VARIATIONAL APPROXIMATION FOR COMPLEX REGRESSION MODELS

TAN SIEW LI, LINDA
(BSc.(Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF STATISTICS AND APPLIED PROBABILITY

NATIONAL UNIVERSITY OF SINGAPORE

2013

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Tan Siew Li, Linda
21 June 2013

Acknowledgements

First and foremost, I wish to express my sincere gratitude and heartfelt thanks to my supervisor, Associate Professor David Nott. He has been very kind, patient and encouraging in his guidance and I have learnt very much about carrying out research from him. I thank him for introducing me to the topic of variational approximation, for the many motivating discussions and for all the invaluable advice and timely feedback. This thesis would not have been possible without his help and support.

I want to take this opportunity to thank Associate Professors Yang Yue and Sanjay Chaudhuri for helping me embark on my PhD studies, and Professor Loh Wei Liem for his kind advice and encouragement. I am very grateful to Ms Wong Yean Ling, Professor Robert Kohn and especially Associate Professor Fred Leung for their continual help and kind support. I have had a wonderful learning experience at the Department of Statistics and Applied Probability and for this I would like to offer my special thanks to the faculty members and support staff.

I thank the Singapore Delft Water Alliance for providing partial financial support during my PhD studies as part of the tropical reservoir research programme and for supplying the water temperature data set for my research. I also thank Dr. David Burger and Dr. Hans Los for their help and feedback on my work in relation to the water temperature data set. I thank Professor Matt Wand for his interest and valuable comments on our work in nonconjugate variational message passing and am very grateful to him for making available to us his preliminary results on fully simplified multivariate normal updates in nonconjugate variational message passing.

I wish to thank my parents who have always supported me in what I do. They are always there when I needed them and I am deeply grateful for their unwavering love and care for me. Finally, I want to thank my husband and soul mate, Taw Kuei for his love, understanding and support through all the difficult times. He has always been my source of inspiration and my pillar of support. Without him, I would not have embarked on this journey or be able to made it through.

Contents

Summary

The trend towards collecting large data sets driven by technology has resulted in the need for fast computational approximations and more flexible models. My thesis reflects these themes by considering very flexible regression models and developing fast variational approximation methods for fitting them.

First, we consider mixtures of heteroscedastic regression models where the response distribution is a normal mixture, with the component means, variances and mixing weights all varying as a function of the covariates. Fast variational approximation methods are developed for fitting these models. The advantages of our approach as compared to computationally intensive Markov chain Monte Carlo (MCMC) methods are compelling, particularly for time series data where repeated refitting for model choice and diagnostics is common. This basic variational approximation can be further improved by using stochastic approximation to perturb the initial solution.

Second, we propose a novel variational greedy algorithm for fitting mixtures of linear mixed models, which performs parameter estimation and model selection simultaneously, and returns a plausible number of mixture components automatically. In cases of weak identifiability of model parameters, we use hierarchical centering to reparametrize the model and show that there is a gain in efficiency in variational algorithms similar to that in MCMC algorithms. Related to this, we prove that the approximate rate of convergence of variational algorithms by Gaussian approximation is equal to that of the corresponding Gibbs sampler. This result suggests that reparametrizations can lead to improved convergence in variational algorithms just as in MCMC algorithms.

Third, we examine the performance of the centered, noncentered and partially noncentered parametrizations, which have previously been used to accelerate MCMC and expectation maximization algorithms for hierarchical models, in the context of variational Bayes for generalized linear mixed models (GLMMs). We demonstrate how GLMMs can be fitted using nonconjugate variational message passing and show that the partially noncen-

tered parametrization is able to automatically determine a parametrization close to optimal and accelerate convergence while yielding more accurate approximations statistically. We also demonstrate how the variational lower bound, produced as part of the computation, can be useful for model selection.

Extending recently developed methods in stochastic variational inference to nonconjugate models, we develop a stochastic version of nonconjugate variational message passing for fitting GLMMs that is scalable to large data sets, by optimizing the variational lower bound using stochastic natural gradient approximation. In addition, we show that diagnostics for prior-likelihood conflict, which are very useful for Bayesian model criticism, can be obtained from nonconjugate variational message passing automatically. Finally, we demonstrate that for moderate-sized data sets, convergence can be accelerated by using the stochastic version of nonconjugate variational message passing in the initial stage of optimization before switching to the standard version.

List of Tables

List of Figures

List of Abbreviations

AIC	Akaike information criterion
ARI	Adjusted Rand index
AWBM	Australian water balance model
BIC	Bayesian information criterion
EM	Expectation maximization
GLMM	Generalized linear mixed model
GO	Gene ontology
LPDS	Log predictive density score
MCMC	Markov chain Monte Carlo
MHR	Mixture of heteroscedastic regression
MLMM	Mixture of linear mixed models
VB	Variational Bayes
VGA	Variational greedy algorithm

Chapter 1

Introduction

Technological advances have enabled the collection of larger data sets which presents new challenges in the development of statistical methods and computational algorithms for their analysis. As data sets grow in size and complexity, there is a need for (i) more flexible models to capture and describe more accurately the relationship between responses and predictors and (ii) fast computational approximations to maintain efficiency and relevance. This thesis seeks to address these needs by considering some very flexible regression models and developing fast variational approximation methods for fitting them. We adopt a Bayesian approach to inference which allows uncertainty in unknown model parameters to be quantified.

This chapter is organized as follows. Section 1.1 briefly reviews variational approximation methods and describes how they are useful in Bayesian inference. Section 1.2 highlights the main contributions of this thesis and Section 1.3 describes the notation and distributional definitions used in this thesis.

1.1 Variational Approximation

In recent years, variational approximation has emerged as an attractive alternative to Markov chain Monte Carlo (MCMC) and Laplace approximation methods for posterior estimation in Bayesian inference. Being a fast, deterministic and flexible technique, it requires much less computation time than MCMC methods, especially for complex models. It does not restrict the posterior to a Gaussian form as in Laplace approximation and the convergence is easy to monitor. However, unlike MCMC methods which can in principle be made arbitrarily accurate by increasing the simulation sample size, variational approximation methods are limited in how closely they can approximate the true posterior.

Variational approximation methods originated in statistical physics and have mostly been developed in the machine learning community (e.g. Jordan *et al.*, 1999; Ueda and Ghahramani, 2002; Winn and Bishop, 2005). However, research in variational methods is currently very active in both machine learning and statistics (e.g. Braun and McAuliffe, 2010; Ormerod and Wand, 2012). In particular, variational Bayes computational methods are attracting increasing interest because of their ability to scale to large high-dimensional data (Hoffman *et al.*, 2010; Wang *et al.*, 2011).

1.1.1 Bayesian inference

First, let us consider how variational approximation can be applied in Bayesian inference. Suppose we have a model where y denotes the observed data, θ denotes the set of unknown parameters and $p(\theta)$ represents a prior distribution placed on the unknown parameters. Bayesian inference is based on the posterior distribution of the unknown parameters, $p(\theta|y)$, which is often intractable. In variational approximation, we approximate $p(\theta|y)$ by a $q(\theta)$ for which inference is more tractable. It is common to assume, for instance, that $q(\theta)$ belongs to some parametric distribution or that $q(\theta)$ factorizes into $\prod_{i=1}^{m} q_i(\theta_i)$ for some partition $\{\theta_1, \ldots, \theta_m\}$ of θ. We attempt to make $q(\theta)$ a good approximation to $p(\theta|y)$ by minimizing the Kullback-Leibler divergence between them. The Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|y)$ is

$$\int q(\theta) \log \frac{q(\theta)}{p(\theta|y)} \, d\theta = \int q(\theta) \log \frac{q(\theta)}{p(y|\theta)p(\theta)} \, d\theta + \log p(y), \tag{1.1}$$

where $p(y) = \int p(y|\theta)p(\theta) \, d\theta$ is the marginal likelihood. As the Kullback-Leibler divergence is non-negative, we have

$$\begin{aligned} \log p(y) &\geq \int q(\theta) \log \frac{p(y|\theta)p(\theta)}{q(\theta)} \, d\theta \\ &= E_q\{\log p(y, \theta)\} - E_q\{\log q(\theta)\} \\ &= \mathcal{L}, \end{aligned} \tag{1.2}$$

where E_q denotes expectation with respect to the variational approximation $q(\theta)$ and $\mathcal{L}$ is a lower bound on the log marginal likelihood. From (1.1), the difference between the lower bound and the log marginal likelihood is the Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|y)$. Maximization of the lower bound $\mathcal{L}$ is thus equivalent to minimization of the Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|y)$. The lower bound $\mathcal{L}$ is sometimes used

as an approximation to the log marginal likelihood for Bayesian model selection purposes (see Section 1.1.3).

Variational approximations are often useful in Bayesian predictive inference. Let y^* denote a future response. Bayesian predictive inference is based on the predictive distribution

$$p(y^*|y) = \int p(y^*|\theta, y)p(\theta|y)\, d\theta. \tag{1.3}$$

The first component of uncertainty in $p(y^*|y)$ is the inherent randomness in y^* which would still be around if θ were known and this is captured by $p(y^*|\theta, y)$ in the integrand. The second component of uncertainty is parameter uncertainty which is captured by $p(\theta|y)$. For large data sets, the parameter uncertainty is small and substituting $p(\theta|y)$ with the variational posterior $q(\theta)$ in (1.3) is an attractive means of obtaining predictive inference, provided that $q(\theta)$ gives good point estimation. Moreover, this still accounts to some extent for parameter uncertainty.

The independence and distributional assumptions made in variational approximations may not be realistic and it has been shown in the context of Gaussian mixture models that factorized variational approximations have a tendency to underestimate the posterior variance (Wang and Titterington, 2005; Bishop, 2006). However, variational approximation can often lead to good point estimates, reasonable estimates of marginal posterior distributions and excellent predictive inferences compared to other approximations, particularly in high dimensions. Blei and Jordan (2006), for instance, showed that predictive distributions based on variational approximations to the posterior were very similar to those obtained by MCMC for Dirichlet process mixture models. Braun and McAuliffe (2010) reported similar findings in large-scale models of discrete choice although they observed that the variational posterior is more concentrated around the mode than the MCMC posterior, a familiar underdispersion effect noted above.

1.1.2 Variational Bayes

The restriction that the variational approximation $q(\theta)$ factorizes as $q(\theta) = \prod_{i=1}^{m} q_i(\theta_i)$ for some partition $\{\theta_1, \ldots, \theta_m\}$ of θ, is known as "mean field" approximation in Physics (Parisi, 1988). Approximate Bayesian inference under this product density assumption is also known as variational Bayes (VB). A very early instance of VB applied to mixture of regression models (Jacobs *et al.*, 1991; Jordan and Jacobs, 1994) was presented in Waterhouse

et al. (1996) and the VB framework was first proposed formally by Attias (1999). VB has since been applied to many models in different applications (e.g. McGrory and Titterington, 2007; Faes *et al.*, 2011). Maximization of the lower bound $\mathcal{L}$ with respect to each of $q_1, \ldots, q_m$ in VB leads to optimal densities satisfying

$$q_i(\theta_i) \propto \exp\{E_{-\theta_i} \log p(y, \theta)\}, \tag{1.4}$$

for each $i = 1, \ldots, m$, where $E_{-\theta_i}$ denotes expectation with respect to the density $\prod_{j \neq i} q_j(\theta_j)$ (see, e.g. Ormerod and Wand, 2010). If conjugate priors are used, the optimal densities q_i will have the same form as the prior so that it suffices to update the parameters of q_i (Winn and Bishop, 2005).

Suppose the Bayesian model $p(y, \theta)$ is represented by a directed graph with nodes representing the variables and arrows expressing the probabilistic relationship between variables. In VB, optimization of the variational posterior can be decomposed into local computations that involve only neighbouring nodes. This leads to fast computational algorithms. Winn and Bishop (2005) developed an algorithm called variational message passing that allows VB to be applied to a very general class of conjugate-exponential models (Attias, 2000; Ghahramani and Beal, 2001) without having to derive application-specific updates. In this algorithm, "messages" are passed between nodes in the graph, and the posterior distribution associated with any particular node can be updated once the node has received messages from all of its neighbouring nodes. Knowles and Minka (2011) proposed an algorithm called nonconjugate variational message passing to extend variational message passing to nonconjugate models.

For computational efficiency, VB methods often rely on analytic solutions to integrals and conjugacy in the posterior. This limits the type of approximations and posteriors VB can handle. Recent developments in VB methods seek to overcome this restriction by branching out into stochastic optimization (e.g., Paisley *et al.*, 2012; Salimans and Knowles, 2012). More details are given in Section 5.1. Wand *et al.* (2011) developed some strategies to handle models whose VB parameter updates do not admit closed form solutions by making use of auxiliary variables, quadrature schemes and finite mixture approximations of difficult density functions.

1.1.3 Variational approach to Bayesian model selection

Variational methods provide an important approach to model selection and a number of innovative automated model selection procedures that follow a variational approach have been developed for Gaussian mixture models.

First, let us review briefly the Bayesian approach to model selection, which is usually based traditionally on the Bayes factor. Suppose there are k candidate models, $M_1, \ldots, M_k$. Let $p(M_j)$ and $p(y|M_j)$ denote the prior probability and marginal likelihood of model M_j respectively. Applying Bayes' rule, the posterior probability of model M_j is

$$p(M_j|y) = \frac{p(M_j)p(y|M_j)}{\sum_{l=1}^{k} p(M_l)p(y|M_l)}.$$

To compare any two models, say M_i and M_j, we consider the posterior odds in favour of model M_i:

$$\frac{p(M_i|y)}{p(M_j|y)} = \frac{p(M_i)p(y|M_i)}{p(M_j)p(y|M_j)}.$$

The ratio of the marginal likelihoods, $\frac{p(y|M_i)}{p(y|M_j)}$, is the Bayes factor and can be considered as the strength of evidence provided by the data in favour of model M_i over M_j. Therefore, model comparison can be performed using marginal likelihoods once a prior has been specified on the models. See O'Hagan and Forster (2004) for a review of Bayes factors and alternative methods for Bayesian model choice.

Computing marginal likelihoods for complex models is not straightforward (see, e.g., Frühwirth-Schnatter, 2004) and in the variational approximation literature, it is common to replace the log marginal likelihood with the variational lower bound to obtain approximate posterior model probabilities. Corduneanu and Bishop (2001) verified through experiments and comparisons with cross-validation that the variational lower bound is a good score for model selection in Gaussian mixture models. Bishop and Svensén (2003) also considered the use of the variational lower bound in model selection for mixture of regression models. By considering models with varying number of mixture components and multiple runs from random starting points (as the lower bound has many local modes), they demonstrated that the lower bound attained its maximum value when the number of mixture components was optimal.

In mixture models, there are many equivalent modes that arise from component relabelling. For instance, if there are k components, then there

will be $k!$ different modes with equivalent parameter settings. However, variational inference tends to approximate the posterior distribution in one of the modes and ignore others when there is multimodality (Bishop, 2006). This failure to approximate all modes of the true posterior leads to underestimation of the log marginal likelihood by the lower bound. Bishop (2006) suggests adding $\log k!$ to the lower bound when using it for model comparison. See Bishop (2006) and Paquet *et al.* (2009) for further discussion. In Chapter 3, we do not attempt any adjustment when using the lower bound in the variational greedy algorithm as we find that the $\log k!$ correction tends to be too large when k is large and modes overlap.

Another advantage of variational methods is the potential for simultaneous parameter estimation and model selection. Attias (1999) observed that when mixture models are fitted using VB, competition between components with similar parameters will result in weightings of redundant components decreasing to zero. This component elimination property was used by several authors to develop algorithms with automatic model selection for Gaussian mixtures. For instance, Corduneanu and Bishop (2001) estimate mixing coefficients by optimizing a variational lower bound on the log marginal likelihood, where all parameters except the mixing coefficients are integrated out. They demonstrated that by initializing the algorithm with a large number of components, mixture components whose weightings become sufficiently small can be removed, leading to automatic model selection. McGrory and Titterington (2007) considered a similar approach using a different model hierarchy and extended the deviance information criterion of Spiegelhalter *et al.* (2002a) to VB methods. These were used to validate the automatic model selection in VB. On the other hand, Ueda and Ghahramani (2002) proposed using a VB split and merge EM (expectation maximization) procedure to optimize an objective function that can perform model selection and parameter estimation for Gaussian mixtures simultaneously. Building upon past split operations proposed previously (see also Ghahramani and Beal, 2000), Wu *et al.* (2012) proposed a new goodness-of-fit measure for evaluating mixture models and developed a split and eliminate VB algorithm which identifies components fitted poorly using two types of split operations. All poorly fitted components were then split at the same time. No merge moves are required as the algorithm makes use of the component elimination property associated with VB. Constantinopoulos and Likas (2007) observed that in the component elimination approach of McGrory and Titterington (2007), the number of components in the re-

sulting mixture can be sensitive to the prior on the precision matrix. They proposed an incremental approach where components are added to the mixture following a splitting test which takes into account characteristics of the precision matrix of the component being tested.

1.2 Contributions

In this thesis, we consider some highly flexible models, namely, mixture of heteroscedastic regression (MHR) models, mixture of linear mixed models (MLMM) and the generalized linear mixed model (GLMM). Fast variational approximation methods are developed for fitting them. We also investigate the use of reparametrization techniques and stochastic approximation methods for improving the convergence of variational algorithms.

Chapter 2 considers the problem of regression density estimation and the use of MHR models to flexibly estimate a response distribution smoothly as a function of covariates. In a MHR model, the response distribution is a normal mixture, with the component means, variances and mixture weights all varying as a function of covariates. We develop fast variational approximation methods for inference in MHR models, where the variational lower bound is in closed form. Our motivation is that alternative computationally intensive MCMC methods are difficult to apply when it is desired to fit models repeatedly in exploratory analysis and in cross-validation for model choice. We also improve the basic variational approximation by using stochastic approximation methods to perturb the initial solution so as to attain higher accuracy. The advantages of variational methods as compared to MCMC methods in model choice are illustrated with real examples.

In Chapter 3, we consider MLMMs which are very useful for clustering grouped data. The conventional approach to estimating MLMMs is by likelihood maximization through the EM algorithm. A suitable number of components is then determined by comparing different mixture models using penalized log-likelihood criteria such as BIC (Bayesian information criterion). Our motivation for fitting MLMMs with variational methods is that parameter estimation and model selection can be performed simultaneously. We describe a variational approximation for MLMMs where the variational lower bound is in closed form, allowing for fast evaluation and develop a novel variational greedy algorithm for model selection and learning of the mixture components. This approach handles algorithm initialization and returns a plausible number of mixture components automatically. In cases of weak identifiability of certain model parameters, we use hierar-

chical centering to reparametrize the model and show empirically that there is a gain in efficiency in variational algorithms similar to that in MCMC algorithms. Related to this, we prove that the approximate rate of convergence of variational algorithms by Gaussian approximation is equal to that of the corresponding Gibbs sampler, which suggests that reparametrizations can lead to improved convergence in variational algorithms just as in MCMC algorithms.

We turn to GLMMs in Chapter 4. We show how GLMMs can be fitted using nonconjugate variational message passing and demonstrate that this algorithm is faster than MCMC methods by an order of magnitude which is especially important in large scale applications. In addition, we examine the effects of reparametrization techniques such as centering, noncentering and partial noncentering in the context of VB for GLMMs. These techniques have been used to accelerate convergence for hierarchical models in MCMC and EM algorithms but are still not well studied for VB methods. The use of different parametrizations for VB has not only computational but also statistical implications as different parametrizations are associated with different factorized posterior approximations. We show that the partially noncentered parametrization can adapt to the quantity of information in the data and automatically determine a parametrization close to optimal. Moreover, partial noncentering can accelerate convergence and produce more accurate posterior approximations than centering or noncentering. Standard model selection criteria such as AIC (Akaike information criteria) or BIC are difficult to apply to GLMMs and we demonstrate how the variational lower bound, a by-product of the nonconjugate variational message passing algorithm, can be useful for model selection.

The nonconjugate variational message algorithm for GLMMs has to iterate between updating local variational parameters associated with individual observations and global variational parameters and becomes increasingly inefficient for large data sets. In Chapter 5, we extend stochastic variational inference for conjugate-exponential models to nonconjugate models and present a stochastic version of nonconjugate variational message passing for fitting GLMMs that is scalable to large data sets. This is achieved by combining updates in nonconjugate variational message passing with stochastic natural gradient optimization of the variational lower bound. In addition, we show that diagnostics for prior-likelihood conflict, which are very useful for model criticism, can be obtained from nonconjugate variational message passing automatically, as an alternative to simulation-based,

computationally intensive MCMC methods. Finally, we demonstrate that for moderate-sized data sets, convergence can be accelerated by using the stochastic version of nonconjugate variational message passing in the initial stage of optimization before switching to the standard version.

The materials presented in this thesis have either been published or submitted for publication. Results in Chapter 2, Chapter 3 and Chapter 4 have been published in Nott *et al.* (2012), Tan and Nott (2013a) and Tan and Nott (2013b) respectively. Results in Chapter 5 are covered in Tan and Nott (2013c) which has been submitted for publication.

1.3 Notation

Here we introduce some notation that will apply throughout the thesis.

The determinant of a square matrix A is denoted by $|A|$ and the transpose of any matrix B is denoted by B^T. We use 1_d to denote the $d\times 1$ column vector with all entries equal to 1 and I_d to denote the $d \times d$ identity matrix. Let $a = [a_1, a_2, a_3]^T$ and $b = [b_1, b_2, b_3]^T$. We adopt the convention that scalar functions such as $\exp(\cdot)$ applied to vector arguments are evaluated element by element. For example, $\exp(a) = [\exp(a_1), \exp(a_2), \exp(a_3)]^T$. We use $\odot$ to denote element by element multiplication of two vectors. For example, $a \odot b = [a_1b_1, a_2b_2, a_3b_3]$. The kronecker product between any two matrices is denoted by $\otimes$.

For a $d \times d$ square matrix A, we let $\text{diag}(A)$ denote the $d \times 1$ vector containing the diagonal entries of A and $\text{vec}(A)$ denotes the $d^2 \times 1$ vector obtained by stacking the columns of A under each other, from left to right in order. In addition, $\text{vech}(A)$ denotes the $\frac{1}{2}d(d+1) \times 1$ vector obtained from $\text{vec}(A)$ by eliminating all supradiagonal elements of A. See Magnus and Neudecker (1988) for more details. On the other hand, if a is a $d \times 1$ vector, $\text{diag}(a)$ is used to denote the $d \times d$ diagonal matrix with diagonal entries given by the vector a.

We let $N(\mu, \Sigma)$ denote the normal distribution with mean μ and covariance matrix Σ. The Gaussian density of a random variable x with mean μ and standard deviation σ is denoted by $\phi(x; \mu, \sigma)$. Let $\Gamma(\cdot)$ denote the Gamma function given by $\Gamma(x) = \int_0^\infty u^{x-1} \exp(-u)\, du$ and $\psi(\cdot)$ denote the digamma function given by $\psi(x) = \frac{d}{dx} \log \Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$. We use $IG(\alpha, \lambda)$ to denote the inverse gamma distribution with density function $\frac{\lambda^\alpha}{\Gamma(\alpha)} x^{-(\alpha+1)} \exp\left(-\frac{\lambda}{x}\right)$ defined for $x > 0$. We use $IW(\nu, S)$ to denote the

inverse-Wishart distribution with density function given by

$$\left\{2^{\frac{\nu r}{2}}\pi^{\frac{r(r-1)}{4}}\prod_{l=1}^{r}\Gamma\left(\frac{\nu+1-l}{2}\right)\right\}^{-1}|S|^{\frac{\nu}{2}}|D|^{-\frac{\nu+r+1}{2}}\exp\{-\tfrac{1}{2}\mathrm{tr}(SD^{-1})\},$$

for an $r \times r$ matrix D. The degrees of freedom is ν and S is a symmetric, positive definite $r \times r$ scale matrix.

Chapter 2

Regression density estimation with variational methods and stochastic approximation

In this chapter, we consider the problem of regression density estimation, that is, how to model a response distribution so that it varies smoothly as a function of the covariates. Finite mixture models provide an important approach to regression density estimation and here we consider mixture of heteroscedastic regression (MHR) models where the response distribution is a normal mixture, with the component means, variances and mixing weights all varying with covariates. Each component is described by a heteroscedastic linear regression model and the component weights by a multinomial logit model. This allowance for heteroscedasticity is important as simulations by Villani *et al.* (2009) showed that when models with homoscedastic components are used to model heteroscedastic data, their performance become worse as the number of covariates increases. There is also a limit as to how much their performance can be improved by merely increasing the number of mixture components. Another advantage of MHR models is that the same level of performance can be achieved with fewer components as was shown in Li *et al.* (2011) using the benchmark LIDAR data. This makes estimating and interpreting the mixture model an easier task. Moreover, MHR models can also be used for fitting homoscedastic data (see Villani *et al.*, 2009).

Fitting mixture models with MCMC methods can be computationally intensive, especially when models have to be fitted repeatedly in exploratory analysis or model choice using cross-validation. We develop fast variational approximation methods for fitting MHR models where the variational lower bound is in closed form and updates can be computed effi-

ciently. We demonstrate the advantages of our approach as compared to MCMC methods in model choice and evaluation. The advantages are significant for time series data, where model refitting is common in repeated one-step ahead prediction (Geweke and Amisano, 2010) and rolling window computations to check for model stability (Pesaran and Timmermann, 2002). Variational methods are particularly suitable for this type of refitting as variational parameters obtained from a previous fit can be used to initialize the next one. The computational speed up arising from such "warm starts" are quantified in an example. Finally, we propose to improve the basic variational approximation by integrating out the mixture component indicators from the posterior and perturbing the initial solution using stochastic approximation methods (see, e.g. Spall, 2003). Results indicate that the stochastic approximation correction is very helpful in attaining better accuracy and requires less computation time than MCMC methods.

This chapter is organized as follows. Section 2.1 provides some background. Section 2.2 defines MHR models and Section 2.3 describes fast variational methods for fitting them. Section 2.4 discusses model choice using a variational approach and Section 2.5 describes how the basic variational approximation can be improved by using a stochastic approximation correction. Section 2.6 considers examples involving real data and Section 2.7 concludes.

Results presented in this chapter have been published in Nott *et al.* (2012).

2.1 Background

MHR models extend conventional mixture of regression models by allowing the component models to be heteroscedastic. In machine learning, mixtures of regression models are commonly referred to as mixtures of experts (Jacobs *et al.*, 1991; Jordan and Jacobs, 1994), in which the individual component distributions are called experts and the mixing coefficients are termed gating functions. Mixtures of regression models are also known as concomitant variable mixture regression models in marketing (e.g. Wedel, 2002) or as mixtures of generalized linear models when the individual component distributions are generalized linear models. Previously, Villani *et al.* (2009) have considered MHR models where the means, variances and mixing probabilities are modelled using spline basis function expansions with a variable selection prior. Bayesian inference was obtained by using MCMC methods in Villani *et al.* (2009).

Mixtures of regression models are highly flexible and can be fitted using likelihood maximization through the EM algorithm (e.g. Jordan and Jacobs, 1994). Recent Bayesian approaches use MCMC methods for inference (e.g. Peng *et al.*, 1996; Wood *et al.*, 2002; Geweke and Keane, 2007). A number of authors have also considered variational methods although they did not consider heteroscedastic components (Waterhouse *et al.*, 1996; Ueda and Ghahramani, 2002; Bishop and Svensén, 2003). Innovative approaches to model selection that follow from variational methods have been proposed for mixtures of regression models as well as Gaussian mixtures and a brief review is given in Section 1.1.3.

Jiang and Tanner (1999) study the rate at which mixtures of regression models approximate the true density and the consistency of maximum likelihood estimation in the case where the response follows a one-parameter exponential family regression model. Norets (2010) showed that a large class of conditional densities can be approximated in the sense of the Kullback-Leibler distance by using different types of finite smooth normal mixtures and derived approximation error bounds. Some insights on when additional flexibility might be most usefully employed in the mean, variance and gating functions are also provided.

Research in Bayesian nonparametric approaches to regression density estimation relating to mixtures of regression models is currently very active (e.g. MacEachern, 1999; De Iorio *et al.*, 2004; Griffin and Steel, 2006; Dunson *et al.*, 2007). Instead of considering finite mixtures of regressions, it is possible to place a prior such as the Dirichlet process prior on the mixing distribution. For some common priors, the resulting model can be considered as mixtures with an infinite number of components. This approach avoids the difficulty of determining a suitable number of mixture components, although a finite mixture may be easier to interpret and communicate to scientific practitioners.

A central approach to stochastic optimization is the root-finding stochastic approximation algorithm of Robbins and Monro (1951). Here we consider optimization of the variational lower bound through stochastic gradient approximation (see, e.g. Spall, 2003). A similar approach was proposed by Ji *et al.* (2010), but we offer several improvements, such as an improved gradient estimate and a strategy of perturbing only the mean and scale of an initial variational approximation. Perturbing an existing solution keeps the dimension of optimization low which is important for a fast and stable implementation. Ji *et al.* (2010) also propose using Monte Carlo samples

to optimize upper and lower bounds on the marginal likelihood.

2.2 Mixtures of heteroscedastic regression models

Suppose that responses $y_1, \ldots, y_n$ are observed. For each $i = 1, \ldots, n$, y_i is modelled by a MHR model of the form:

$$y_i|\delta_i, \beta, \alpha \sim N\big(x_i^T \beta_{\delta_i},\ \exp(u_i^T \alpha_{\delta_i})\big),$$

where δ_i is a categorical latent variable with k categories, $\delta_i \in \{1, \ldots, k\}$, $x_i = [x_{i1}, \ldots, x_{ip}]^T$ and $u_i = [u_{i1}, \ldots, u_{im}]^T$ are vectors of covariates, and $\beta_j = [\beta_{j1}, \ldots, \beta_{jp}]^T$ and $\alpha_j = [\alpha_{j1}, \ldots, \alpha_{jm}]^T$, $j = 1, \ldots, k$, are vectors of unknown parameters. Conditional on $\delta_i = j$, the response follows a heteroscedastic linear model with mean $x_i^T \beta_j$ and log variance $u_i^T \alpha_j$. The mixing distribution for δ_i is

$$P(\delta_i = j|\gamma) = p_{ij}(\gamma) = \frac{\exp(\gamma_j^T v_i)}{\sum_{l=1}^k \exp(\gamma_l^T v_i)}, \quad j = 1, \ldots, k,$$

where $v_i = [v_{i1}, \ldots, v_{ir}]^T$ is a vector of covariates, γ_1 is set as identically zero for identifiability, $\gamma_j = [\gamma_{j1}, \ldots, \gamma_{jr}]^T$, $j = 2, \ldots, k$, are vectors of unknown parameters and $\gamma = [\gamma_2^T, \ldots, \gamma_k^T]^T$. With this prior, the responses are modelled as a mixture of heteroscedastic linear regressions where the mixture weights vary with covariates. For Bayesian inference, we specify the following independent prior distributions on the unknown parameters: $\beta_j \sim N(\mu^0_{\beta_j}, \Sigma^0_{\beta_j})$ and $\alpha_j \sim N(\mu^0_{\alpha_j}, \Sigma^0_{\alpha_j})$ for $j = 1, \ldots, k$ and $\gamma \sim N(\mu^0_\gamma, \Sigma^0_\gamma)$. Let $y = [y_1, \ldots, y_n]^T$, $X = [x_1, \ldots, x_n]^T$, $U = [u_1, \ldots, u_n]^T$, $V = [v_1, \ldots, v_n]^T$, $\delta = [\delta_1, \ldots, \delta_n]^T$, $\beta = [\beta_1^T, \ldots \beta_k^T]^T$, $\alpha = [\alpha_1^T, \ldots \alpha_k^T]^T$ and $\theta = \{\delta, \beta, \alpha, \gamma\}$ denote the set of all unknown parameters. Fast variational approximation methods for MHR models are described in the next section. Variational inference has been considered for mixtures of regression models but not for the case of heteroscedastic mixture components and we demonstrate that a variational lower bound can still be computed in closed form in this case.

