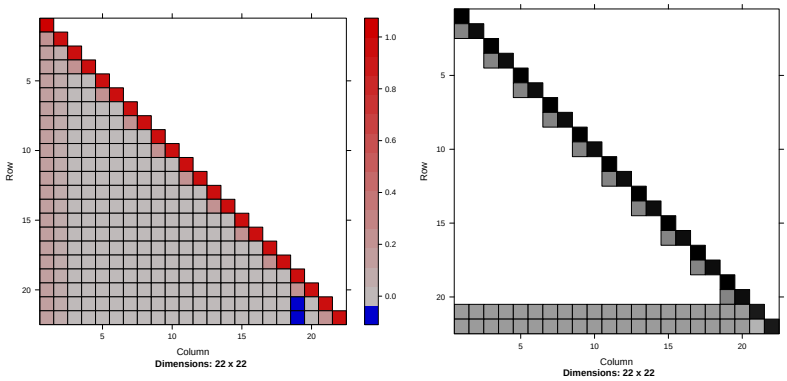

FIGURE 2. Cholesky factor of Inverse Covariance matrix of approximate posterior for $\boldsymbol{\nu}$ – Fixed effects before random effects and random before fixed effects.

4.4.2. Precision parameterisation. The GVA is fit by maximising the Gaussian Variational Lower Bound, which is parameterised by a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Lambda}$. The simplest parameterisation of $\boldsymbol{\mu}$ is the natural parameterisation, but the covariance matrix has many possible parameterisations. Covariance matrices are positive semi-definite, and hence symmetric, so they have

a unique Cholesky factorisation. Parameterising the covariance matrix in terms of the Cholesky factor allows us to represent the square covariance matrix using only a lower triangular matrix with half as many non-zero elements. Thus the Cholesky factor is a convenient way to parameterising covariance matrices.

Another advantage of parameterising using the precision matrix is that the covariance matrix contains the marginal covariances between the elements of $\boldsymbol{\nu}$, while the precision matrix contains the conditional covariances between those elements. In generalised linear mixed models, fixed and random effects are conditionally independent, implying sparsity in the precision matrix although not necessarily in the covariance matrix.

The variational lower bound of a GVA takes the form given below.

$$\begin{aligned} \log \underline{p}(\mathbf{y};\boldsymbol{\mu},\boldsymbol{\Lambda}) = \; & \mathbf{y}^\top \mathbf{C}\boldsymbol{\mu} - \mathbf{1}^\top B(\mathbf{C}\boldsymbol{\mu}, \mathrm{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)) + \mathbf{1}^\top c(\mathbf{y}) \\ & -\tfrac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \tfrac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}) \\ & +\tfrac{1}{2}\log|\boldsymbol{\Lambda}| - \tfrac{1}{2}\log|\boldsymbol{\Sigma}| + \tfrac{d}{2}. \end{aligned} \tag{46}$$

Let $\boldsymbol{\Omega} = \boldsymbol{\Lambda}^{-1}$, the precision matrix. Then if we reparameterise the variational lower bound in terms of $\boldsymbol{\Omega}$ we obtain the function below.

$$\begin{aligned} F(\boldsymbol{\Omega}) = \; & \mathbf{y}^\top \mathbf{C}\boldsymbol{\mu} - \mathbf{1}^\top B(\mathbf{C}\boldsymbol{\mu}, \mathrm{diag}(\mathbf{C}\boldsymbol{\Omega}^{-1}\mathbf{C}^\top)) + \mathbf{1}^\top c(\mathbf{y}) \\ & -\tfrac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \tfrac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Omega}^{-1}) \\ & -\tfrac{1}{2}\log|\boldsymbol{\Omega}| - \tfrac{1}{2}\log|\boldsymbol{\Sigma}| + \tfrac{d}{2}. \end{aligned} \tag{47}$$

When the variational lower bound is optimised, by the first-order optimality conditions, $\frac{\partial F}{\partial \boldsymbol{\Omega}_{jk}} = \mathbf{0}$. Then using matrix calculus and the properties of the trace

operator

$$\begin{aligned}\frac{\partial F}{\partial \boldsymbol{\Omega}_{jk}} &= -\tfrac{1}{2}\text{tr}(\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}) + \tfrac{1}{2}\text{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}\boldsymbol{\Omega}^{-1}) \\ &\quad -\tfrac{1}{2}\text{tr}\{\mathbf{C}^\top \text{diag}(B^{(2)}(\mathbf{C}\boldsymbol{\mu}, \text{diag}(\mathbf{C}\boldsymbol{\Omega}^{-1}\mathbf{C}^\top)))\mathbf{C}\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}\boldsymbol{\Omega}^{-1}\} \\ &= -\tfrac{1}{2}[\text{tr}(\boldsymbol{\Omega}^{-1}\boldsymbol{\Omega}\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}) - \text{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}\boldsymbol{\Omega}^{-1}) \\ &\quad +\text{tr}\{\boldsymbol{\Omega}^{-1}\mathbf{C}^\top \text{diag}(B^{(2)}(\mathbf{C}\boldsymbol{\mu}, \text{diag}(\mathbf{C}\boldsymbol{\Omega}^{-1}\mathbf{C}^\top)))\mathbf{C}\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}\}] \\ &= -\tfrac{1}{2}\text{tr}\{\boldsymbol{\Omega}^{-1}[\boldsymbol{\Omega} - \mathbf{C}^\top \text{diag}(B^{(2)}(\mathbf{C}\boldsymbol{\mu}, \text{diag}(\mathbf{C}\boldsymbol{\Omega}^{-1}\mathbf{C}^\top)))\mathbf{C} - \boldsymbol{\Sigma}^{-1}]\boldsymbol{\Omega}^{-1}\tfrac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}}\}\end{aligned}$$

As $\boldsymbol{\Omega}^{-1} \neq \mathbf{0}$ and $\frac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\Omega}_{jk}} \neq \mathbf{0}$, this implies $\boldsymbol{\Omega} = \mathbf{C}^\top \text{diag}(B^{(2)})\mathbf{C} + \boldsymbol{\Sigma}^{-1}$. Thus the sparsity of $\boldsymbol{\Omega}$ depends on the structure of $\mathbf{C}$ and $\boldsymbol{\Sigma}$, which depends on the model specified.

We optimise over the space $(\boldsymbol{\mu}, \overline{\mathbf{R}})$ as in the Section 4.4, but now the elements of the Cholesky factor are parameterised as

$$\mathbf{R}_{ij} = \begin{cases} \exp(-\overline{\mathbf{R}}_{ij}), & i = j \\ \overline{\mathbf{R}}_{ij}, & i > j \\ 0, & \text{otherwise.} \end{cases}$$

This new choice of parameterisation allows us to calculate $\frac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)$ by solving the linear systems $\mathbf{R}\mathbf{a} = \mathbf{C}_i, i = 1, \ldots, n$ for $\mathbf{a}$ and then calculating $\mathbf{a}^\top\mathbf{a}$ where $\mathbf{C}_i =$ the ith row of $\mathbf{C}$, rather than calculating $\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)$ directly.

A final advantage of this parameterisation is its' greater numerical accuracy. Matrix multiplication and back substitution are both equally numerically accurate and stable - as shown in Golub and Van Loan (2013) §2.7.8 and §3.1.2 or Trefethen and Bau (1997) Lecture 17, and the precision matrix will be sparse due to the specification of the mixed model/conditional independence. This implies that

the numerical accuracy of the inversion will be higher as there are fewer non-zero entries in the Cholesky factor of the precision matrix than of the Cholesky factor of the covariance matrix. Thus parameterising the variational lower bound in terms of the precision matrix will have the same or higher numerical accuracy than parameterising in terms of the covariance matrix.

Finally, we have outlined several versions of the Gaussian variational approximation. Sometimes these give qualitatively different results and the question arises how we would choose amongst them. While we do not have theory to support this, our pragmatic advice is to choose a method we would choose the method (amongst those that converge) is the method that achieves the highest lower bound value.

4.5. Numerical results

The accuracy of each of the model fitting algorithms presented in Section 4.3 was assessed by comparing the approximating distribution of each parameter with the posterior distribution of MCMC samples of that parameter. One million MCMC samples were generated using `RStan` (Carpenter et al., 2016; Stan Development Team, 2016). The accuracy of examples using random intercept, random slope and spline models were evaluated using this method.

Algorithm	Mean (seconds)	Standard deviation (seconds)
Laplace's method	0.37	0.07
GVA covariance parameterisation	2.04	1.24
GVA precision parameterisation	0.38	0.66
GVA fixed point	0.05	0.07

TABLE 1. Table of results - Speed.

	Laplace's Method	GVA ($\mathbf{\Lambda} = \mathbf{R}\mathbf{R}^\top$)	GVA NP ($\mathbf{\Lambda} = (\mathbf{R}\mathbf{R}^\top)^{-1}$)	GVA FP
$\boldsymbol{\beta}_1$	85%	90%	91%	90%
$\boldsymbol{\beta}_2$	76%	98%	99%	99%
Mean of $\mathbf{u}$'s	81%	94%	94%	94%
$\sigma^2_{\mathbf{u}_1}$	66%	66%	66%	66%
ρ	99%	99%	99%	99%

TABLE 2. Table of accuracy - Random intercept model.

4.5.1. Simulated data. For each of these simulations, the model is as presented in Section 4.2. Several common application scenarios were simulated and their accuracy evaluated. A random intercept model was simulated with $\boldsymbol{\beta} = (2, 1)^\top$, $\rho = 0.5$, $m = 20$, $n_i = 10$ and $b = 1$. The results are presented in Table 2. A random slope model was simulated with $\boldsymbol{\beta} = (2, 1)^\top$, $\rho = 0.5$, $m = 20$, $n_i = 10$ and $b = 2$. The results are presented in Table 3. Spline model was fit to a data set generated from the function $3 + 3\sin(\pi x)$ on the interval $[-1, 1]$. The resulting model fits are presented in Figure 4.5.1.

To assess the speed of each approach, a test case was constructed of a random slope model with $m = 50$ groups, each containing $n_i = 100$ individuals. A model was then fit to this data set ten times using each algorithm, and the results averaged. These results are presented in Table 1.

The median accuracy of the algorithms was assessed by running them on 100 randomly generated data sets. The results are presented in Figure 3 and Figure 4.