2.3 Variational approximation

We consider a variational approximation to the joint posterior $p(\theta|y)$ of the form

$$q(\theta) = q(\delta)q(\beta)q(\alpha)q(\gamma), \tag{2.1}$$

where

$$q(\delta) = \prod_{i=1}^{n} q(\delta_i), \quad q(\beta) = \prod_{j=1}^{k} q(\beta_j), \quad q(\alpha) = \prod_{j=1}^{k} q(\alpha_j) \tag{2.2}$$

and $q(\beta_j)$ is $N(\mu^q_{\beta_j}, \Sigma^q_{\beta_j})$, $q(\alpha_j)$ is $N(\mu^q_{\alpha_j}, \Sigma^q_{\alpha_j})$, $q(\gamma)$ is a delta function placing point mass of 1 on μ^q_γ, and $q(\delta_i = j) = q_{ij}$ for $i = 1, \ldots, n$, $j = 1, \ldots, k$, with $\sum_{j=1}^{k} q_{ij} = 1$ for each i. Bishop (2006) noted that q_{ij} can be interpreted as a measure of the responsibility undertaken by component j in explaining the ith observation. Here a parametric form is chosen for $q(\theta)$ and we attempt to make $q(\theta)$ a good approximation to $p(\theta|y)$ by choosing the variational parameters to minimize the Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|y)$. From (1.2), this is equivalent to maximizing the variational lower bound $\mathcal{L}$ with respect to the variational parameters.

We note that the product forms of $q(\delta)$, $q(\beta)$ and $q(\alpha)$ assumed in (2.2) also arise as optimal solutions of the product restriction in (2.1) through application of (1.4). The densities assumed for $q(\beta_j)$ and $q(\delta_i)$ are also the optimal densities which arise through application of (1.4). The optimal densities of $q(\alpha_j)$ and $q(\gamma)$ do not belong to recognizable densities however and we have assumed specific parametric forms for them. In particular, a degenerate point mass variational posterior has been assumed for γ so that computation of the lower bound is tractable. We suggest a method for relaxing $q(\gamma)$ to be a normal distribution after first describing a variational algorithm which uses the point mass form for $q(\gamma)$.

Unlike previous developments of variational methods for mixture models with homoscedastic components (e.g. Bishop and Svensén, 2003), it is not straightforward to derive a closed form of the variational lower bound in the heteroscedastic case and we also have to handle optimization of the variance parameters, $\mu^q_{\alpha_j}$ and $\Sigma^q_{\alpha_j}$, in the variational posterior. These variance parameters cannot be optimized in closed form and we develop computationally efficient approximate methods for dealing with them.

At the moment, we are considering only a fixed point estimate for γ. Suppose $\theta_{-\gamma}$ denotes the set of unknown parameters excluding γ. We have $p(y|\gamma) = \int p(y|\theta)p(\theta_{-\gamma}|\gamma)\, d\theta_{-\gamma}$ and

$$\begin{aligned}
\log p(\gamma)p(y|\gamma) &= \log \int p(y|\theta)p(\theta)\, d\theta_{-\gamma} \\
&= \log \int q(\theta_{-\gamma}) \frac{p(y,\theta)}{q(\theta_{-\gamma})}\, d\theta_{-\gamma} \\
&\geq \int q(\theta_{-\gamma}) \log \frac{p(y,\theta)}{q(\theta_{-\gamma})}\, d\theta_{-\gamma} \text{ (by Jensen's inequality)}
\end{aligned} \tag{2.3}$$

This implies that $\mathcal{L} = E_q\{\log p(y,\theta)\} - E_q\{\log q(\theta_{-\gamma})\}$ where $E_q(\cdot)$ denotes expectation with respect to $q(\theta)$, gives a lower bound on $\sup_\gamma \log p(\gamma)p(y|\gamma)$. This lower bound can be computed in closed form (see details in Appendix A) and is given by

$$\begin{aligned}\mathcal{L} = \tfrac{1}{2}\sum_{j=1}^{k}\Big\{ &\log|{\Sigma^0_{\beta_j}}^{-1}\Sigma^q_{\beta_j}| - \mathrm{tr}({\Sigma^0_{\beta_j}}^{-1}\Sigma^q_{\beta_j}) - (\mu^q_{\beta_j}-\mu^0_{\beta_j})^T{\Sigma^0_{\beta_j}}^{-1}(\mu^q_{\beta_j}-\mu^0_{\beta_j}) \\ &+ \log|{\Sigma^0_{\alpha_j}}^{-1}\Sigma^q_{\alpha_j}| - \mathrm{tr}({\Sigma^0_{\alpha_j}}^{-1}\Sigma^q_{\alpha_j}) - (\mu^q_{\alpha_j}-\mu^0_{\alpha_j})^T{\Sigma^0_{\alpha_j}}^{-1}(\mu^q_{\alpha_j}-\mu^0_{\alpha_j})\Big\} \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\Big\{\log p_{ij}(\mu^q_\gamma) - \tfrac{1}{2}u_i^T\mu^q_{\alpha_j} - \tfrac{1}{2}w_{ij}\exp\left(\tfrac{1}{2}u_i^T\Sigma^q_{\alpha_j}u_i - u_i^T\mu^q_{\alpha_j}\right) \\ &- \log q_{ij}\Big\} - \tfrac{n}{2}\log 2\pi + \tfrac{(p+m)k}{2} + \log p(\mu^q_\gamma), \end{aligned} \tag{2.4}$$

where $w_{ij} = (y_i - x_i^T\mu^q_{\beta_j})^2 + x_i^T\Sigma^q_{\beta_j}x_i$ and $p(\mu^q_\gamma)$ is the prior distribution for γ evaluated at μ^q_γ.

The variational parameters to be optimized consist of $\mu^q_{\beta_j}$, $\Sigma^q_{\beta_j}$, $\mu^q_{\alpha_j}$, $\Sigma^q_{\alpha_j}$ for $j = 1,\ldots,k$, μ^q_γ and q_{ij} for $i = 1,\ldots,n$, $j = 1,\ldots,k$. We optimize the lower bound with respect to each of these sets of parameters with the others held fixed in a gradient ascent algorithm. This leads to the iterative scheme in Algorithm 1. The updates in steps 1 and 5 can be derived using vector differential calculus (see, e.g. Wand, 2002) or from application of (1.4).

Algorithm 1: Variational approximation for MHR model

Generate an initial clustering of the data. Initialize $\mu^q_{\alpha_j} = 0$ and $\Sigma^q_{\alpha_j} = 0$ for $j = 1,\ldots,k$ and q_{ij} as 1 if the ith observation lies in cluster j and 0 otherwise for $i = 1,\ldots,n$, $j = 1,\ldots,k$.
Cycle:

1. For $j = 1,\ldots,k$,
   - $\Sigma^q_{\beta_j} \leftarrow \left({\Sigma^0_{\beta_j}}^{-1} + X^TD_jX\right)^{-1}$,
   - $\mu^q_{\beta_j} \leftarrow \Sigma^q_{\beta_j}\left({\Sigma^0_{\beta_j}}^{-1}\mu^0_{\beta_j} + X^TD_jy\right)$,

   where D_j is a $n\times n$ diagonal matrix with the ith diagonal entry given by $q_{ij}\exp\left(\frac{1}{2}u_i^T\Sigma^q_{\alpha_j}u_i - u_i^T\mu^q_{\alpha_j}\right)$.
2. For $j = 1,\ldots,k$, set $\mu^q_{\alpha_j}$ to be the conditional mode of the lower bound with other variational parameters fixed at current values.

3. For $j = 1, \ldots, k$, $\Sigma^q_{\alpha_j} \leftarrow \left(\Sigma^{0\ -1}_{\alpha_j} + U^T W_j U\right)^{-1}$, where W_j is a $n \times n$ diagonal matrix with ith diagonal entry given by $\frac{1}{2} q_{ij} w_{ij} \exp(-u_i^T \mu^q_{\alpha_j})$. This update is performed only if it leads to a higher lower bound.

4. Set μ^q_γ to be the conditional mode of the lower bound fixing other variational parameters at their current values.

5. For $i = 1, \ldots, n$, $j = 1, \ldots, k$, $q_{ij} \leftarrow \dfrac{p_{ij}(\mu^q_\gamma) \exp(b_{ij})}{\sum_{l=1}^k p_{il}(\mu^q_\gamma) \exp(b_{il})}$, where

   $b_{il} = -\frac{1}{2} u_i^T \mu^q_{\alpha l} - \frac{1}{2} w_{ij} \exp\left(\frac{1}{2} u_i^T \Sigma^q_{\alpha l} u_i - u_i^T \mu^q_{\alpha l}\right)$ for $l = 1, \ldots, k$.

until the increase in $\mathcal{L}$ is negligible.

Consider the update of $\mu^q_{\alpha_j}$ in step 2. As a function of $\mu^q_{\alpha_j}$, the lower bound is (ignoring irrelevant additive constants)

$$-\tfrac{1}{2} \sum_{i=1}^n q_{ij} \Big\{ u_i^T \mu^q_{\alpha_j} + w_{ij} \exp\left(\tfrac{1}{2} u_i^T \Sigma^q_{\alpha_j} u_i - u_i^T \mu^q_{\alpha_j}\right) \Big\} - \tfrac{1}{2} (\mu^q_{\alpha_j} - \mu^0_{\alpha_j})^T \Sigma^{0\ -1}_{\alpha_j} (\mu^q_{\alpha_j} - \mu^0_{\alpha_j}).$$

This is the log posterior of a generalized linear model with normal prior $N(\mu^0_{\alpha_j}, \Sigma^0_{\alpha_j})$, gamma responses w_{ij} and coefficients of variation $\sqrt{\frac{2}{q_{ij}}}$. The log of the mean is $u_i^T \mu^q_{\alpha_j} - \frac{1}{2} u_i^T \Sigma^q_{\alpha_j} u_i$ where $-\frac{1}{2} u_i^T \Sigma^q_{\alpha_j} u_i$ define an offset. Although the mode has no closed form expression it can be easily found using an iteratively weighted least squares approach (McCullagh and Nelder, 1989; West, 1985) or some other numerical optimization technique.

We have used an approximation in the update of $\Sigma^q_{\alpha_j}$ in step 3 and our motivation comes from the following. Suppose we relax the restriction that $q(\alpha_j)$ is a normal distribution. From (1.4), the optimal $q(\alpha_j)$ which maximizes the lower bound would satisfy

$$q(\alpha_j) \propto \exp\Big[-\tfrac{1}{2} \sum_{i=1}^n q_{ij} \big\{ u_i^T \alpha_j + w_{ij} \exp(-u_i^T \alpha_j) \big\} - \tfrac{1}{2} (\alpha_j - \mu^0_{\alpha_j})^T \Sigma^{0\ -1}_{\alpha_j} (\alpha_j - \mu^0_{\alpha_j}) \Big]. \tag{2.5}$$

If $\mu^q_{\alpha_j}$ is close to the mode, we can obtain a normal approximation to $q(\alpha_j)$ by taking the mean as $\mu^q_{\alpha_j}$ and the covariance matrix as the negative inverse Hessian of the log of (2.5) at $\mu^q_{\alpha_j}$. The negative inverse Hessian at $\mu^q_{\alpha_j}$ works out to be $(\Sigma^{0\ -1}_{\alpha_j} + U^T W_j Z)^{-1}$ with W_j defined as in step 3 of Algorithm 1. Waterhouse *et al.* (1996) used a similar reasoning in approximating the posterior distribution of the mixing weights model parameters

for a homoscedastic mixture model. The update in step 3 is performed only if it improves the lower bound.

For the update of μ_γ^q in step 5, note that as a function of μ_γ^q, the lower bound is (ignoring irrelevant additive constants)

$$\log p(\mu_\gamma^q) + \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij} \log p_{ij}(\mu_\gamma^q).$$

This is the log posterior for a Bayesian multinomial regression with normal prior on μ_γ^q and where the ith response is $[q_{i1}, \ldots, q_{ik}]^T$. In a typical multinomial regression, only one component of this pseudo-response vector would be 1 with the other terms 0 and although this is not the case here, iteratively weighted least squares (or some other numerical optimization algorithm) can be used for finding the mode.

At convergence, we suggest replacing the point estimate variational posterior for γ with a normal approximation, where the mean is μ_γ^q and the covariance matrix Σ_γ^q is the negative inverse Hessian of the Bayesian multinomial log posterior considered in step 4 of Algorithm 1. The justification for this approximation is similar to our justification for the update of $\Sigma_{\alpha_j}^q$ in step 3 of Algorithm 1. Waterhouse *et al.* (1996) discuss a similar approximation which they use at every step of their iterative algorithm while we use only a one-step approximation after first using a point estimate for the posterior distribution for γ. With this normal approximation, the variational lower bound on $\log p(y)$ is the same as (2.4), except that $\sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij} \log p_{ij}(\mu_\gamma^q) + \log p(\mu_\gamma^q)$ has to be replaced with

$$\sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij} E_q\big\{\log p_{ij}(\gamma)\big\} - \tfrac{1}{2}(\mu_\gamma^q - \mu_\gamma^0)^T {\Sigma_\gamma^0}^{-1}(\mu_\gamma^q - \mu_\gamma^0) - \tfrac{1}{2}\log|\Sigma_\gamma^0| - \tfrac{1}{2}\mathrm{tr}\left({\Sigma_\gamma^0}^{-1}\Sigma_\gamma^q\right) + \tfrac{1}{2}\log|\Sigma_\gamma^q| + \tfrac{r(k-1)}{2}.$$

The expectation in the first term is not available in closed form and we replace $E_q\big\{\log p_{ij}(\gamma)\big\}$ with $\log p_{ij}(\mu_\gamma^q)$ to obtain an estimate $\mathcal{L}^*$ which might be used as an approximation to $\log p(y)$.

The iterative scheme in Algorithm 1 guarantees convergence only to a local mode and we suggest running the algorithm from multiple starting points to deal with the issue of multiple modes. For the examples in Section 2.6, we consider random clusterings in the initialization where each observation is randomly and equally likely to be assigned to any of the mixture components. For each random clustering, we would perform a "short run",

where Algorithm 1 is terminated once the increase in the lower bound is less than 1. From a total of 20 of these "short runs", we select the one with the highest attained lower bound and follow only this run to full convergence. This "short runs" strategy is similar to one that is recommended for initialization of the EM algorithm, for maximum likelihood estimation of Gaussian mixture models, by Biernacki *et al.* (2003).

We also observed that sometimes, components may "fall out" during the fitting process, in the sense that q_{ij} will go to zero for all observations i, for some mixture component j. This phenomenon is dependent on the initial clustering and is likely to happen when Algorithm 1 is initialized with a larger than required number of components. McGrory and Titterington (2007) propose using this component elimination feature to perform model selection in the fitting of Gaussian mixtures using VB (see Section 1.1.3). We focus on model choice using cross-validation for MHR models.

It has been observed (e.g. Qi and Jaakkola, 2006), that the convergence of VB algorithms can be very slow when parameters are highly correlated between the blocks used in the variational factorization. This can happen, for instance, when two mixture components are very similar. This is a complex problem and we do not see any easy solution. One possible solution is to integrate out the mixture indicators and use larger blocks for the remaining parameters in the blockwise gradient ascent. However, this will incur a greater computational burden and require the introduction of new approximations to the variational lower bound.

Finally, we note that as the posteriors of β, α and γ are of the same form as their priors, it might be possible to implement Algorithm 1 sequentially for very large data sets. For instance, the data set can be split into smaller batches and the variational posterior approximation learnt from a previous batch can be used as the prior for processing the next one. There may be difficulties with the naive implementation of this idea, however, as the learning may get stuck in a local mode corresponding to the first solution found. We did not implement this idea for the examples in Section 2.6. Honkela and Valpola (2003) discuss an online version of VB learning which is based on maintaining a decaying history of previous samples so that the system is able to forget old solutions in favour of new better ones. Sato (2001) proposed a similar online model selection algorithm based on VB.

2.4 Model choice

Marginal likelihood is a popular approach to Bayesian model comparison. However, Li *et al.* (2010) noted that the marginal likelihood can be sensitive to the prior in the context of density estimation as the prior is not very informative. They argue that cross-validation is a better tool for assessing predictive performance as dependence on the prior is reduced when a subset of the data has been used to update the vague prior. Following Li *et al.* (2010), we carry out model selection for MHR models using likelihood cross-validation. This approach can be computationally expensive and we demonstrate the advantages of using variational approximation as compared to MCMC-based methods for this purpose. In this section, we describe briefly how model selection is carried out using cross-validation.

2.4.1 Cross-validation

In B-fold cross-validation, the data is split randomly into B roughly equal parts, $F_1, \ldots, F_B$, which serve as the test sets. The training sets, $T_1, \ldots, T_B$ are constructed by leaving out $F_1, \ldots, F_B$ from the complete data set respectively. Let y_{F_b} and y_{T_b} denote observations in F_b and T_b respectively. One useful measure of predictive performance that can be used for model choice is the log predictive density score (LPDS) defined as

$$\text{LPDS} = \frac{1}{B}\sum_{b=1}^{B} \log p(y_{F_b}|y_{T_b}),$$

where

$$p(y_{F_b}|y_{T_b}) = \int p(y_{F_b}|\theta)p(\theta|y_{T_b})\, d\theta. \tag{2.6}$$

Here, we assume that y_{F_b} and y_{T_b} are conditionally independent given θ, the set of unknown parameters. This assumption is usually not valid for time series data and modified approaches appropriate for that case are discussed later. For MHR models, $p(y_{F_b}|\theta)$ can be written as

$$p(y_{F_b}|\theta) = \prod_{i \in \text{ index set of } F_b} \left\{ \sum_{j=1}^{k} p_{ij}(\gamma)\phi\big(y_i;\, x_i^T\beta_j, \exp(u_i^T\alpha_j)\big) \right\}.$$

For MCMC-based methods, the integral in (2.6) can be estimated using samples $\theta_1, \ldots, \theta_S$ from the posterior so that

$$p(y_{F_b}|y_{T_b}) \approx \frac{1}{S}\sum_{s=1}^{S} p(y_{F_b}|\theta_s).$$

In the variational approach, we replace $p(\theta|y_{T_b})$ with the variational approximation $q(\theta)$ learned from the training set T_b, and generate $\theta_1, \ldots, \theta_S$, randomly from $q(\theta)$ instead. We use $S = 1000$ for later examples.

2.4.2 Model choice in time series

In Section 2.6.2, we consider autoregressive time series models in the form of MHR models. The cross-validation approach described above is not natural in the time series context and we consider the approach of Geweke and Keane (2007) and Li *et al.* (2010) described below. Let $y_{\leq T} = (y_1, \ldots, y_T)$ denote a training set of T initial observations. Predictive performance for the purpose of model comparison is measured using the logarithmic score for the subsequent T^* observations $y_{>T} = (y_{T+1}, \ldots, y_{T+T^*})$ defined as

$$\text{LPDS} = \sum_{i=1}^{T^*} \log p(y_{T+i}|y_{\leq T+i-1}) \tag{2.7}$$

and

$$p(y_{T+i}|y_{\leq T+i-1}) = \int p(y_{T+i}|\theta, y_{\leq T+i-1}) p(\theta|y_{\leq T+i-1})\, d\theta. \tag{2.8}$$

In (2.8), $p(\theta|y_{\leq T+i-1})$ denotes the posterior distribution for the set of unknown parameters θ based on observed data available at time $T + i - 1$. Note that (2.7) contains T^* terms and from (2.8), each of these terms depends on a different posterior based on an increasing set of observed data. Geweke and Keane (2007) noted that the most accurate way of computing the LPDS is to run an MCMC sampler separately for each of the T^* terms to estimate the posterior distribution required in each case. This procedure is highly demanding computationally and may not be feasible if T^* is large or if the MCMC scheme is slow to converge. While it might be possible to reuse the MCMC samples for successive terms by using ideas from importance sampling, it is difficult to carry out such ideas reliably (see, e.g. Vehtari and Lampinen, 2002, for discussion). To reduce computation time, Li *et al.* (2010) suggest approximating $p(\theta|y_{\leq T+i-1})$ with $p(\theta|y_{\leq T})$ for each of the T^* terms when T is large compared to T^*. They presented some empirical support for the accuracy of this approximation by comparison with

a scheme where the posterior was updated sequentially at every tenth observation in a financial time series example. Finally, the integral in (2.8) can be estimated similarly using the Monte Carlo method described in Section 2.4.1 and we use $S = 1000$ for the examples in Section 2.6.2.

We note that the variational approach is very efficient for carrying out sequential updating. Besides being faster than MCMC, variational approximation can also benefit from a "warm start" since the variational parameters obtained from the fit at a previous time step can be used to initialize optimization at the next time step so that the time to convergence is reduced. This makes variational approaches ideally suited to model choice based on one-step ahead predictions and the LPDS for time series data.

2.5 Improving the basic approximation

It is well known that factorized variational approximations have a tendency to underestimate the variance of posterior distributions (e.g. Wang and Titterington, 2005; Bishop, 2006). Here, we propose a novel approach to improve the accuracy of estimates obtained from variational approximation by using stochastic approximation methods to perturb the initial solution. Ji *et al.* (2010) independently proposed a Monte Carlo stochastic approximation for maximizing the lower bound numerically, which is similar to our approach. However, we offer some improvements on their implementation such as an improved gradient estimate in the stochastic approximation procedure and the idea of perturbing only the mean and scale of an initial variational approximation. The methods described in this section assume that an initial variational approximation has been obtained using Algorithm 1 and serve only to improve the approximations of the posterior distributions of β, α and γ.

2.5.1 Integrating out the latent variables

In Section 2.2, the MHR model was specified using latent variables δ. These latent variables can be integrated out of the model to give

$$p(y_i|\alpha, \beta, \lambda) = \sum_{j=1}^{k} p_{ij}(\gamma)\phi\big(y_i;\, x_i^T\beta_j, \exp(u_i^T\alpha_j)\big)$$

for $i = 1, \ldots, n$. We consider a variational approximation of the form $q(\beta, \alpha, \gamma) = q(\beta)q(\alpha)q(\gamma)$ for the remaining unknown parameters β, α and γ, where $q(\beta) = \prod_{j=1}^{k} q(\beta_j)$ and $q(\alpha) = \prod_{j=1}^{k} q(\alpha_j)$. We assume that $q(\beta_j)$

is $N(\mu^q_{\beta_j} + m^q_{\beta_j}, S^q_{\beta_j}\Sigma^q_{\beta_j}S^q_{\beta_j})$, $q(\alpha_j)$ is $N(\mu^q_{\alpha_j} + m^q_{\alpha_j}, S^q_{\alpha_j}\Sigma^q_{\alpha_j}S^q_{\alpha_j})$ and $q(\gamma)$ is $N(\mu^q_\gamma + m^q_\gamma, S^q_\gamma\Sigma^q_\gamma S^q_\gamma)$ where $\mu^q_{\beta_j}$, $\mu^q_{\alpha_j}$, μ^q_γ, $\Sigma^q_{\beta_j}$, $\Sigma^q_{\alpha_j}$, Σ^q_γ are the converged values from Algorithm 1, $m^q_{\beta_j}$, $m^q_{\alpha_j}$, m^q_γ are vectors which serve as mean corrections and $S^q_{\beta_j}$, $S^q_{\alpha_j}$, S^q_γ are diagonal matrices which help to adjust the posterior variance in the initial variational approximation. As this variational approximation is of the same form as before for the parameters β, α and γ, it might seem like the optimal choices for $m^q_{\beta_j}$, $m^q_{\alpha_j}$, m^q_γ are zero vectors and for $S^q_{\beta_j}$, $S^q_{\alpha_j}$, S^q_γ, identity matrices. However, this is not the case as the latent variables δ have been integrated out from the model. The optimization problem considered here is thus different from before, with no independence assumptions made about the distribution of δ. We consider the following parametrization for the mean and variance corrections:

$$
\begin{aligned}
m^q_{\beta_j} &= d^q_{\beta_j} \odot \sqrt{\operatorname{diag}(\Sigma^q_{\beta_j})}, & S^q_{\beta_j} &= \operatorname{diag}(\exp(v^q_{\beta_j})),\\
m^q_{\alpha_j} &= d^q_{\alpha_j} \odot \sqrt{\operatorname{diag}(\Sigma^q_{\alpha_j})}, & S^q_{\alpha_j} &= \operatorname{diag}(\exp(v^q_{\alpha_j})),\\
m^q_\gamma &= d^q_\gamma \odot \sqrt{\operatorname{diag}(\Sigma^q_\gamma)}, & S^q_\gamma &= \operatorname{diag}(\exp(v^q_\gamma)),
\end{aligned}
$$

where $d^q_{\beta_j}$, $d^q_{\alpha_j}$, d^q_γ, $v^q_{\beta_j}$, $v^q_{\alpha_j}$ and v^q_γ are vectors, for $j = 1, \ldots, k$. The parameters to be adjusted in the variational approximation are thus $d^q_{\beta_j}$, $d^q_{\alpha_j}$, d^q_γ, $v^q_{\beta_j}$ $v^q_{\alpha_j}$ and v^q_γ for $j = 1, \ldots, k$. Adjusting only the means and variances with other parameters held fixed helps to keep the optimization problem low-dimensional, with subsequent reduction in computation time.

Integrating out the latent variables means that less restrictions have to be imposed on the variational approximation. This can help to reduce the Kullback-Leibler divergence between the true posterior and the variational approximation, which leads to an improved lower bound on the log marginal likelihood. However, integrating out the latent variables also moves us out of the context of a tractable lower bound. Next, we describe how the root-finding stochastic approximation algorithm (Robbins and Monro, 1951) can be used for optimizing the lower bound with respect to parameters in the variational approximation. The methods described in Section 2.5.2 are applicable in a general context (not limited to MHR models) and are particularly useful when the lower bound is intractable.

2.5.2 Stochastic gradient algorithm

Let us consider again the general setting where θ denotes the set of unknown parameters, with prior $p(\theta)$ and likelihood $p(y|\theta)$. Let $q(\theta|\lambda)$, assumed to belong to some parametric family with parameters λ, be the variational

approximation of the true posterior $p(\theta|y)$. The lower bound $\mathcal{L}$ in (1.2) then becomes a function of λ such that

$$\mathcal{L}(\lambda) = \int q(\theta|\lambda) \log \frac{p(\theta)p(y|\theta)}{q(\theta|\lambda)}\, d\theta$$

and we are interested in determining the optimal λ which maximizes the lower bound. By converting this problem into one of finding a root of the equation $g(\lambda) \equiv \frac{\partial}{\partial\lambda} L(\lambda) = 0$ and supposing noisy estimates of $g(\lambda)$ are available, we can then make use of the stochastic gradient form of stochastic approximation (see Spall, 2003) for root-finding. Stochastic approximation is a powerful tool for root-finding and optimization, and there is strong theoretical support for its performance. Spall (2003) presents sufficient conditions for the convergence of the stochastic approximation algorithm and one of them requires the noisy estimates of $g(\lambda)$ to be unbiased. As $\mathcal{L}(\lambda)$ is an expectation with respect to $q(\theta|\lambda)$, this condition is satisfied in our case provided it is valid to interchange the derivative $\frac{\partial}{\partial\lambda}$ and the integral. In particular, we have

$$g(\lambda) = \int \log\left\{\frac{p(\theta)p(y|\theta)}{q(\theta|\lambda)}\right\} \frac{\partial \log q(\theta|\lambda)}{\partial\lambda} q(\theta|\lambda)\, d\theta,$$

since

$$\int \frac{\partial \log q(\theta|\lambda)}{\partial\lambda} q(\theta|\lambda)\, d\theta = 0.$$

An unbiased estimate of the gradient $g(\lambda)$ can thus be computed as

$$\hat{g}(\lambda, \theta') = \left[\log\left\{\frac{p(\theta')p(y|\theta')}{q(\theta'|\lambda)}\right\} - c\right] \frac{\partial \log q(\theta'|\lambda)}{\partial\lambda}, \tag{2.9}$$

where θ' is generated from $q(\theta|\lambda)$ and c can be chosen arbitrarily. In addition, we note that

$$\log p(y) = \log \frac{p(\theta)p(y|\theta)}{p(\theta|y)}$$

for every θ. This suggests that if $q(\theta|\lambda)$ is a good approximation to $p(\theta|y)$ (as it might be near the optimal λ), then the term

$$\left[\log\left\{\frac{p(\theta')p(y|\theta')}{q(\theta'|\lambda)}\right\} - c\right]$$

in the gradient estimate will be nearly constant and equal to $\log p(y) - c$, and hence the variance of the gradient estimate will contain a factor roughly equal to $\{\log p(y) - c\}^2$. This suggests that when λ is close to optimal,

taking c close to $\log p(y)$ may help to reduce fluctuations in the gradient estimates. Ji *et al.* (2010) considered a similar approach for optimizing the lower bound but they use $c = 1$, obtained by differentiating directly under the integral sign. From simulations we have conducted (results not shown), choosing $c = 1$ is usually suboptimal as it can result in gradient estimates with very high variance (since $\{\log p(y) - 1\}^2$ is large when $\log p(y)$ is large). Ji *et al.* (2010) counteract variability in the gradient estimates by using multiple simulations from $q(\theta|\lambda)$. In our application to MHR models, we initialize c as $\mathcal{L}^*$, the estimate of $\log p(y)$ from Algorithm 1. As the stochastic approximation algorithm proceeds, we update c with the latest estimate of $\log p(y)$. This is described in more detail later.

With an unbiased estimate of the gradient, we can now use the stochastic gradient algorithm (Algorithm 2) for optimizing the lower bound.

Algorithm 2: Stochastic gradient approximation for MHR model

Let $\lambda^{(1)}$ be some initial estimate of λ.
For $t = 1, \ldots, N$,

1. Simulate $\theta^{(t)} \sim q(\theta|\lambda^{(t)})$.
2. Set $\lambda^{(t+1)} = \lambda^{(t)} + a_t \, \hat{g}(\lambda^{(t)}, \theta^{(t)})$.

Spall (2003, p. 106) presents sufficient conditions for the strong convergence of the iterates $\{\lambda^{(t)}\}$ and one of them, regarding unbiasedness of the gradient estimates, has been discussed earlier. Another condition requires that the gain sequence $\{a_t\}$ satisfy:

$$a_t \to 0, \quad \sum_{t=0}^{\infty} a_t = \infty \ \text{ and } \ \sum_{t=0}^{\infty} a_t^2 < \infty. \tag{2.10}$$

This criteria gives a balance to $\{a_t\}$ so that the gain goes to zero fast enough to dampen out noise effects when optimal λ is close, but sufficiently slow to avoid false convergence. The remaining two conditions place some restrictions on the shape and magnitude of the gradients and are more difficult to verify. In practice, these conditions (which are sufficient but not necessary) serve more as guidelines and Spall (2003) notes that many practical applications have produced good results even when one or more of the conditions are not satisfied. Note that step 2 of Algorithm 2 can be interpreted as a stochastic version of a gradient ascent algorithm update, where step sizes decrease according to a_t.

In the examples, we use a gain sequence of the form $a_t = a/(A + I_t)^\alpha$, where a, A and α are constants to be chosen. We have found it helpful to adapt the step size at each iteration using the method of Delyon and Juditsky (1993), which generalizes the method of Kesten (1958) to the multivariate case. Some extensions of this idea have also been considered in the adaptive MCMC literature (e.g. Andrieu and Thoms, 2008, p. 357). Suppose λ can be partitioned into $\{\lambda_1, \ldots, \lambda_m\}$. We let I_t for λ_l be equal to the number of sign changes in the gradient estimate for λ_l up to iteration t, for each $l = 1, \ldots, m$. Intuitively, sign changes occur more frequently when we are close to the mode so that step sizes should decrease more rapidly when this happens.

The total number of iterations, N, is usually determined according to some computational budget. It is also possible to use stopping criteria based on some notion that the iterates $\{\lambda^{(t)}\}$ have "stabilized". See Spall (2003) for more discussion. An estimate of the log marginal likelihood can also be obtained from the stochastic approximation iterates using

$$\frac{1}{N - N_0} \sum_{i=N_0+1}^{N} \log \frac{p(\theta^{(i)})p(y|\theta^{(i)})}{q(\theta^{(i)}|\lambda^{(i)})}, \tag{2.11}$$

which requires negligible additional computation. Here N_0 denotes the number of initial iterates to discard where we are not yet close to the optimal solution. In our gradient estimate, there is a constant c that we have argued should be chosen to be an estimate of the log marginal likelihood. In our examples, we initialize c as the estimate of the log marginal likelihood from Algorithm 1, and at iteration $t > 1$ of Algorithm 2, we use (2.11) as the estimate for c with $N_0 = 0$ and $N = t - 1$.

The stochastic approximation approach discussed in this section can be used in general for learning parametric variational posteriors and Algorithm 2 is easy to implement provided $q(\theta|\lambda)$ is easy to simulate from.

2.5.3 Computing unbiased gradient estimates

To use Algorithm 2, we have to compute unbiased estimates of the gradients. From (2.9), we need $\frac{\partial \log q(\beta_j)}{\partial d^q_{\beta_j}}$, $\frac{\partial \log q(\alpha_j)}{\partial d^q_{\alpha_j}}$, $\frac{\partial \log q(\gamma)}{\partial d^q_{\gamma}}$, $\frac{\partial \log q(\beta_j)}{\partial v^q_{\beta_j}}$, $\frac{\partial \log q(\alpha_j)}{\partial v^q_{\alpha_j}}$ and $\frac{\partial \log q(\gamma)}{\partial v^q_{\gamma}}$ for $j = 1, \ldots, k$.

It can be shown that

$$\frac{\partial \log q(\beta_j)}{\partial d^q_{\beta_j}} = \sqrt{\text{diag}(\Sigma^q_{\beta_j})} \odot (S^q_{\beta_j}\Sigma^q_{\beta_j}S^q_{\beta_j})^{-1}(\beta_j - m^q_{\beta_j} - \mu^q_{\beta_j}),$$
$$\frac{\partial \log q(\alpha_j)}{\partial d^q_{\alpha_j}} = \sqrt{\text{diag}(\Sigma^q_{\alpha_j})} \odot (S^q_{\alpha_j}\Sigma^q_{\alpha_j}S^q_{\alpha_j})^{-1}(\alpha_j - m^q_{\alpha_j} - \mu^q_{\alpha_j}),$$
$$\frac{\partial \log q(\gamma)}{\partial d^q_{\gamma}} = \sqrt{\text{diag}(\Sigma^q_{\gamma})} \odot (S^q_{\gamma}\Sigma^q_{\gamma}S^q_{\gamma})^{-1}(\gamma - m^q_{\gamma} - \mu^q_{\gamma}),$$
$$\frac{\partial \log q(\beta_j)}{\partial v^q_{\beta_j}} = \text{diag}\big\{(S^q_{\beta_j}\Sigma^q_{\beta_j}S^q_{\beta_j})^{-1}(\beta_j - m^q_{\beta_j} - \mu^q_{\beta_j})(\beta_j - m^q_{\beta_j} - \mu^q_{\beta_j})^T - I\big\},$$
$$\frac{\partial \log q(\alpha_j)}{\partial v^q_{\alpha_j}} = \text{diag}\big\{(S^q_{\alpha_j}\Sigma^q_{\alpha_j}S^q_{\alpha_j})^{-1}(\alpha_j - m^q_{\alpha_j} - \mu^q_{\alpha_j})(\alpha_j - m^q_{\alpha_j} - \mu^q_{\alpha_j})^T - I\big\},$$
$$\frac{\partial \log q(\gamma)}{\partial v^q_{\gamma}} = \text{diag}\left\{(S^q_{\gamma}\Sigma^q_{\gamma}S^q_{\gamma})^{-1}(\gamma - m^q_{\gamma} - \mu^q_{\gamma})(\gamma - m^q_{\gamma} - \mu^q_{\gamma})^T - I\right\},$$

for $j = 1, \ldots, k$. We initialize $d^q_{\beta_j}$, $d^q_{\alpha_j}$, d^q_{γ}, $v^q_{\beta_j}$ $v^q_{\alpha_j}$ and v^q_{γ} as zero vectors in Algorithm 2 for $j = 1, \ldots, k$.

2.6 Examples

Algorithm 1 was initialized using the "short runs" strategy discussed in Section 2.3 and was considered to have converged fully when the relative increase in the lower bound $\mathcal{L}$ between successive iterations is less than 10^{-6}. For the MCMC approach, we considered a random walk Metropolis-Hastings algorithm for the MHR model with latent variables integrated out. The proposal covariances were taken from the fit obtained using variational approximation and parameters were updated in blocks corresponding to the factorized variational posterior. All code was written in the R language and run on an Intel Core i5-2500 3.30 GHz processor workstation.

2.6.1 Emulation of a rainfall-runoff model

In this example, we use MHR models to emulate a deterministic rainfall-runoff model, which is a simplification of the Australian water balance model (AWBM, Boughton, 2004). Our goal is to develop a computationally cheap statistical surrogate for the original model for some characteristic of the model output. Using the emulator in applications where the deterministic model is expensive to run or has to be run many times (e.g. in model calibration) allows similar results to be achieved with computation time reduced by an order of magnitude. O'Hagan (2006) gives an overview of statistical analysis of computer models and model emulation. In the sta-

Model	A	B	C	D	E
$\mathcal{L}^*$ (variational)	-803.4	-688.4	-678.5	-682.8	-729.0
LPDS (variational)	-65.9	-54.5	-51.5	-52.1	-57.2
LPDS (MCMC)	-65.5	-54.2	-51.2	-51.4	-57.4

Table 2.1: Rainfall-runoff data. Marginal log-likelihood estimates from variational approximation (first row), ten-fold cross-validaton LPDS estimated by variational approximation (second row) and MCMC (third row).

tistical literature, Gaussian process models that interpolate model output are often used to construct emulators, but it is often recommended that an independent noise term be included in the model (Pepelyshev, 2010).

The AWBM uses time series of rainfall and evapotranspiration data to estimate catchment streamflow and is widely used in Australia for estimating catchment water yield or design flood estimation. The model has three parameters — the maximum storage capacity S, the base flow index BFI and the baseflow recession factor K. We have model simulations for close to eleven years of average monthly potential evapotranspiration and daily rainfall data for the Barrington River catchment, located in New South Wales, Australia*. The model was run for 500 different values of the parameters (S, K, BFI) generated using a maximin Latin hypercube design. We consider the AWBM streamflow response at a time of peak rainfall input as the response y, and S and K as predictors. The parameter BFI is omitted as the model output at this time is fairly insensitive to it. A small amount of independent normal random noise with standard deviation 0.01 was added to y to avoid degeneracies in the variance model in regions of the space where the response tends to be identically zero.

We consider fitting five models to the data. The first four are MHR models with both predictors, S and K, in the mean and variance models. We label these as models A, B, C and D having 2, 3, 4 and 5 mixture components respectively. The fifth model, model E, has four mixture components but only an intercept in the variance model and is thus homoscedastic. For the normal prior distributions, we used $\mu^0_{\beta_j} = 0$, $\Sigma^0_{\beta_j} = 10000I$, $\mu^0_{\alpha_j} = 0$, $\Sigma_{\alpha_j} = 100I$, $\mu^0_\gamma = 0$ and $\Sigma^0_\gamma = 100I$, where dimensions of the mean vectors and covariance matrices depend on the model fitted.

Table 2.1 shows the estimates of marginal log-likelihoods estimated from variational approximation (first row) and ten-fold cross-validation LPDS values computed using variational approximation (second row) and MCMC (third row). We focus on model selection for MHR models using

*We thank Lucy Marshall for supplying this data set.

Model		A	B	C	D	E
Full data	variational	88	146	215	274	254
	MCMC	330	473	650	825	659
Cross-validation	variational	121	184	281	393	276
	MCMC	2941	4409	5979	7626	5929

Table 2.2: Rainfall-runoff data. CPU times (in seconds) for full data and cross-validation calculations using variational approximation and MCMC.

cross-validation as discussed in Section 2.4. There is very good agreement between the LPDS estimated by variational approximation and MCMC, and both methods indicate that model C, a mixture with 4 heteroscedastic components, is adequate. The MCMC results for model D need to be treated with some caution as there is very slow mixing in the MCMC scheme here due to the use of too many mixture components and hence a poorly identified model. On the other hand, one of the mixture components was automatically eliminated when model D was fitted using variational approximation as the mixing weights for all observations went to zero for one of the components. It is interesting to note that model C also has the highest estimated marginal log-likelihood. The fit of model C obtained using variational approximation is summarized in Figure 2.1. Here, each observation has been assigned to the mixture component it is most likely to belong to and observations for each mixture component have been plotted along with the fitted mean and standard deviation. The different rows correspond to different mixture components.

The CPU times taken to fit the full data set and implement ten-fold cross-validation using both variational approximation and MCMC are shown in Table 2.2. We note that there are some difficulties in comparing MCMC with variational approximation in this manner as the run time of Algorithm 1 depends on the initialization and stopping rule, and the rate of convergence is problem-dependent. Similarly, computation time for MCMC depends on the number of simulations, length of burn-in required to achieve convergence and the sampling algorithm — factors which are also problem specific. The MCMC algorithms were run for 10000 iterations with the first 1000 iterations discarded as burn-in both for fitting the full data and in the cross-validation calculations. Such short run times are only possible because our MCMC scheme uses a very good proposal based on the fit from variational approximation. This MCMC algorithm generally mixes rapidly and initial values were also based on the variational approximation so that the length of burn-in is short. For cross-validation calculations us-

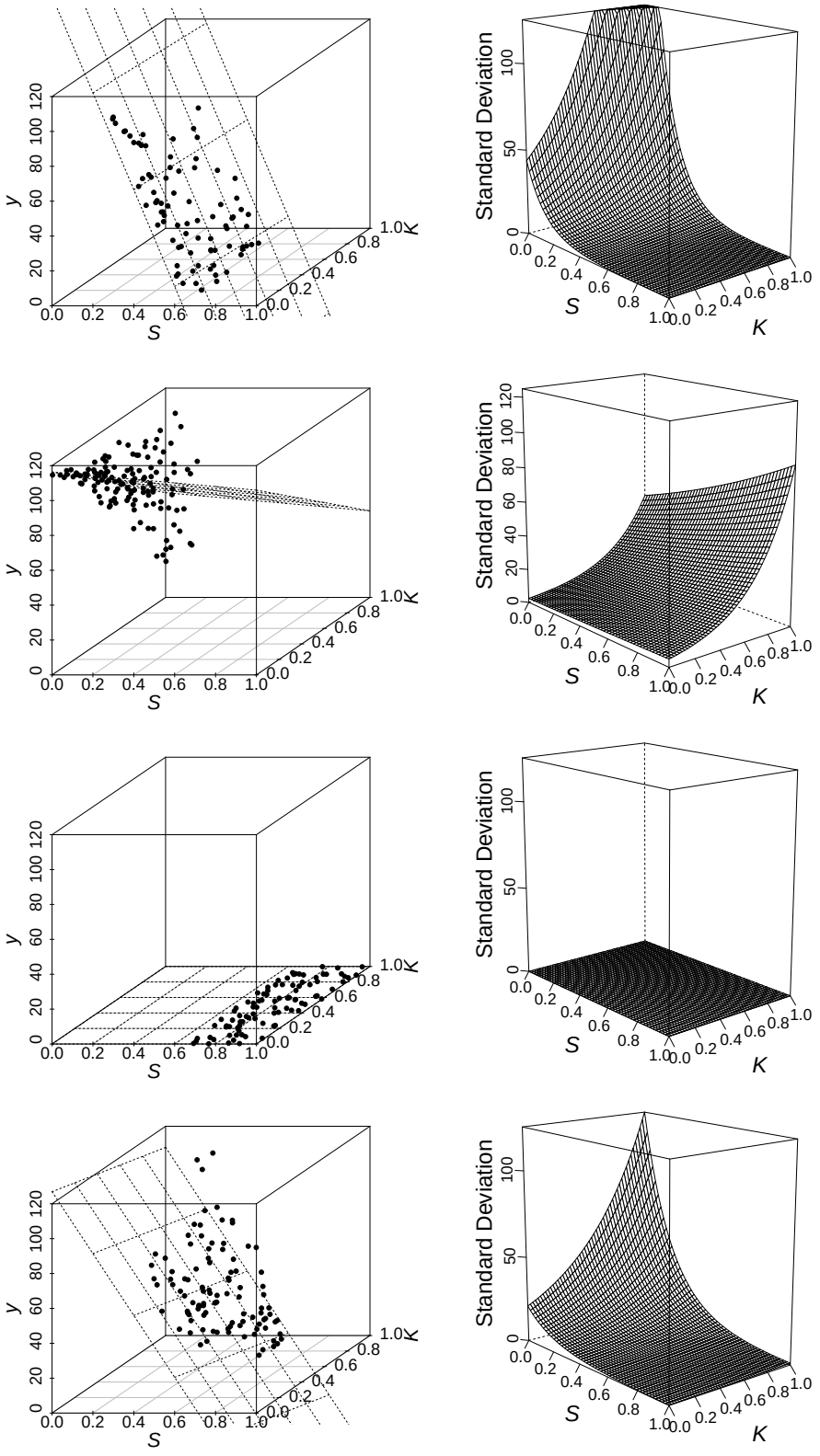

Figure 2.1: Rainfall-runoff data. Fitted component means (first column) and standard deviations (second column) for model C from variational approximation. Different rows correspond to different mixture components.

ing variational approximation, the "short runs" strategy was applied only in the fitting of the first training set. For subsequent training sets, the initialization of Algorithm 1 was based on the fit from the previous training set. Table 2.2 indicates a roughly 20 fold speed up for all models, by using variational approximation in the cross-validation computations when using just 10000 iterations in the MCMC sampling. This is a rather conservative estimate of the benefits and is consistent with other comparisons in the variational approximation literature. Furthermore, difficulties in assessing convergence in the MCMC approach are avoided by the variational method. We note that it is very difficult to use MCMC methods in cross-validatory approaches to model comparison as repeated MCMC runs for model fits to different parts of the data and for many models are very computationally intensive. This example demonstrates the advantage of fast variational approximation in inference due to its ability to fit many models for model assessment and exploratory analysis.

For model C, we use the stochastic gradient algorithm (Algorithm 2) to improve the basic variational approximation obtained from Algorithm 1. We set $N = 10000$ in Algorithm 2. For the gain sequences, we let $a = 0.4$, $A = 10000$, $\alpha = 0.8$ for the mean adjustment parameters and $\alpha = 0.9$ for the variance adjustment parameters. We are looking at just one of the modes here and there are no issues of label switching in MCMC as the modes corresponding to relabelling are well seperated. Computation of the stochastic approximation correction took 166 seconds of CPU time. Figure 2.2 shows the marginal posterior distributions for the parameters in the mixing weights model obtained using MCMC (solid lines), simple variational approximation (dashed lines) and variational approximation with stochastic approximation correction (dot-dashed lines). The stochastic approximation correction is helpful for obtaining an improved approximation for at least some of the parameters, with the estimated posterior marginals from stochastic approximation generally being closer to the Monte Carlo estimated marginals than the marginals from basic variational approximation. There is little improvement in estimation of the marginal posteriors for the mean and variance parameters by the stochastic approximation correction (results not shown). Similar benefits in estimation of the mixing weights parameters have been observed in other examples that we have considered.

To investigate the performance of ten-fold cross-validation in model choice using a variational approach, we simulate fifty data sets from model

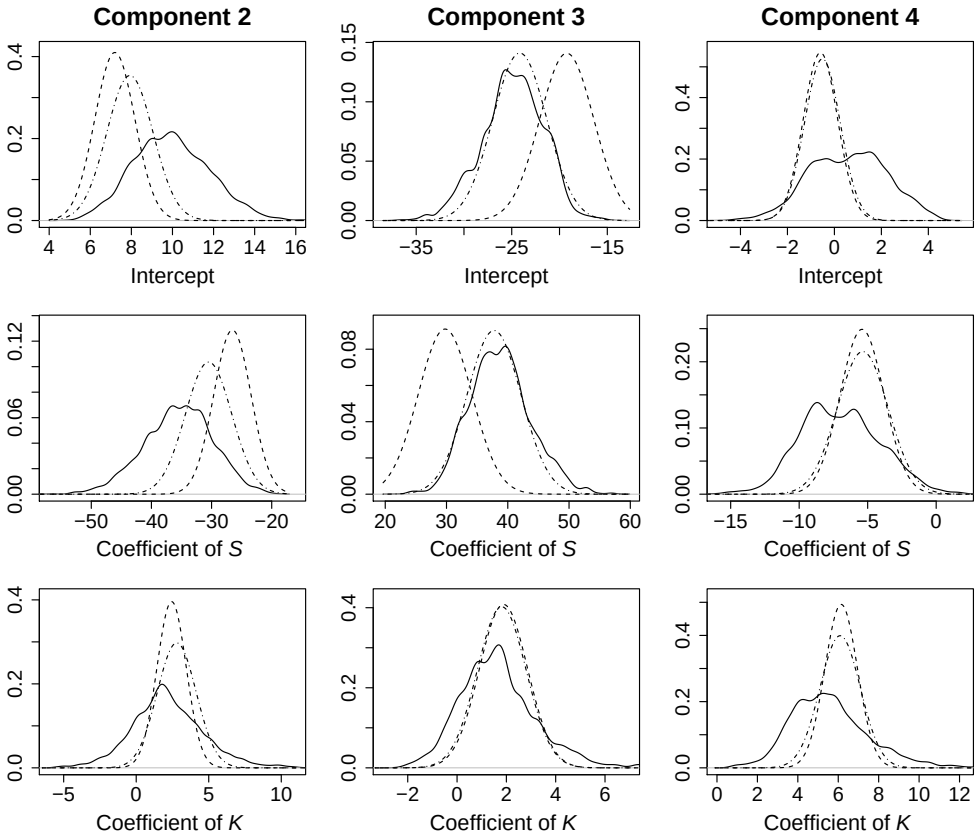

Figure 2.2: Rainfall-runoff data. Marginal posterior distributions for parameters in the mixing weights estimated by MCMC (solid line), simple variational approximation (dashed line) and variational approximation with stochastic approximation correction (dot-dashed line). Columns are different components and the first, second and third rows correspond to the intercept and coefficients for S and K respectively.

C, using as parameters the variational posterior means obtained by using Algorithm 1 to fit model C to the real data. For each simulated data set, we compute ten-fold cross-validation LPDS using a variational approach for MHR models with the number of mixture components ranging from 2 to 7. Both predictors S and K are included in the mean and variance models. In this case, model C is regarded as the "true" model. Of the 50 simulated data sets, the true model was chosen 32 times, model D (with one extra mixture component) was chosen 17 times and a six component MHR model was chosen once.

2.6.2 Time series example

In this example, we use MHR models to analyze daily returns from the S&P500 stock market index. The response y_t is defined as $\log(p_t/p_{t-1})$ where p_t is the closing S&P500 index on day t. Following Li *et al.* (2010),

Number of mixture components	1	2	3	4
No sequential updating (MCMC)	-477.8	-471.2	-469.0	-470.6
No sequential updating (variational)	-478.0	-470.1	-470.1	-471.7
Sequential updating (variational)	-477.7	-470.0	-470.1	-473.3

Table 2.3: Time series data. LPDS computed with no sequential updating (posterior not updated after end of training period) using MCMC algorithm (first line) and variational method (second line). LPDS computed with sequential updating using variational method (third line).

we consider data from 4646 trading days (from 1 January 1990 to 29 May 2008) as the training set for model estimation and data from the subsequent 199 trading days (from 30 May 2008 to 13 March 2009) as the test set for performing model selection. Li *et al.* (2010) note that the choice of the last 199 observations in the series for validation is a difficult test for candidate models because this period covers the recent financial crisis where there is unusually high volatility. Previously, Villani *et al.* (2009) showed that the heteroscedastic components of a smooth adaptive Gaussian mixtures model were able to provide a better fit to a data set of daily returns from the S&P500 stock market index than the smoothly mixing regression model (with homoscedastic components) considered by Geweke and Keane (2007). Li *et al.* (2010) generalized the Gaussian components of the smooth adaptive Gaussian mixtures model (Villani *et al.*, 2009) to asymmetric t-densities so that skewness and excess kurtosis can be captured.

We consider as predictors, `LastWeek` (average of returns for last 5 trading days), `LastMonth` (average of returns for last 20 trading days) and `MaxMin95`, defined as $(1-\varsigma)\sum_{s=0}^{\infty}\varsigma^s(\log p_{t-1-s}^{(h)} - \log p_{t-1-s}^{(l)})$ where $p_t^{(h)}$ and $p_t^{(l)}$ are the highest and lowest values of the index on day t and $\varsigma = 0.95$. These covariates were found to be significant by Li *et al.* (2010) in fitting a one-component split-t model where the location, scale, skewness and degrees of freedom are all functions of covariates. All covariates were standardized to lie in $[-1, 1]$ as in Li *et al.* (2010). We consider MHR models with only an intercept term in the mean model as the level of stock market returns are generally not predictable (see Villani *et al.*, 2009; Li *et al.*, 2010), but an intercept as well as the covariates `LastWeek`, `LastMonth` and `MaxMin95` in the variance model and mixing weights model. We consider models with number of mixture components ranging from 1 to 4.

Table 2.3 shows the LPDS values computed using MCMC (first row) and variational approximation (second row), by means of the approxima-

Number of mixture components	1	2	3	4
Initial fit (MCMC)	504	2463	3427	4417
Initial fit (variational)	1	739	1022	1442
Initial fit + validation (variational)	250	1902	2552	4754

Table 2.4: Time series data. Rows 1–3 shows respectively the CPU times (seconds) taken for initial fit using MCMC, initial fit using variational approximation, and initial fit plus sequential updating for cross-validation using variational approximation.

tion of Li *et al.* (2010), where the posterior is not updated after the end of the training period. The third row shows the LPDS computed using variational approximation with sequential updating of the posterior at each time point. Based on the largest LPDS, it seems that a two-component mixture provides an adequate model. The CPU times (in seconds) taken to compute the LPDS using MCMC and variational approximation are shown in Table 2.4. The first row shows the time taken to obtain an initial fit using the MCMC algorithm. For each of the models, we run the MCMC algorithm for 10000 iterations with the first 1000 iterations discarded as burn-in. The second row shows the time taken to obtain an initial fit using variational approximation and the third row shows the total time taken to compute the LPDS values with sequential updating using variational approximation (initial fit plus sequential updating). In this case there is a roughly 200 fold speed up from employing the variational method as compared to MCMC in sequential updating. Note that the total time taken to compute LPDS with sequential updating, using the variational method (initial fit plus validation) is close to the time taken to obtain just the initial fit using the MCMC algorithm. We need to multiply the computational cost for the initial MCMC fit by approximately $T^* = 199$ to get the computational cost for the complete computations.

Another area where MCMC methods may not be feasible for analyzing time series data is in rolling window computations, where parameter estimates for the model within different windows are examined to check for structural breaks and model instability. We illustrate this application here for the two component MHR model. Consider windows of size $M = 500$. First, we fit the model to the first M observations. Next, we advance the rolling window by 50 observations, that is, we refit the model to observations 51 to $M + 50$. We continue in this way, advancing the rolling window by 50 observations at each step. Figure 2.3 shows the estimated lower 1% and 5% quantiles of the predictive densities for the covariate values at times

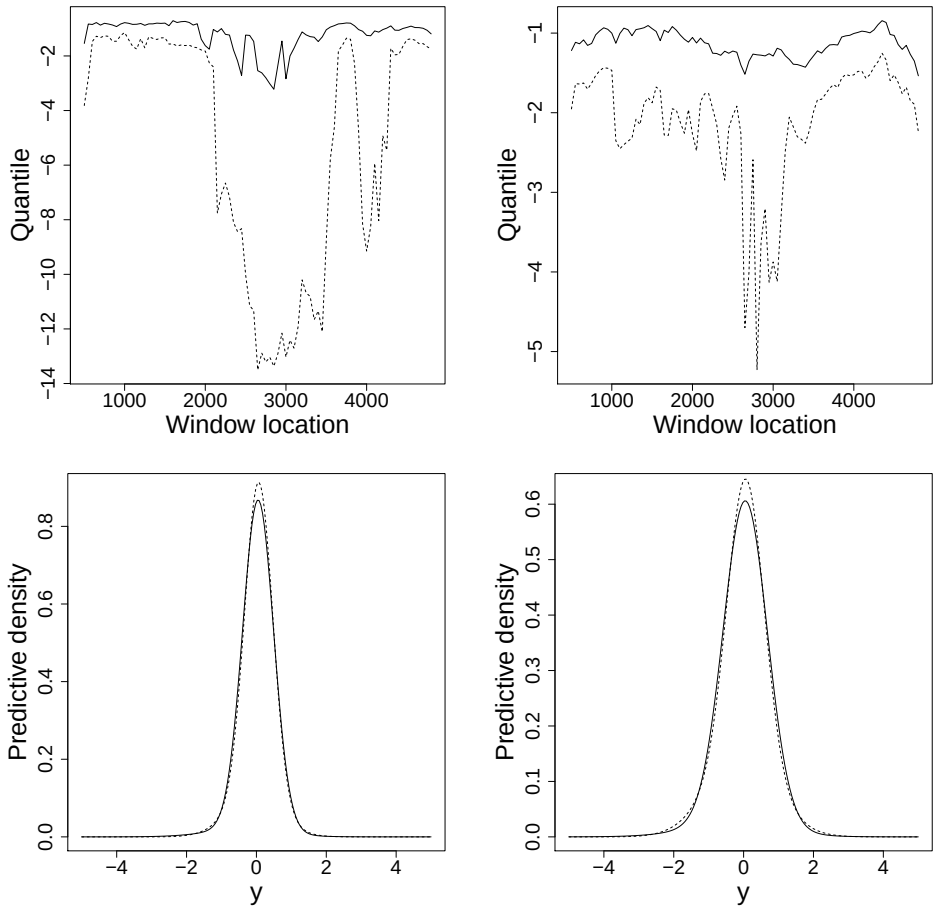

Figure 2.3: Time series data. Estimated 1% (dashed line) and 5% (solid line) quantiles of predictive densities for covariate values at $t = 1000$ (top left) and $t = 4000$ (top right) plotted against the upper edge of the rolling window. Also shown are the estimated predictive densities for covariate values at $t = 1000$ and $t = 4000$ (bottom left and right respectively) estimated based on the entire training data set using MCMC (solid line) and variational approximation (dashed line).

$t = 1000$ and $t = 4000$ versus the upper edge of the rolling window. There is some evidence of model instability and structural change. Also shown in Figure 2.3 are the predictive densities for the same covariates estimated based on the entire training data using MCMC (solid lines) and variational approximation (dashed lines). The MCMC and variational predictive densities are nearly indistinguishable so that the variational approximation provides excellent predictive inference here.

2.7 Conclusion

In this chapter, we have developed fast variational approximation methods for fitting MHR models. The benefits of the variational approach as

compared to MCMC methods are illustrated in problems where repeated refitting of models is required, such as in exploratory analysis and cross-validation approaches to model choice. We have also described how the basic variational approximation can be improved by using stochastic approximation methods to perturb the initial solution. There are several promising avenues for future research. While we have emphasized the advantages of using a variational approach as compared to MCMC methods in model refitting, MCMC methods and variational methods can be complementary. For instance, variational methods can be used to provide initial values and good proposal distributions for MCMC schemes. This may be helpful in reducing the length of burn-in and number of simulations required. This strategy is sometimes called variational MCMC (de Freitas *et al.*, 2001). The combination of variational methods with stochastic approximation has the potential to broaden the applicability of such an approach. It might be possible to combine variational methods or the stochastic approximation approach of Section 2.5, with MCMC methods applied to a subset of the data. A rough idea of the correlation structure in the posterior can be obtained by running MCMC for a subset and the means and variances can be adjusted using stochastic approximation approaches similar to those we have described. There are many issues to be addressed in practice with such an approach however. Another interesting extension that we have not pursued for MHR models is to allow some of the coefficients in the components to be shared across components. Villani *et al.* (2009) reported that they have found such restrictions for the variance models to be useful in practice.

Chapter 3

Variational approximation for mixtures of linear mixed models

Mixtures of linear mixed models (MLMMs) provide a formal mathematical framework for the clustering of grouped data, which may be correlated or replicated, and allow for the incorporation of covariate information. They have been applied in the clustering of gene expression profiles in microarray analysis (e.g. Celeux *et al.*, 2005) and electrical load series for electric utility planning (Coke and Tsao, 2010). Here, we consider MLMMs where the response distribution is a normal mixture, with mixture weights varying as a function of the covariates. Cluster-specific random effects are included in the model so that observations from the same cluster are correlated.

MLMMs can be estimated by likelihood maximization through the EM algorithm and a suitable number of components is determined conventionally by comparing different mixture models using penalized log-likelihood criteria such as BIC (e.g. Ng *et al.*, 2006). Here, we propose fitting MLMMs with variational methods that can perform parameter estimation and model selection simultaneously. First, we describe a variational approximation for MLMMs where the variational lower bound is in closed form, allowing for fast evaluation. A novel variational greedy algorithm (VGA) is then developed for model selection and learning of the mixture components. Initialization is handled within the VGA and a plausible number of mixture components is returned automatically at the end of the algorithm together with the fitted model. The greedy approach developed here is not limited to MLMMs and can be adapted to fit other mixture models using variational methods.

In cases of weak identifiability of certain model parameters, we use hierarchical centering to reparametrize the model and show empirically that there is a gain in efficiency in variational algorithms similar to that in

MCMC algorithms. Hierarchical centering was first proposed by Gelfand *et al.* (1995) who showed that such reparametrizations of normal linear mixed models gave improved convergence in MCMC algorithms. We consider a case of partial centering, a second case of full centering, and derive the corresponding variational algorithms. Related to this, we prove that the approximate rate of convergence of VB algorithms by Gaussian approximation is equal to that of the corresponding Gibbs sampler. Previously, Sahu and Roberts (1999) showed that the approximate rate of convergence of the Gibbs sampler by Gaussian approximation is equal to that of the corresponding EM algorithm and hence improvement strategies for one algorithm can be used for the other. As reparametrizations using hierarchical centering can lead to improved convergence in the Gibbs sampler, this result suggests that convergence in variational algorithms may be improved through reparametrizations just as in MCMC algorithms.

This chapter is organized as follows. Section 3.1 provides some background. Section 3.2 introduces MLMMs and Section 3.3 describes fast variational methods for fitting them. Section 3.4 discusses reparametrization of MLMMs through hierarchical centering. Section 3.5 describes the variational greedy algorithm. Section 3.6 contains theoretical results on the rate of convergence of VB algorithms by Gaussian approximation. Section 3.7 considers examples involving real and simulated data and Section 3.8 concludes.

The results presented in this chapter have been published in Tan and Nott (2013a).

3.1 Background

In microarray analysis, clustering of gene expression profiles is a valuable exploratory tool for identifying meaningful relationships between genes. In the model-based cluster analysis context, Luan and Li (2003) studied clustering of genes in the mixture model framework using a mixed-effects model with B-splines. Celeux *et al.* (2005) proposed using MLMMs to account for data variability in repeated measurements. Both of these approaches require the independence assumption for genes which may not hold in practice for all pairs of genes (McLachlan *et al.*, 2004). In contrast, Ng *et al.* (2006) considered MLMMs with cluster-specific random effects which allow genes within a cluster to be correlated. Similar models were considered by Booth *et al.* (2008), who proposed a stochastic search algorithm for finding partitions of the data with high posterior probability through maximization of

an objective function. For the clustering of electrical load series, Coke and Tsao (2010) developed random effects mixture models with antedependence models for the non-stationary random effects.