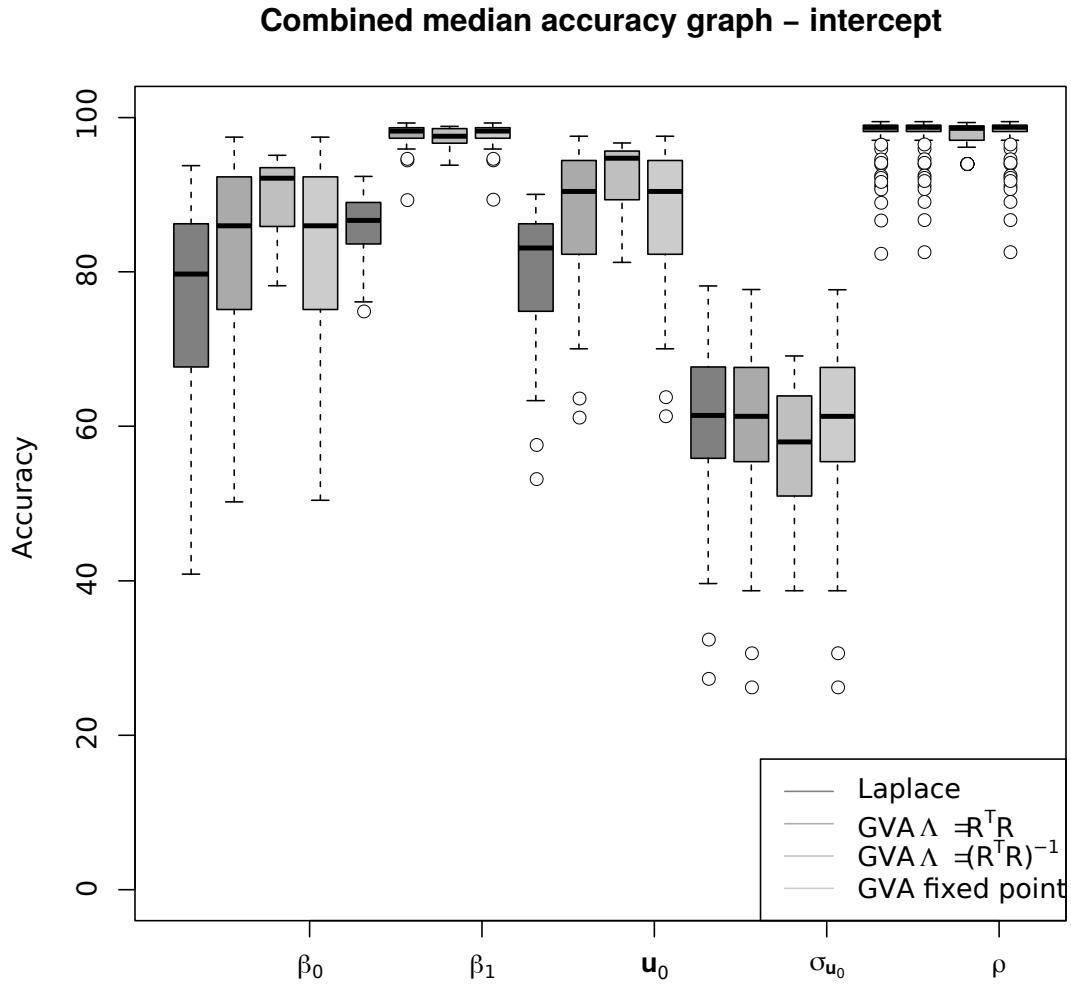

FIGURE 3. Boxplots of accuracies of the parameter estimates for a random intercept model after 100 repeated runs on simulated data. We see that the accuracy of the parameter estimates is quite stable, and the median accuracies are high.

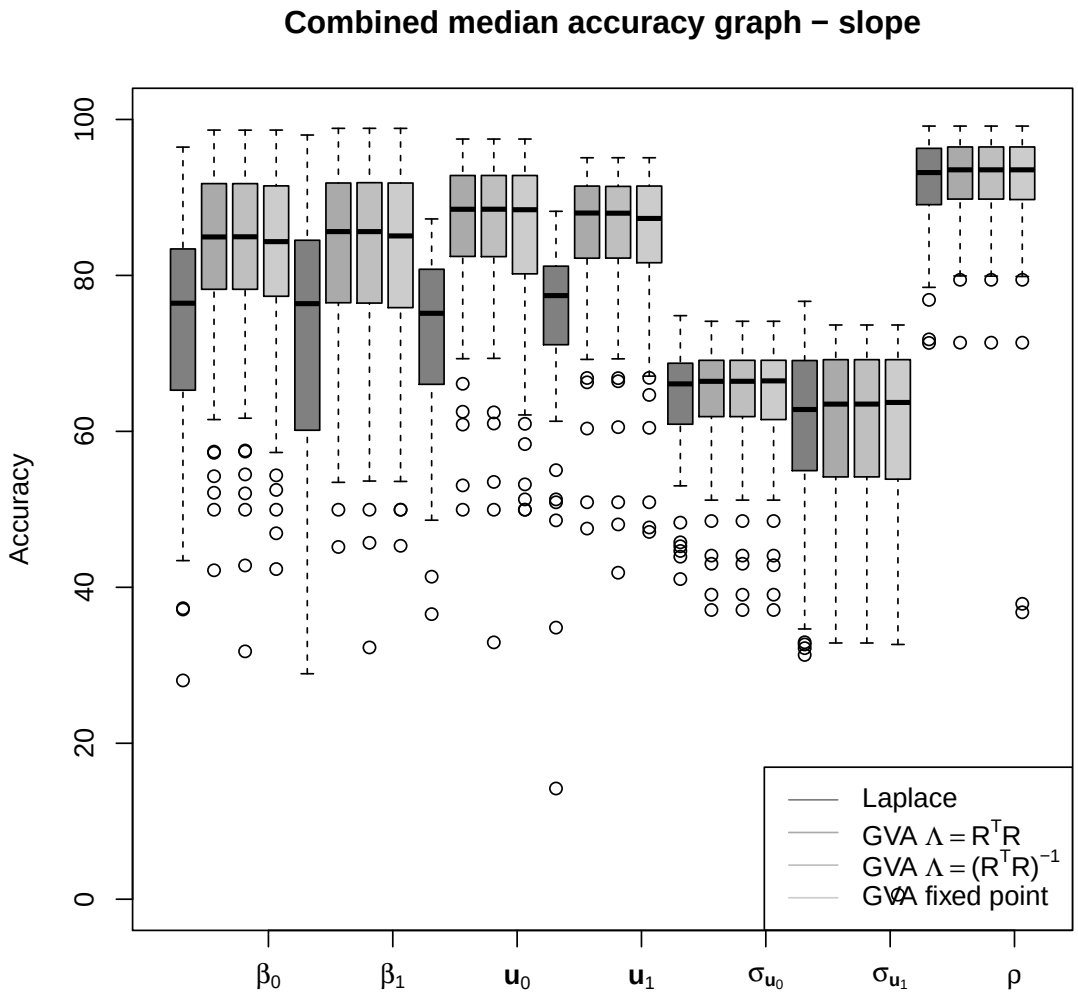

FIGURE 4. Boxplots of accuracies of the parameter estimates for a random slope model after 100 repeated runs on simulated data. We see that the accuracy of the parameter estimates is quite stable, and the median accuracies are high.

	Laplace's Method	GVA ($\mathbf{\Lambda} = \mathbf{R}\mathbf{R}^\top$)	GVA ($\mathbf{\Lambda} = (\mathbf{R}\mathbf{R}^\top)^{-1}$)	GVA FP
$\boldsymbol{\beta}_1$	67%	88%	88%	88%
$\boldsymbol{\beta}_2$	70%	89%	88%	89%
Mean of $\mathbf{u}$	70%	91%	91%	91%
$\sigma^2_{\mathbf{u}_1}$	71%	73%	73%	73%
$\sigma^2_{\mathbf{u}_2}$	68%	69%	69%	69%
ρ	91%	90%	90%	90%

TABLE 3. Table of accuracy - Random slope model.

4.5.2. Numerical stability of the parameterisation. The stability of this scheme was tested by calculating the accuracy of the approximations fit with a range of safe exponential thresholds, the results of which are presented in Figure 4.5.2. The variational approximation was found to be stable, with the accuracy largely insensitive to the choice of threshold.

We repeated our numerical experiments with the new parameterisation, varying the threshold within reasonable bounds and found that the numerical experiments no longer resulted in overflow, and that the numerical accuracy of the approximation was still very good.

The stability of the GVA algorithm with the parameterisation $\mathbf{\Lambda} = (\mathbf{R}^\top\mathbf{R})^{-1}$ depends on the threshold chosen for the safe exponential function. When the threshold is set to 2, the algorithm is stable for all starting points within the grid except 6. When the threshold is set to ∞, equivalent to using the naive $\exp$ parameterisation, the algorithm encounters numerical errors for every starting point on the grid.

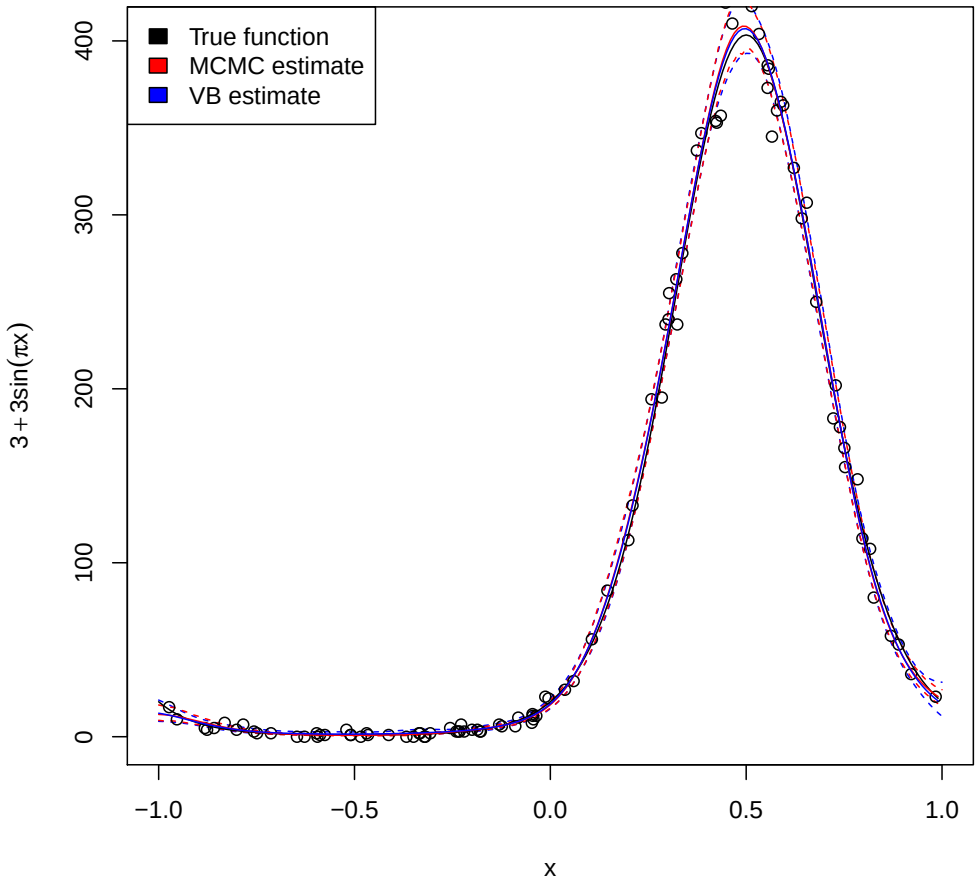

FIGURE 5. Comparison of VB and MCMC spline fits with the true function.

4.5.3. Stability of the GVA precision parameterisation algorithm for different starting points. The numerical stability of each fitting algorithm in Section 4.3 was assessed by initialising each algorithm from a range of different starting points. Errors due to numerical instability and the fitted $\boldsymbol{\mu}$ were recorded for each starting point.

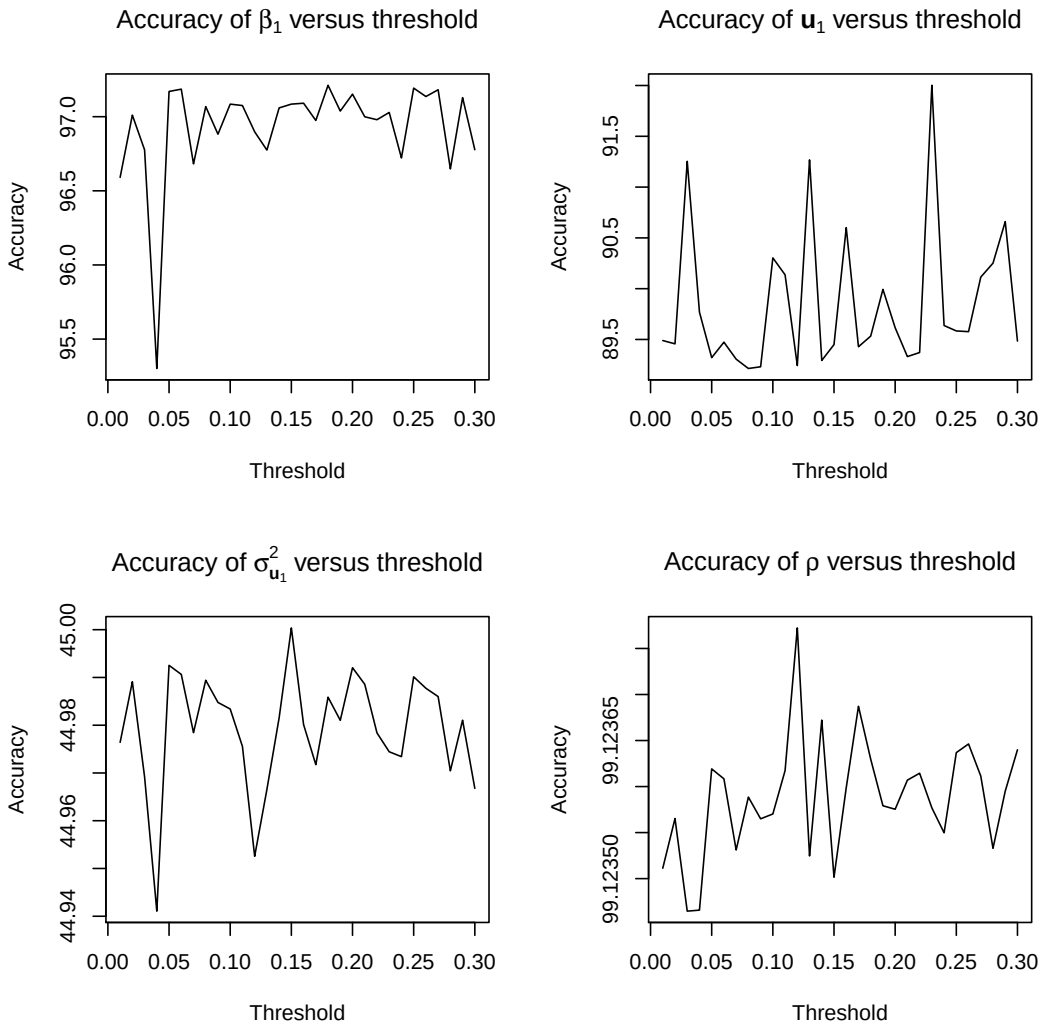

FIGURE 6. Accuracy of approximation of parameters versus the safe exponential threshold.

A data set of 100 individuals in ten groups ($m = 10$) was generated from a model with a fixed intercept and slope, and a random intercept. $\boldsymbol{\mu}$ was initialised from a grid of points on the interval $[-4.5, 5]$ for intercept and slope, spaced 0.1 apart. The error counts are presented in Table 4. Plots of the starting locations

Algorithm	Error count
Laplace's algorithm	12
GVA $\mathbf{\Lambda} = \mathbf{R}^\top \mathbf{R}$	1306
GVA $\mathbf{\Lambda} = (\mathbf{R}^\top \mathbf{R})^{-1}$	6
GVA NR fixed point	992

TABLE 4. Count of numerical errors for each algorithm during stability tests.

which resulted in numerical errors when the fitting algorithm was run are presented in 4.5.3.

The GVA algorithm with the $\mathbf{\Lambda} = (\mathbf{R}\mathbf{R}^\top)^{-1}$ parameterisation was less prone to instability due to starting point when the safe exponential parameterisation was used then when it was not used, as can be seen from Figure 4.5.3.

4.5.4. Stability of the GVA fixed point algorithm for different starting points. The naive fixed point algorithm was extremely unstable for many starting points, as can be seen from Figure 4.5.4. The variant of the algorithm which checked whether the inversion of the $\mathbf{\Lambda}_{\mathbf{uu}}$ block of $\mathbf{\Lambda}$ was performed successfully was much more stable, and did not suffer from any numeric errors at all over the range of starting points we tested. The algorithm is able to abort safely, and allow the Variational Bayes algorithm to update the other parameters before trying to fit the Gaussian component of the model again until the correct parameters are accurately estimated.

4.6. Applications

We now present numerical results for the application of our model fitting algorithms to several publicly available data sets.

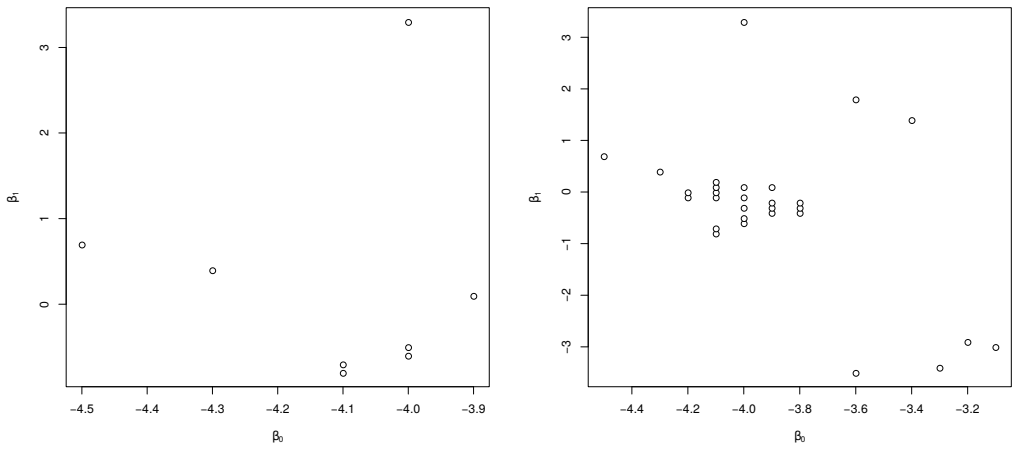

FIGURE 7. Starting locations which caused the GVA fitting algorithm to fail with numeric errors. The true model had fixed parameters $\boldsymbol{\beta} = (2, 1)^\top$ and random intercepts. There were ten groups in the hierarchical model each with ten individuals $(m = 10, n_i = 10)$. In the left figure the starting points which lead to numeric errors when the safe exponential was used are shown, while in the right figure the starting points which lead to numeric errors when the safe exponential was not used are plotted.

4.6.1. Poisson example without zero-inflated component – Police stops. The data set used for this example was the police stop example from Chapter 15 of Gelman and Hill (2007). The model fit was

$$\mathbf{y}_{ep} \sim \text{Poisson}(n_{ep} e^{\boldsymbol{\nu}})$$

where $\boldsymbol{\nu} = \beta_0 + \beta_e \text{ethnicity}_e + \alpha_c \text{crime} + \mathbf{u}_p$, with priors

$$\boldsymbol{\alpha} \sim \text{N}(0, \sigma^2_{\boldsymbol{\alpha}}), \qquad \boldsymbol{\beta} \sim \text{N}(0, \sigma^2_{\boldsymbol{\beta}}), \quad \text{and} \quad \mathbf{u}_p \sim \text{N}(0, \sigma^2_{\mathbf{u}}),$$

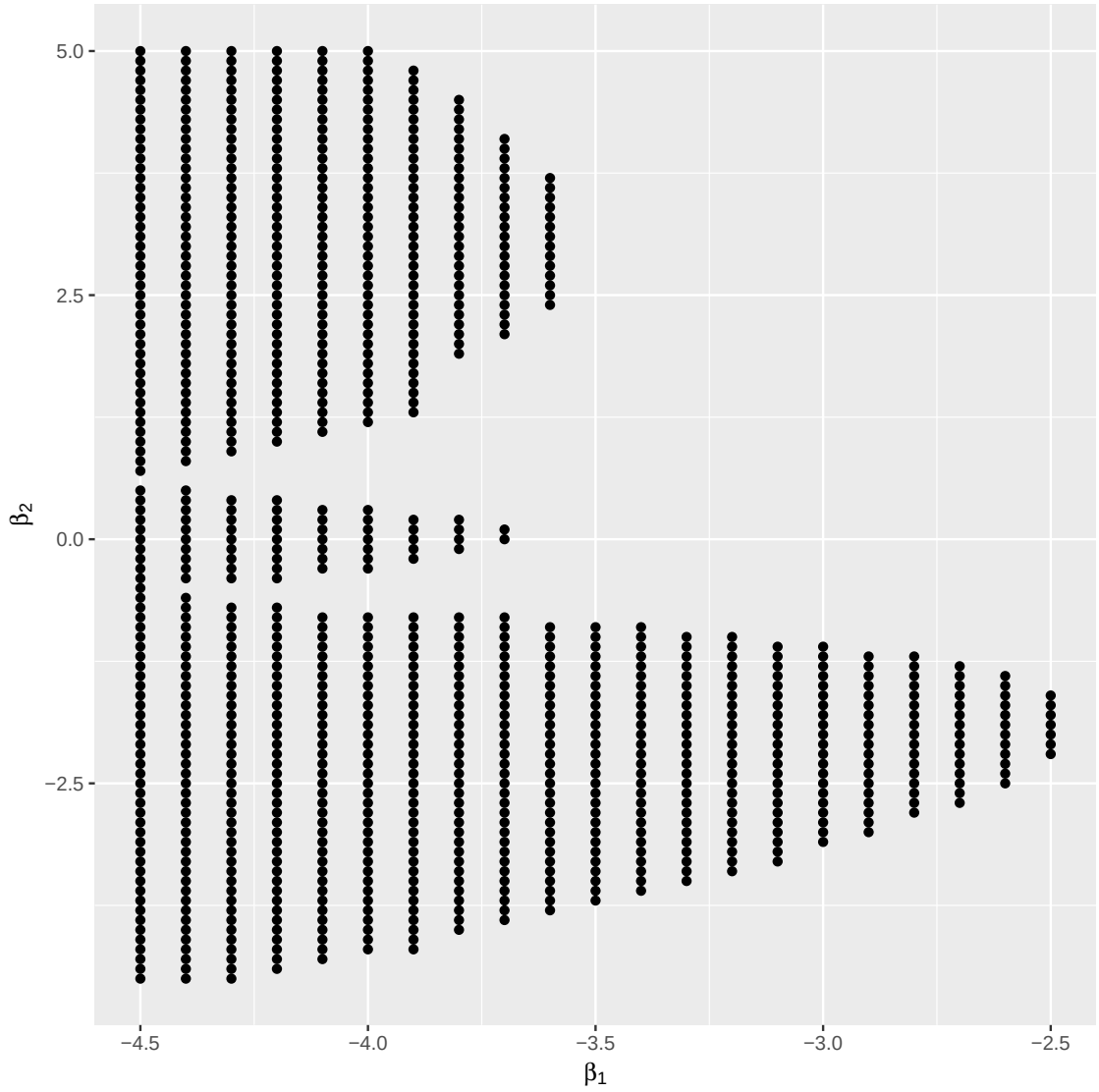

FIGURE 8. Starting locations which caused the fixed point fitting algorithm to fail with numeric errors. The true model had fixed parameters $\boldsymbol{\beta} = (2, 1)^\top$ and random intercepts. There were ten groups in the hierarchical model each with ten individuals ($m = 10, n_i = 10$).