The EM algorithm was used for the estimation of MLMMs in Luan and Li (2003), Celeux *et al.* (2005) and Coke and Tsao (2010). Ng *et al.* (2006) developed a program called EMMIX-WIRE (EM-based MIXture analysis WIth Random Effects) for clustering correlated and replicated data. In these articles, the optimal number of components was determined by comparing different mixture models using BIC. The EM algorithm can be sensitive to initialization and is commonly run from multiple starting values to avoid convergence to local optima. Scharl *et al.* (2010) studied the performance of different EM algorithm initialization strategies for mixtures of regression models and Biernacki *et al.* (2003) compared simple initialization strategies for Gaussian mixtures. Verbeek *et al.* (2003) discussed a greedy approach to the learning of Gaussian mixtures which resolves sensitivity to initialization and is useful in finding the optimal number of components.

We propose fitting MLMMs with variational methods using a greedy algorithm. Previously, Ormerod and Wand (2010) have illustrated the use of variational methods in fitting Gaussian linear mixed models. Armagan and Dunson (2011) used variational methods to obtain sparse approximate Bayes inference in the analysis of large longitudinal data sets using linear mixed models. Recently, Ormerod and Wand (2012) introduced Gaussian variational approximation for fitting generalized linear mixed models. The variational algorithm suffers from problems of local optima as well and initialization strategies for the EM algorithm can often be adapted for use with the variational algorithm. For example, a "short runs" strategy was discussed in Section 2.3, where the variational algorithm is initialized randomly from multiple starting points, stopped prematurely, and only the short run with the highest attained value of the variational lower bound is followed to convergence. This is similar to a strategy recommended by Biernacki *et al.* (2003) for initialization of the EM algorithm.

A key advantage of variational methods is the potential for simultaneous parameter estimation and model selection. A number of such methods have been developed for fitting Gaussian mixtures and a brief review is given in Section 1.1.3. In particular, McGrory and Titterington (2007) described a variational optimization technique where the algorithm is initialized with a large number of components and mixture components whose weightings become sufficiently small are dropped out as the optimization proceeds,

leading to automatic model selection. We have attempted this component elimination approach for some of the examples in this chapter (results not shown) and observed some difficulties in the implementation. First, clustering results tend to be sensitive to the initialization and strategies to avoid convergence to local optima, such as using multiple starting points are necessary, which adds to the computational burden. Second, the choice of the initial number of mixture components was observed to have an impact on the resulting number of components and it may not be easy in some cases to determine a suitable initial number. Finally, initializing the algorithm with a large number of mixture components can be computationally expensive for large data sets.

We develop a novel VGA for fitting MLMMs. Starting with one component, the VGA adds new components to the mixture after searching for the optimal way to split components in the current mixture. While this bottom-up approach resolves the difficulty of estimating the upper bound of the number of mixture components, it can become time-consuming when the number of components is large, since a larger number of components have to be tested to find the optimal way of splitting each one. Some measures are introduced to keep the search time short and the component elimination property of variational approximation is used to sieve out components which resist splitting. Greedy approaches for fitting Gaussian mixtures have been considered for instance, by Verbeek *et al.* (2003) using the EM algorithm and Constantinopoulos and Likas (2007) using variational methods.

3.2 Mixtures of linear mixed models

The MLMM we are considering is a generalization of that proposed by Ng *et al.* (2006), where units from the same cluster share cluster-specific random effects and are hence correlated. Unlike Ng *et al.* (2006), our model can fit data where the number of observations on each unit are not equal and we allow the mixture weights to vary with covariates between clusters.

Suppose we observe n multivariate reponses $y_i = [y_{i1}, \ldots, y_{in_i}]^T$, $i = 1, \ldots, n$, and $N = \sum_{i=1}^{n} n_i$. Let the number of mixture components be k and δ_i, $i = 1, \ldots, n$, be latent variables indicating which mixture component the ith cluster corresponds to, $\delta_i \in \{1, \ldots, k\}$. Conditional on $\delta_i = j$,

$$y_i = X_i\beta_j + W_i a_i + V_i b_j + \epsilon_i, \tag{3.1}$$

where X_i, W_i and V_i are design matrices of dimensions $n_i \times p$, $n_i \times s_1$ and

$n_i \times s_2$ respectively, β_j, $j = 1, \ldots, k$, are $p \times 1$ vectors of fixed effects, a_i, $i = 1, \ldots, n$, are $s_1 \times 1$ vectors of random effects, b_j, $j = 1, \ldots, k$, are $s_2 \times 1$ vectors of random effects and ϵ_i, $i = 1, \ldots, n$, are vectors of random errors. We assume that the random effects a_i, $i = 1, \ldots, n$, b_j, $j = 1, \ldots, k$, and the error vectors ϵ_i, $i = 1, \ldots, n$, are mutually independent. The fixed effects, the distribution of the random effects and the distribution of the error terms are all mixture component specific. Given that $\delta_i = j$, a_i and b_j are distributed as $N(0, \sigma^2_{a_j} I_{s_1})$ and $N(0, \sigma^2_{b_j} I_{s_2})$ respectively. The error vector ϵ_i is distributed as $N(0, \Sigma_{ij})$ where $\Sigma_{ij} = \text{blockdiag}(\sigma^2_{j1} I_{\kappa_{i1}}, \ldots, \sigma^2_{jg} I_{\kappa_{ig}})$, a block diagonal with the lth block equal to $\sigma^2_{jl} I_{\kappa_{il}}$. Here g is constant for all i and $\sum_{l=1}^{g} \kappa_{il} = n_i$ for each $i = 1, \ldots, n$. In microarray experiments for instance, this specification provides increased flexibility as the error variance of each mixture component is allowed to vary between different experiments, say, by setting g to be the total number of experiments. We assume that

$$P(\delta_i = j|\gamma) = p_{ij}(\gamma) = \frac{\exp(u_i^T \gamma_j)}{\sum_{l=1}^{k} \exp(u_i^T \gamma_l)},$$

where $u_i = [u_{i1}, \ldots, u_{id}]^T$ is a vector of covariates, $\gamma_1 = 0$ for identifiability, $\gamma_j = [\gamma_{j1}, \ldots, \gamma_{jd}]^T$ are vectors of unknown parameters for $j = 2, \ldots, k$ and $\gamma = [\gamma_2^T, \ldots, \gamma_k^T]^T$. This model for the mixture component indicators allows mixture weights to vary with covariates across clusters. For Bayesian inference, we assume the following priors on unknown parameters: $\gamma \sim N(0, \Sigma_\gamma)$, $\beta_j \sim N(0, \Sigma_{\beta_j})$, $\sigma^2_{a_j} \sim IG(\alpha_{a_j}, \lambda_{a_j})$ and $\sigma^2_{b_j} \sim IG(\alpha_{b_j}, \lambda_{b_j})$ for $j = 1, \ldots, k$, and $\sigma^2_{jl} \sim IG(\alpha_{jl}, \lambda_{jl})$ for $j = 1, \ldots, k$, $l = 1, \ldots, g$. The hyperparameters α_{a_j}, λ_{a_j}, α_{b_j}, λ_{b_j}, α_{jl}, λ_{jl}, Σ_γ and Σ_{β_j}, $j = 1, \ldots, k$, $l = 1, \ldots, g$, are considered known. Let $\beta = [\beta_1^T, \ldots, \beta_k^T]^T$, $a = [a_1^T, \ldots, a_n^T]^T$, $b = [b_1^T, \ldots, b_k^T]^T$, $\sigma^2_a = [\sigma^2_{a_1}, \ldots, \sigma^2_{a_k}]^T$, $\sigma^2_b = [\sigma^2_{b_1}, \ldots, \sigma^2_{b_k}]^T$, $\sigma^2_j = [\sigma^2_{j1}, \ldots, \sigma^2_{jg}]^T$ for $j = 1, \ldots, k$, $\sigma^2 = [\sigma_1^{2T}, \ldots, \sigma_k^{2T}]^T$ and $\delta = [\delta_1, \ldots, \delta_n]^T$ so that $\theta = \{\beta, a, b, \sigma^2_a, \sigma^2_b, \sigma^2, \gamma, \delta\}$ denotes the set of all unknown parameters in the MLMM. We describe a variational approximation for the joint posterior distribution $p(\theta|y)$ in the next section.

For the specification of the inverse gamma priors, we consider an approach used by Fong *et al.* (2010) which is based on the following lemma.

Lemma 3.1. Let $u|\sigma^2 \sim N(0, \sigma^2)$ and $\sigma^2 \sim IG(\alpha, \lambda)$. The marginal distribution of u obtained by integrating over σ^2 is a non-standardized Student's t distribution with location parameter 0, scale parameter $\sqrt{\frac{\lambda}{\alpha}}$ and degrees of freedom 2α.

The density of a non-standardized Student's t with location parameter μ,

scale parameter σ and degrees of freedom ν is given by

$$\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\left\{1+\frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right\}^{-\frac{\nu+1}{2}}.$$

Fong *et al.* (2010) suggested that to choose a prior for a single random effect u, one can give a range for u, specify the degrees of freedom ν, and then solve for α and λ. Here we obtain a crude estimate of the random effects in (3.1) by considering the residuals from a least squares regression of $y = [y_1, \ldots, y_n]$ against $X = [X_1^T, \ldots, X_n^T]^T$. We fix the shape parameter α as 2, since $IG(2, \lambda)$ has an infinite variance but a finite mean at λ. This specification allows the prior to be centered on a reasonable belief while maintaining a large prior variance (see, e.g., Finley *et al.*, 2008). We estimate λ by fitting a non-standardized Student's t to the residuals with location parameter 0 and degrees of freedom 4. This can be done in `R` using the function `fitdistr()` from the package `MASS` (Venables and Ripley, 2002) to estimate the scale. For convenience, we used the same priors for $\sigma^2_{a_j}$ and $\sigma^2_{b_j}$ for $j = 1, \ldots, k$ and σ^2_{jl} for $j = 1, \ldots, k$, $l = 1, \ldots, g$.

3.3 Variational approximation

We consider a variational approximation to $p(\theta|y)$ of the form

$$q(\theta) = q(\beta)q(a)q(b)q(\sigma^2, \sigma_a^2, \sigma_b^2)q(\delta)q(\gamma). \tag{3.2}$$

Application of (1.4) leads to optimal densities of the form:

$$q(\beta) = \prod_{j=1}^{k} q(\beta_j), \quad q(a) = \prod_{i=1}^{n} q(a_i), \quad q(b) = \prod_{j=1}^{k} q(b_j), \quad q(\delta) = \prod_{i=1}^{n} q(\delta_i)$$

and $q(\sigma^2, \sigma_a^2, \sigma_b^2) = q(\sigma^2)q(\sigma_a^2)q(\sigma_b^2)$, where

$$q(\sigma_a^2) = \prod_{j=1}^{k} q(\sigma^2_{a_j}), \quad q(\sigma_b^2) = \prod_{j=1}^{k} q(\sigma^2_{b_j}), \quad q(\sigma^2) = \prod_{j=1}^{k}\prod_{l=1}^{g} q(\sigma^2_{jl}).$$

It also follows from (1.4) that $q(\beta_j)$ is $N(\mu^q_{\beta_j}, \Sigma^q_{\beta_j})$, $q(a_i)$ is $N(\mu^q_{a_i}, \Sigma^q_{a_i})$, $q(b_j)$ is $N(\mu^q_{b_j}, \Sigma^q_{b_j})$, $q(\sigma^2_{a_j})$ is $IG(\alpha^q_{a_j}, \lambda^q_{a_j})$, $q(\sigma^2_{b_j})$ is $IG(\alpha^q_{b_j}, \lambda^q_{b_j})$, $q(\sigma^2_{jl})$ is $IG(\alpha^q_{jl}, \lambda^q_{jl})$ and $q(\delta_i = j) = q_{ij}$ with $\sum_{j=1}^{k} q_{ij} = 1$ for each $i = 1, \ldots, n$. The value of q_{ij} can be interpreted as a measure of the responsibility undertaken by component j in explaining the ith observation (see Bishop, 2006). The

optimal $q(\gamma)$ does not belong to any recognizable density family and we assume that $q(\gamma)$ is a delta function placing a point mass of 1 on μ_γ^q. A degenerate point mass has been assumed for $q(\gamma)$ so that computation of the lower bound is tractable.

We have assumed in the variational posterior that the distributions of the fixed effects, random effects, variance parameters, latent variables, and mixing weights model parameters are independent of each other. Similar independence assumptions have been made in the case of the linear mixed model by Armagan and Dunson (2011). It is also possible to consider the fixed effects β and the random effects a and b as a single block and replace $q(\beta)q(a)q(b)$ by $q(\beta, a, b)$ as in Ormerod and Wand (2010). This results in a less restricted factorization with dependence structure between β, a and b preserved and a higher lower bound can be achieved. However, this will involve dealing with high dimensional sparse covariance matrices which create a greater computational burden, although matrix inversion results can be used for the blocked matrices to attain better computational efficiency. We have decided to use a factorized form for faster computation and better scalability to larger data sets (see Armagan and Dunson, 2011).

Let $\theta_{-\gamma}$ denote the set of unknown parameters excluding γ. From the argument in (2.3), $\mathcal{L} = E_q\{\log p(y,\theta)\} - E_q\{\log q(\theta_{-\gamma})\}$ gives a lower bound on $\sup_\gamma \log p(\gamma)p(y|\gamma)$, where $E_q(\cdot)$ denotes expectation with respect to $q(\theta)$. The lower bound $\mathcal{L}$ can be computed in closed form, and is given by (details in Appendix B)

$$
\begin{aligned}
\mathcal{L} = \sum_{j=1}^{k} \Big[& \tfrac{1}{2}\log|\Sigma_{\beta_j}^{-1}\Sigma_{\beta_j}^q| - \tfrac{1}{2}\mathrm{tr}(\Sigma_{\beta_j}^{-1}\Sigma_{\beta_j}^q) - \tfrac{1}{2}{\mu_{\beta_j}^q}^T\Sigma_{\beta_j}^{-1}\mu_{\beta_j}^q + \tfrac{1}{2}\log|\Sigma_{b_j}^q| + \alpha_{b_j}^q \\
& - \tfrac{\alpha_{b_j}^q}{2\lambda_{b_j}^q}\big\{{\mu_{b_j}^q}^T\mu_{b_j}^q + \mathrm{tr}(\Sigma_{b_j}^q)\big\} + \alpha_{b_j}\log\lambda_{b_j} - \alpha_{b_j}^q\log\lambda_{b_j}^q - \tfrac{\lambda_{a_j}\alpha_{a_j}^q}{\lambda_{a_j}^q} - \tfrac{\lambda_{b_j}\alpha_{b_j}^q}{\lambda_{b_j}^q} \\
& + \alpha_{a_j}^q + \log\tfrac{\Gamma(\alpha_{b_j}^q)}{\Gamma(\alpha_{b_j})} + \tfrac{s_1\sum_{i=1}^n q_{ij}}{2}\{\psi(\alpha_{a_j}^q) - \log\lambda_{a_j}^q\} + \psi(\alpha_{a_j}^q)(\alpha_{a_j} - \alpha_{a_j}^q) \\
& + \alpha_{a_j}\log\tfrac{\lambda_{a_j}}{\lambda_{a_j}^q} + \log\tfrac{\Gamma(\alpha_{a_j}^q)}{\Gamma(\alpha_{a_j})}\Big] + \sum_{j=1}^{k}\sum_{l=1}^{g}\Big[\alpha_{jl}\log\tfrac{\lambda_{jl}}{\lambda_{jl}^q} + \log\tfrac{\Gamma(\alpha_{jl}^q)}{\Gamma(\alpha_{jl})} - \tfrac{\lambda_{jl}\alpha_{jl}^q}{\lambda_{jl}^q} \\
& + \alpha_{jl}^q + \psi(\alpha_{jl}^q)(\alpha_{jl} - \alpha_{jl}^q) + \tfrac{\sum_{i=1}^n \kappa_{il}q_{ij}}{2}\big\{\psi(\alpha_{jl}^q) - \log\lambda_{jl}^q\big\}\Big] + \log p(\mu_\gamma^q) \\
& - \sum_{i=1}^{n}\sum_{j=1}^{k}\tfrac{q_{ij}}{2}\Big[\xi_{ij}^T{\Sigma_{ij}^q}^{-1}\xi_{ij} + \mathrm{tr}({\Sigma_{ij}^q}^{-1}\Lambda_{ij}) + \tfrac{\alpha_{a_j}^q}{\lambda_{a_j}^q}\big\{{\mu_{a_i}^q}^T\mu_{a_i}^q + \mathrm{tr}(\Sigma_{a_i}^q)\big\}\Big] \\
& + \tfrac{k(p+s_2)+ns_1-N\log(2\pi)}{2} + \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\log\tfrac{p_{ij}(\mu_\delta^q)}{q_{ij}} + \tfrac{1}{2}\sum_{i=1}^{n}\log|\Sigma_{a_i}^q|, \qquad (3.3)
\end{aligned}
$$

where $p(\mu_\gamma^q)$ denotes the prior distribution for γ evaluated at μ_γ^q, $\xi_{ij} = y_i - X_i\mu_{\beta_j}^q - W_i\mu_{a_i}^q - V_i\mu_{b_j}^q$, $\Lambda_{ij} = X_i\Sigma_{\beta_j}^q X_i^T + W_i\Sigma_{a_i}^q W_i^T + V_i\Sigma_{b_j}^q V_i^T$ and ${\Sigma_{ij}^q}^{-1} = \text{blockdiag}\left(\frac{\alpha_{j1}^q}{\lambda_{j1}^q} I_{\kappa_{i1}}, \ldots, \frac{\alpha_{jg}^q}{\lambda_{jg}^q} I_{\kappa_{ig}}\right)$.

The updates of the variational parameters, $\mu_{\beta_j}^q$, $\Sigma_{\beta_j}^q$, $\mu_{b_j}^q$, $\Sigma_{b_j}^q$, $\alpha_{a_j}^q$, $\lambda_{a_j}^q$, $\alpha_{b_j}^q$, $\lambda_{b_j}^q$, for $j = 1, \ldots, k$, $\mu_{a_i}^q$, $\Sigma_{a_i}^q$, for $i = 1, \ldots, n$, α_{jl}^q, λ_{jl}^q, for $j = 1, \ldots, k$, $l = 1, \ldots, g$ and q_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, k$, can be determined from (1.4) and obtained using the iterative scheme in Algorithm 3. The update for μ_γ^q can be obtained by maximizing the variational lower bound $\mathcal{L}$ with respect to μ_γ^q. All updates are available in closed form except for μ_γ^q.

An alternative approach for deriving the variational updates, that is presented in Tan and Nott (2013a), is to assume parametric forms for the factors in the variational posterior $q(\theta)$. The forms of the optimal densities can be deduced from (1.4) and the fact that the model has conjugate priors. The variational lower bound $\mathcal{L}$ can then be computed as a function of the variational parameters and maximizing $\mathcal{L}$ with respect to these parameters, say, by using methods in vector differential calculus (see Wand, 2002), gives the required updates.

Algorithm 3: Variational approximation for MLMM

Initialize: q_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, k$, $\frac{\alpha_{jl}^q}{\lambda_{jl}^q}$ for $j = 1, \ldots, k$, $l = 1, \ldots, g$, $\mu_{a_i}^q$ for $i = 1, \ldots, n$ and $\mu_{b_j}^q$, $\frac{\alpha_{a_j}^q}{\lambda_{a_j}^q}$, $\frac{\alpha_{b_j}^q}{\lambda_{b_j}^q}$ for $j = 1, \ldots, k$.
Cycle:

1. For $j = 1, \ldots, k$,
   - $\Sigma_{\beta j}^q \leftarrow (\Sigma_{\beta j}^{-1} + \sum_{i=1}^n q_{ij} X_i^T {\Sigma_{ij}^q}^{-1} X_i)^{-1}$,
   - $\mu_{\beta j}^q \leftarrow \Sigma_{\beta j}^q \sum_{i=1}^n q_{ij} X_i^T {\Sigma_{ij}^q}^{-1}(y_i - W_i\mu_{a_i}^q - V_i\mu_{b_j}^q)$.
2. For $i = 1, \ldots, n$,
   - $\Sigma_{a_i}^q \leftarrow \left\{\sum_{j=1}^k q_{ij} \frac{\alpha_{a_j}^q}{\lambda_{a_j}^q} I_{s_1} + W_i^T(\sum_{j=1}^k q_{ij} {\Sigma_{ij}^q}^{-1}) W_i\right\}^{-1}$,
   - $\mu_{a_i}^q \leftarrow \Sigma_{a_i}^q \sum_{j=1}^k q_{ij} W_i^T {\Sigma_{ij}^q}^{-1}(y_i - X_i\mu_{\beta_j}^q - V_i\mu_{b_j}^q)$.
3. For $j = 1, \ldots, k$,
   - $\Sigma_{b_j}^q \leftarrow \left(\frac{\alpha_{b_j}^q}{\lambda_{b_j}^q} I_{s_2} + \sum_{i=1}^n q_{ij} V_i^T {\Sigma_{ij}^q}^{-1} V_i\right)^{-1}$,
   - $\mu_{b_j}^q \leftarrow \Sigma_{b_j}^q \sum_{i=1}^n q_{ij} V_i^T {\Sigma_{ij}^q}^{-1}(y_i - X_i\mu_{\beta_j}^q - W_i\mu_{a_i}^q)$.

4. Set μ_γ^q to be the conditional mode of the lower bound, fixing other variational parameters at their current values. As a function of μ_γ^q, the lower bound is the log posterior for a Bayesian multinomial regression with the ith response being $(q_{i1}, \ldots, q_{ik})^T$ and a normal prior on μ_γ^q. The usual iteratively weighted least squares algorithm (or other numerical optimization algorithm) can be used for finding the mode.

5. For $i = 1, \ldots, n$, $j = 1, \ldots, k$, $q_{ij} \leftarrow \frac{p_{ij}(\mu_\gamma^q)\exp(c_{ij})}{\sum_{l=1}^k p_{il}(\mu_\gamma^q)\exp(c_{il})}$ where

$$c_{ij} = \tfrac{1}{2}\sum_{l=1}^{g} \kappa_{il}\{\psi(\alpha_{jl}^q) - \log\lambda_{jl}^q\} - \tfrac{1}{2}\{\mathrm{tr}({\Sigma_{ij}^q}^{-1}\Lambda_{ij}) + \xi_{ij}^T{\Sigma_{ij}^q}^{-1}\xi_{ij}\}$$
$$+ \tfrac{s_1}{2}\{\psi(\alpha_{a_j}^q) - \log\lambda_{a_j}^q\} - \tfrac{\alpha_{a_j}^q}{2\lambda_{a_j}^q}\{{\mu_{a_i}^q}^T\mu_{a_i}^q + \mathrm{tr}(\Sigma_{a_i}^q)\}.$$

6. For $j = 1, \ldots, k$,

   - $\alpha_{a_j}^q \leftarrow \alpha_{a_j} + \frac{s_1}{2}\sum_{i=1}^n q_{ij}$,
   - $\lambda_{a_j}^q \leftarrow \lambda_{a_j} + \frac{1}{2}\sum_{i=1}^n q_{ij}\{{\mu_{a_i}^q}^T\mu_{a_i}^q + \mathrm{tr}(\Sigma_{a_i}^q)\}$.

7. For $j = 1, \ldots, k$,

   - $\alpha_{b_j}^q \leftarrow \alpha_{b_j} + \frac{s_2}{2}$,
   - $\lambda_{b_j}^q \leftarrow \lambda_{b_j} + \frac{1}{2}\{{\mu_{b_j}^q}^T\mu_{b_j}^q + \mathrm{tr}(\Sigma_{b_j}^q)\}$.

8. For $j = 1, \ldots, k$, $l = 1, \ldots, g$,

   - $\alpha_{jl}^q \leftarrow \alpha_{jl} + \frac{1}{2}\sum_{i=1}^n q_{ij}\kappa_{il}$,
   - $\lambda_{jl}^q \leftarrow \lambda_{jl} + \frac{1}{2}\sum_{i=1}^n q_{ij}\{(\xi_{ij})_{\kappa_{il}}^T(\xi_{ij})_{\kappa_{il}} + \mathrm{tr}(\Lambda_{ij})_{\kappa_{il}}\}$,

   where $((\xi_{ij})_{\kappa_{i1}}, \ldots, (\xi_{ij})_{\kappa_{ig}})$ is the partition of ξ_{ij} corresponding to $(\kappa_{i1}, \ldots, \kappa_{ig})$ and $(\Lambda_{ij})_{\kappa_{il}}$ is the diagonal block of Λ_{ij} with rows and columns corresponding to the position of κ_{il} within $(\kappa_{i1}, \ldots, \kappa_{ig})$.

until the increase in $\mathcal{L}$ is negligible.

In the examples, when Algorithm 3 is used in conjunction with the VGA described in Section 3.5 to fit a one-component mixture, for $j = 1$, we set $\frac{\alpha_{a_j}^q}{\lambda_{a_j}^q} = \frac{\alpha_{b_j}^q}{\lambda_{b_j}^q} = 1$, $\frac{\alpha_{jl}^q}{\lambda_{jl}^q} = 1$ for $l = 1, \ldots, g$, $\mu_{b_j}^q = 0$, $\mu_{a_i}^q = 0$ for $i = 1, \ldots, n$, and $q_{ij} = 1$ for $i = 1, \ldots, n$ for initialization.

The form of $q(\gamma)$ can be relaxed to be a normal distribution at convergence using methods similar to that described in Section 2.3. Suppose

$q(\gamma)$ is not subjected to any distributional restriction, the optimal choice for this term is given by

$$q(\gamma) \propto \exp\left\{\sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\log p_{ij}(\gamma) - \tfrac{1}{2}\gamma^T\Sigma_\gamma^{-1}\gamma\right\}. \tag{3.4}$$

If μ_γ^q is close to the mode, we can get a normal approximation to $q(\gamma)$ by setting μ_γ^q as the mean and the covariance matrix Σ_γ^q as the negative inverse Hessian of the log of (3.4), which is the Bayesian multinomial log posterior considered in step 4 of Algorithm 3. Waterhouse *et al.* (1996) outlined a similar idea which they used at every step of their iterative algorithm. We recommend using a delta function approximation first in Algorithm 3 and then doing a one-step approximation after the algorithm has converged. Using the normal approximation $N(\mu_\gamma^q, \Sigma_\gamma^q)$ as the variational posterior for $q(\gamma)$, the variational lower bound $\mathcal{L}$ is the same as in (3.3) except that $\sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\log p_{ij}(\mu_\gamma^q) + \log p(\mu_\gamma^q)$ is replaced with

$$\sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}E_q\left\{\log p_{ij}(\gamma)\right\} + \tfrac{1}{2}\log|\Sigma_\gamma^{-1}\Sigma_\gamma^q| - \tfrac{1}{2}\mu_\gamma^{qT}\Sigma_\gamma^{-1}\mu_\gamma^q - \tfrac{1}{2}\mathrm{tr}(\Sigma_\gamma^{-1}\Sigma_\gamma^q) + \tfrac{d(k-1)}{2}.$$

The expectation of the first term, $E_q\left\{\log p_{ij}(\gamma)\right\}$, is not available in closed form and we replace it with $\log p_{ij}(\mu_\gamma^q)$ to obtain an approximation $\mathcal{L}^*$ to $\log p(y)$. We shall later use $\mathcal{L}^*$ as a model selection criterion in the VGA.

3.4 Hierarchical centering

In later examples, we encounter situations where there is weak identification of certain model parameters and Algorithm 3 converges slowly. We apply hierarchical centering and show empirically that there is a gain in efficiency in variational algorithms through hierarchical centering reparametrization, similar to that in MCMC algorithms. Some theoretical support for this observation is given in Section 3.6.

We consider a case of partial centering in which $X_i = W_i$ and a second case of full centering in which $X_i = W_i = V_i$ in (3.1). In the first case, we introduce $\eta_i = \beta_j + a_i$ conditional on $\delta_i = j$ so that (3.1) is reparametrized as

$$y_i = X_i\eta_i + V_i b_j + \epsilon_i$$

and η_i is "centered" about β_j, with $\eta_i \sim N(\beta_j, \sigma_{a_j}^2 I_p)$. If we let $\eta = (\eta_1^T, \ldots, \eta_n^T)^T$, then η replaces a in the set of unknown parameters θ. Replac-

ing $q(a)$ in (3.2) with $q(\eta)$ with other assumptions unchanged, the optimal $q(\eta)$ is $\prod_{i=1}^{n} q(\eta_i)$, where $q(\eta_i)$ is $N(\mu_{\eta_i}^q, \Sigma_{\eta_i}^q)$ for $i = 1, \ldots, n$. In the second case of full centering, we introduce $\rho_i = \nu_j + a_i$ and $\nu_j = \beta_j + b_j$, conditional on $\delta_i = j$ so that (3.1) is reparametrized as

$$y_i = X_i \rho_i + \epsilon_i,$$

with ρ_i "centered" about ν_j and ν_j "centered" about β_j. We have $\rho_i \sim N(\nu_j, \sigma_{a_j}^2 I_p)$ and $\nu_j \sim N(\beta_j, \sigma_{b_j}^2 I_p)$. If we let $\rho = (\rho_1^T, \ldots, \rho_n^T)^T$ and $\nu = (\nu_1^T, \ldots, \nu_k^T)^T$, then ρ and ν replace a and b in the set of unknown parameters θ. Replacing $q(a)$ and $q(b)$ in (3.2) with $q(\rho)$ and $q(\nu)$ with other assumptions unchanged, the optimal densities for $q(\rho)$ and $q(\nu)$ turn out to be $\prod_{i=1}^{n} q(\rho_i)$ and $\prod_{j=1}^{k} q(\nu_j)$ respectively, where $q(\rho_i)$ is $N(\mu_{\rho_i}^q, \Sigma_{\rho_i}^q)$ for $i = 1, \ldots, n$ and $q(\nu_j)$ is $N(\mu_{\nu_j}^q, \Sigma_{\nu_j}^q)$ for $j = 1, \ldots, k$.

The resulting iterative schemes for the first case with partial centering and the second case with full centering are given in Algorithms 4 and 5 respectively. The variational posterior for γ can be relaxed to be a normal distribution at convergence and similar adjustments, as discussed in Section 3.3, apply to the variational lower bounds for Algorithms 4 and 5.

Algorithm 4: Variational approximation for MLMM with partial centering

Initialize: q_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, k$, $\frac{\alpha_{jl}^q}{\lambda_{jl}^q}$ for $j = 1, \ldots, k$, $l = 1, \ldots, g$ and $\mu_{b_j}^q$, $\mu_{\beta_j}^q$, $\frac{\alpha_{a_j}^q}{\lambda_{a_j}^q}$, $\frac{\alpha_{b_j}^q}{\lambda_{b_j}^q}$ for $j = 1, \ldots, k$.
Cycle:

1. For $i = 1, \ldots, n$,
   - $\Sigma_{\eta_i}^q \leftarrow \left\{ \sum_{j=1}^{k} q_{ij} \frac{\alpha_{a_j}^q}{\lambda_{a_j}^q} I_p + X_i^T (\sum_{j=1}^{k} q_{ij} {\Sigma_{ij}^q}^{-1}) X_i \right\}^{-1}$,
   - $\mu_{\eta_i}^q \leftarrow \Sigma_{\eta_i}^q \sum_{j=1}^{k} q_{ij} \left\{ \frac{\alpha_{a_j}^q}{\lambda_{a_j}^q} \mu_{\beta_j}^q + X_i^T {\Sigma_{ij}^q}^{-1} (y_i - V_i \mu_{b_j}^q) \right\}$.
2. For $j = 1, \ldots, k$,
   - $\Sigma_{\beta_j}^q \leftarrow \left(\Sigma_{\beta_j}^{-1} + \frac{\alpha_{a_j}^q}{\lambda_{a_j}^q} \sum_{i=1}^{n} q_{ij} I_p \right)^{-1}$,
   - $\mu_{\beta_j}^q \leftarrow \Sigma_{\beta j}^q \frac{\alpha_{a_j}^q}{\lambda_{a_j}^q} \sum_{i=1}^{n} q_{ij} \mu_{\eta_i}^q$.
3. For $j = 1, \ldots, k$,
   - $\Sigma_{b_j}^q \leftarrow \left(\frac{\alpha_{b_j}^q}{\lambda_{b_j}^q} I_{s_2} + \sum_{i=1}^{n} q_{ij} V_i^T {\Sigma_{ij}^q}^{-1} V_i \right)^{-1}$,
   - $\mu_{b_j}^q \leftarrow \Sigma_{b_j}^q \sum_{i=1}^{n} q_{ij} V_i^T {\Sigma_{ij}^q}^{-1} \left(y_i - X_i \mu_{\eta_i}^q \right)$.