Covariate	Posterior Mean	Lower 95% CI	Upper 95% CI	Accuracy
Intercept [African-Americans]	4.04	3.98	4.07	85%
β_2 [hispanics]	-0.45	-0.46	-0.43	99%
β_3 [whites]	-1.38	-1.40	-1.37	99%
α_1 [weapons crimes]	0.58	0.57	0.59	90%
α_2 [property crimes]	-0.19	-0.21	-0.17	92%
α_3 [drug crimes]	-0.75	-0.77	-0.73	95%
Random intercept	1.32	-0.19	2.20	87%
$\sigma^2_{\mathbf{u}}$	8.57	1.02	24.35	67%

TABLE 5. Table of results - Police stops.

Algorithm	Time in seconds
Laplace	0.07
GVA precision paramaterisation	0.90
GVA fixed point	0.06

TABLE 6. Table of speeds - Police stops.

the index p corresponds to each precinct, e is the index of ethnicity (African-Americans, hispanics or whites), and c is the index of category of crime (violent crimes, weapons crimes, property crimes or drug crimes). The random intercepts u_p allow for variation in the base rates of stops across precincts, the coefficients β_j measure the effect of ethnicity on the rate of police stops and the coefficients α_k measure the effect of each type of crime on the rate. The model finds the relationship between the number of police stops in each precinct and ethnicity for each type of crime.

The model was fit using the GVA algorithm with the $\mathbf{\Lambda} = (\mathbf{R}^\top \mathbf{R})^{-1}$ parameterisation, using the prior $a_\rho = 3$, $b_\rho = 1$ on ρ. Accuracy of the approximation was assessed by comparing the fitted distribution for each parameter to a kernel density estimate of the parameter's distribution from 1 million samples from the equivalent model using Stan. The results are presented in Table 5 and Figure 9.

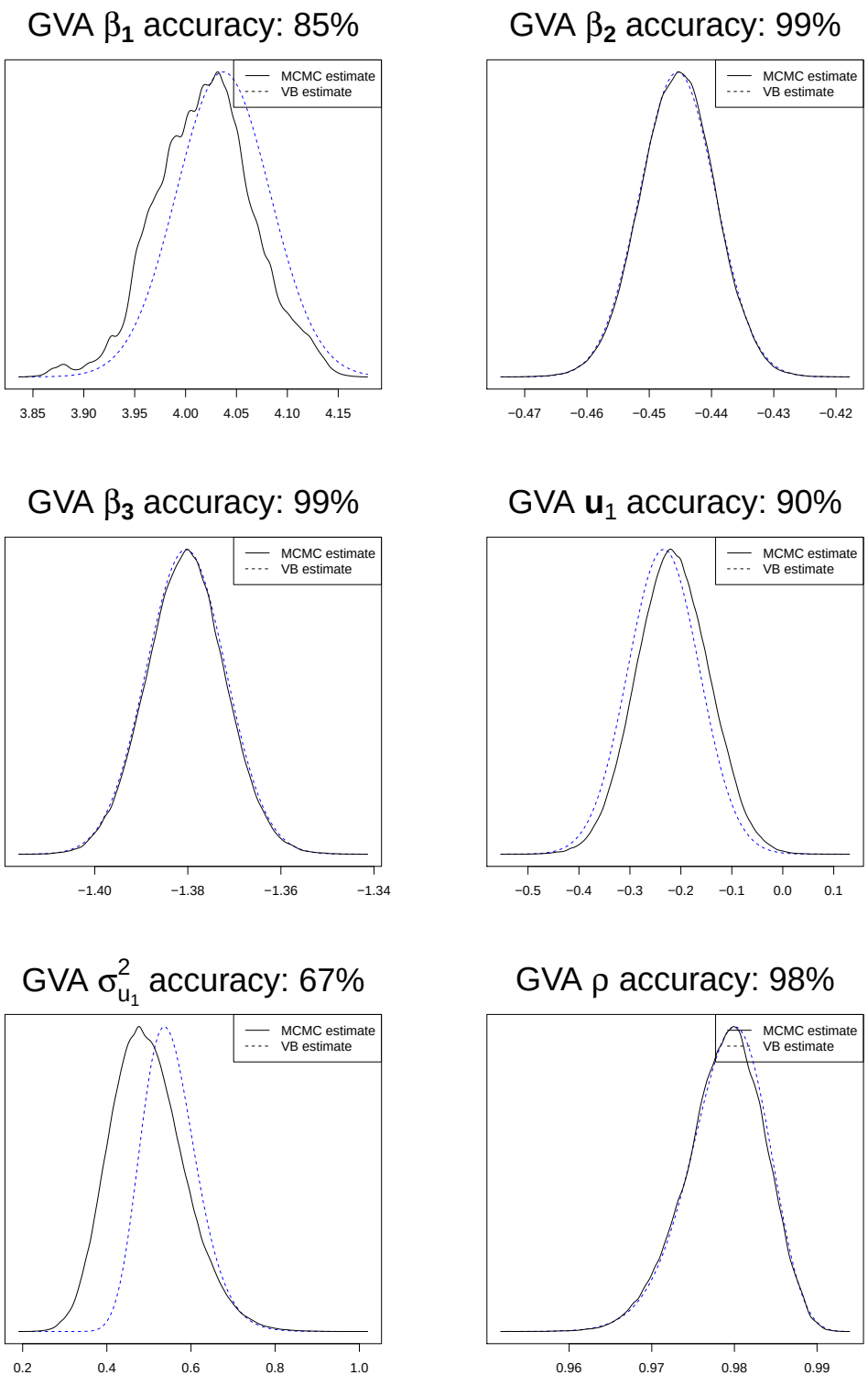

FIGURE 9. Accuracy of parameter estimates for police stops.

4.6.2. Zero–inflated example – Cockroaches in apartments. The model described in this section was fit to the cockroach data set from Section 6.7 of Gelman and Hill (2007), taken from a study on the effect of integrated pest management in controlling cockroach levels in urban apartments. The data set contains data on 160 treatment and 104 control apartments, along with the response y_i in each apartment of the number of cockroaches caught in a set of traps. The apartments had the traps deployed for different numbers of days, referred to as trap days, which was handled by using a log offset (Agresti, 2002). The predictors in the data set included the pre-treatment roach level, a treatment indicator, the time of the observation and an indicator for whether the apartment is in a senior building restricted to the elderly.

In the example application presented in this paper, the zero component represents an apartment completely free of roaches, while the non-zero component represents an apartment where roaches have been able to live and reproduce, possibly in spite of pest control treatment aimed at preventing them from doing so. The model fit was

$$y_i = \begin{cases} 0, & \text{if } R_i = 0, \text{ and} \\ \text{Poisson}(e^{\mathbf{X}_i\boldsymbol{\beta}+\mathbf{Z}_i\mathbf{u}}), & \text{if } R_i = 1, \end{cases}$$

with priors

$$R_i \sim \text{Bernoulli}(\rho), \quad \rho \sim \text{Beta}(a, b), \quad \boldsymbol{\beta} \sim \text{N}(\mathbf{0}, \sigma^2_{\boldsymbol{\beta}}\mathbf{I}),$$
$$\mathbf{u} \sim \text{N}(\mathbf{0}, \boldsymbol{\Sigma}) \quad \text{and} \quad \boldsymbol{\Sigma} \sim \text{Inverse-Wishart}(\boldsymbol{\Psi}, v)$$

Covariate	Posterior Mean	Lower 95% CI	Upper 95% CI	Accuracy
Intercept	3.42	3.20	3.65	96%
Time	−0.14	−0.05	−0.02	98%
Time:Treatment	−0.31	−0.43	−0.14	99%
Random intercept	−1.60	−1.71	−1.49	98%
$\sigma^2_{\mathbf{u}_1}$	3.29	2.02	8.48	64%
ρ	0.51	0.50	0.55	63%

TABLE 7. The posterior means, 95% credible intervals and accuracy of the fixed and random effects, $\sigma^2_{\mathbf{u}_1}$ and ρ for the Roach model.

Algorithm	Time in seconds
Laplace	0.68
GVA	2.02
GVA inv. param	1.70
GVA fixed point	0.17

TABLE 8. The runtimes in seconds for fitting algorithms when fitting the roach model.

and prior hyperparameters $a = 1$, $b = 1$, $\sigma^2_{\boldsymbol{\beta}} = 10^5$, $\boldsymbol{\Psi} = 10^{-5}\mathbf{I}$ and $v = 2$. These priors were chosen to be vaguely informative for the variance components and a uniform prior for the zero-inflation proportion latent variable ρ. The fixed effects covariates included in the model were time in days and time in days $\times$ pest control treatment. A random intercept to account for variation between the apartment buildings was included.

The GVA algorithm with the $\boldsymbol{\Lambda} = (\mathbf{R}^\top\mathbf{R})^{-1}$ parameterisation was used to fit a random intercept model to the Roaches data set provided in Gelman and Hill (2007). The fitted coefficients and accuracy results are presented in Table 7.

4.6.3. Example - Biochemists. The model described in this section was fit to the biochemistry data set analysed by Long (1990). The sample was taken from 915 biochemistry graduate students. The outcome $\mathbf{y}_i$ is the number of articles

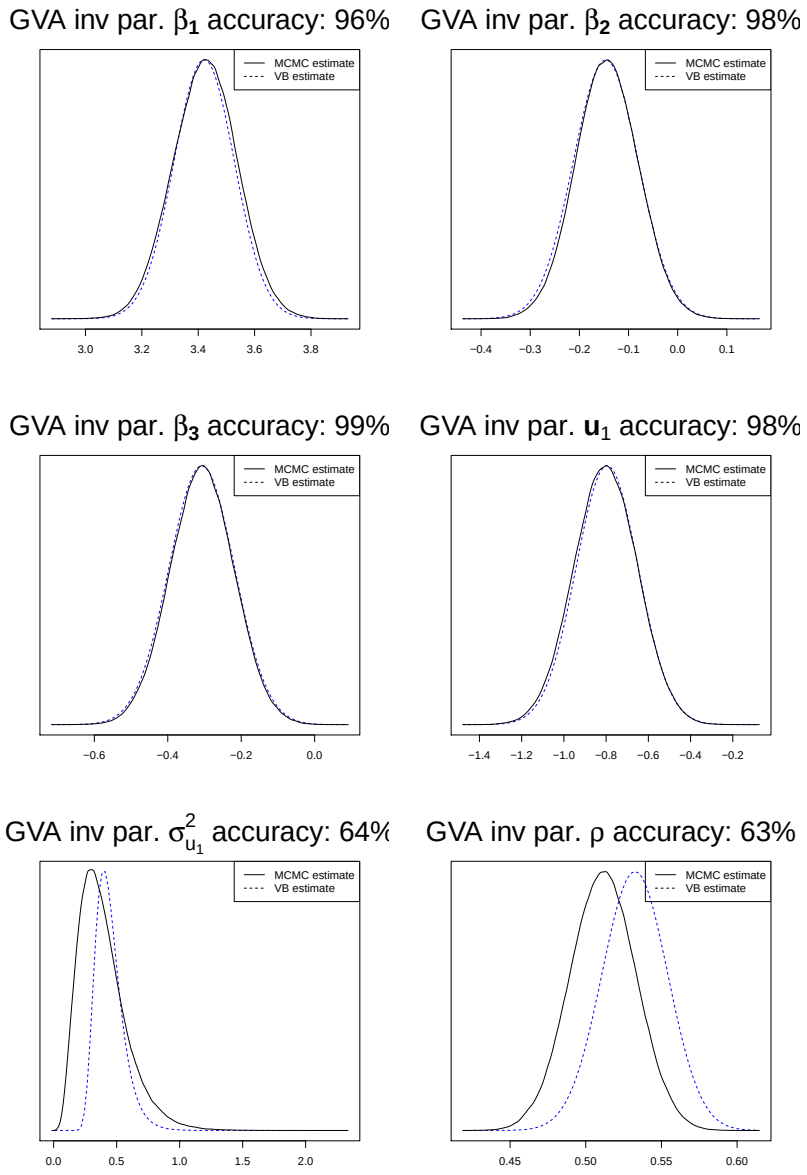

FIGURE 10. Accuracy graphs for roach model. The height of the graphs have been scaled to be the same height.

published in the last three years of the PhD. The covariates were the gender of the student, coded 1 for female and 0 for male, the marital status of the student (1 for

Covariate	Posterior Mean	Lower 95% CI	Upper 95% CI	Accuracy
Intercept	0.86	0.65	1.06	95%
Female	−0.18	−0.29	−0.08	95%
Married	0.06	−0.05	0.18	96%
Children under age 6	−0.08	−0.15	−0.01	97%
PhD	0.03	−0.02	−0.01	97%

TABLE 9. The posterior means, 95% credible intervals and accuracy of the fixed effects for the Biochemists model.

married, 0 for unmarried), the number of children under age six and the prestige of the PhD program.

In this example application, the zero component represents the number of biochemists who did not publish any articles during the last three years of their PhD. Examination of the data reveals that this number is higher than would be expected if the data followed a purely Poisson distribution – 30% of biochemistry graduate students published no articles in their final years whereas a Poisson distribution would predict only 18%. This justifies our choice of model. The model fit was

$$y_i = \begin{cases} 0, & \text{if} R_i = 0, \text{ and} \\ \text{Poisson}(e^{\boldsymbol{\nu}}), & \text{if} \quad R_i = 1, \end{cases}$$

where $\boldsymbol{\nu} = \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2$female $+ \boldsymbol{\beta}_3$married $+ \boldsymbol{\beta}_4$children under age 6 $+ \boldsymbol{\beta}_5$PhD, with priors $R_i \sim$ Bernoulli$(\rho), \rho \sim$ Beta(A, B) and $\boldsymbol{\beta} \sim$ N$(0, \sigma^2_{\boldsymbol{\beta}}\mathbf{I})$ and $A = 1$, $B = 1$ and $\sigma^2_{\boldsymbol{\beta}} = 10,000$. The model was fit using the GVA precision parameterisation algorithm. The resulting model fit is presented in Table 4.6.3 The accuracy of the parameter estimates is presented in Figure 4.6.3. As this is a fixed effects model with a large number of samples relative to the number of parameters being fit, we are able to estimate all of the parameters with great accuracy.

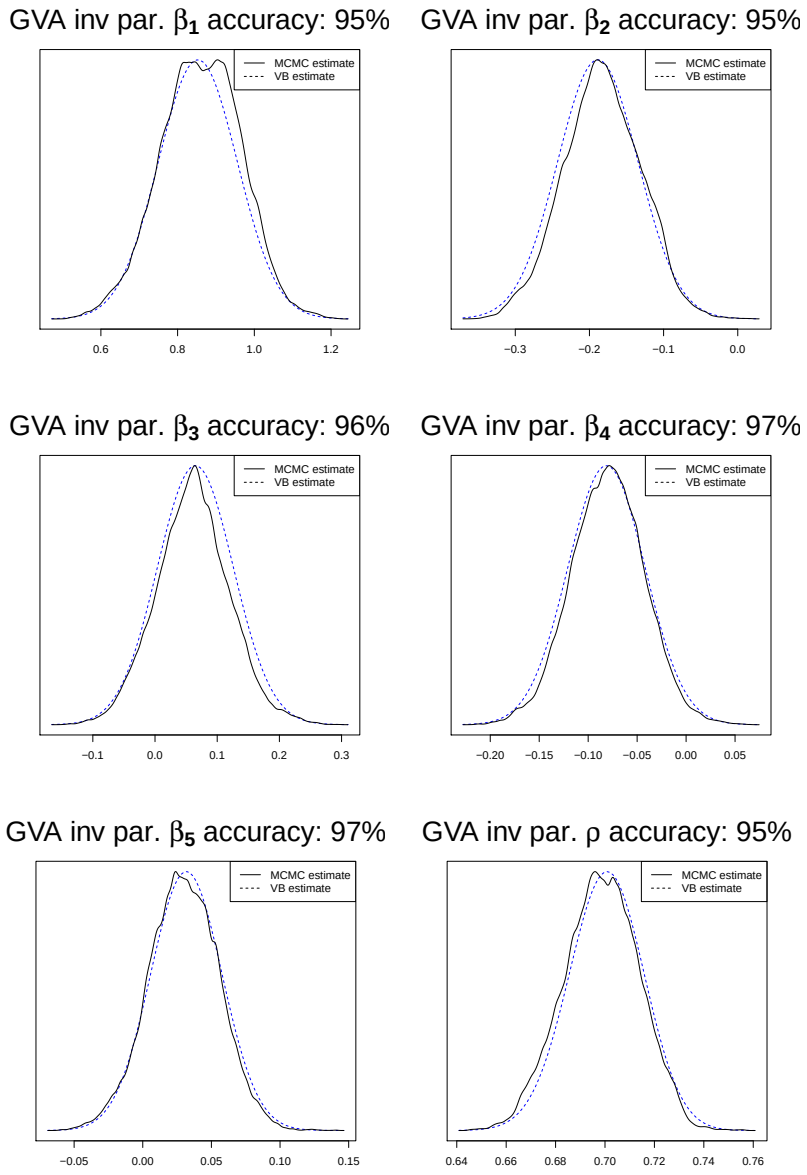

FIGURE 11. Accuracy of the approximations of the parameters fit to the biochemists data.

4.6.4. Example - Owls. The model described in this section was fit to the Owls data set taken from Zuur et al. (2009). The sample consists of 599 observations

Algorithm	Time in seconds
Laplace	0.12
GVA	0.60
GVA inv. param	0.53
GVA fixed point	0.07

TABLE 10. The run times in seconds for fitting algorithms when fitting the Biochemists model.

made of owls grouped across 25 nests.The fixed covariates fit in the model were food treatment (Deprived or Satiated), a categorical variable, and arrival time, a continuous covariate. The variation between the 25 different nests sampled from was modelled by a random intercept $\mathbf{u}$. The model fit was

$$y_i = \begin{cases} 0, & \text{if} R_i = 0, \text{ and} \\ \text{Poisson}(e^{\boldsymbol{\nu}}), & \text{if } \ R_i = 1, \end{cases}$$

where $\boldsymbol{\nu} = \boldsymbol{\beta}_2 \text{I(Food Treatment = Satiated)} + \boldsymbol{\beta}_3 \text{I(Arrival Time)} + \mathbf{u}_n$ and n is the n-th nest. We specified the priors $R_i \sim \text{Bernoulli}(\rho)$, $\rho \sim \text{Beta}(A, B)$, $\boldsymbol{\beta} \sim \text{N}(\mathbf{0}, \sigma^2_{\boldsymbol{\beta}}\mathbf{I})$, $\mathbf{u} \sim \text{N}(0, \sigma^2_{\mathbf{u}})$ and $\sigma^2_{\mathbf{u}} \sim \text{Inverse-Gamma}(s, t)$ with $\sigma^2_{\boldsymbol{\beta}} = 10,000$, $A = 1$, $B = 1$, $s = 10^{-2}$ and $r = 10^{-2}$ on the parameters in the model.

The model was fit using the GVA precision parameterisation algorithm. The accuracy of the parameter estimates is shown in Figure 12, while the runtime of the algorithms is shown in Table 12. We draw attention to the difference in runtimes between the covariance and precision parameterisations. The algorithm using the precision parameterisation fits the model significantly faster with a runtime of 1.88 seconds versus 5.66 seconds for the covariance parameterisation.

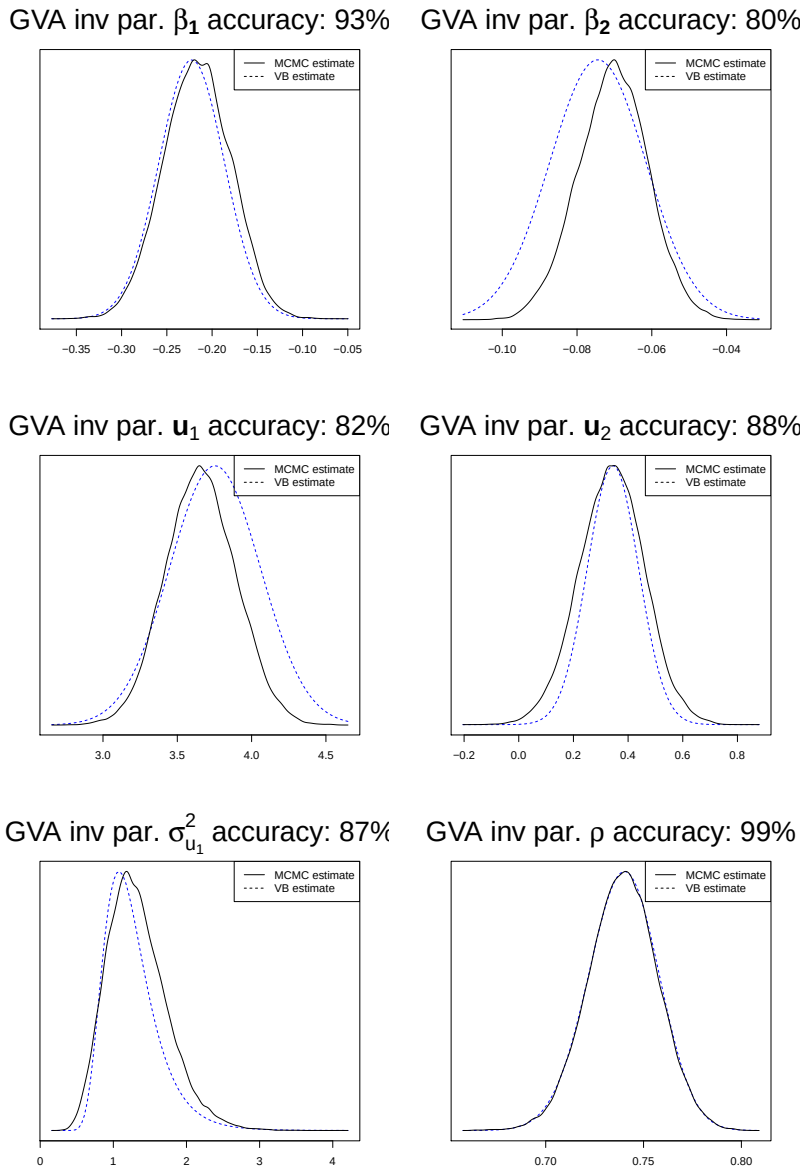

FIGURE 12. Accuracy of the approximations of the parameters fit to the Owls data.