4. Same as step 4 in Algorithm 3.

5. For $i = 1, \ldots, n$, $j = 1, \ldots, k$, $q_{ij} \leftarrow \frac{p_{ij}(\mu_\gamma^q)\exp(c_{ij})}{\sum_{l=1}^k p_{il}(\mu_\gamma^q)\exp(c_{il})}$, where

$$\begin{aligned} c_{ij} = &-\tfrac{1}{2}\big[\omega_{ij}^T {\Sigma_{ij}^q}^{-1}\omega_{ij} + \mathrm{tr}\big\{{\Sigma_{ij}^q}^{-1}(X_i\Sigma_{\beta_j}^q X_i^T + V_i\Sigma_{b_j}^q V_i^T)\big\}\big] \\ &+ \tfrac{p}{2}\big\{\psi(\alpha_{a_j}^q) - \log\lambda_{a_j}^q\big\} + \sum_{l=1}^g \tfrac{\kappa_{il}}{2}\left\{\psi(\alpha_{jl}^q) - \log\lambda_{jl}^q\right\} \\ &- \tfrac{\alpha_{a_j}^q}{2\lambda_{a_j}^q}\big\{(\mu_{\eta_i}^q - \mu_{\beta_j}^q)^T(\mu_{\eta_i}^q - \mu_{\beta_j}^q) + \mathrm{tr}(\Sigma_{\eta_i}^q + \Sigma_{\beta_j}^q)\big\}. \end{aligned}$$

6. For $j = 1, \ldots, k$,

   - $\alpha_{a_j}^q \leftarrow \alpha_{a_j} + \frac{p}{2}\sum_{i=1}^n q_{ij}$,
   - $\lambda_{a_j}^q \leftarrow \lambda_{a_j} + \sum_{i=1}^n \frac{q_{ij}}{2}\{(\mu_{\eta_i}^q - \mu_{\beta_j}^q)^T(\mu_{\eta_i}^q - \mu_{\beta_j}^q) + \mathrm{tr}(\Sigma_{\eta_i}^q + \Sigma_{\beta_j}^q)\}$.

7. Same as step 7 in Algorithm 3

8. For $j = 1, \ldots, k$, $l = 1, \ldots, g$,

   - $\alpha_{jl}^q \leftarrow \alpha_{jl} + \frac{1}{2}\sum_{i=1}^n q_{ij}\kappa_{il}$,
   - $\lambda_{jl}^q \leftarrow \lambda_{jl} + \sum_{i=1}^n \frac{q_{ij}}{2}\big\{(\omega_{ij})_{\kappa_{il}}^T(\omega_{ij})_{\kappa_{il}} + \mathrm{tr}(X_i\Sigma_{\eta_i}^q X_i^T + V_i\Sigma_{b_j}^q V_i^T)_{\kappa_{il}}\big\}$,

   where $\omega_{ij} = y_i - X_i\mu_{\eta_i}^q - V_i\mu_{b_j}^q$.

until the increase in $\mathcal{L}$ is negligible.

The variational lower bound $\mathcal{L}$ for Algorithm 4 is the same as that in (3.3) except that

$$\tfrac{1}{2}\sum_{i=1}^n \log|\Sigma_{a_i}^q| - \sum_{i=1}^n\sum_{j=1}^k \tfrac{q_{ij}}{2}\left[\xi_{ij}^T {\Sigma_{ij}^q}^{-1}\xi_{ij} + \mathrm{tr}(\Sigma_{ij}^{q\,-1}\Lambda_{ij}) + \tfrac{\alpha_{a_j}^q}{\lambda_{a_j}^q}\big\{{\mu_{a_i}^q}^T\mu_{a_i}^q + \mathrm{tr}(\Sigma_{a_i}^q)\big\}\right]$$

is replaced with

$$\begin{aligned} &\tfrac{1}{2}\sum_{i=1}^n \log|\Sigma_{\eta_i}^q| + \tfrac{\alpha_{a_j}^q}{\lambda_{a_j}^q}\big\{\mathrm{tr}(\Sigma_{\eta_i}^q + \Sigma_{\beta_j}^q) + (\mu_{\eta_i}^q - \mu_{\beta_j}^q)^T(\mu_{\eta_i}^q - \mu_{\beta_j}^q)\big\}\Big] \\ &\qquad - \sum_{i=1}^n\sum_{j=1}^k \tfrac{q_{ij}}{2}\Big[\omega_{ij}^T {\Sigma_{ij}^q}^{-1}\omega_{ij} + \mathrm{tr}\big\{{\Sigma_{ij}^q}^{-1}(X_i\Sigma_{\eta_i}^q X_i^T + V_i\Sigma_{b_j}^q V_i^T)\big\}, \end{aligned}$$

where $\omega_{ij} = y_i - X_i\mu_{\eta_i}^q - V_i\mu_{b_j}^q$. For the examples in Section 3.7, when Algorithm 4 is used in conjunction with the VGA to fit a one-component

mixture ($j = 1$), we set $q_{ij} = 1$ for $i = 1, \ldots, n$, $\frac{\alpha^q_{jl}}{\lambda^q_{jl}} = 1$ for $l = 1, \ldots, g$, $\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} = 1$, $\frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} = 0.1$, $\mu^q_{b_j} = 0$ and $\mu^q_{\beta_j} = 0$ for initialization.

Algorithm 5: Variational approximation for MLMM with full centering

Initialize: q_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, k$, $\mu^q_{\nu_j}$, $\mu^q_{\beta_j}$ $\frac{\alpha^q_{a_j}}{\lambda^q_{a_j}}$ and $\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}}$ for $j = 1, \ldots, k$ and $\frac{\alpha^q_{jl}}{\lambda^q_{jl}}$ for $j = 1, \ldots, k$, $l = 1, \ldots, g$.
Cycle:

1. For $i = 1, \ldots, n$,
   - $\Sigma^q_{\rho_i} \leftarrow \left\{ \sum_{j=1}^k q_{ij} \frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} I_p + X_i^T (\sum_{j=1}^k q_{ij} {\Sigma^q_{ij}}^{-1}) X_i \right\}^{-1}$,
   - $\mu^q_{\rho_i} \leftarrow \Sigma^q_{\rho_i} \sum_{j=1}^k q_{ij} \left(\frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} \mu^q_{\nu_j} + X_i^T {\Sigma^q_{ij}}^{-1} y_i \right)$.
2. For $j = 1, \ldots, k$,
   - $\Sigma^q_{\nu_j} \leftarrow \left\{ \left(\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} + \frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} \sum_{i=1}^n q_{ij} \right) I_p \right\}^{-1}$,
   - $\mu^q_{\nu_j} \leftarrow \Sigma^q_{\nu_j} \left(\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} \mu^q_{\beta_j} + \frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} \sum_{i=1}^n q_{ij} \mu^q_{\rho_i} \right)$.
3. For $j = 1, \ldots, k$,
   - $\Sigma^q_{\beta_j} \leftarrow \left(\Sigma^{-1}_{\beta_j} + \frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} I_p \right)^{-1}$,
   - $\mu^q_{\beta_j} \leftarrow \Sigma^q_{\beta j} \frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} \mu^q_{\nu_j}$.
4. Same as step 4 in Algorithm 3.
5. For $i = 1, \ldots, n$, $j = 1, \ldots, k$, $q_{ij} \leftarrow \frac{p_{ij}(\mu^q_\gamma) \exp(c_{ij})}{\sum_{l=1}^k p_{il}(\mu^q_\gamma) \exp(c_{il})}$ where

$$
\begin{aligned}
c_{ij} = & -\tfrac{1}{2} \big\{ (y_i - X_i \mu^q_{\rho_i})^T {\Sigma^q_{ij}}^{-1} (y_i - X_i \mu^q_{\rho_i}) + \operatorname{tr}({\Sigma^q_{ij}}^{-1} X_i \Sigma^q_{\rho_i} X_i^T) \big\} \\
& - \tfrac{\alpha^q_{a_j}}{2\lambda^q_{a_j}} \big\{ (\mu^q_{\rho_i} - \mu^q_{\nu_j})^T (\mu^q_{\rho_i} - \mu^q_{\nu_j}) + \operatorname{tr}(\Sigma^q_{\rho_i} + \Sigma^q_{\nu_j}) \big\} \\
& + \tfrac{p}{2} \big\{ \psi(\alpha^q_{a_j}) - \log \lambda^q_{a_j} \big\} + \sum_{l=1}^g \tfrac{\kappa_{il}}{2} \big\{ \psi(\alpha^q_{jl}) - \log \lambda^q_{jl} \big\}.
\end{aligned}
$$

6. For $j = 1, \ldots, k$,
   - $\alpha^q_{a_j} \leftarrow \alpha_{a_j} + \frac{p}{2} \sum_{i=1}^n q_{ij}$,
   - $\lambda^q_{a_j} \leftarrow \lambda_{a_j} + \frac{1}{2} \sum_{i=1}^n q_{ij} \big\{ (\mu^q_{\rho_i} - \mu^q_{\nu_j})^T (\mu^q_{\rho_i} - \mu^q_{\nu_j}) + \operatorname{tr}(\Sigma^q_{\rho_i} + \Sigma^q_{\nu_j}) \big\}$.

7. For $j = 1, \ldots, k$,

   - $\alpha^q_{b_j} \leftarrow \alpha_{b_j} + \frac{p}{2}$,
   - $\lambda^q_{b_j} \leftarrow \lambda_{b_j} + \frac{1}{2}\Big\{(\mu^q_{\nu_j} - \mu^q_{\beta_j})^T(\mu^q_{\nu_j} - \mu^q_{\beta_j}) + \mathrm{tr}(\Sigma^q_{\nu_j} + \Sigma^q_{\beta_j})\Big\}$.

8. For $j = 1, \ldots, k$, $l = 1, \ldots, g$,

   - $\alpha^q_{jl} \leftarrow \alpha_{jl} + \frac{1}{2}\sum_{i=1}^n q_{ij}\kappa_{il}$,
   - $\lambda^q_{jl} \leftarrow \lambda_{jl} + \sum_{i=1}^n \frac{q_{ij}}{2}\big\{(y_i - X_i\mu^q_{\rho_i})^T_{\kappa_{il}}(y_i - X_i\mu^q_{\rho_i})_{\kappa_{il}} + \mathrm{tr}(X_i\Sigma^q_{\rho_i}X_i^T)_{\kappa_{il}}\big\}$.

until the increase in $\mathcal{L}$ is negligible.

The variational lower bound $\mathcal{L}$ for Algorithm 5 is the same as in (3.3) except that

$$\begin{aligned}
&\tfrac{1}{2}\sum_{i=1}^n \log|\Sigma^q_{a_i}| + \sum_{j=1}^k \Big[\tfrac{1}{2}\log|\Sigma^q_{b_j}| - \tfrac{\alpha^q_{b_j}}{2\lambda^q_{b_j}}\big\{\mu^{q\,T}_{b_j}\mu^q_{b_j} + \mathrm{tr}(\Sigma^q_{b_j})\big\}\Big] \\
&\quad - \sum_{i=1}^n\sum_{j=1}^k \tfrac{q_{ij}}{2}\Big[\xi^T_{ij}\Sigma^{q\,-1}_{ij}\xi_{ij} + \mathrm{tr}(\Sigma^{q\,-1}_{ij}\Lambda_{ij}) + \tfrac{\alpha^q_{a_j}}{\lambda^q_{a_j}}\big\{\mu^{q\,T}_{a_i}\mu^q_{a_i} + \mathrm{tr}(\Sigma^q_{a_i})\big\}\Big]
\end{aligned}$$

is replaced with

$$\begin{aligned}
&\tfrac{1}{2}\sum_{i=1}^n \log|\Sigma^q_{\rho_i}| - \sum_{i=1}^n\sum_{j=1}^k \tfrac{q_{ij}}{2}\tfrac{\alpha^q_{a_j}}{\lambda^q_{a_j}}\big\{\mathrm{tr}(\Sigma^q_{\rho_i} + \Sigma^q_{\nu_j}) + (\mu^q_{\rho_i} - \mu^q_{\nu_j})^T(\mu^q_{\rho_i} - \mu^q_{\nu_j})\big\} \\
&\quad - \sum_{i=1}^n\sum_{j=1}^k \tfrac{q_{ij}}{2}\Big[\mathrm{tr}\big\{\Sigma^{q\,-1}_{ij}(X_i\Sigma^q_{\rho_i}X_i^T)\big\} + (y_i - X_i\mu^q_{\rho_i})^T\Sigma^{q\,-1}_{ij}(y_i - X_i\mu^q_{\rho_i})\Big] \\
&\quad + \tfrac{1}{2}\sum_{j=1}^k \Big[\log|\Sigma^q_{\nu_j}| - \tfrac{\alpha^q_{b_j}}{\lambda^q_{b_j}}\big\{(\mu^q_{\nu_j} - \mu^q_{\beta_j})^T(\mu^q_{\nu_j} - \mu^q_{\beta_j}) + \mathrm{tr}(\Sigma^q_{\nu_j} + \Sigma^q_{\beta_j})\big\}\Big].
\end{aligned}$$

For the examples in Section 3.7, when Algorithm 5 is used in conjunction with the VGA to fit a one-component mixture ($j = 1$), we set $q_{ij} = 1$ for $i = 1, \ldots, n$, $\frac{\alpha^q_{jl}}{\lambda^q_{jl}} = 10$ for $l = 1, \ldots, g$, $\frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} = 0.1$, $\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} = 0.01$, $\mu^q_{\beta_j} = 0$ and $\mu^q_{\nu_j} = 0$ for initialization. We note that the rate of convergence of Algorithm 5 can be sensitive to the initialization of $\frac{\alpha^q_{jl}}{\lambda^q_{jl}}$, $\frac{\alpha^q_{a_j}}{\lambda^q_{a_j}}$ and $\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}}$ and observed that an initialization satisfying $\frac{\alpha^q_{b_j}}{\lambda^q_{b_j}} < \frac{\alpha^q_{a_j}}{\lambda^q_{a_j}} < \frac{\alpha^q_{jl}}{\lambda^q_{jl}}$ works better.

3.5 Variational greedy algorithm

The VGA carries out model selection and parameter estimation simultaneously and is fully automatic. At the end of the algorithm, a plausible number of mixture components is returned together with the fitted model. The greedy approach described in this section is not limited to MLMMs and can be adapted to fit other mixture models using variational methods easily. In the description of the VGA below, "variational algorithm" refers to either Algorithms 3, 4 or 5 depending on whether any centering (either partial or full) is desired. Let f_k denote the k-component mixture model fitted to the data and C_k denote the set of k components that form the mixture model f_k. The greedy learning procedure is outlined below.

Variational Greedy Algorithm (VGA)

1. Fit a one-component mixture model f_1 to the data using the variational algorithm.

2. Find the optimal way of splitting each of the components that form the current mixture model f_k. This is done in the following manner. For each component $c_{j^*} \in C_k$, form

$$A_{j^*} = \left\{ i \in \{1, \ldots, n\} | j^* = \underset{1 \leq j \leq k}{\arg\max}\, q_{ij} \right\},$$

   where $\{q_{ij} | i = 1, \ldots, n, j = 1, \ldots, k\}$ are the responsibilities from f_k. For each $m = 1, \ldots, M$,

   - randomly partition A_{j^*} into two disjoint subsets A_{j1^*} and A_{j2^*}. Form a $(k+1)$-component mixture by splitting c_{j^*} into two subcomponents, c_{j1^*} and c_{j2^*}, while keeping the remaining $(k-1)$ components in C_k fixed. For $i \in A_{j^*}$ and $l \in \{1, 2\}$, let q_{ij} of c_{jl^*} be equal to the responsibilities of c_{j^*} in f_k if the ith observation lies in A_{jl^*} and zero otherwise. For $i \notin A_{j^*}$, let q_{ij} of c_{j1^*} be equal to the responsibilities of c_{j^*} in f_k and q_{ij} of c_{j2^*} be zero. The rest of the variational parameters of c_{j1^*} and c_{j2^*} which are required for initialization of the variational algorithm are set as equal to that of c_{j^*}.
   - Using this setting as initialization, apply a "partial" variational algorithm to the $(k+1)$-component mixture. Here, variational parameters of components in $C_k - c_{j^*}$ are not updated as we are only interested in learning the optimal way of splitting c_{j^*}.

For each component $c_{j^*} \in C_k$, choose the run with the highest attained lower bound among M runs as that yielding the optimal way of splitting c_{j^*}. Let $\mathcal{L}_{j^*}$ denote the lower bound and $f_{j^*}^{\text{split}}$ denote the $(k+1)$-component mixture model corresponding to the optimal way of splitting c_{j^*}.

3. The components in C_k are then sorted in descending order according to $\mathcal{L}_{j^*}$ and then split in order, starting with the component with the highest $\mathcal{L}_{j^*}$. After the lth split, the total number of components in the mixture is $k+l$. Let f_{k+l}^{temp} denote the mixture model obtained after l splits. Suppose that at the $(l+1)th$ split, the component in C_k being split is c_{j^*}. We apply a "partial" variational algorithm again, keeping fixed variational parameters of components awaiting to be split. For the initialization, we let the variational parameters of c_{j1^*} and c_{j2^*} be equal to those in $f_{j^*}^{\text{split}}$ and the variational parameters of all other components be equal to those in f_{k+l}^{temp} if $l \geq 1$ and $f_{j^*}^{\text{split}}$ if $l = 0$. A split is considered successful if the estimated log marginal likelihood $\mathcal{L}^*$ increases after the split. This process of splitting components is terminated once an unsuccessful split is encountered.

4. If the total number of successful splits in step 3 is s, then a $(k+s)$-component model f_{k+s}^{temp} is obtained at the end of step 3. We apply the variational algorithm on f_{k+s}^{temp} until convergence updating all variational parameters this time to obtain mixture model f_{k+s}.

5. Repeat steps 2–4 until all splits of the current mixture model are unsuccessful.

For the partitioning of A_{j^*} in step 2, we have experimented with several dissimilarity measures based on Euclidean distance as well as variability-weighted similarity measures (Yeung *et al.*, 2003) in the case of repeated data. Generally, the VGA performed better when a random partition was used. Methods such as k-means clustering are also difficult to apply when there is missing data. We note that the partitioning of A_{j^*} into two disjoint subsets in step 2 serves only as an initialization to the "partial" variational algorithm to be carried out in search of the optimal way to split component c_{j^*}. Suppose an outright partitioning of the data is obtained by assigning observation i to the j^*th component if $j^* = \arg\max_{1 \leq j \leq k} q_{ij}$ where $\{q_{ij} | i = 1, \ldots, n, k = 1, \ldots, k\}$ are the responsibilities of f_k. We emphasize that it is possible for observations that have been assigned to different components

at any particular stage to be assigned to the same component again at the next stage of the VGA. This is due to the updating of the responsibilities q_{ij} of all components which have been split in step 3 and that of all existing components in step 4.

The amount of computation is greatly reduced by the use of a "partial" variational algorithm as the algorithm converges quickly when the variational parameters of all other components (except for the two subcomponents arising from the component being split) are fixed. In step 2, we are looking for the run with the highest attained lower bound out of M runs and it may not be computationally efficient to continue every run to full convergence. We suggest using "short runs" in this search step. For the examples in section 3.7, we set M as 5 and each of the M runs is terminated when the increment in the lower bound is less than 1. For steps 1, 3 and 4, the variational algorithm is considered to have converged when the absolute relative change in the lower bound $\mathcal{L}$ is less than 10^{-5}. Suppose we are trying to split a component c_{j^*} into two subcomponents c_{j1^*} and c_{j2^*}. After applying "partial" variational algorithm, the responsibilities q_{ij} of one of the two subcomponents sometimes reduce to zero for all of $i = 1, \ldots, n$, so that it is effectively removed. When this happens on the attempt leading to the highest variational lower bound among all M attempts to split c_{j^*}, we suggest omitting c_{j^*} in future splitting tests provided the responsibilities of c_{j^*} remain unchanged. This reduces the number of components we need to test for splitting and can be very useful when the number of components grows to a large number.

Due to the random partitions in step 2, repeated applications of the VGA may not return the same number of mixture components. However, empirical results indicate that the variation is relatively small compared to the number of components returned. If the user finds certain clusters to be very similar and suspect that the VGA may have overestimated the number of components, some optional merge moves may be carried out as we later demonstrate in Section 3.7. A merge move is considered successful if the estimated log marginal likelihood increases when two components are merged. While the VGA has been applied repeatedly in the examples for the purpose of analysing its performance, the user need only apply it once and may consider some merge moves if he/she finds clusters which are very similar. If multiple applications are used, we suggest using the estimated log marginal likelihood as a guideline to select the clustering solution. We observed that reparametrization using hierarchical centering increases the

efficiency of the VGA and a larger gain may be expected for mixtures with a larger number of components. The quality of the clustering results also seems to improve with hierarchical centering with a higher estimated log marginal likelihood being attained.

3.6 Rate of convergence

In this section, we show that the approximate rate of convergence of the variational algorithm by Gaussian approximation is equal to that of the corresponding Gibbs sampler. As reparametrizations using hierarchical centering can lead to improved convergence in the Gibbs sampler, this result lends insight into how such reparametrizations can increase the efficiency of variational algorithms in the context of MLMMs. This is because the joint posterior of the fixed and random effects in a linear mixed model is Gaussian (with Gaussian priors and Gaussian random effects distributions) when the variance parameters are known.

Let the complete data be $Y_{\text{aug}} = (Y_{\text{obs}}, Y_{\text{mis}})$ where Y_{obs} is the observed data and Y_{mis} is the missing data. Let the complete data likelihood be $p(Y_{\text{aug}}|\theta)$ where θ is a $p \times 1$ vector. Let Y_{mis} be a $r \times 1$ vector. Suppose the prior for θ is $p(\theta) \propto 1$ and the target distribution is $p(\theta, Y_{\text{mis}}|Y_{\text{obs}}) = N\left(\left(\begin{smallmatrix}\mu_1\\ \mu_2\end{smallmatrix}\right), \Sigma\right)$, where $\Sigma = \left(\begin{smallmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{smallmatrix}\right)$. Let $H = \Sigma^{-1} = \left(\begin{smallmatrix}H_{11} & H_{12}\\ H_{21} & H_{22}\end{smallmatrix}\right)$. It can be shown that

$$
\begin{aligned}
p(Y_{\text{mis}}|\theta, Y_{\text{obs}}) &= N\left(\mu_2 - H_{22}^{-1}H_{21}(\theta - \mu_1), H_{22}^{-1}\right) \quad \text{and}\\
p(\theta|Y_{\text{mis}}, Y_{\text{obs}}) &= N\left(\mu_1 - H_{11}^{-1}H_{12}(Y_{\text{mis}} - \mu_2), H_{11}^{-1}\right).
\end{aligned}
$$

Sahu and Roberts (1999) showed that under such conditions, the rate of convergence of the EM algorithm alternating between the two components θ and Y_{mis} is equal to the rate of convergence of the corresponding two-block Gibbs sampler. This rate is given by $\rho(B^{\text{EM}})$, where $B^{\text{EM}} = H_{11}^{-1}H_{12}H_{22}^{-1}H_{21}$ and $\rho(\cdot)$ denotes the spectral radius of a matrix.

In the variational approach, we seek an approximation $q(\theta, Y_{\text{mis}})$ to the true posterior $p(\theta, Y_{\text{mis}}|Y_{\text{obs}})$ for which the Kullback-Leibler divergence between q and $p(\theta, Y_{\text{mis}}|Y_{\text{obs}})$ is minimized subject to the restriction that $q(\theta, Y_{\text{mis}})$ can be factorized as $q(\theta)q(Y_{\text{mis}})$. The optimal densities are

$$
\begin{aligned}
q(Y_{\text{mis}}) &= N\left(\mu_2 - H_{22}^{-1}H_{21}(\mu_\theta^q - \mu_1),\ H_{22}^{-1}\right) \quad \text{and}\\
q(\theta) &= N\left(\mu_1 - H_{11}^{-1}H_{12}(\mu_{Y_{\text{mis}}}^q - \mu_2),\ H_{11}^{-1}\right),
\end{aligned}
$$

where μ_θ^q and $\mu_{Y_{\text{mis}}}^q$ denote the mean of $q(\theta)$ and $q(Y_{\text{mis}})$ respectively. Starting with some initial estimate for μ_θ^q, we can iteratively update the parameters μ_θ^q and $\mu_{Y_{\text{mis}}}^q$ until convergence. Let $\mu_\theta^{q\,(t)}$ and $\mu_{Y_{\text{mis}}}^{q\,(t)}$ denote the tth iterates. It can be shown that

$$\mu_{Y_{\text{mis}}}^{q\,(t+1)} = H_{22}^{-1} H_{21} H_{11}^{-1} H_{12}\, \mu_{Y_{\text{mis}}}^{q\,(t)} + \left(I_r - H_{22}^{-1} H_{21} H_{11}^{-1} H_{12}\right) \mu_2 \quad \text{and}$$
$$\mu_\theta^{q(t+1)} = B^{\text{EM}} \mu_\theta^{q(t)} + \left(I_p - B^{\text{EM}}\right) \mu_1.$$

The matrix rate of convergence of an iterative algorithm for which $\theta^{(t+1)} = M(\theta^{(t)})$ and θ^* is the limit is given by $\text{DM}(\theta^*)$ where $\text{DM}(\theta) = (\frac{\partial M_j(\theta)}{\partial \theta_i})$. A measure of the actual observed rate of convergence is given by the largest eigenvalue of $\text{DM}(\theta^*)$ (Meng, 1994). The rate of convergence of $\mu_\theta^{q\,(t)}$ is therefore $\rho(B^{\text{EM}})$. Since $H_{22}^{-1} H_{21} H_{11}^{-1} H_{12}$ and B^{EM} share the same eigenvalues, the rate of convergence of $\mu_{Y_{\text{mis}}}^{q\,(t)}$ is also $\rho(B^{\text{EM}})$. The overall rate of convergence of the variational algorithm is thus $\rho(B^{\text{EM}})$.

Suppose we impose a tougher restriction on $q(\theta, Y_{\text{mis}})$. For a partition of θ into m groups such that $\theta = [\theta_1^T, \ldots, \theta_m^T]^T$ with θ_i a $r_i \times 1$ vector and $\sum r_i = p$, we assume that $q(\theta, Y_{\text{mis}})$ can be factorized as $\prod_{i=1}^m q(\theta_i) q(Y_{\text{mis}})$. The optimal density of $q(Y_{\text{mis}})$ remains unchanged. Let $\mu_1 = (\mu_{11}, \ldots, \mu_{1m})$ and

$$H_{11} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \ldots & \Lambda_{1m} \\ \Lambda_{21} & \Lambda_{22} & \ldots & \Lambda_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{m1} & \Lambda_{m2} & \ldots & \Lambda_{mm} \end{bmatrix}.$$

be partitioned according to $\theta = (\theta_1, \ldots, \theta_m)$. The optimal density of $q(\theta_i)$ is

$$N\Big(\mu_{1i} - \Lambda_{ii}^{-1}\Big\{ \sum_{j \neq i} \Lambda_{ij}(\mu_{\theta_j}^q - \mu_{1j}) + H_{12i}(\mu_{Y_{\text{mis}}}^q - \mu_2)\Big\},\, \Lambda_{ii}^{-1}\Big)$$

where H_{12i} denotes the ith row of H_{12}, for $i = 1, \ldots, m$. This leads to the following iterative scheme. After initializing $\mu_{\theta_i}^q$, $i = 1, \ldots, m$, we cycle though updates:

- $\mu_{Y_{\text{mis}}}^q \leftarrow \mu_2 - H_{22}^{-1} H_{21}(\mu_\theta^q - \mu_1)$,
- $\mu_{\theta_i}^q \leftarrow \mu_{1i} - \Lambda_{ii}^{-1}\big\{ \sum_{j \neq i} \Lambda_{ij}(\mu_{\theta_j}^q - \mu_{1j}) + H_{12i}(\mu_{Y_{\text{mis}}}^q - \mu_2)\big\}$ for $i = 1, \ldots, m$,

till convergence. Consider the $(t+1)th$ iteration. For notational simplicity, we replace $(\mu_{\theta_i}^{q\,(t)} - \mu_{1i})$ by $\lambda_{\theta_i}^{q\,(t)}$, $(\mu_\theta^{q\,(t)} - \mu_1)$ by $\lambda_\theta^{q\,(t)}$ and $(\mu_{Y_{\text{mis}}}^{q\,(t)} - \mu_2)$ by

$\lambda_{Y_{\text{mis}}}^{q\,(t)}$. Since $\lambda_{Y_{\text{mis}}}^{q\,(t+1)} = -H_{22}^{-1}H_{21}\lambda_\theta^{q(t)}$, we have

$$\begin{bmatrix} \Lambda_{11} & 0 & \dots & 0 \\ \Lambda_{21} & \Lambda_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{m1} & \Lambda_{m2} & \dots & \Lambda_{mm} \end{bmatrix} \begin{bmatrix} \lambda_{\theta_1}^{q\,(t+1)} \\ \lambda_{\theta_2}^{q\,(t+1)} \\ \vdots \\ \lambda_{\theta_m}^{q\,(t+1)} \end{bmatrix} + \begin{bmatrix} 0 & \Lambda_{12} & \dots & \Lambda_{1m} \\ 0 & 0 & \dots & \Lambda_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \lambda_{\theta_1}^{q\,(t)} \\ \lambda_{\theta_2}^{q\,(t)} \\ \vdots \\ \lambda_{\theta_m}^{q\,(t)} \end{bmatrix} = H_{11}B^{\text{EM}}\lambda_\theta^{q(t)}.$$

Let

$$L = \begin{bmatrix} \Lambda_{11} & 0 & \dots & 0 \\ \Lambda_{21} & \Lambda_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{m1} & \Lambda_{m2} & \dots & \Lambda_{mm} \end{bmatrix}$$

be the lower triangular block matrix of H_{11} and $U = L - H_{11}$. Then

$$\begin{aligned} L\lambda_\theta^{q(t+1)} - U\lambda_\theta^{q(t)} &= H_{11}B^{\text{EM}}\lambda_\theta^{q(t)} \\ \Leftrightarrow \lambda_\theta^{q(t+1)} &= L^{-1}U\lambda_\theta^{q(t)} + L^{-1}(L-U)B^{\text{EM}}\lambda_\theta^{q(t)} \\ \Leftrightarrow \lambda_\theta^{q(t+1)} &= [B_{\text{aug}} + (I_p - B_{\text{aug}})B^{\text{EM}}]\lambda_\theta^{q(t)} \end{aligned}$$

where $B_{\text{aug}} = L^{-1}U$. Therefore the rate of convergence of $\lambda_\theta^{q\,(t)}$ and hence, that of μ_θ^q is $\rho(B_{\text{aug}} + (I_p - B_{\text{aug}})B^{\text{EM}})$. As the rate of convergence of $\theta^{(t)}$ is defined as $\lim_{t\to\infty} \frac{\left\|\theta^{(t+1)}-\theta^*\right\|}{\left\|\theta^{(t)}-\theta^*\right\|}$, the rate of convergence of $\lambda_{Y_{\text{mis}}}^{q\,(t)}$ and hence $\mu_{Y_{\text{mis}}}^{q\,(t)}$ is given by

$$\begin{aligned} \lim_{t\to\infty} \frac{\left\|\lambda_{Y_{\text{mis}}}^{q\,(t+1)} - \lambda_{Y_{\text{mis}}}^{q*}\right\|}{\left\|\lambda_{Y_{\text{mis}}}^{q\,(t)} - \lambda_{Y_{\text{mis}}}^{q*}\right\|} &= \lim_{t\to\infty} \frac{\left\|-H_{22}^{-1}H_{21}\lambda_\theta^{q(t)} + H_{22}^{-1}H_{21}\lambda_\theta^{q*}\right\|}{\left\|-H_{22}^{-1}H_{21}\lambda_\theta^{q(t-1)} + H_{22}^{-1}H_{21}\lambda_\theta^{q*}\right\|} \\ &= \lim_{t\to\infty} \frac{\left\|\lambda_\theta^{q(t)} - \lambda_\theta^{q*}\right\|}{\left\|\lambda_\theta^{q(t-1)} - \lambda_\theta^{q*}\right\|}, \end{aligned}$$

which is equal to the rate of convergence of $\mu_\theta^{q\,(t)}$. The overall rate of convergence of the variational algorithm is thus $\rho(B_{\text{aug}} + (I_p - B_{\text{aug}})B^{\text{EM}})$ which is equal to the rate of convergence of the Gibbs sampler that sequentially updates components of θ, and then block updates Y_{mis} derived by Sahu and Roberts (1999). Although the theory developed may not be directly applicable to linear mixed models with unknown variance components as well as MLMMs in general, it suggests to consider hierarchical centering in the context of variational algorithms and examples in Section 3.7 show that there is some gain in efficiency due to the reparametrizations.