Covariate	Posterior Mean	Lower 95% CI	Upper 95% CI	Accuracy
Satiated	-0.22	-0.21	-0.21	93%
Arrival Time	-0.07	-0.07	-0.07	80%
Random intercept (nest)	0.34	-5.28	5.96	82%
$\sigma^2_{\mathbf{u}_1}$	7.90	3.21	468.12	87%
ρ	0.74	0.70	0.77	99%

TABLE 11. The posterior means, 95% credible intervals and accuracy of the fixed and random effects, $\sigma^2_{\mathbf{u}_1}$ and ρ for the Owls model.

Algorithm	Time in seconds
Laplace	0.78
GVA covariance parameterisation	13.95
GVA precision parameterisation	2.15
GVA fixed point	0.25

TABLE 12. The run times of the fitting algorithms for the Owls model in seconds.

4.7. Conclusion

We described a Variational Bayes approximation to Zero-Inflated Poisson regression models which allows such models to be fit with considerable generality. We have also devised and extensively tested a number of alternative approaches for fitting such models, and extended one of these alternative approaches with a new parameterisation. Using MCMC methods as the gold standard to test against, we have assessed the accuracy and computational speed of these algorithms.

We applied our model fitting algorithms to a number of data sets to fit a range of models. The Cockroaches model in Section 4.6.2 had few fixed covariates, a random intercept for each apartment building and incorporated zero-inflation. The Police stops model in Section 4.6.1 was a pure Poisson mixed model, with no zero-inflation and a random intercept for precincts/locality. The Biochemists model in Section 4.6.3 was zero-inflated with fixed effects. The Owls model in

Section 4.6.4 was zero-inflated, with a random intercepts for each nest. There were a large number of nests ($m = 27$). We were able to estimate the variance component for this model very accurately.

The use of Mean Field Variational Bayes allows estimation of Bayesian ZIP models in a fraction of the time taken to fit the same model using even the best MCMC methods available, with only a small loss of accuracy. This is of great utility in applications where speed matters, such as when applied statisticians are comparing and choosing amongst many candidate models, as is typical in practice.

The new parameterisation of GVA using the Cholesky factorisation of the inverse of $\mathbf{\Lambda}$ presented in Section 4.4 provides significant advantages when used to estimate mixed models.

Mixed models have covariance matrices with a block structure, due to the dependence structure of the random effects. The precision parameterisation presented in this chapter is able to preserve this sparsity within the structure of the Cholesky factors of the inverses of the covariance matrices use in the variational lower bound by re-ordering the rows and columns of the matrices so that the random effects blocks appear first. The Owls example presented in this chapter shows the computational advantages of this approach when the number of groups m in the model is large ($m = 27$ in this case) – as the covariance parameterisation takes 46 seconds to fit whereas the inverse parameterisation only takes 3 seconds. This clearly demonstrates advantage of using sparsity to reduce the dimension of the optimisation problem to be solved when models are being fit – as only the non-zero values in the covariance matrices need to be optimised over. This allows

models to be fit more quickly, and with greatly improved numerical stability and without loss of accuracy.

While all of the fitting algorithms presented in this chapter except the Laplace's approximation algorithm were able to fit ZIP random and fixed effects models with high accuracy, and the Gaussian inverse parameterisation and fixed point algorithms were able to do so at high speed, they could be numerically unstable depending on the data the model was being fit to and their starting points. In the case of the Gaussian inverse parameterisation algorithm, the source of the problem was tracked down to the exponential function used in the parameterisation of the diagonal of the Cholesky factor of the precision matrix combined with the exponential that arises in the derivation of the Gaussian variational lower bound for Poisson mixed models – leading to frequent numeric overflows during the fitting process. This problem, once discovered, was mitigated by replacing the exponential parameterisation of the diagonal of the Cholesky factor with a piecewise function which is exponential beneath a threshold and quadratic above that threshold. This was shown to greatly increase the numeric stability of the GVA inverse parameterisation for a range of starting points.

Some of the algorithms which we experimented with were found to be very sensitive to their initial conditions. While these algorithms are typically initialised with a starting point as close as possible to the final solution, this gives some sense of the stability of each algorithm. We were able to develop a variant of the algorithm that employs a parameterisation which is much more numerically stable, and demonstrate this numerical stability for a range of models.

4.A. Calculation of the variational lower bound

The variational lower bound is $\mathbb{E}_q\{\log p(\mathbf{y},\boldsymbol{\theta}) - \log q(\boldsymbol{\theta})\} = T_1 + T_2 + T_3$, where

$$
\begin{aligned}
T_1 &= \mathbb{E}_q[\log p(\mathbf{y},\boldsymbol{\nu}) - \log q(\boldsymbol{\nu})] \\
&= \mathbf{y}\mathbf{P}\mathbf{C}\boldsymbol{\mu} - \mathbf{p}^\top \exp\left\{\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\mathrm{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\right\} - \mathbf{1}^\top \log\Gamma(\mathbf{y}+\mathbf{1}) \\
&\quad + \tfrac{p+m}{2}(1+\log 2\pi) + \tfrac{1}{2}\log|\boldsymbol{\Lambda}|, \\
T_2 &= \mathbb{E}_q\left\{\log p(\boldsymbol{\Sigma}_{\mathbf{uu}}) - \log q(\boldsymbol{\Sigma}_{\mathbf{uu}})\right\} \\
&= \mathbb{E}_q\big\{v/2(\log|\Psi| - \log|\Psi + \boldsymbol{\mu}_{\mathbf{u}}\boldsymbol{\mu}_{\mathbf{u}}^\top + \boldsymbol{\Lambda}_{\mathbf{uu}}|) + \tfrac{1}{2}\log 2 + \tfrac{1}{2}\log|\boldsymbol{\Sigma}_{\mathbf{uu}}| \\
&\qquad + \log\Gamma_{p+1}(v/2) - \log\Gamma_p(v/2) + \tfrac{1}{2}\mathrm{tr}((\boldsymbol{\mu}_{\mathbf{u}}\boldsymbol{\mu}_{\mathbf{u}}^\top + \boldsymbol{\Lambda}_{\mathbf{uu}})\boldsymbol{\Sigma}_{\mathbf{uu}}^{-1})\big\} \\
&= v/2\big(\log|\Psi| - \log|\Psi + \boldsymbol{\mu}_{\mathbf{u}}\boldsymbol{\mu}_{\mathbf{u}}^\top + \boldsymbol{\Lambda}_{\mathbf{uu}}|\big) + \tfrac{1}{2}\log 2 \\
&\quad + \tfrac{1}{2}\mathbb{E}_q\log|\boldsymbol{\Sigma}_{\mathbf{uu}}| + \log\Gamma_{p+1}(v/2) - \log\Gamma_p(v/2) \\
&\quad + \tfrac{1}{2}\mathrm{tr}\big[\mathbf{I}_m + \Psi(\Psi + \boldsymbol{\mu}_{\mathbf{u}}\boldsymbol{\mu}_{\mathbf{u}}^\top + \boldsymbol{\Lambda}_{\mathbf{uu}})^{-1}/(v+p+2)\big] \\
T_3 &= -\mathbf{p}^\top\log\mathbf{p} - (\mathbf{1}-\mathbf{p})^\top\log(\mathbf{1}-\mathbf{p}) - \log\mathrm{Beta}(\alpha_\rho,\beta_\rho) + \log\mathrm{Beta}(\alpha_q,\beta_q)
\end{aligned}
$$

with $\mathbb{E}_q\log|\boldsymbol{\Sigma}_{\mathbf{uu}}| = m\log 2 + \log\left|\Psi + \boldsymbol{\mu}_{\mathbf{u}}\boldsymbol{\mu}_{\mathbf{u}}^\top + \boldsymbol{\Lambda}_{\mathbf{uu}}\right| + \sum_{i=1}^m \Psi\left(\frac{v-i+1}{2}\right)$.

4.B. Calculation of derivatives

4.B.1. Derivatives for Laplace-Gaussian variational approximation.

$$
\begin{aligned}
\tfrac{\partial \log p(\boldsymbol{\mu},\boldsymbol{\Lambda};\mathbf{y})}{\partial\boldsymbol{\mu}} &\approx \mathbf{P}\mathbf{C}(\mathbf{y} - \exp(\mathbf{C}\boldsymbol{\mu})) - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \text{ and} \\
\tfrac{\partial \log p(\boldsymbol{\mu},\boldsymbol{\Lambda};\mathbf{y})}{\partial\boldsymbol{\Lambda}} &\approx -\mathbf{C}^\top\mathrm{diag}(\mathbf{p}e^{(\mathbf{C}\boldsymbol{\mu})})\mathbf{C} - \boldsymbol{\Sigma}^{-1}.
\end{aligned}
$$

4.B.2. Derivatives for parameterisation $\boldsymbol{\Lambda} = \mathbf{R}\mathbf{R}^\top$.

$$
\begin{aligned}
\tfrac{\partial \log p(\boldsymbol{\mu},\boldsymbol{\Lambda};\mathbf{y})}{\partial\boldsymbol{\mu}} &= \mathbf{P}\mathbf{C}(\mathbf{y} - \mathbf{C}^\top\exp(\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\mathrm{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top))) - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \text{ and} \\
\tfrac{\partial \log p(\boldsymbol{\mu},\boldsymbol{\Lambda};\mathbf{y})}{\partial\boldsymbol{\Lambda}} &= \left\{\boldsymbol{\Lambda}^{-1} - \mathbf{P}\mathbf{C}^\top\exp(\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\mathrm{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top))\mathbf{P}\mathbf{C}) - \boldsymbol{\Sigma}^{-1}\right\}\mathbf{R}.
\end{aligned}
$$

4.B.3. Derivatives for fixed point approach.

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{\mu},\boldsymbol{\Lambda};\mathbf{y})}{\partial \boldsymbol{\mu}} &= \mathbf{C}^\top \mathbf{p}\left[\mathbf{y} - \mathbf{C}\exp\{\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\}\right] - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \text{ and}\\ \frac{\partial \log p(\boldsymbol{\mu},\boldsymbol{\Lambda};\mathbf{y})}{\partial \boldsymbol{\Lambda}} &= -\mathbf{C}^\top \text{diag}[\mathbf{p}^\top \exp\{\mathbf{C}\boldsymbol{\mu} + \tfrac{1}{2}\text{diag}(\mathbf{C}\boldsymbol{\Lambda}\mathbf{C}^\top)\}] - \boldsymbol{\Sigma}^{-1}.\end{aligned}$$

CHAPTER 5

Future Directions

We conclude by briefly summarising the content of the thesis and outline potential research directions we seek to pursue.

5.1. Calculating Bayes factors for g-priors

In Chapter 2 we reviewed the prior structures that lead to closed form expressions for Bayes factors for linear models. We have described ways that each of these priors, except for the hyper-g/n prior can be evaluated in a numerically stable manner and have implemented a package `blma` for performing full exact Bayesian model averaging using this methodology. Our package is competitive with `BAS` and `BMS` in terms of computational speed, is numerically more stable and accurate, and offers some different priors structures not offered in `BAS`. Our package is much faster than `BayesVarSelect` and is also numerically more stable and accurate.

We are currently working on several extensions to this work. Firstly, we are working on a parallel implementation of the package which will allow for exact Bayesian inference for problems roughly the size $p \approx 30$. While a prototype of this work has been developed which runs on Linux using `OpenMP`, and displayed good performance as the number of cores used increased, we found it difficult to get the parallel code to work reliably on Macintosh and Windows computers.

As the majority of R users use these two platforms, we feel this is an important technical issue to resolve. Our algorithm could also be ported to GPUs.

Secondly, we are currently implementing Markov Chain Monte Carlo (MCMC) and population based MCMC methods for exploring the model space when $p > 30$. We can also see several obvious paths to extend this work to Generalised Linear Models - either using the approach described in Li and Clyde (2015) or by using Laplace approximations.

Thirdly, we are working on fast numerically stable quadrature based methods for the hyper-g/n and Zellner-Siow based priors. Further we are deriving exact expressions for parameter posterior distributions under some of the prior structures we have considered here. Many of these parameter posterior distributions are expressed in terms of special functions whose numerical evaluation must be handled with care.

5.2. Particle Variational Approximation

In Chapter 3 we developed PVA, a fast method for approximate Bayesian model averaging. There are several planned future extensions to this work. Firstly, we would like to generalise the PVA approach to linear models and generalised linear models, to be able to perform model selection for regression models applicable to a wider range of types of data. The computational approach would again either calculate Bayes factors based on the ideas of Li and Clyde (2015) or by using Laplace approximations. Additional care needs to be exercised for these models as the likelihood can often become irregular for a significant portion of models in the process of model selection.

Secondly, although the algorithm already runs in parallel on multicore CPUs using `OpenMP`, we believe even greater gains in performance could be achieved by porting the algorithm to run on GPUs, or by using distributed computing such as `OpenMPI`.

Thirdly, and most excitingly, we could examine modifications to the PVA algorithm itself. The way the algorithm currently ensures diversity amongst the particles in the population is to reward increases in entropy, weighted by the hyperparameter λ. The current version of the algorithm hard codes λ to 1, but it would be interesting to alter λ and as in the Population EM algorithm of Ročková (2017) and observe the effect on model selection performance. The algorithm also currently maintains diversity in the population by maintaining uniqueness of every particle within the population. It would be interesting to relax this constraint and compare the effect on model performance.

5.3. Zero-inflated models via Gaussian Variational Approximation

This chapter presents the essential ideas necessary for a high performance implementation for model fitting of ZIP regression models. The majority of the performance improvements over existing approaches come from avoiding unnecessary matrix inversion, which is a computationally expensive and numerically unstable process taking $\mathcal{O}(p^3)$ flops, and from constructing and calculating with sparse matrices. The gains of these approaches, particularly from sparse matrix techniques, can be difficult to fully realise in R without expert knowledge of the underlying implementation and libraries.

Our application of these ideas to Andrew Gelman's data showed that the new parameterisation very effectively speeds up fitting zero-inflated mixed models to real world data with a large number of groups, while still maintaining excellent accuracy versus an MCMC approach. This demonstrates the applicability of the ideas presented within this chapter to real world data sets.

The first directions for future research stemming from this chapter would be generalising the approximation to other zero-inflated models which handle overdispersion in the data without the need for a random intercept, such as the zero-inflated negative binomial model.

Furthermore, much more exploration could be done on alternative parameterisations of the covariance matrix in the Gaussian Variational Approximation (GVA). The specific parameterisation of the diagonal of the Cholesky factor as a piecewise exponential/quadratic polynomial function was chosen largely for convenience.

The current mean field update and GVA algorithms use the entire sample. For large samples in the Big Data era, this may not be computationally feasible. Other authors such as Tan and Nott (2018) have used doubly stochastic algorithms which both sub-sample the data and use noise to approximate the integral expression for the expectation of the variational lower bound. The sub-sampling in particular is very appealing in a Big Data context. We wish to experiment with this class of algorithm, and compare the performance and accuracy of this kind of doubly stochastic algorithm with the more traditional mean field and GVA algorithms presented in Chapter 4.

Bibliography

Abramowitz, M., Stegun, I. A., 1972. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. New York: Dover Publications.

Agresti, A., 2002. Categorical Data Analysis. Vol. 13.

Akaike, H., 1974. A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control 19 (6), 716–723.

Barber, R. F., Drton, M., Tan, K. M., 2016. Laplace approximation in high-dimensional Bayesian regression. Abel Symposia 11 (2012), 15–36.

Barbieri, M. M., Berger, J. O., 2004. Optimal predictive model selection. The Annals of Statistics 32 (3), 870–897.

Bartlett, M. S., 1957. A comment on D.V. Lindley's statistical paradox. Biometrika 44 (3), 533–534.

Bayarri, M. J., Berger, J. O., Forte, A., García-Donato, G., 2012. Criteria for Bayesian model choice with application to variable selection. The Annals of Statistics 40 (3), 1550–1577.

Berger, J. O., Pericchi, L. R., 2001. Objective Bayesian methods for model selection: Introduction and comparison. IMS Lecture Notes - Monograph Series 38, 135–207.

Berger, J. O., Pericchi, L. R., Varshavsky, J. A., 1998. Bayes factors and marginal distributions in invariant situations. Sankhyā: The Indian Journal of Statistics,

Series A 60 (3), 307–321.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

Bové, D. S., Colavecchia, F. D., Forrey, R. C., Gasaneo, G., Michel, N. L. J., Shampine, L. F., Stoitsov, M. V., Watts, H. A., 2013. appell: Compute Appell's F1 hypergeometric function. R package version 0.0-4.
URL https://CRAN.R-project.org/package=appell

Boyd, S., Vandenberghe, L., 2010. Convex Optimization. Vol. 25.

Breheny, P., Huang, J., 2011. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Annals of Applied Statistics 5 (1), 232–253.

Breiman, L., 1996. Heuristics of instability in model selection. The Annals of Statistics 24 (6), 2350–2383.

Butler, R. W., 2007. Saddlepoint Approximations with Applications. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Carbonetto, P., Stephens, M., 2011. Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. Bayesian Analysis 6 (4), 1–42.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Li, P., Riddell, A., 2016. Journal of Statistical Software Stan : A Probabilistic Programming Language. Journal of Statistical Software VV (Ii).

Casella, G., Moreno, E., 2006. Objective Bayesian Variable Selection. Journal of the American Statistical Association 101 (473), 157–167.

Castillo, I., Schmidt-Hieber, J., van der Vaart, A., 10 2015. Bayesian linear regression with sparse priors. Ann. Statist. 43 (5), 1986–2018.

Challis, E., Barber, D., 2013. Gaussian Kullback-Leibler Approximate Inference. Journal of Machine Learning Research 14, 2239–2286.

Chen, M. H., Huang, L., Ibrahim, J. G., Kim, S., 2008. Bayesian variable selection and computation for generalized linear models with conjugate priors. Bayesian Analysis 3 (3), 585–614.

Chipman, H., George, E. I., Mcculloch, R. E., Clyde, M., Foster, D. P., Stine, R., 2014. The Practical Implementation of Bayesian Model Selection. Lecture Notes-Monograph Series 38 (2001), 65–134.

Claeskens, G., Hjort, N. L., 2008. Model selection and model averaging. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge Univ. Press, Leiden.

Clyde, M., 2017. BAS: Bayesian Adaptive Sampling for Bayesian Model Averaging. R package version 1.4.4.

Cox, D., 2005. Frequentist and Bayesian statistics: a critique. Proceedings of the Statistical Problems in Particle . . . , 8–11.

Croissant, Y., 2016. Ecdat: Data Sets for Econometrics. R package version 0.3-1. URL `https://CRAN.R-project.org/package=Ecdat`

Cui, W., George, E. I., 2008. Empirical Bayes vs. fully Bayes variable selection. Journal of Statistical Planning and Inference 138 (4), 888–900.

de Boor, C., 1972. On calculating with B-splines. Journal of Approximation Theory 6 (1), 50–62.

Efron, B., 2013. Estimation and Accuracy after Model Selection. Journal of the American Statistical Association 1459 (October), 130725111823001.

Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood. Journal of the American Statistical Association 96 (456), 1348–1360.

Fernández, C., Ley, E., Steel, M. F. J., 2001. Benchmark priors for Bayesian model averaging. Journal of Econometrics 100 (2), 381 – 427.

Foster, D. P., George, E. I., 1994. The Risk inflation criterion for multiple regression. Annals of Statistics 22 (4), 1947–1975.

Gandomi, A., Haider, M., 2015. Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management 35 (2), 137–144.

García-Donato, G., Forte, A., 2016. BayesVarSel: Bayes Factors, Model Choice and Variable Selection in Linear Models. R package version 1.7.0.
URL `https://CRAN.R-project.org/package=BayesVarSel`

Gelman, A., Hill, J., 2007. Data analysis using regression and multilevel/hierarchical models. Vol. 625.

George, E. I., McCulloch, R. E., 1993. Variable selection via Gibbs sampling. Journal of the American Statistical Association 88 (423), 881–889.

George, E. I., McCulloch, R. E., 1997. Approaches for Bayesian Variable Selection. Statistica Sinica 7, 339–373.

Gershman, S. J., Hoffman, M. D., Blei, D. M., 2012. Nonparametric Variational Inference. International Conference on Machine Learning, 1–8.

Geweke, J., 1996. Variable selection and model comparison in regression.

Ghosh, S. K., Mukhopadhyay, P., Lu, J.-C., 2006. Bayesian analysis of zero-inflated regression models. Journal of Statistical Planning and Inference 136, 1360–1375.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 13 (5), 533 – 549, applications of Integer Programming.

URL http://www.sciencedirect.com/science/article/pii/0305054886900481

Golub, G. H., Van Loan, C. F., 2013. Matrix Computations (4th Ed.). Johns Hopkins University Press, Baltimore, MD, USA.

Goodnight, J. H., 1979. A tutorial on the sweep operator. The American Statistician 33 (3), 149–158.

Gradshteyn, I., Ryzhik, I., 2007. Tables of Integrals, Series, and Products. Academic Press.

Hall, D. B., 2000. Zero-Inflated Poisson and Binomial Regression with Random Effects: A Case Study. Biometrics 56 (4), 1030–1039.

Hall, P., Ormerod, J., Wand, M., 2011. Theory of Gaussian variational approximation for a Poisson mixed model. Statistica Sinica 21 (1), 369–389.

Hankin, R. K. S., October 2006. Special functions in R: introducing the gsl package. R News 6.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hoerl, A. E., Kennard, R. W., 1970. Ridge Regression: Biased Estimation for Problems Nonorthogonal. Technometrics 12 (1), 55–67.

Hoeting, J. a., Madigan, D., Raftery, A. E., Volinsky, C. T., 1999. Bayesian model averaging: a tutorial. Statistical Science 14 (4), 382–417.

Ishwaran, H., Rao, J. S., 2005. Spike and slab variable selection: Frequentist and Bayesian strategies. The Annals of Statistics 33 (2), 730–773.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2014. An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated.

Johnson, N. L., Kotz, S., Balakrishnan, N., 1995. Continuous Univariate Distributions, Volume 2 (2nd Edition). Wiley.