3.7 Examples

To illustrate the methods proposed, we apply VGA using Algorithms 3, 4 and 5 on three real data sets. We also consider a simulated data set created by Yeung *et al.* (2003) where there is independent external knowledge on which objects should cluster together. In Sections 3.7.2, 3.7.3 and 3.7.4, we compare results obtained without applying hierarchical centering with those obtained via either partial centering or full centering. We observed that hierarchical centering was able to not only increase efficiency but also produce better clustering results. In the examples below, an outright partitioning of the data is obtained by assigning observation i to the j^*th component if $j^* = \arg\max_{1 \leq j \leq k} q_{ij}$, where $\{q_{ij} | i = 1, \ldots, n, j = 1, \ldots, k\}$ are the responsibilities from the variational posterior of the mixture model. All code was written in the R language and run on a dual processor Windows PC 3GHz workstation.

3.7.1 Time course data

Using DNA microarrays and samples from yeast cultures synchronized by three independent methods, Spellman *et al.* (1998) identified 800 genes that meet an objective minimum criterion for cell cycle regulation. We consider the 18 α-factor synchronization where the yeast cells were sampled at 7 min intervals for 119 mins and a subset of 612 genes that have no missing gene expression data across all 18 time points. This data set was analyzed by Luan and Li (2003) and Ng *et al.* (2006) previously and is available online at `http://www.molbiolcell.org/content/9/12/3273/suppl/DC1`.

Our aim is to obtain an optimal clustering of these genes using VGA. Following Ng *et al.* (2006), we take $n = 612$, X_i to be an 18×2 matrix with the $(l+1)$th row $(l = 0, \ldots, 17)$ as $(\cos\{2\pi(7l)/\omega\}, \sin\{2\pi(7l)/\omega\})$, where $\omega = 53$ is the period of the cell cycle, $W_i = 1_{18}$, $V_i = I_{18}$ and $u_i = 1$ for $i = 1, \ldots, n$. For the error terms, we take $g = 1$ and $\kappa_{i1} = 18$ for $i = 1, \ldots, n$, so that the error variance of each mixture component is constant across the 18 time points. We used the following priors, $\gamma \sim N(0, 1000I)$, $\beta_j \sim N(0, 1000I)$ for $j = 1, \ldots, k$, and $IG(2, 0.25)$ for $\sigma^2_{a_j}$, $\sigma^2_{b_j}$, $j = 1, \ldots, k$ and σ^2_{jl}, $j = 1, \ldots, k$, $l = 1, \ldots, g$.

Applying VGA using Algorithm 3 ten times, we obtained a 15-component mixture three times, a 17-component mixture five times and a 18-component mixture twice. After applying merge moves to clusters which appear similar, three of the 17-component mixtures were reduced to 16-component

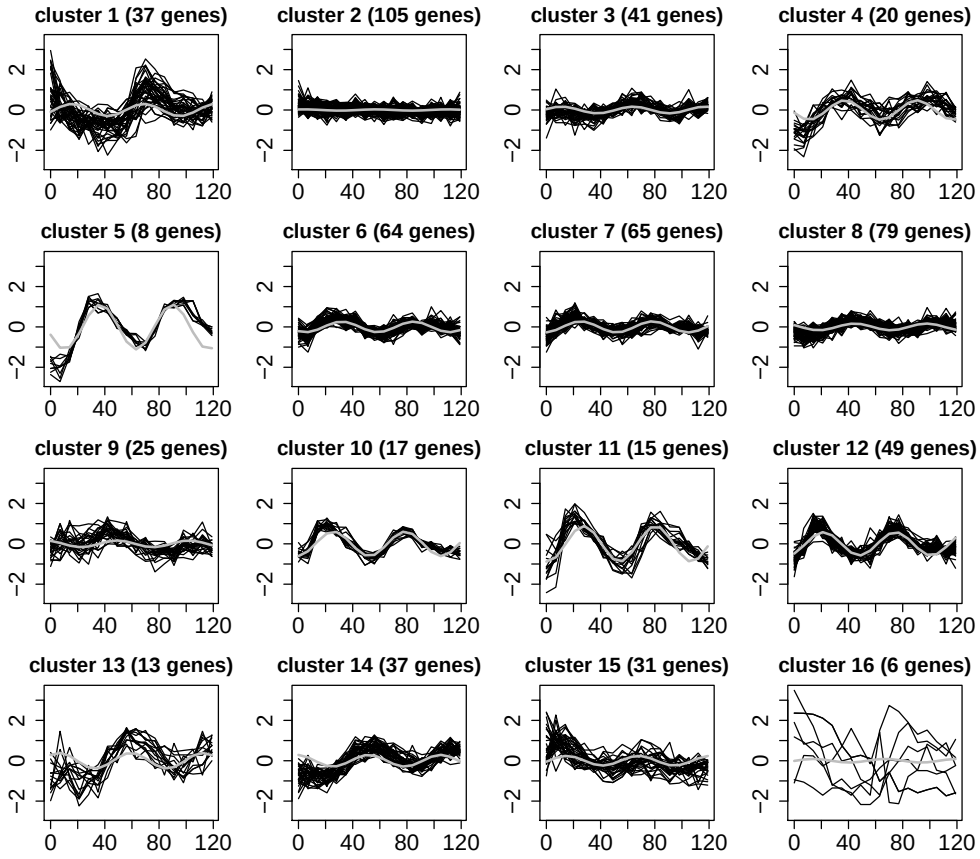

Figure 3.1: Time course data. Clustering results obtained after applying one merge move to a 17-component mixture produced by VGA using Algorithm 3. The x-axis are the time points and y-axis are the gene expression levels. Line in grey is the posterior mean of the fixed effects given by $X_i \mu^q_{\beta_j}$.

mixtures and both of the 18-component mixtures were reduced to 17-component mixtures. We report in Figure 3.1 the clustering for a 16-component mixture, obtained after applying one merge move to a 17-component mixture produced by VGA. For this clustering, we attempted further merge moves such as merging cluster 13 with 14, cluster 10 with 12 and cluster 8 with 9. These merge moves did not result in any further increase in the estimated log marginal likelihood.

While it is possible for the VGA to overestimate the number of mixture components, the variation in the number of mixture components returned by the VGA is relatively small and merge moves can be considered when very similar clusters are encountered. For this data set, the number of clusters returned by VGA was generally larger than that obtained by Ng *et al.* (2006) where BIC was used for model selection and the optimal number of clusters was reported as 12. Any interpretation of the differences in results would need to be pursued with the help of subject matter experts.

It may also be argued that the ability to estimate the "true model" is not a chief concern in clustering applications where interpretability of the results in the substantive scientific context is the primary motivation.

3.7.2 Synthetic data set

We consider a synthetic data set created by Yeung *et al.* (2003) which consist of 400 data points (genes), 20 attributes (experiments), 4 repeated measurements and 6 clusters. Clusters 1–4 are periodic sine functions each of size 67 and clusters 5–6 are linear each of size 66. For gene i from cluster j, the rth repeated measurement at experiment t is y_{itr}, which is generated randomly from a normal distribution with mean ϕ_{it} and standard deviation σ_{it}. The mean ϕ_{it} is defined as

$$\phi_{it} = \begin{cases} \sin(\frac{2\pi t}{10} - \omega_j) & \text{if } j = 1, \ldots, 4, \\ \frac{t}{20} & \text{if } j = 5, \\ -\frac{t}{20} & \text{if } j = 6, \end{cases}$$

where ω_j is a random phase shift between 0 and 2π and σ_{it} represents randomly sampled error from the yeast galactose data of Ideker *et al.* (2001). The synthetic data set we used is shown in Figure 3.2, sorted according to the true clusterings, and can be accessed from `http://expression.washington.edu/publications/kayee/yeunggb2003/` under the filename "syn_sine_5_mult1".

We take $n = 400$, $y_i = (y_{i11}, \ldots, y_{i14}, \ldots, y_{i,20,1}, \ldots, y_{i,20,4})$, X_i to be a 80×20 matrix where

$$X_i = \begin{bmatrix} 1_4 & 0_4 & \cdots & 0_4 \\ 0_4 & 1_4 & \cdots & 0_4 \\ \vdots & \vdots & \ddots & \vdots \\ 0_4 & 0_4 & \cdots & 1_4 \end{bmatrix}$$

$W_i = X_i$ and $V_i = I_{80}$ for $i = 1, \ldots, n$. For the error terms, we set $g = 20$ with $\kappa_{il} = 4$, $i = 1, \ldots, n$, $l = 1, \ldots, g$, so that the error variance of each mixture component is allowed to vary between different experiments. We used the following priors, $\gamma \sim N(0, 1000I)$, $\beta_j \sim N(0, 1000I)$ for $j = 1, \ldots, k$, and $IG(2, 0.74)$ for $\sigma^2_{a_j}$, $\sigma^2_{b_j}$, $j = 1, \ldots, k$ and σ^2_{jl}, $j = 1, \ldots, k$, $l = 1, \ldots, g$.

Applying VGA using Algorithm 4 (with partial centering) five times, we obtained a 6-component mixture three times and a 7-component mixture

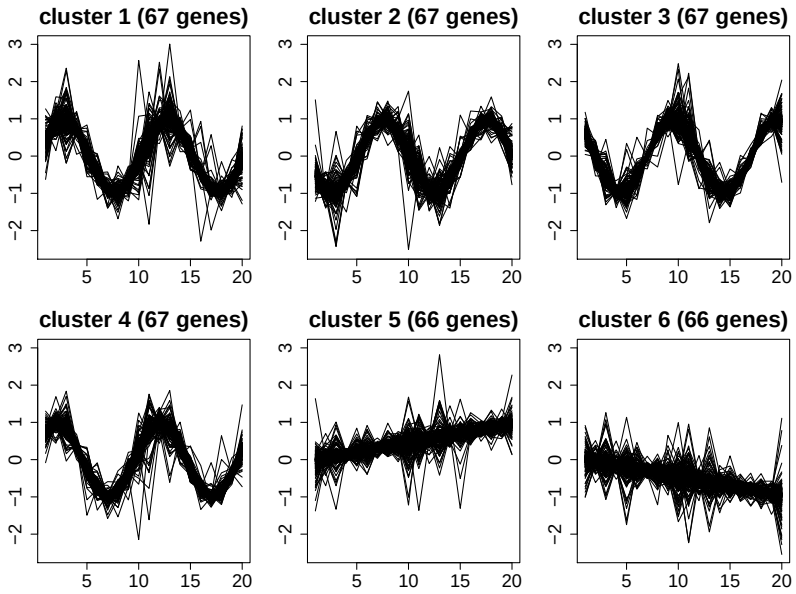

Figure 3.2: Expression profiles of synthetic data set sorted according to the true clusterings. The x-axis are the experiments and y-axis are the gene expression levels.

twice. Further merge moves were considered for the two 7-component mixtures but these were unsuccessful. For assessing the degree of agreement between the clustering of the fitted model relative to the true grouping of the 400 genes, we use the Adjusted Rand Index (ARI, Hubert and Arabie, 1985). The ARI can be used for comparing partitions with different number of clusters, with a value between 0 and 1, and is 1 when two partitions are in complete agreement. A higher value indicates better agreement between the two partitions. We compute the ARI for each of the five trials, which gave an average of 0.99. On the other hand, applying VGA using Algorithm 3 (without hierarchical centering) five times produced a 2-component mixture with an ARI less than 0.01 each time. Hierarchical centering thus produced much better clustering results in this case although it is difficult to compare the efficiency of Algorithms 1 and 2 due to the large difference in number of components returned.

3.7.3 Water temperature data

We consider the daily average water temperature readings during the period 9 September 2010 - 10 August 2011 collected at a monitoring station at Upper Peirce Reservoir, Singapore. No data were available during the periods 23 December 2010 - 28 December 2010, 10 February 2010 - 23 February 2010 and 14 April 2011 - 10 May 2011. Readings were collected

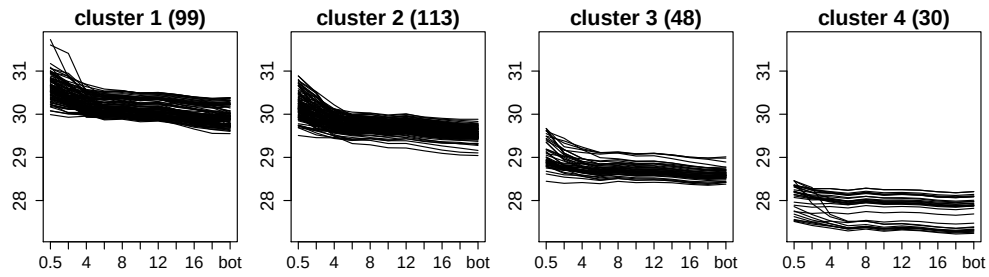

Figure 3.3: Clustering results for water temperature data. The x-axis is the depth and y-axis is the water temperature.

at eleven depths from the water surface; 0.5 m, 2 m, 4 m, 6 m, 8 m, 10 m, 12 m, 14 m, 16 m, 18 m and at the bottom*. Using data from the remaining 290 days, we apply the VGA to obtain a clustering of this data. We take $n = 290$, $n_i = 11$ and $X_i = W_i = V_i = I_{11}$ for $i = 1, \ldots, n$. We set $g = 11$ with $\kappa_{il} = 1$ for $i = 1, \ldots, n$, $l = 1, \ldots, g$, so that the error variance of each mixture component is allowed to be different at different depths. For the mixture weights, we set $u_i = [1, i, i^2, i^3]$, $i = 1, \ldots, n$, and subsequently rescale columns 2–4 in the matrix $U = [u_1^T, \ldots, u_n^T]^T$ to take values between -1 and 1. We used the following priors, $\gamma \sim N(0, 1000I)$, $\beta_j \sim N(0, 10000I)$ for $j = 1, \ldots, k$, and $IG(2, 0.8)$ for $\sigma^2_{a_j}$, $\sigma^2_{b_j}$, $j = 1, \ldots, k$ and σ^2_{jl}, $j = 1, \ldots, k$, $l = 1, \ldots, g$.

Applying VGA using Algorithm 5 (with full centering) five times, we obtained a 4-component model each time with similar results. The clustering of a typical 4-component fitted model is shown in Figure 3.3 and the fitted probabilities from the mixing weights model are shown in Figure 3.4. For comparison, we apply VGA with Algorithm 3 (without hierarchical centering) five times. A 4-component mixture model was obtained on all five attempts. The average CPU time taken to fit a 4-component model using VGA with Algorithm 3 was 725 seconds compared to 469 seconds by Algorithm 5. In this example, hierarchical centering reparametrization has helped to improve the rate of convergence with the computation time reduced by 35%. The average log marginal likelihood attained using Algorithm 5 was -789, which is higher than the average of -837 obtained using Algorithm 3.

The Upper Peirce Reservoir uses aeration devices intended to mix the water at different depths, with the aim of controlling outbreaks of phytoplankton and algal scums. On days when these aeration devices are opera-

*We thank Singapore Delft Water Alliance for supplying the water temperature data set.

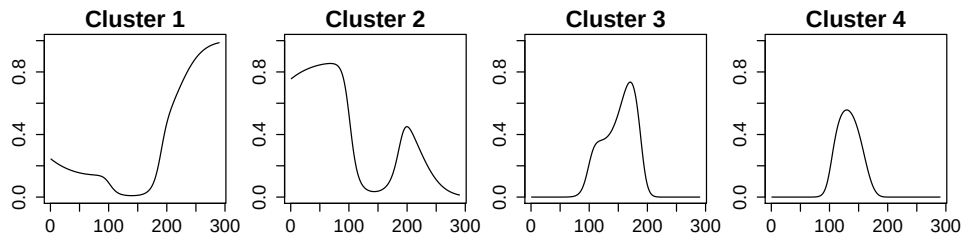

Figure 3.4: Water temperature data. Fitted probabilities from mixing weights model for clusters 1 to 4. The x-axis are days numbered 1 to 290 and y-axis are the probabilities.

tional, it is expected that there will be less stratification of the temperature with depth. Accurate records of the operation of the aeration devices were not available to us and there is some interest in seeing whether the clusters divide into more or less stratified components giving some insight into when the aeration devices were used.

3.7.4 Yeast galactose data

The yeast galactose data of Ideker *et al.* (2001) has four replicate hybridizations for each of 20 cDNA array experiments. We consider a subset of 205 genes previously analyzed by Yeung *et al.* (2003) and Ng *et al.* (2006) whose expression patterns reflect four functional categories in the gene ontology (GO) listings (Ashburner *et al.*, 2000). Approximately 8% of the data are missing and Yeung *et al.* (2003) used a k-nearest neighbour method to impute the missing data values. Yeung *et al.* (2003) and Ng *et al.* (2006) evaluated the performance of their clustering algorithms by how closely the clusters compared with the four categories in the GO listings. They used the ARI to assess the degree of agreement between their partitions and the four functional categories.

We use this example to illustrate the way that our model can make use of covariates in the mixing weights, unlike previous analyses of this data set. In particular, we use the GO listings as covariates in the mixture weights. Let u_i be a vector of length $d = 4$ where the lth element is 1 if the functional category of gene i is l and 0 otherwise. Instead of looking at the data with missing values imputed by the k-nearest neighbour method, we consider the original data containing 8% missing values, since our model has the capability to handle missing data. This data set can be accessed from `http://expression.washington.edu/publications/kayee/yeunggb2003/gal205.txt`. Taking $n = 205$ genes, let y_{itr} denote the rth repetition of the expression profile for gene i at experiment t,

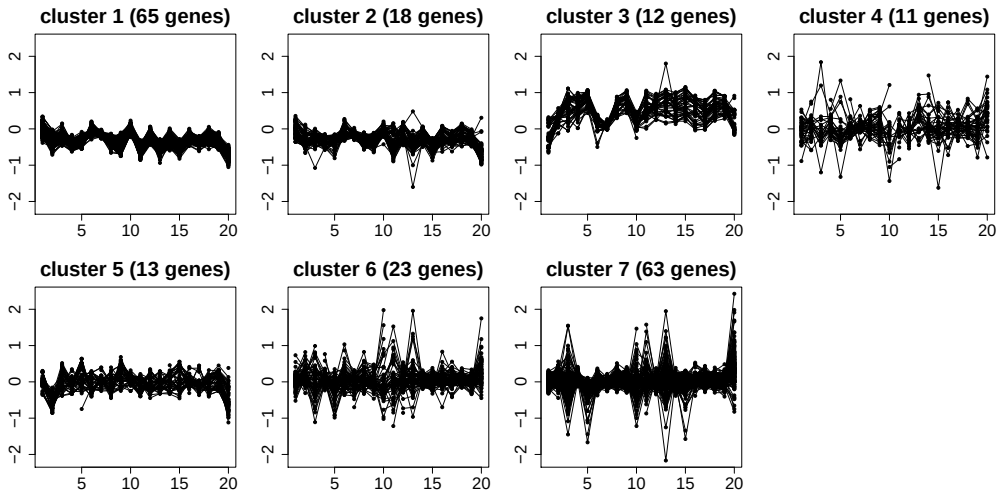

Figure 3.5: Clustering results for yeast galactose data obtained from VGA using Algorithm 4. The x-axis are the experiments and y-axis are the gene expression profiles. GO listings were used as covariates in the mixture weights.

$0 \leq r \leq 4$, and R_{it} denote the number of replicate hybridizations data available for gene i in experiment t, $i = 1, \ldots, 205$, $t = 1, \ldots, 20$. For each $i = 1, \ldots, n$, y_i is a vector of n_i observations where $n_i = \sum_{t=1}^{20} R_{it}$ and $y_i = (y_{i11}, \ldots, y_{i14}, \ldots, y_{i,20,1}, \ldots, y_{i,20,4})^T$, with missing observations omitted. V_i is a $n_i \times 80$ matrix obtained from I_{80} by removing the (tr)th row if the observation for experiment t at the rth repetition is not available. X_i is a $n_i \times 20$ matrix,

$$X_i = \begin{bmatrix} 1_{R_{i1}} & 0_{R_{i1}} & \ldots & 0_{R_{i1}} \\ 0_{R_{i2}} & 1_{R_{i2}} & \ldots & 0_{R_{i2}} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{R_{i20}} & 0_{R_{i20}} & \ldots & 1_{R_{i20}} \end{bmatrix}$$

and $W_i = X_i$. For the error terms, we set $g = 20$ with $\kappa_{il} = R_{il}$, $i = 1, \ldots, n$, $l = 1, \ldots, g$, so that the error variance of each mixture component is allowed to vary between different experiments. We used the following priors, $\delta \sim N(0, 1000I)$, $\beta_j \sim N(0, 1000I)$ for $j = 1, \ldots, k$, and $IG(2, 0.12)$ for $\sigma^2_{a_j}$, $\sigma^2_{b_j}$, $j = 1, \ldots, k$ and σ^2_{jl}, $j = 1, \ldots, k$, $l = 1, \ldots, g$.

Applying VGA using Algorithm 4 (with partial centering) for five times, we obtained a 7-component mixture on all five trials with similar results. The clustering of a 7-component mixture with the highest estimated log marginal likelihood among the five trials is shown in Figure 3.5. Some merge moves such as merging cluster 1 with 2, cluster 4 with 7 or cluster 4

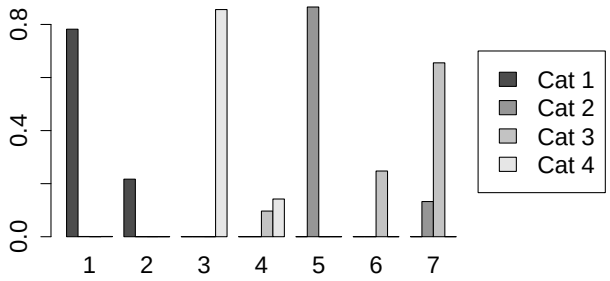

Figure 3.6: Yeast galactose data. Fitted probabilities from gating function. The x-axis are the clusters and y-axis are the probabilities.

with 6 were considered but these did not result in a higher estimated log marginal likelihood. The same holds for the other 7-component mixtures. The number of optimal clusters obtained using VGA is the same as that reported in Ng *et al.* (2006) although there are slight differences in the clusterings. In particular, instead of having one cluster containing all the genes from Category 4, we observed that two or three of the genes in Category 4 were consistently separated from the cluster containing the remaining genes from Category 4. Fitted probabilities from the gating function are shown in Figure 3.6. These were obtained by substituting δ with μ_δ^q from the variational posterior into $P(\delta_i = j) = p_{ij} = \frac{\exp(u_i^T \delta_j)}{\sum_{l=1}^k \exp(u_i^T \delta_l)}$ which represents the probability that observation i belongs to component j of the mixture, conditional on the category that observation i belongs to in the GO listings.

To investigate the impact of reparametrizing the model using hierarchical centering, we applied VGA using Algorithm 3 five times. This time, we obtained a 6-component mixture twice and a 7-component mixture thrice. The average estimated log marginal log likelihood attained by Algorithm 3 was 7901 which is lower than the average of 8201 attained by Algorithm 4. For fitting a 7-component model, VGA with Algorithm 3 took an average of 3418 seconds, while Algorithm 4 took an average of 1758 seconds. While these results may not be conclusive, the gain in efficiency in using Algorithm 4 over Algorithm 3 is clear. By using hierarchical centering, the computation time was reduced by nearly half in this example.

3.8 Conclusion

In this chapter, we have proposed fitting MLMMs with variational methods and developed an efficient VGA which is able to perform parameter estimation and model selection simultaneously. This greedy approach handles

initialization automatically and returns a plausible value for the number of mixture components. The experiments we have conducted showed that the VGA does not systematically underestimate nor overestimate the number of mixture components. For the simulated data set considered, VGA was able to return mixture models where the number of mixture components is very close to the true number of components. We further showed empirically that hierarchical centering can help to improve the rate of convergence in variational algorithms and return better clustering results. Some theoretical support was also provided for this observation. Implementation of the VGA is straightforward as no further derivation is required once the basic variational algorithms are available. This greedy approach is not limited to MLMMs and could potentially be extended to fitting other mixture models using variational methods. The R codes for implementing the VGA using algorithms 3, 4 and 5 and the water temperature data set are available online as supplemental materials of Tan and Nott (2013a).

Chapter 4

Variational inference for generalized linear mixed models using partially noncentered parametrizations

The effects of different parametrizations on the convergence of Bayesian computational algorithms for hierarchical models are well explored. Techniques such as centering, noncentering and partial noncentering have been used to accelerate convergence in MCMC and EM algorithms, but are still not well studied for VB methods. The use of different parametrizations for VB has not only computational but also statistical implications as different parametrizations are associated with different factorized posterior approximations. Here, we examine the use of partially noncentered parametrizations in the context of VB for generalized linear mixed models (GLMMs). First, we show how to implement an algorithm developed recently in machine learning called nonconjugate variational message passing (Knowles and Minka, 2011) for fitting GLMMs. Second, we show that the partially noncentered parametrization is able to adapt to the quantity of information in the data so that it is not necessary to make a choice in advance between centering and noncentering, with the data determining automatically a parametrization close to optimal. Third, we show that that in addition to accelerating convergence, partial noncentering is a good strategy statistically for VB in terms of producing more accurate approximations to the posterior than either centering or noncentering. Finally, we demonstrate how the variational lower bound, which is produced as part of the computation, can be useful for model selection. Note that the terms partial noncentering and partially noncentered introduced in this chapter do not have the same meaning as the term partial centering used in Chapter 3 and should not be confused.

This chapter is organized as follows. Section 4.1 provides some background and motivation for considering partial noncentering in the VB context. Section 4.2 specifies the GLMM and priors used. Section 4.3 describes a partially noncentered parametrization for GLMMs. Section 4.4 outlines the nonconjugate variational message passing algorithm for fitting GLMMs. Section 4.5 discusses briefly the use of the variational lower bound for model selection. Section 4.6 considers examples including real and simulated data and Section 4.7 concludes.

The results presented in this chapter have been published in Tan and Nott (2013b).

4.1 Background and motivation

GLMMs extend generalized linear models by the inclusion of random effects to account for correlation of observations in grouped data and are of wide applicability. Estimation of GLMMs using maximum likelihood is challenging as the integral over random effects is intractable and methods involving numerical quadrature or MCMC to approximate these integrals are computationally intensive. Various approximate methods such as penalized quasi-likelihood (Breslow *et al.*, 1993), Laplace approximation and its extension (Raudenbush *et al.*, 2000) and Gaussian variational approximation (Ormerod and Wand, 2012) have been developed. Fong *et al.* (2010) considered a Bayesian approach using integrated nested Laplace approximations. Stochastic approximation has also been used in conjunction with MCMC (Zhu *et al.*, 2002) and the EM algorithm (Jank, 2006) to fit GLMMs. We demonstrate how to fit GLMMs using nonconjugate variational message passing, focusing on Poisson and logistic mixed models and their applications in longitudinal data analysis. A brief review of VB methods and variational message passing is given in Section 1.1.

The convergence of MCMC algorithms depends greatly on the choice of parametrization and simple reparametrizations can often give improved convergence. The literature on parametrization of hierarchical models including partial noncentering techniques for accelerating MCMC algorithms is inspired by earlier similar work for the EM algorithm (see, e.g. Meng and van Dyk, 1997; Liu and Wu, 1999). Gelfand *et al.* (1995, 1996) proposed hierarchical centering for normal linear mixed models and GLMMs to improve the slow mixing in MCMC algorithms due to high correlations between model parameters. Papaspiliopoulos *et al.* (2003, 2007) demonstrated that centering and noncentering play complementary roles in boosting MCMC

efficiency and neither are uniformly effective. They considered the partially noncentered parametrization which is data dependent and lies on the continuum between the centered and noncentered parametrizations. Extending this idea, Christensen *et al.* (2006) devised reparametrization techniques to improve performance for Hastings-within-Gibbs algorithms for spatial GLMMs. Yu and Meng (2011) introduced a strategy for boosting MCMC efficiency via interweaving the centered and noncentered parametrizations to reduce dependence between draws. Parameter-expanded VB methods were proposed by Qi and Jaakkola (2006) to reduce coupling in updates and speed up VB.

The idea of partial noncentering is to introduce a tuning parameter via reparametrization of the model and then seek its optimal value for fastest convergence. For the normal hierarchical model, Papaspiliopoulos *et al.* (2003) showed that the partially noncentered parametrization has convergence properties superior to that of the centered and noncentered parametrizations for the Gibbs sampler. In Section 3.6, we have shown that the rate of convergence of an algorithm based on VB is equal to that of the corresponding Gibbs sampler when the target distribution is Gaussian. This implies that partial noncentering will similarly outperform centering and noncentering in the context of VB for the normal hierarchical model and provides motivation to consider partial noncentering in the VB context. We illustrate this idea with the following example.

4.1.1 Motivating example: linear mixed model

Consider the linear mixed model

$$y_i = X_i\beta + X_i u_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2 I), \quad i = 1, \ldots, n, \tag{4.1}$$

where y_i is a vector of length n_i, β is a vector of length r of fixed effects, X_i is a $n_i \times r$ matrix of covariates and u_i is a vector of length r of random effects independently distributed as $N(0, D)$. For simplicity, we specify a constant prior on β and assume σ^2 and D are known. Let

$$\alpha_i = \beta + u_i \quad \text{and} \quad \tilde{\alpha}_i = \alpha_i - W_i\beta, \quad i = 1, \ldots, n,$$

where W_i is an $r \times r$ tuning matrix to be specified. $W_i = 0$ corresponds to the centered and $W_i = I$ to the noncentered parametrization. We have

$$y_i = X_i W_i \beta + X_i \tilde{\alpha}_i + \epsilon_i \quad \text{and} \quad \tilde{\alpha}_i \sim N\left((I - W_i)\beta, D\right)$$

for each $i = 1, \ldots, n$. This is the partially noncentered parametrization and the set of unknown parameters is $\theta = \{\beta, \tilde{\alpha}\}$ where $\tilde{\alpha} = [\tilde{\alpha}_1^T, \ldots, \tilde{\alpha}_n^T]^T$.