Johnstone, I. M., Titterington, D. M., Adragni, K. P., Cook, R. D., Banks, D. L., House, L., Killhoury, K., Barber, D., Beal, M. J., Ghahramani, Z., Belabbas, M.-A., Wolfe, P. J., Belkin, M., Niyogi, P., Benjamini, Y., Hochberg, Y., Benjamini, Y., Heller, R., Yekutieli, D., Bickel, P., Bickel, P. J., Levina, E., Bickel, P. J., Ritov, Y., Tsybakov, A. B., Bickel, P. J., Brown, J. B., Huang, H., Li, Q., Bishop, C. M., Breiman, L., Buja, A., Cook, D., Asimov, D., Hurley, D., Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D. F., Wickham, H., Candès, E. J., Tao, T., Candès, E., Tao, T., Chipman, H. A., George, E. I., McCulloch, R., Cook, R. D., Dawid, A. P., Dettling, M., Donoho, D. L., Donoho, D. L., Grimes, C., Donoho, D. L., Jin, J., Donoho, D., Jin, J., Donoho, D., Tanner, J., Karoui, N. E., Fan, J., Lv, J., Graunt, J., Hamilton, W. C., Hastie, T., Tibshirani, R., Hastie, T., Tibshirani, R., Friedman, J. H., Hoerl, A. E., Kennard, R. W., Huber, P. J., Ingster, Y. I., Pouet, C., Tsybakov, A. B., Jin, J., Johnstone, I. M., Johnstone, I. M., Johnstone, I. M., Lu, A. Y., Jolliffe, I. T., Lindsay, B. G., Kettenring, J., Siegmund, D. O., Nadler, B., Onatski, A., Ravikumar, P., Liu, H., Lafferty, J., Wasserman, L., Roweis, S. T., Saul, L. K., Tenenbaum, J., DeSilva, V., Langford, J., Tibshirani, R., Titterington, D. M., Wainwright, M. J., Wegman, E. J., Wegman, E. J., Solka,

J. L., 2009. Statistical challenges of high-dimensional data. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences 367 (1906), 4237–53.

Jordan, M., 2010. Stat260/cs 294 bayesian modeling and inference lecture notes. URL https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/

Kachman, S. D., 2000. an Introduction To Generalized Linear Mixed Models. Proceedings of a symposium at the organizational, 59–63.

Kärkkäinen, H. P., Sillanpää, M. J., 2012. Robustness of Bayesian Multilocus Association Models to Cryptic Relatedness. Annals of Human Genetics 76 (6), 510–523.

Kass, R. E., Raftery, A., 1995. Bayes factors. Journal of the American Statistical Association 91 (6), 773–795.

Kass, R. E., Raftery, A. E., 1993. Bayes Factors and Model Uncertainty. Technical Report (254), 1–73.

Kass, R. E., Wasserman, L., 1995. A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. Journal of the American Statistical Association 90 (431), 928–934.

Kim, A. S. I. K., Wand, M. P., 2017. On Expectation Propagation for Generalized , Linear and Mixed Models, 1–27.

Kingma, D. P., Welling, M., 2013. Auto-Encoding Variational Bayes (Ml), 1–14.

Kleinman, K., Lazarus, R., Platt, R., 2004. A Generalized Linear Mixed Models Approach for Detecting Incident Clusters of Disease in Small Areas, with an Application to Biological Terrorism. American Journal of Epidemiology 159 (3),

217–224.

Koopman, B. O., 1935. On Distributions Admitting a Sufficient Statistics. Transactions of the American Mathematical Society 222, 399–409.

Lambert, D., 1992. Zero-Inflated Poisson Regression, With an Application To Defects in Manufacturing. Technometrics 34 (1), 1–14.

Lee, A. H., Wang, K., Scott, J. A., Yau, K. K. W., McLachlan, G. J., 2006. Multi-level zero-inflated Poisson regression modelling of correlated count data with excess zeros. Statistical methods in medical research 15 (1), 47–61.

Li, Y., Clyde, M. A., 2015. Mixtures of g -priors in Generalized Linear Models. arXiv 1503.06913.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., Berger, J. O., 2008. Mixtures of g priors for Bayesian variable selection. Journal of the American Statistical Association 103 (481), 410–423.

Lindley, D. V., 1957. A statistical paradox. Biometrika 44, 187–192.

Liu, D. C., Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. Mathematical programming 45 (1-3), 503–528.

Lo, S., Andrews, S., 2015. To transform or not to transform: using generalized linear mixed models to analyse reaction time data. Frontiers in Psychology 6 (August), 1–16.

URL http://journal.frontiersin.org/Article/10.3389/fpsyg.2015.01171/abstract

Long, J. S., 06 1990. The origins of sex differences in science 68.

MacKay, D. J. C., 2002. Information Theory, Inference & Learning Algorithms. Cambridge University Press, New York, NY, USA.

Madigan, D., Raftery, A. E., 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. Journal of the American Statistical Association 89, 1335–1346.

Maruyama, Y., George, E. I., 2011. Fully Bayes factors with a generalized g-prior. Annals of Statistics 39 (5), 2740–2765.

Min, Y., Agresti, A., 2005. Random effect models for repeated measures of zero-inflated count data. Statistical Modelling 5 (1), 1–19.

Minka, T. P., 2001. A family of algorithms for approximate Bayesian inference. Ph.D. Thesis, 1–482.

Minka, T. P., 2013. Expectation Propagation for approximate Bayesian inference.

Mitchell, T. J., Beauchamp, J. J., 1988. Bayesian variable selection in linear regression. Journal of the American Statistical Association 83, 1023–1032.

Murphy, K. P., 2012. Machine Learning: A Probabilistic Perspective. The MIT Press.

Nadarajah, S., 2015. On the computation of Gauss hypergeometric functions. The American Statistician 69 (2), 146–148.

Nengjun Yi, H. M., 2013. Bayesian Methods for High Dimensional Linear Models. Journal of Biometrics & Biostatistics.

Nocedal, J., Wright, S., 2006. Numerical optimization. Springer Science & Business Media.

Opper, M., Archambeau, C., 2009. The variational Gaussian approximation revisited. Neural computation 21 (3), 786–792.

Ormerod, J. T., Stewart, M., Yu, W., Romanes, S. E., Oct. 2017. Bayesian hypothesis tests with diffuse priors: Can we have our cake and eat it too? ArXiv e-prints.

Ormerod, J. T., Wand, M. P., 2010. Explaining variational approximations. The American Statistician 64, 140–153.

Ormerod, J. T., Wand, M. P., 2012. Gaussian Variational Approximate Inference for Generalized Linear Mixed Models. Journal of Computational and Graphical Statistics 21 (1), 2–17.

O'Sullivan, F., 1986. A Statistical Perspective on Ill-Posed Inverse Problems. Statistical Science 1 (4), 505–527.

Papaspiliopoulos, O., Rossell, D., 2016. Scalable Bayesian variable selection and model averaging under block orthogonal design.

Pearson, J. W., Olver, S., Porter, M. A., 2017. Numerical methods for the computation of the confluent and gauss hypergeometric functions. Numerical Algorithms 74 (3), 821–866.

Pinheiro, J. C., Bates, D. M., 2000. Mixed-effects models in S and S-PLUS. Springer, New York, NY [u.a.].

Pitman, E. J. G., 1936. Sufficient statistics and intrinsic accuracy. Mathematical Proceedings of the Cambridge Philosophical Society 32 (4), 567–579.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 2007a. Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd Edition. Cambridge University Press, New York, NY, USA.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 2007b. Numerical Recipes in C (3rd Ed.): The Art of Scientific Computing. Cambridge University Press, New York, NY, USA.

Prudnikov, A. P., Brychkov, Y. A., Marichev, O. I., 1986. Integrals and Series (Vols. 1âĂŞ3). Gordon and Breach.

R Core Team, 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL https://www.R-project.org/

Raftery, A. E., Madigan, D., Hoeting, J. A., 1997. Bayesian Model Averaging for Linear Regression. Journal of the American Statistical Association 92, 179–191.

Redmond, M., Baveja, A., 2002. A data-driven software tool for enabling cooperative information sharing among police departments. European Journal of Operational Research 141 (3), 660–678.

Rijsbergen, C. J. V., 1979. Information Retrieval, 2nd Edition. Butterworth-Heinemann, Newton, MA, USA.

Rohde, D., Wand, M. P., 2015. Semiparametric Mean Field Variational Bayes : General Principles and Numerical Issues Semiparametric Mean Field Variational Bayes : (January), 1–41.

Ročková, V., 2017. Particle em for variable selection. Journal of the American Statistical Association 0, 0–0.

Ročková, V., George, E. I., 2014. EMVS: The EM approach to Bayesian variable selection. Journal of the American Statistical Association 109 (506), 828–846.

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

Ruppert, D., Wand, M. P., Carroll, R. J., 2003. Semiparametric Regression. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Schelldorfer, J., Bühlmann, P., van de Geer, S., 2010. Estimation for High-Dimensional Linear Mixed-Effects Models Using L_1-Penalization 2 (20), 1–30.

Scott, J. G., Berger, J. O., 10 2010. Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. Ann. Statist. 38 (5), 2587–2619.

Shankar, V., Milton, J., Mannering, F., 1997. Modeling accident frequencies as zero-altered probability processes: An empirical inquiry. Accident Analysis and Prevention 29 (6), 829–837.

Stan Development Team, 2016. {RStan}: the {R} interface to {Stan}.

Tan, L. S. L., Nott, D. J., Mar 2018. Gaussian variational approximation with sparse precision matrices. Statistics and Computing 28 (2), 259–275.

Teh, Y. W., Newman, D., Welling, M., 2006. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. Neural Information Processing Systems, 1353—-1360.

Tibshirani, R., 1996. Regression Selection and Shrinkage via the Lasso.

Tierney, L., Kadane, J. B., 1986. Accurate approximations for posterior moments and marginal densities.

Trefethen, L. N., Bau, D., 1997. Numerical Linear Algebra. SIAM.

Vatsa, R., Wilson, S., 2014. Variational Bayes Approximation for Inverse Non-Linear Regression (JANUARY 2014), 76–84.

Venables, W. N., Ripley, B. D., 2002. Modern Applied Statistics with S, 4th Edition. Springer, New York.

Wand, M. P., Ormerod, J. T., 2008. On semiparametric regression with O'Sullivan penalized splines. Australian & New Zealand Journal of Statistics 50 (2), 179–198.

Wang, X., George, E. I., 2007. Adaptive Bayesian criteria in variable selection for generalized linear models. Statistics Sinica 17, 667–690.

Weisstein, E. W., 2009. Appell Hypergeometric Function.

Xu, S., 2007. An empirical Bayes method for estimating epistatic effects of quantitative trait loci. Biometrics 63 (2), 513–521.

Yau, K. K. W., Wang, K., Lee, A. H., 2003. Zero-Inflated Negative Binomial Mixed Regression Modeling of Over-Dispersed Count Data with Extra Zeros. Biometrical Journal 45 (4), 437–452.

You, C., Ormerod, J. T., 2014. On Variational Bayes Estimation and Variational Bayes Information Criteria for Linear Regression Models 61 (2).

Zellner, A., 1986. On assessing prior distributions and Bayesian regression analysis with g-prior distributions. Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti, 233–243.

Zellner, A., Siow, A., 1980a. Posterior odds ratio for selected regression hypothesis. Bayesian Statistics (1978), 585–648.

Zellner, A., Siow, A., 1980b. Posterior odds ratios for selected regression hypotheses. Trabajos de Estadistica Y de Investigacion Operativa 31 (1), 585–603.

Zeugner, S., Feldkircher, M., 2015. Bayesian model averaging employing fixed and flexible priors: The BMS package for R. Journal of Statistical Software 68 (4), 1–37.

Zhang, C. H., 2010. Nearly unbiased variable selection under minimax concave penalty. Vol. 38.

Zhao, Y., Staudenmayer, J., Coull, B. A., Wand, M. P., 2006. General Design Bayesian Generalized Linear Mixed Models. Statistical Science 21 (1), 35–51.

Zuur, A. F., Ieno, E. N., Walker, N., Saveliev, A. A., Smith, G. M., 2009. Mixed effects models and extensions in ecology with R. Statistics for Biology and Health. Springer New York, New York, NY.