Let $y = [y_1, \ldots, y_n]^T$ denote the observed data. Of interest is the posterior distribution of θ, $p(\theta|y)$. Suppose we use VB and approximate $p(\theta|y)$ with $q(\theta) = q(\beta)q(\tilde{\alpha})$. From (1.4), the optimal densities can be derived to be $q(\beta) = N(\mu_\beta^q, \Sigma_\beta^q)$ and $q(\tilde{\alpha}) = \prod_{i=1}^n q(\tilde{\alpha}_i)$ where $q(\tilde{\alpha}_i) = N(\mu_{\tilde{\alpha}_i}^q, \Sigma_{\tilde{\alpha}_i}^q)$. The variational parameters μ_β^q, Σ_β^q and $\mu_{\tilde{\alpha}_i}^q$, $\Sigma_{\tilde{\alpha}_i}^q$, $i = 1, \ldots, n$, are interdependent and can be computed using the iterative scheme in Algorithm 6.

Algorithm 6: VB for linear mixed model

Initialize $\mu_{\tilde{\alpha}_i}^q$ and $\Sigma_{\tilde{\alpha}_i}^q$ for $i = 1, \ldots, n$.
Cycle:

1.
   - $\Sigma_\beta^q \leftarrow \left[\sum_{i=1}^n \left\{(I - W_i)^T D^{-1}(I - W_i) + \frac{1}{\sigma^2} W_i^T X_i^T X_i W_i\right\}\right]^{-1}$,
   - $\mu_\beta^q \leftarrow \Sigma_\beta^q \sum_{i=1}^n \left[\frac{1}{\sigma^2} W_i^T X_i^T y_i + \{D^{-1}(I - W_i) - \frac{1}{\sigma^2} X_i^T X_i W_i\}^T \mu_{\tilde{\alpha}_i}^q\right]$.

2. For $i = 1, \ldots, n$,
   - $\Sigma_{\tilde{\alpha}_i}^q \leftarrow \left(D^{-1} + \frac{1}{\sigma^2} X_i^T X_i\right)^{-1}$,
   - $\mu_{\tilde{\alpha}_i}^q \leftarrow \Sigma_{\tilde{\alpha}_i}^q \left[\frac{1}{\sigma^2} X_i^T y_i + \{D^{-1}(I - W_i) - \frac{1}{\sigma^2} X_i^T X_i W_i\} \mu_\beta^q\right]$.

until convergence.

Observe that Algorithm 6 converges in one iteration if $D^{-1}(I - W_i) = \frac{1}{\sigma^2} X_i^T X_i W_i$ for each i, that is, if

$$W_i = (\tfrac{1}{\sigma^2} X_i^T X_i + D^{-1})^{-1} D^{-1}, \text{ for } i = 1, \ldots, n. \tag{4.2}$$

For this specification of the tuning parameters, partial noncentering gives more rapid convergence than centering or noncentering. Moreover, it can be shown that the true posteriors are recovered in this partially noncentered parametrization so that a better fit is achieved than in the centered or noncentered parametrizations. This example suggests that with careful tuning of W_i, $i = 1, \ldots, n$, the partially noncentered parametrization can potentially outperform the centered and noncentered parametrizations in the VB context.

4.2 Generalized linear mixed models

Consider clustered data where y_{ij} denotes the jth response from cluster i, $i = 1, \ldots, n$, $j = 1, \ldots, n_i$. Conditional on the r-dimensional random effects u_i drawn independently from $N(0, D)$, y_{ij} is independently distributed from some exponential family distribution with density

$$f(y_{ij}|u_i) = \exp\left\{\frac{y_{ij}\zeta_{ij} - b(\zeta_{ij})}{a(\varphi)} + c(y_{ij}, \varphi)\right\}, \tag{4.3}$$

where ζ_{ij} is the canonical parameter, φ is the dispersion parameter, and $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ are functions specific to the family. The conditional mean of y_{ij}, $\mu_{ij} = E(y_{ij}|u_i)$, is assumed to depend on the fixed and random effects through the linear predictor

$$\eta_{ij} = X_{ij}^{R^T}\beta^R + X_{ij}^{G^T}\beta^G + X_{ij}^{R^T}u_i,$$

with $g(\mu_{ij}) = \eta_{ij}$ for some known link function, $g(\cdot)$. Here, X_{ij}^R and $X_{ij} = \left[X_{ij}^{R^T},\ X_{ij}^{G^T}\right]^T$ are $r \times 1$ and $p \times 1$ vectors of covariates and $\beta = \left[\beta^{R^T}, \beta^{G^T}\right]^T$ is a $p \times 1$ vector of fixed effects. We have considered the above breakdown for the linear predictor to allow for centering (see Zhao *et al.*, 2006). For the ith cluster, let $y_i = [y_{i1}, \ldots, y_{in_i}]^T$, $X_i^R = [X_{i1}^R, \ldots, X_{in_i}^R]^T$, $X_i^G = [X_{i1}^G, \ldots, X_{in_i}^G]^T$, $X_i = [X_{i1}, \ldots, X_{in_i}]^T$ and $\eta_i = [\eta_{i1}, \ldots, \eta_{in_i}]^T$. We assume that the first column of X_i^R is 1_{n_i} if X_i^R is not a zero matrix.

We focus on responses from the Bernoulli and Poisson families. If $y_{ij} \sim$ Bernoulli(μ_{ij}), then $b(x) = \log\{1 + \exp(x)\}$, $c(x) = 0$ and $\text{logit}(\mu_{ij}) = \eta_{ij}$. For Poisson responses, we allow for an offset $\log E_{ij}$. If $y_{ij} \sim$ Poisson(μ_{ij}), then $b(x) = \exp(x)$, $c(x) = -\log(x!)$ and $\log \mu_{ij} = \log E_{ij} + \eta_{ij}$. For Bayesian inference, we specify prior distributions on the fixed effects β and random effects covariance matrix D. The dispersion parameter is one for responses from the Bernoulli and Poisson families so we do not consider a prior for φ. We assume a diffuse prior, $N(0, \Sigma_\beta)$, for β and an independent inverse Wishart prior, $IW(\nu, S)$, for D. Following the suggestion by Kass and Natarajan (2006), we set $\nu = r$ and let the scale matrix S be determined from first-stage data variability. In particular, $S = r\hat{R}$ where

$$\hat{R} = c\left(\frac{1}{n}\sum_{i=1}^{n} X_i^{R^T} M_i(\hat{\beta}) X_i^R\right)^{-1}, \tag{4.4}$$

$M_i(\hat{\beta})$ denotes the $n_i \times n_i$ diagonal generalized linear model weight matrix

with diagonal elements $[\varphi\, v(\hat{\mu}_{ij})\, g'(\hat{\mu}_{ij})^2]^{-1}$, $v(\cdot)$ is the variance function of $f(.)$ in (4.3) and $g(\cdot)$ is the link function. Here, $\hat{\mu}_{ij} = g^{-1}(X_{ij}^T\hat{\beta} + X_{ij}^{R^T}\hat{u}_i)$ where $\hat{u}_i$ is set as 0 for all i and $\hat{\beta}$ is an estimate of the regression coefficients from the generalized linear model obtained by pooling all data and setting $u_i = 0$ for all i. The value of c is an inflation factor representing the amount by which within-cluster variability should be increased in determining $\hat{R}$. We used $c = 1$ for all examples in Chapters 4 and 5.

4.3 Partially noncentered parametrizations for generalized linear mixed models

We introduce the following partially noncentered parametrization for the GLMM. The linear predictor is $\eta_i = X_i^R\beta^R + X_i^G\beta^G + X_i^R u_i$ for each $i = 1, \ldots, n$. Let

$$\begin{aligned} X_i^G\beta^G &= X_i^{G_1}\beta^{G_1} + X_i^{G_2}\beta^{G_2} \\ &= 1_{n_i}\, x_i^{G_1 T}\beta^{G_1} + X_i^{G_2}\beta^{G_2}, \end{aligned}$$

where β^{G_1} is a vector of length g_1 consisting of all parameters corresponding to subject specific covariates (that is, the rows of $X_i^{G_1}$ are all the same and equal to the vector $x_i^{G_1}$ say). Recall that the first column of X_i^R is 1_{n_i} if X_i^R is not a zero matrix. We have

$$\eta_i = X_i^R(C_i\beta^{RG_1} + u_i) + X_i^{G_2}\beta^{G_2}$$

$$\text{where} \quad C_i = \begin{bmatrix} I_r & x_i^{G_1 T} \\ & 0 \end{bmatrix} \quad \text{and} \quad \beta^{RG_1} = \begin{bmatrix} \beta^R \\ \beta^{G_1} \end{bmatrix}.$$

Let $\alpha_i = C_i\beta^{RG_1} + u_i$ and $\tilde{\alpha}_i = \alpha_i - W_i C_i \beta^{RG_1}$ where W_i is an $r \times r$ matrix to be specified. The proportion of $C_i\beta^{RG_1}$ subtracted from each α_i is allowed to vary with i as in Papaspiliopoulos *et al.* (2003) to reflect the varying informativity of each response y_i about the underlying α_i. $W_i = 0$ corresponds to the centered and $W_i = I$ to the noncentered parametrization. Finally,

$$\begin{aligned} \eta_i &= X_i^R(\tilde{\alpha}_i + W_i C_i \beta^{RG_1}) + X_i^{G_2}\beta^{G_2} \\ &= V_i\beta + X_i^R\tilde{\alpha}_i, \end{aligned} \tag{4.5}$$

where $V_i = [X_i^R W_i C_i \quad X_i^{G_2}]$. Let $\tilde{W}_i = \left[(I - W_i)C_i \quad 0_{r\times(p-r-g_1)}\right]$ for $i = 1, \ldots, n$. We then have $\tilde{\alpha}_i \sim N(\tilde{W}_i\beta,\, D)$. We refer to (4.5) as the partially

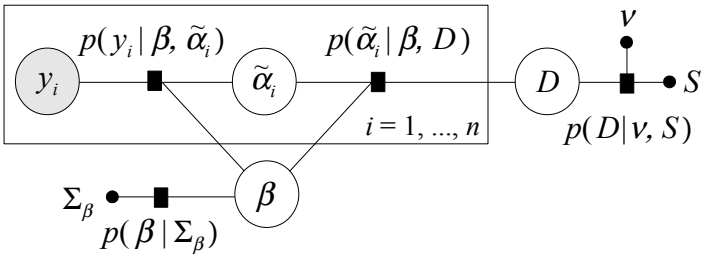

Figure 4.1: Factor graph for $p(y, \theta)$ in (4.6). Filled rectangles denote factors and circles denote variables (shaded for observed variables). Smaller filled circles denote constants or hyperparameters. The box represents a plate which contains variables and factors to be replicated. Number of repetitions is indicated in lower right corner.

noncentered parametrization. Let $\tilde{\alpha} = [\tilde{\alpha}_1^T, \ldots, \tilde{\alpha}_n^T]^T$ and $\theta = \{\beta, D, \tilde{\alpha}\}$ denote the set of unknown parameters in the GLMM. We have

$$p(y, \theta) = \left\{ \prod_{i=1}^{n} p(y_i|\beta, \tilde{\alpha}_i) p(\tilde{\alpha}_i|\beta, D) \right\} p(\beta|\Sigma_\beta) p(D|\nu, S). \tag{4.6}$$

Figure 4.1 shows the factor graph for $p(y, \theta)$ where there is a node (circle) for every variable, which is shaded in the case of observed variables and a node (filled rectangle) for each factor in the joint distribution. Constants or hyperparameters are denoted with smaller filled circles. Each factor node is connected by undirected links to all of the variable nodes on which that factor depends (see Bishop, 2006). Next, we consider specification of the tuning parameter W_i, referring to the linear mixed model in Section 4.1.1 which is a special case of the GLMM in (4.3) with an identity link.

4.3.1 Specification of tuning parameters

It is interesting to note that for the linear mixed model in (4.1), the expression for W_i leading to optimal performance in VB and the Gibbs sampling algorithm is exactly the same (see Papaspiliopoulos *et al.*, 2003). Gelfand *et al.* (1995) also noted the importance of W_i in assessing convergence properties of the centered parametrization. They showed that $|W_i| < 1$ for all i and $|W_i|$ is close to zero (centering is more efficient) when $|D|$ is large. On the other hand, $|W_i|$ is close to 1 (noncentering works better) when the error variance is large. Outside the Gaussian context, Papaspiliopoulos *et al.* (2003) considered partial noncentering for the spatial GLMM and specified the tuning parameters by using a quadratic expansion of the log-likelihood to obtain an indication of the information present in y_i. If we let

$\ell = \log p(y_i|\beta, \alpha_i)$ denote the log-likelihood and $\mathcal{I}_f = -\frac{\partial^2 \ell}{\partial \alpha_i \partial \alpha_i^T}$, then W_i in (4.2) can be expressed as

$$W_i = \left(\mathcal{I}_f + D^{-1}\right)^{-1} D^{-1}. \tag{4.7}$$

We use (4.7) to extend partially noncentered parametrizations to GLMMs and consider the specification of W_i for responses from the Bernoulli and Poisson families.

Recall that the linear predictor η_i can be expressed as $X_i^R \alpha_i + X_i^{G_2} \beta^{G_2}$. Let $E_i = [E_{i1}, \ldots, E_{in_i}]^T$. For Poisson responses with the log link function, we have

$$\begin{aligned} \ell &= y_i^T (\log E_i + \eta_i) - E_i^T \exp(\eta_i) - 1_{n_i}^T \log(y_i!) \\ \text{and } \mathcal{I}_f &= \sum_{j=1}^{n_i} E_{ij} \exp(\eta_{ij}) X_{ij}^R X_{ij}^{R^T} \approx \sum_{j=1}^{n_i} y_{ij} X_{ij}^R X_{ij}^{R^T} \end{aligned} \tag{4.8}$$

if we approximate the conditional mean μ_{ij} with the response. For Bernoulli responses with the logit link function, we have

$$\begin{aligned} \ell &= y_i^T \eta_i - 1_{n_i}^T \log\left\{1_{n_i} + \exp(\eta_i)\right\} \\ \text{and} \quad \mathcal{I}_f &= \sum_{j=1}^{n_i} \frac{\exp(\eta_{ij})}{\{1 + \exp(\eta_{ij})\}^2} X_{ij}^R X_{ij}^{R^T}. \end{aligned} \tag{4.9}$$

The specification of W_i depends on the random effects covariance D and for Bernoulli responses, on the linear predictor η_i as well. Later in Algorithm 8, we initialize W_i by considering $\eta_i = X_i \beta + X_i^R u_i$ and using estimates of D, β and u_i from penalized quasi-likelihood. Subsequently, we can either keep W_i as fixed or update them by replacing D with $\frac{S^q}{\nu^q - r - 1}$, assuming the variational posterior of D is $IW(\nu^q, S^q)$ and η_i with $V_i \mu_\beta^q + X_i^R \mu_{\tilde{\alpha}_i}^q$, where μ_β^q and $\mu_{\tilde{\alpha}_i}^q$ are the variational posterior means of β and $\tilde{\alpha}_i$ respectively. This can be done at the beginning of each cycle after new estimates of μ_β^q, $\mu_{\tilde{\alpha}_i}^q$, ν^q and S^q are obtained (see Algorithm 8, step 1).

4.4 Variational inference for generalized linear mixed models

In this section, we describe how the nonconjugate variational message passing algorithm (Knowles and Minka, 2011) can be used to fit GLMMs. In VB, the posterior distribution $p(\theta|y)$ is approximated by a $q(\theta)$ which is

assumed to be factorized as $\prod_{i=1}^{m} q_i(\theta_i)$ for some partition $\{\theta_1, \ldots, \theta_m\}$ of θ. For conjugate-exponential models, the optimal densities q_i will have the same form as the prior so that it suffices to update the parameters of q_i, such as in Algorithm 6. Variational message passing (Winn and Bishop, 2005) is an algorithm which allows VB to be applied to conjugate-exponential models without having to derive application-specific updates. In the case of GLMMs where the responses are from the Bernoulli or Poisson families, the factor $p(y_i|\beta, \tilde{\alpha}_i)$ of $p(y, \theta)$ in (4.6) is nonconjugate with respect to the prior distributions over β and $\tilde{\alpha}_i$ for each $i = 1, \ldots, n$. Therefore, if we apply VB and assume say $q(\theta) = q(\beta)q(D)\prod_{i=1}^{n} q(\tilde{\alpha}_i)$, the optimal densities for $q(\beta)$ and $q(\tilde{\alpha}_i)$ will not belong to recognizable density families.

In nonconjugate variational message passing, besides assuming that $q(\theta)$ must factorize into $\prod_{i=1}^{m} q_i(\theta_i)$ for some partition $\{\theta_1, \ldots, \theta_m\}$ of θ, we impose an additional restriction that each q_i must belong to some exponential family. In this way, we only have to find the parameters of each q_i that maximizes the variational lower bound $\mathcal{L}$ in (1.2). Suppose each q_i can be written in the form

$$q_i(\theta_i) = \exp\{\lambda_i^T t_i(\theta_i) - h_i(\lambda_i)\},$$

where λ_i is the vector of natural parameters and $t_i(\cdot)$ are the sufficient statistics. We wish to maximize $\mathcal{L}$ with respect to the variational parameters $\lambda_1, \ldots, \lambda_m$, which are also natural parameters of $q_1(\theta_1), \ldots, q_m(\theta_m)$ respectively. In the following, we show that nonconjugate variational message passing can be interpreted as fixed-point iterations where updates are obtained from the condition that the gradient of $\mathcal{L}$ with respect to each λ_i is zero when $\mathcal{L}$ is maximized.

From (1.2), the gradient of $\mathcal{L}$ with respect to λ_i is

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = \frac{\partial}{\partial \lambda_i} E_q\{\log p(y, \theta)\} - \frac{\partial}{\partial \lambda_i} E_q\{\log q(\theta)\}. \tag{4.10}$$

Consider the first term in (4.10). Suppose $p(y, \theta) = \prod_a f_a(y, \theta)$. We have $E_q\{\log p(y, \theta)\} = \sum_a S_a$ where $S_a = E_q\{\log f_a(y, \theta)\}$. Note that each S_a is a function of the natural parameters $\lambda_1, \ldots, \lambda_m$. Since we have assumed that θ_i is independent of all θ_j where $j \neq i$ in the variational approximation q, the only terms in $\sum_a S_a$ which depend on λ_i are the factors f_a connected

to θ_i in the factor graph of $p(y, \theta)$. Therefore,

$$\frac{\partial}{\partial \lambda_i} E_q\{\log p(y, \theta)\} = \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \lambda_i}, \tag{4.11}$$

where the summation is over all factors in $N(\theta_i)$, the neighbourhood of θ_i in the factor graph. For the second term in (4.10), we have $E_q\{\log q(\theta)\} = \sum_{l=1}^{m} E_q\{\log q_l(\theta_l)\}$ where the only term in the sum that depends on λ_i is the ith term. Hence,

$$\begin{aligned} \frac{\partial}{\partial \lambda_i} E_q\{\log q(\theta)\} &= \frac{\partial}{\partial \lambda_i} \left\{ \lambda_i^T \frac{\partial h_i(\lambda_i)}{\partial \lambda_i} - h_i(\lambda_i) \right\} \\ &= \mathcal{V}_i(\lambda_i) \lambda_i. \end{aligned} \tag{4.12}$$

Here, we have used the fact that $E_q\{t_i(\theta_i)\} = \frac{\partial h_i(\lambda_i)}{\partial \lambda_i}$ and $\mathcal{V}_i(\lambda_i) = \frac{\partial^2 h_i(\lambda_i)}{\partial \lambda_i \partial \lambda_i^T}$ denotes the variance-covariance matrix of $t(\theta_i)$. Note that $\mathcal{V}_i(\lambda_i)$ is symmetric positive semi-definite. Putting (4.11) and (4.12) together, the gradient of the lower bound is

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \lambda_i} - \mathcal{V}_i(\lambda_i) \lambda_i \tag{4.13}$$

and is zero when $\lambda_i = \mathcal{V}_i(\lambda_i)^{-1} \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \lambda_i}$, provided $\mathcal{V}_i(\lambda_i)$ is invertible. This condition is used as a fixed-point iteration to obtain updates to λ_i in nonconjugate variational message passing (Algorithm 7).

Algorithm 7: Nonconjugate variational message passing

Initialize λ_i for $i = 1, \ldots, m$.
Cycle:
For $i = 1, \ldots, m$,

$$\lambda_i \leftarrow \mathcal{V}_i(\lambda_i)^{-1} \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \lambda_i} \tag{4.14}$$

until convergence.

The update in (4.14) can be simplified when the factor f_a is conjugate to $q_i(\theta_i)$, that is, f_a has the same functional form as $q_i(\theta_i)$ with respect to θ_i. Let $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_m)$. Suppose

$$f_a(y, \theta) = \exp\{g_a(y, \theta_{-i})^T t_i(\theta_i) - h_a(y, \theta_{-i})\}.$$

Then $\frac{\partial S_a}{\partial \lambda_i} = \mathcal{V}_i(\lambda_i) E_q\{g_a(y, \theta_{-i})\}$, where $E_q\{g_a(y, \theta_{-i})\}$ does not depend on

λ_i. When every factor in the neighbourhood of θ_i is conjugate to $q_i(\theta_i)$,

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = \mathcal{V}_i(\lambda_i)\left[\sum\nolimits_{a \in N(\theta_i)} E_q\{g_a(y, \theta_{-i})\} - \lambda_i\right] \tag{4.15}$$

and (4.14) reduces to

$$\lambda_i \leftarrow \sum_{a \in N(\theta_i)} E_q\{g_a(y, \theta_{-i})\}. \tag{4.16}$$

These are the updates in variational message passing. Nonconjugate variational message passing thus reduces to variational message passing for conjugate factors (see also Knowles and Minka, 2011). Unlike variational message passing however, the Kullback-Leibler divergence is not guaranteed to decrease at each step and sometimes convergence problems may be encountered. Knowles and Minka (2011) suggested using damping to fix convergence problems. We did not encounter any convergence issues for the examples in Section 4.6. Moreover, whenever Algorithm 7 converges, it will be to a local maximum of the lower bound as the algorithm becomes highly unstable near any local minimum (Knowles and Minka, 2011).

4.4.1 Updates for multivariate Gaussian distribution

While the updates in Algorithm 7 are in terms of the natural parameters λ_i, it might be more convenient to express $\frac{\partial S_a}{\partial \lambda_i}$ in terms of the mean and covariance of q_i when q_i is Gaussian. Knowles and Minka (2011) have considered the univariate case and Wand (2013) derived fully simplified updates for the multivariate case. Here, we give only a brief outline of the derivation of the multivariate Gaussian updates. Magnus and Neudecker (1988) is a good reference for the matrix differential calculus techniques involved in the derivation.

Suppose $q_i(\theta_i) = N(\mu^q_{\theta_i}, \Sigma^q_{\theta_i})$ where θ_i is a vector of length d. We can write $q_i(\theta_i)$ as

$$\exp\left\{\lambda_i^T \begin{bmatrix} \text{vech}(\theta_i \theta_i^T) \\ \theta_i \end{bmatrix} - h_i(\lambda_i)\right\} \quad \text{where} \quad \lambda_i = \begin{bmatrix} -\frac{1}{2} D_d^T \text{vec}({\Sigma^q_{\theta_i}}^{-1}) \\ {\Sigma^q_{\theta_i}}^{-1} \mu^q_{\theta_i} \end{bmatrix}$$

and $h_i(\lambda_i) = \frac{1}{2} {\mu^q_{\theta_i}}^T {\Sigma^q_{\theta_i}}^{-1} \mu^q_{\theta_i} + \frac{1}{2} \log|\Sigma^q_{\theta_i}| + \frac{d}{2} \log(2\pi)$. The matrix D_d is a unique $d^2 \times \frac{d}{2}(d+1)$ matrix that transforms $\text{vech}(A)$ into $\text{vec}(A)$ for any $d \times d$ symmetric square matrix A, that is, $D_d \text{vech}(A) = \text{vec}(A)$. Let D_d^+ denote the Moore-Penrose inverse of D_d. If we let $\lambda_{i1} = -\frac{1}{2} D_d^T \text{vec}({\Sigma^q_{\theta_i}}^{-1})$

and $\lambda_{i2} = {\Sigma^q_{\theta_i}}^{-1}\mu^q_{\theta_i}$, $\frac{\partial S_a}{\partial \lambda_i}$ can be expressed as

$$\begin{bmatrix} \frac{\partial S_a}{\partial \lambda_{i1}} \\ \frac{\partial S_a}{\partial \lambda_{i2}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathrm{vec}(\Sigma^q_{\theta_i})}{\partial \lambda_{i1}} & \frac{\partial \mu^q_{\theta_i}}{\partial \lambda_{i1}} \\ \frac{\partial \mathrm{vec}(\Sigma^q_{\theta_i})}{\partial \lambda_{i2}} & \frac{\partial \mu^q_{\theta_i}}{\partial \lambda_{i2}} \end{bmatrix} \begin{bmatrix} \frac{\partial S_a}{\partial \mathrm{vec}(\Sigma^q_{\theta_i})} \\ \frac{\partial S_a}{\partial \mu^q_{\theta_i}} \end{bmatrix} = U(\lambda_i) \begin{bmatrix} \frac{\partial S_a}{\partial \mathrm{vec}(\Sigma^q_{\theta_i})} \\ \frac{\partial S_a}{\partial \mu^q_{\theta_i}} \end{bmatrix},$$

where

$$U(\lambda_i) = \begin{bmatrix} 2D_d^+(\Sigma^q_{\theta_i} \otimes \Sigma^q_{\theta_i}) & 2D_d^+(\mu^q_{\theta_i} \otimes \Sigma^q_{\theta_i}) \\ 0 & \Sigma^q_{\theta_i} \end{bmatrix}.$$

Moreover, $\mathcal{V}_i(\lambda_i) = \frac{\partial^2 h_i(\lambda_i)}{\partial \lambda_i \partial \lambda_i^T}$ can be derived to be

$$\begin{bmatrix} 2D_d^+(\mu^q_{\theta_i}{\mu^q_{\theta_i}}^T \otimes \Sigma^q_{\theta_i} + \Sigma^q_{\theta_i} \otimes \mu^q_{\theta_i}{\mu^q_{\theta_i}}^T + \Sigma^q_{\theta_i} \otimes \Sigma^q_{\theta_i}){D_d^+}^T & 2D_d^+(\mu^q_{\theta_i} \otimes \Sigma^q_{\theta_i}) \\ \{2D_d^+(\mu^q_{\theta_i} \otimes \Sigma^q_{\theta_i})\}^T & \Sigma^q_{\theta_i} \end{bmatrix}.$$

From (4.14), we have

$$\lambda_i \leftarrow \mathcal{V}_i(\lambda_i)^{-1}U(\lambda_i) \sum_{a \in N(\theta_i)} \begin{bmatrix} \frac{\partial S_a}{\partial \mathrm{vec}(\Sigma^q_{\theta_i})} \\ \frac{\partial S_a}{\partial \mu^q_{\theta_i}} \end{bmatrix}$$

$$\text{where} \quad \mathcal{V}_i(\lambda_i)^{-1}U(\lambda_i) = \begin{bmatrix} D_d^T & 0 \\ -2({\mu^q_{\theta_i}}^T \otimes I){D_d^+}^T D_d^T & I \end{bmatrix}.$$

Wand (2013) showed that the updates simplify to

$$\Sigma^q_{\theta_i} \leftarrow -\frac{1}{2}\left[\mathrm{vec}^{-1}\left(\sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \mathrm{vec}(\Sigma^q_{\theta_i})}\right)\right]^{-1} \quad \text{and}$$

$$\mu^q_{\theta_i} \leftarrow \mu^q_{\theta_i} + \Sigma^q_{\theta_i} \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \mu^q_{\theta_i}}. \tag{4.17}$$

4.4.2 Nonconjugate variational message passing for generalized linear mixed models

We consider a variational approximation for the GLMM of the form

$$q(\theta) = q(\beta)q(D)\prod_{i=1}^{n} q(\tilde{\alpha}_i), \tag{4.18}$$

where $q(\beta)$ is $N\left(\mu^q_\beta, \Sigma^q_\beta\right)$, $q(D)$ is $IW(\nu^q, S^q)$, and $q(\tilde{\alpha}_i)$ is $N\left(\mu^q_{\tilde{\alpha}_i}, \Sigma^q_{\tilde{\alpha}_i}\right)$, all belonging to the exponential family. Here, we approximate the posterior distributions of β and $\tilde{\alpha}_i$ by Gaussian distributions which are often reasonable and supported by the asymptotic normality of the posterior. Our results

also indicate that Gaussian approximation performs reasonably well as an approximation to the posterior in finite samples. See Gelman *et al.* (2004) for further discussion as well as counterexamples. The posterior distribution for D is approximated by an inverse Wishart which can be shown to be the optimal density under only the VB assumption $q(\theta) = q(\beta)q(D)q(\tilde{\alpha})$. The nonconjugate variational message passing algorithm for GLMMs is outlined in Algorithm 8. For responses from the Poisson family,

$$F_{ij} = E_{ij}\kappa_{ij} \quad \text{and} \quad G_i = E_i \odot \kappa_i$$

for $i = 1, \ldots, n$, $j = 1, \ldots, n_i$, where κ_{ij} is the jth element of $\kappa_i = \exp\{V_i\mu_\beta^q + X_i^R\mu_{\tilde{\alpha}_i}^q + \frac{1}{2}\text{diag}(V_i\Sigma_\beta^q V_i^T + X_i^R\Sigma_{\tilde{\alpha}_i}^q X_i^{RT})\}$. For Bernoulli responses,

$$F_{ij} = B^{(2)}(\mu_{ij}^q, \sigma_{ij}^q) \quad \text{and} \quad G_i = B^{(1)}(\mu_i^q, \sigma_i^q)$$

for $i = 1, \ldots, n$, $j = 1, \ldots, n_i$, where μ_{ij}^q is the jth element of $\mu_i^q = V_i\mu_\beta^q + X_i^R\mu_{\tilde{\alpha}_i}^q$, σ_{ij}^q is the jth element of $\sigma_i^q = \sqrt{\text{diag}(V_i\Sigma_\beta^q V_i^T + X_i^R\Sigma_{\tilde{\alpha}_i}^q X_i^{RT})}$ and

$$B^{(r)}(\mu, \sigma) = \int_{-\infty}^{\infty} b^{(r)}(\sigma x + \mu)\tfrac{1}{\sqrt{2\pi}}\exp(-x^2)\, dx,$$

where $b(x) = \log\{1 + \exp(x)\}$ and $b^{(r)}(x)$ denotes the rth derivative of $b(\cdot)$ with respect to x. If μ and σ are vectors, say $\mu = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ and $\sigma = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$, then $B^{(r)}(\mu, \sigma) = \begin{bmatrix} B^{(r)}(1,4) \\ B^{(r)}(2,5) \\ B^{(r)}(3,6) \end{bmatrix}$.

Algorithm 8: Nonconjugate variational message passing for GLMMs

Initialize μ_β^q, Σ_β^q, S^q and $\mu_{\tilde{\alpha}_i}^q$, $\Sigma_{\tilde{\alpha}_i}^q$, W_i for $i = 1, \ldots, n$. Set $\nu^q = n + \nu$. Cycle:

1. Update W_i and hence V_i for $i = 1, \ldots, n$. (Optional)
2. - $\Sigma_\beta^q \leftarrow \left(\Sigma_\beta^{-1} + \nu^q \sum_{i=1}^n \tilde{W}_i^T S^{q-1}\tilde{W}_i + \sum_{i=1}^n \sum_{j=1}^{n_i} F_{ij}V_{ij}V_{ij}^T\right)^{-1}$,
   - $\mu_\beta^q \leftarrow \mu_\beta^q + \Sigma_\beta^q \Big\{ -\Sigma_\beta^{-1}\mu_\beta^q + \nu^q \sum_{i=1}^n \tilde{W}_i^T S^{q-1}(\mu_{\tilde{\alpha}_i}^q - \tilde{W}_i\mu_\beta^q) + \sum_{i=1}^n V_i^T(y_i - G_i)\Big\}$.
3. For $i = 1, \ldots, n$,
   - $\Sigma_{\tilde{\alpha}_i}^q \leftarrow \left(\nu^q S^{q-1} + \sum_{j=1}^{n_i} F_{ij}X_{ij}^R X_{ij}^{RT}\right)^{-1}$,

- $\mu^q_{\tilde{\alpha}_i} \leftarrow \mu^q_{\tilde{\alpha}_i} + \Sigma^q_{\tilde{\alpha}_i}\big\{ -\nu^q S^{q-1}(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta) + X_i^{R^T}(y_i - G_i)\big\}$.

4. $S^q \leftarrow S + \sum_{i=1}^n \big\{(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta)(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta)^T + \Sigma^q_{\tilde{\alpha}_i} + \tilde{W}_i\Sigma^q_\beta\tilde{W}_i^T\big\}$.

until the absolute relative change in the lower bound $\mathcal{L}$ is negligible.

The updates in Algorithm 8 can be obtained from the formulae in (4.16) and (4.17). Consider the parameters ν^q and S^q of $q(D)$. The factors connected to D are $p(D|\nu, S)$ and $p(\tilde{\alpha}_i|\beta, D)$, $i = 1, \ldots, n$, which are all conjugate factors. Therefore, updates for $q(D)$ can be obtained from (4.16) or by setting $q(D) \propto \exp\{E_{-D} \log p(y, \theta)\}$ as in VB. The shape parameter ν^q can be shown to be deterministic: $\nu^q = n + \nu$ and the update for S^q is given in step 4 of Algorithm 8. The updates of the parameters of $q(\beta)$ and $q(\tilde{\alpha}_i)$, $i = 1, \ldots, n$, have to be computed using (4.17) as $p(y_i|\beta, \tilde{\alpha}_i)$ is connected to β and $\tilde{\alpha}_i$ is a nonconjugate factor. The factors connected to β are $p(\beta|\Sigma_\beta)$, $p(\tilde{\alpha}_i|\beta, D)$ and $p(y_i|\beta, \tilde{\alpha}_i)$ for $i = 1, \ldots, n$ (see Figure 4.1). Let $S_\beta = E_q\{\log p(\beta|\Sigma_\beta)\}$, $S_{\tilde{\alpha}_i} = E_q\{\log p(\tilde{\alpha}_i|\beta, D)\}$ and $S_{y_i} = E_q\{\log p(y_i|\beta, \tilde{\alpha}_i)\}$ for $i = 1, \ldots, n$, where E_q denotes expectation with respect to q. We have

$$\sum_{a \in N(\beta)} \frac{\partial S_a}{\partial \mathrm{vec}(\Sigma^q_\beta)} = \frac{\partial S_\beta}{\partial \mathrm{vec}(\Sigma^q_\beta)} + \sum_{i=1}^n \frac{\partial S_{\tilde{\alpha}_i}}{\partial \mathrm{vec}(\Sigma^q_\beta)} + \sum_{i=1}^n \frac{\partial S_{y_i}}{\partial \mathrm{vec}(\Sigma^q_\beta)},$$

$$\sum_{a \in N(\beta)} \frac{\partial S_a}{\partial \mu^q_\beta} = \frac{\partial S_\beta}{\partial \mu^q_\beta} + \sum_{i=1}^n \frac{\partial S_{\tilde{\alpha}_i}}{\partial \mu^q_\beta} + \sum_{i=1}^n \frac{\partial S_{y_i}}{\partial \mu^q_\beta},$$

and the simplified updates for Σ^q_β and μ^q_β are given in step 2 of Algorithm 8. The factors connected to $\tilde{\alpha}_i$ are $p(\tilde{\alpha}_i|\beta, D)$ and $p(y_i|\beta, \tilde{\alpha}_i)$ for each $i = 1, \ldots, n$ (see Figure 4.1). Hence

$$\sum_{a \in N(\tilde{\alpha}_i)} \frac{\partial S_a}{\partial \mathrm{vec}(\Sigma^q_{\tilde{\alpha}_i})} = \frac{\partial S_{\tilde{\alpha}_i}}{\partial \mathrm{vec}(\Sigma^q_{\tilde{\alpha}_i})} + \frac{\partial S_{y_i}}{\partial \mathrm{vec}(\Sigma^q_{\tilde{\alpha}_i})} \quad \text{and}$$

$$\sum_{a \in N(\tilde{\alpha}_i)} \frac{\partial S_a}{\partial \mu^q_{\tilde{\alpha}_i}} = \frac{\partial S_{\tilde{\alpha}_i}}{\partial \mu^q_{\tilde{\alpha}_i}} + \frac{\partial S_{y_i}}{\partial \mu^q_{\tilde{\alpha}_i}}.$$

The simplified updates for $\Sigma^q_{\tilde{\alpha}_i}$ and $\mu^q_{\tilde{\alpha}_i}$ are given in step 3 of Algorithm 8. See Appendix C for the evaluation of S_β, $S_{\tilde{\alpha}_i}$ and S_{y_i}. All gradients can be computed using vector differential calculus (see Magnus and Neudecker, 1988).

For responses from the Poisson family, S_{y_i} can be evaluated in closed form. However, S_{y_i} cannot be evaluated analytically for Bernoulli responses. Knowles and Minka (2011) discussed several alternatives in handling this

integral. One could construct a bound on $\log(1+e^x)$ such as the "quadratic" bound (Jaakkola and Jordan, 2000) or the "tilted" bound (Saul and Jordan, 1998). We observed a negative bias in the estimates for the random effects variances when using the "tilted bound" in Algorithm 8. This negative bias decreases as the cluster size increases (see Rijmen and Vomlel, 2008). Hence, we use quadrature to compute the expectation and gradients. Following Ormerod and Wand (2012), we reduce all high-dimensional integrals to univariate ones and evaluate these efficiently using adaptive Gauss-Hermite quadrature (Liu and Pierce, 1994). The details are given in Appendix D.

While the updates in Algorithm 8 can be simplified if $W_i = I$ (noncentered) or 0 (centered) and are more complex in the partially noncentered case, the reduction in efficiency is minimal. Moreover, with a good initialization, it is feasible to keep W_i fixed throughout the course of running Algorithm 8 so that no additional computation time is used in updating W_i. We use the fit from penalized quasi-likelihood implemented via the function `glmmPQL()` in the `R` package `MASS` (Venables and Ripley, 2002) to initialize Algorithm 8. In our experiments, the lower bound computed at the end of each cycle of updates is usually on an increasing trend although there might be some instability at the beginning. In cases where the algorithm does not converge, we found that changing the initialization can help to alleviate the situation. Although the lower bound is not guaranteed to increase at the end of each cycle, we continue to use it as a means of monitoring convergence and Algorithm 8 is terminated when the absolute relative change in the lower bound is less than 10^{-6}. The lower bounds for the logistic and Poisson GLMMs are presented in Appendix C.

4.5 Model selection

At the point of convergence of Algorithm 8, the lower bound on the log marginal likelihood, $\log p(y)$, is maximized. This variational lower bound is often tight and can be useful for model selection. In Section 4.6.5, we demonstrate how the variational lower bound, a by-product of Algorithm 8, can be used in place of the log marginal likelihood to obtain approximate posterior model probabilities, assuming all models considered are equally probable. See Section 1.1.3 for a brief discussion on the role of marginal likelihood in Bayesian model selection.

We note that standard model selection criteria such as AIC or BIC are difficult to apply to GLMMs as it is not straightforward to determine the degrees of freedom of a GLMM. Yu and Yau (2012) developed a condi-

tional Akaike information criterion for GLMMs which takes into account estimation uncertainty in variance component parameters. Overstall and Forster (2010) considered a default strategy for Bayesian model selection addressing issues of prior specification and computation. See also Cai and Dunson (2008) for a review of variable selection methods for GLMMs.

4.6 Examples

We investigate the performance of Algorithm 8 using different parametrizations by considering a simulation study and some real data sets. When using partial noncentering, we can either initialize the tuning parameters, W_i for $i = 1, \ldots, n$, and keep them fixed or update them at the beginning of each cycle (see Algorithm 8, step 1). Such updates are particularly useful when a good initialization is lacking. We present results for both cases. There may not be significant improvement in updating W_i in the examples below as the initialization using penalized quasi-likelihood is already good.

We assess the performance of Algorithm 8 using different parametrizations by using MCMC as a "gold standard". Fitting via MCMC was performed in WinBUGS (Lunn *et al.*, 2000) through `R` by using `R2WinBUGS` (Sturtz *et al.*, 2005) as an interface. WinBUGS automatically implements a Markov chain simulation for the posterior distribution after the user specifies a model and starting values (see, e.g. Gelman *et al.*, 2004). We used the centered parametrization when specifying the model in WinBUGS as this produced better mixing than the noncentered parametrization for most of the examples considered (see also Brown and Zhou, 2010). The MCMC algorithm was initialized similarly using the fit from penalized quasi-likelihood. In each case, three chains were run simultaneously to assess convergence, each with 50000 iterations, and the first 5000 iterations were discarded in each chain as burn-in. A thinning factor of 10 was applied to reduce dependence between draws. The posterior means and standard deviations reported were based on the remaining 13500 iterations. The computation times reported for MCMC are the times taken for updating in WinBUGS. We used the same priors for MCMC and Algorithm 8. For the fixed effects, we used a $N(0, 1000I)$ prior. All code was written in the R language and run on a dual processor Windows PC 3.30 GHz workstation.

4.6.1 Simulated data

In this simulation study, we consider the Poisson random intercept model

$$y_{ij}|u_i \sim \text{Poisson}\left(\exp(\beta_0 + \beta_1 x_{ij} + u_i)\right)$$

and the logistic random intercept model

$$y_{ij}|u_i \sim \text{Bernoulli}\left(\frac{\exp(\beta_0 + \beta_1 x_{ij} + u_i)}{1 + \exp(\beta_0 + \beta_1 x_{ij} + u_i)}\right),$$

where $u_i \sim N(0, \sigma^2)$. For the Poisson random intercept model, we set $x_{ij} = j - 1$ for $i = 1, \ldots, 100$, $j = 1, 2$, and used $\beta_0 = \beta_1 = -0.5$, $\sigma = 0.1$. For the logistic random intercept model, we set $x_{ij} = \frac{j}{8}$ for $i = 1, \ldots, 50$, $j = 1, \ldots, 8$, and used $\beta_0 = 0$, $\beta_1 = 5$, $\sigma = \sqrt{1.5}$. Similar settings have been considered by Ormerod and Wand (2012). For each model, 100 data sets were generated. No convergence issues were encountered for these simulated data but experience with other simulated data sets (not shown) indicate that problems may arise when the covariance matrix of the fixed effects estimated from penalized quasi-likelihood is nearly singular or when the standard deviation of the random effects are very close to zero. In such cases, we can use alternative means of initialization such as estimates from the generalized linear model obtained by setting the random effects as zero. The expression in (4.4) can also serve as a prior guess for D (see Kass and Natarajan, 2006). Table 4.1 reports the estimates from penalized quasi-likelihood and the posterior means and standard deviations estimated by Algorithm 8 (using different parametrizations) and MCMC. Results are averaged over the 100 sets of simulated data. We have also included root mean squared errors computed as $\sqrt{\frac{1}{100}\sum_{l=1}^{100}(\hat{\vartheta}_l - \vartheta_l^0)^2}$ for an estimate $\hat{\vartheta}_l$ from the lth simulated data set obtained from penalized quasi-likelihood or Algorithm 8, where ϑ_l^0 is the corresponding estimate from MCMC regarded as the "gold standard".

For the Poisson model, the posterior means of the fixed effects and random effects estimated using the centered and noncentered parametrizations are quite close and also close to that of MCMC. However, the posterior standard deviations of the fixed effects are underestimated in the centered parametrization and the noncentered parametrization does better. The average time to convergence was shorter with noncentering and a higher lower bound was attained on average. We observed that the partially noncentered parametrization where tuning parameters were not updated took on aver-

	PQL	NCP	CP	PNCP: W_i fixed	PNCP: W_i updated	MCMC
Poisson						
β_0	−0.54	−0.63	−0.63	−0.63	−0.63	−0.64
	(0.11)	(0.01)	(0.01)	(0.01)	(0.01)	
sd(β_0)	0.13	0.13	0.05	0.13	0.13	0.15
	(0.02)	(0.02)	(0.10)	(0.02)	(0.02)	
β_1	−0.48	−0.49	−0.50	−0.49	−0.49	−0.48
	(0.01)	(<.005)	(0.01)	(<.005)	(<.005)	
sd(β_1)	0.19	0.21	0.16	0.20	0.19	0.21
	(0.03)	(<.005)	(0.05)	(0.01)	(0.02)	
σ	0.27	0.48	0.50	0.49	0.49	0.50
	(0.35)	(0.02)	(0.01)	(0.01)	(0.01)	
sd(σ)	—	0.03	0.04	0.03	0.03	0.11
	—	(0.08)	(0.07)	(0.08)	(0.08)	
Time	0.1	3.6	4.3	3.5	4.0	60.1
$\mathcal{L}$	—	−196.0	−197.0	−196.0	−196.0	—
Logistic						
β_0	−0.10	−0.07	−0.07	−0.07	−0.07	−0.05
	(0.06)	(0.02)	(0.02)	(0.02)	(0.02)	
sd(β_0)	0.32	0.33	0.17	0.30	0.30	0.38
	(0.07)	(0.06)	(0.21)	(0.09)	(0.08)	
β_1	5.02	5.20	5.24	5.23	5.21	5.23
	(0.27)	(0.04)	(0.02)	(0.02)	(0.04)	
sd(β_1)	0.63	0.77	0.41	0.50	0.50	0.85
	(0.24)	(0.09)	(0.45)	(0.37)	(0.36)	
σ	1.25	1.18	1.24	1.22	1.22	1.24
	(0.16)	(0.06)	(0.03)	(0.03)	(0.04)	
sd(σ)	—	0.12	0.13	0.12	0.12	0.32
	—	(0.20)	(0.20)	(0.20)	(0.20)	
Time	0.2	3.2	3.1	2.9	3.9	146.6
$\mathcal{L}$	—	−140.4	−141.1	−140.5	−140.5	—

Table 4.1: Results of simulation study showing initialization values from penalized quasi-likelihood (PQL), posterior means and standard deviations (sd) estimated by Algorithm 8 (using the noncentered (NCP), centered (CP) and partially noncentered (PNCP) parametrizations) and MCMC, computation times (seconds) and variational lower bounds ($\mathcal{L}$), averaged over 100 sets of simulated data. Values in () are the corresponding root mean squared errors.

age the least time to converge and produced a fit closer to that of the noncentered parametrization but with improvements in the estimation of the posterior means of the random effects. When the tuning parameters were updated, the fit was just as good although computation time was longer. For the logistic model, centering and noncentering have different merits. While centering produced better estimates of the posterior means, the posterior standard deviations of the fixed effects were underestimated. The partially noncentered parametrization tries to adapt between the centered and noncentered parametrizations, producing better estimates of the posterior means than noncentering and better estimates of the posterior standard deviations than centering. When the tuning parameters were updated, the results leaned more towards the noncentered parametrization and the algorithm took longer to converge. In both cases, Algorithm 8 using the partially noncentered parametrization was faster than MCMC and provided better estimates of the fixed effects and random effects than penalized quasi-likelihood. There are some difficulties, however, in comparing Algorithm 8 and MCMC in this way as the time taken for Algorithm 8 to converge depends on the initialization, stopping rule and the rate of convergence also depends on the problem. Similarly, the updating time taken for MCMC is also problem-dependent and depends on the length of burn-in and number of sampling iterations. In addition, we observed (in simulated data sets not shown) that posterior inferences can be sensitive to prior assumptions on the variance components in Poisson models where many of the counts are close to zero or in binary data where the cluster size is small (see Browne and Draper, 2006; Roos and Held, 2011).

4.6.2 Epilepsy data

Here we consider the epilepsy data of Thall and Vail (1990) which has been analyzed by many authors (e.g. Breslow *et al.*, 1993; Ormerod and Wand, 2012). In this clinical trial, 59 epileptics were randomized to a new antiepileptic drug, progabide, (Trt=1) or a placebo (Trt=0). Before receiving treatment, baseline data on the number of epileptic seizures during the preceding 8-week period were recorded. The logarithm of $\frac{1}{4}$ the number of baseline seizures (Base) and the logarithm of age (Age) were treated as covariates. Counts of epileptic seizures during the two weeks before each of four successive clinic visits (Visit, coded as $\text{Visit}_1 = -0.3$, $\text{Visit}_2 = -0.1$, $\text{Visit}_3 = 0.1$ and $\text{Visit}_4 = 0.3$) were recorded. A binary variable (V4=1 for fourth visit, 0 otherwise) was also considered as a covariate.

We consider models II and IV from Breslow *et al.* (1993). Model II is a Poisson random intercept model where

$$\log \mu_{ij} = \beta_0 + \beta_{\text{Base}}\text{Base}_i + \beta_{\text{Trt}}\text{Trt}_i + \beta_{\text{Base}\times\text{Trt}}\text{Base}_i \times \text{Trt}_i + \beta_{\text{Age}}\text{Age}_i + \beta_{\text{V4}}\text{V4}_{ij} + u_i,$$

for $i = 1, \ldots, n$, $j = 1, \ldots, 4$ and $u_i \sim N(0, \sigma^2)$. Model IV is a Poisson random intercept and slope model of the form

$$\log \mu_{ij} = \beta_0 + \beta_{\text{Base}}\text{Base}_i + \beta_{\text{Trt}}\text{Trt}_i + \beta_{\text{Base}\times\text{Trt}}\text{Base}_i \times \text{Trt}_i + \beta_{\text{Age}}\text{Age}_i + \beta_{\text{Visit}}\text{Visit}_{ij} + u_{1i} + u_{2i}\text{Visit}_{ij},$$

for $i = 1, \ldots, n$, $j = 1, \ldots, 4$ and $\left[\begin{smallmatrix} u_{1i} \\ u_{2i} \end{smallmatrix}\right] \sim N\left(0, \left[\begin{smallmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{21} & \sigma_{22}^2 \end{smallmatrix}\right]\right)$. As the MCMC chains for intercept and Age were mixing poorly, we decided to center the covariate Age. In the analysis that follows, we assume Age_i has been replaced by $\text{Age}_i - \text{mean(Age)}$.

Table 4.2 shows the estimates of the posterior means and standard deviations of the fits from MCMC and Algorithm 8 (using different parametrizations), initialization values from penalized quasi-likelihood and computation times in seconds taken by different methods. All the variational methods are faster than MCMC by an order of magnitude which is especially important in large scale applications. In the noncentered parametrization, the standard deviations of the fixed effects were underestimated and the centered parametrization does better in this aspect. The partially noncentered parametrization produced a fit that is closer to that of the centered parametrization and has improved upon it. In both models, the fits produced by partial noncentering are very close to that produced by MCMC and are superior to that of the centered and noncentered parametrizations. The lower bound attained by partial noncentering is also higher than that of centering and noncentering, giving a tighter bound on the log marginal likelihood. It is important to emphasize that the relevant comparison is of the partially noncentered parametrization to the worst of the centered and noncentered parametrizations, since in general we do not know if centering or noncentering is better without running both algorithms. Partial noncentering on the other hand, automatically chooses a near optimal parametrization. Updating of the tuning parameters helped to improve the fit produced by partial noncentering. Figure 4.2 shows the marginal posterior distributions for parameters in models II and IV estimated by MCMC

	PQL	NCP	CP	PNCP: W_i		MCMC
				fixed	updated	
Model II						
β_0	0.31	0.26	0.27	0.27	0.27	0.26
	0.26	0.11	0.24	0.26	0.27	0.27
β_{Base}	0.88	0.89	0.88	0.88	0.88	0.89
	0.13	0.04	0.13	0.13	0.14	0.14
β_{Trt}	−0.91	−0.94	−0.94	−0.94	−0.94	−0.94
	0.41	0.15	0.36	0.40	0.41	0.42
$\beta_{\text{Base}\times\text{Trt}}$	0.34	0.34	0.34	0.34	0.34	0.34
	0.20	0.06	0.19	0.21	0.21	0.21
β_{Age}	0.54	0.50	0.48	0.48	0.48	0.48
	0.35	0.12	0.33	0.35	0.36	0.37
β_{V4}	−0.16	−0.16	−0.16	−0.16	−0.16	−0.16
	0.08	0.05	0.05	0.05	0.05	0.05
σ	0.44	0.50	0.54	0.53	0.53	0.53
	—	0.05	0.05	0.05	0.05	0.06
$\mathcal{L}$	—	−707.3	−702.0	−701.6	−701.5	—
Time	0.2	1.1	0.4	0.4	0.6	61
Model IV						
β_0	0.27	0.21	0.21	0.21	0.21	0.21
	0.26	0.10	0.24	0.26	0.26	0.27
β_{Base}	0.88	0.89	0.88	0.89	0.89	0.88
	0.13	0.04	0.13	0.13	0.13	0.14
β_{Trt}	−0.92	−0.94	−0.93	−0.93	−0.93	−0.94
	0.41	0.15	0.36	0.40	0.40	0.42
$\beta_{\text{Base}\times\text{Trt}}$	0.35	0.34	0.34	0.34	0.34	0.34
	0.20	0.06	0.19	0.20	0.21	0.22
β_{Age}	0.54	0.49	0.47	0.47	0.47	0.47
	0.35	0.12	0.32	0.35	0.35	0.37
β_{Visit}	−0.28	−0.27	−0.27	−0.27	−0.27	−0.27
	0.16	0.10	0.10	0.14	0.15	0.17
σ_{11}	0.45	0.50	0.53	0.52	0.53	0.53
	—	0.05	0.05	0.05	0.05	0.06
σ_{22}	0.46	0.75	0.77	0.75	0.76	0.76
	—	0.07	0.07	0.07	0.07	0.15
$\mathcal{L}$	—	−701.4	−696.1	−695.3	−695.1	—
Time	0.5	1.5	1.3	1.2	1.4	122

Table 4.2: Epilepsy data. Results for models II and IV showing initialization values from penalized quasi-likelihood (PQL), posterior means and standard deviations (respectively given by the first and second row of each variable) estimated by Algorithm 8 (using the noncentered (NCP), centered (CP) and partially noncentered (PNCP) parametrizations) and MCMC, computation times (seconds) and variational lower bounds ($\mathcal{L}$).

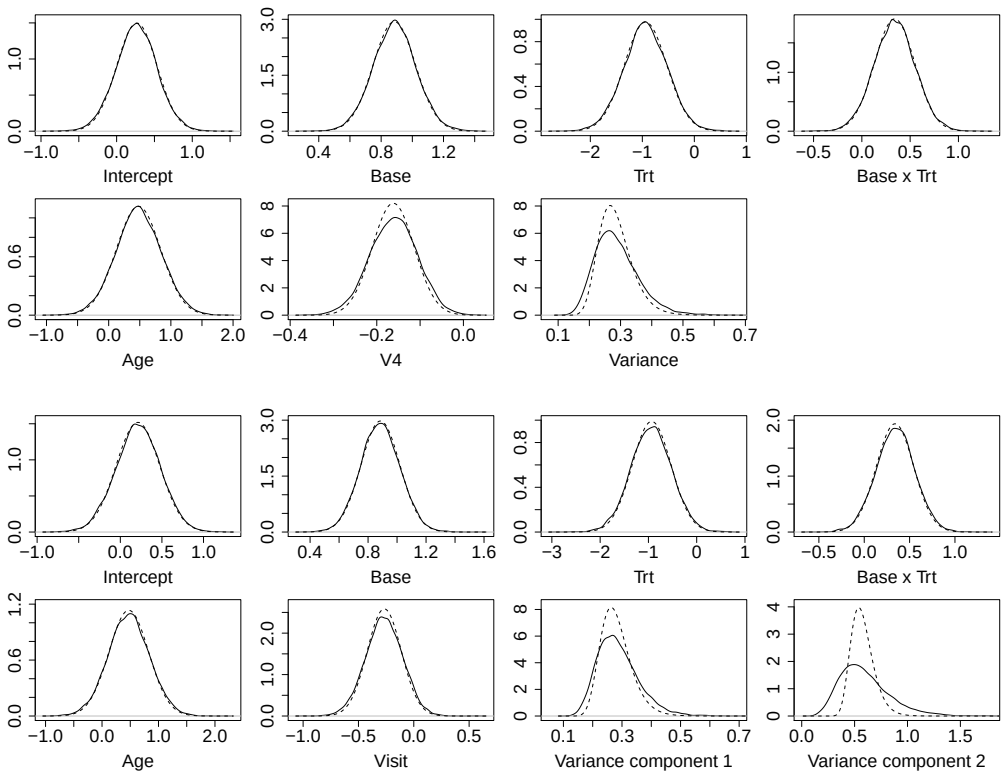

Figure 4.2: Epilepsy data. Marginal posterior distributions of parameters in model II (first two rows) and model IV (last two rows) estimated by MCMC (solid line) and Algorithm 8 using partially noncentered parametrization where tuning parameters are updated (dashed line).

(solid line) and Algorithm 8 using the partially noncentered parametrization where tuning parameters are updated (dashed line). The variational posterior densities of the fixed effects are very close to those obtained via MCMC. For the variance components, there is still some underestimation of the posterior variance.

4.6.3 Toenail data

This data set was obtained from a multicenter study comparing two competing oral antifungal treatments for toenail infection (De Backer *et al.*, 1998). It contains information for 294 patients to be evaluated at seven visits. Not all patients attended all seven planned visits and there were 1908 measurements in total. The patients were randomized into two treatment groups, one group receiving 250 mg per day of terbinafine (Trt=1) and the other group 200 mg per day of itraconazole (Trt=0). Visits were planned at weeks 0, 4, 8, 12, 24, 36 and 48 but patients did not always arrive as scheduled and the exact time in months (t) that they did attend was recorded. The binary response variable (onycholysis) indicates the de-

	PQL	NCP	CP	PNCP: W_i		MCMC
				fixed	updated	
β_0	−0.75	−1.41	−1.44	−1.44	−1.44	−1.65
	0.25	0.17	0.29	0.35	0.32	0.44
β_{Trt}	−0.04	−0.13	−0.13	−0.13	−0.13	−0.17
	0.35	0.25	0.41	0.49	0.45	0.60
β_{t}	−0.30	−0.38	−0.38	−0.38	−0.38	−0.40
	0.03	0.04	0.03	0.03	0.03	0.05
$\beta_{\text{Trt}\times\text{Time}}$	−0.10	−0.13	−0.13	−0.13	−0.13	−0.14
	0.05	0.06	0.04	0.04	0.04	0.07
σ	2.32	3.52	3.56	3.55	3.55	4.10
	—	0.15	0.15	0.15	0.15	0.39
$\mathcal{L}$	—	−664.1	−663.1	−662.7	−662.9	—
Time	2.8	37.9	27.9	26.0	24.1	1072

Table 4.3: Toenail data. Results showing initialization values from penalized quasi-likelihood (PQL), posterior means and standard deviations (respectively given by the first and second row of each variable) estimated by Algorithm 8 (using the noncentered (NCP), centered (CP) and partially noncentered (PNCP) parametrizations) and MCMC, computation times (seconds) and variational lower bounds ($\mathcal{L}$).

gree of separation of the nail plate from the nail-bed (0 if none or mild, 1 if moderate or severe). We consider the following logistic random intercept model,

$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_{\text{Trt}}\text{Trt}_i + \beta_t t_{ij} + \beta_{\text{Trt}\times t}\text{Trt}_i \times t_{ij} + u_i,$$

where $u_i \sim N(0, \sigma^2)$ for $i = 1, \ldots, 294$, $1 \leq j \leq 7$.

Table 4.3 shows the posterior means and standard deviations of the fits from MCMC and Algorithm 8 (using different parametrizations), initialization values from penalized quasi-likelihood and computation time in seconds taken by different methods. Again, the variational methods are faster than MCMC by an order of magnitude. In this example, centering produced a better fit than noncentering and partial noncentering produced a fit closer to that of the centered parametrization but improving it. Partial noncentering also took less time to converge and attained a lower bound higher than that of the centered and noncentered parametrizations. Again, we emphasize that it is not easy to know beforehand which of centering or noncentering will perform better, and a big advantage of partial noncentering is the way that it automatically chooses a good parametrization. In this example, updating the tuning parameters did not result in a better

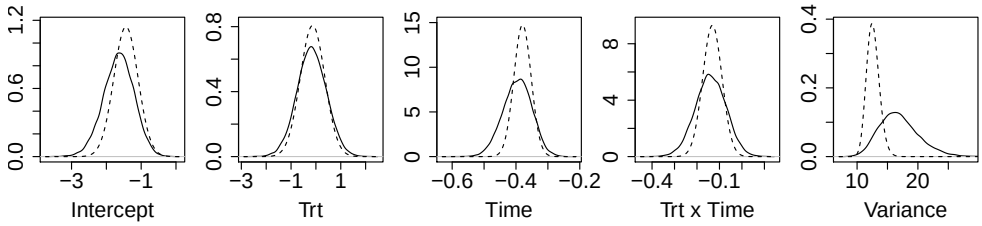

Figure 4.3: Toenail data. Marginal posterior distributions of parameters estimated by MCMC (solid line) and Algorithm 8 using partially noncentered parametrization where tuning parameters are not updated (dashed line).

fit although the time to convergence is reduced. The marginal posterior distributions estimated by MCMC (solid line) and Algorithm 8 using the partially noncentered parametrization where tuning parameters were not updated (dashed line) are shown in Figure 4.3. Compared with the MCMC fit, there is still some underestimation of the variance of the fixed effects particularly for the parameters which could not be centered. Although the partially noncentered parametrization has improved the estimation of random effects from the initial penalized quasi-likelihood fit, there is still some underestimation of the mean and variance of the random effects when compared to the MCMC fit.

4.6.4 Six cities data

In the previous two real data examples, centering performed better than noncentering and partial noncentering was able to improve on the centering results. While centering often performs better than noncentering, we use this example to show that partial noncentering will automatically tend towards noncentering when noncentering is preferred. We consider the six cities data in Fitzmaurice and Laird (1993), where the binary response variable y_{ij} indicates the wheezing status (1 if wheezing, 0 if not wheezing) of the ith child at time-point j, $i = 1, \ldots, 537$, j=1, 2, 3, 4. We use as covariate the age of the child at time-point j, centered at 9 years (Age) and consider the following random intercept and slope model

$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_{\text{Age}}\text{Age}_i + u_{1i} + u_{2i}\text{Age}_i$$

for $i = 1, \ldots, 537$, $j = 1, \ldots, 4$ and $\left[\begin{smallmatrix} u_{1i} \\ u_{2i} \end{smallmatrix}\right] \sim N\left(0, \left[\begin{smallmatrix} \sigma^2_{11} & \sigma_{12} \\ \sigma_{21} & \sigma^2_{22} \end{smallmatrix}\right]\right)$. This model has been considered in Overstall and Forster (2010).

Table 4.4 shows the estimates of the posterior means and standard deviations of the fits from MCMC and Algorithm 8 using different parametriza-

	PQL	NCP	CP	PNCP: W_i		MCMC
				fixed	updated	
β_0	−3.12	−3.05	−3.05	−3.05	−3.05	−3.29
	0.14	0.09	0.09	0.13	0.13	0.25
β_{Age}	−0.24	−0.22	−0.21	−0.22	−0.22	−0.25
	0.08	0.07	0.02	0.07	0.07	0.16
σ_{11}	2.52	2.16	2.16	2.16	2.16	2.48
	—	0.07	0.07	0.07	0.07	0.24
σ_{22}	1.19	0.55	0.56	0.55	0.55	0.61
	—	0.02	0.02	0.02	0.02	0.10
$\mathcal{L}$	—	−833.2	−834.1	−832.8	−832.6	—
Time	3.8	114.7	125.8	110.6	120.6	1010

Table 4.4: Six cities data. Results showing initialization values from penalized quasi-likelihood (PQL), posterior means and standard deviations (respectively given by the first and second row of each variable) estimated by Algorithm 8 (using the noncentered (NCP), centered (CP) and partially noncentered (PNCP) parametrizations) and MCMC, computation times (seconds) and variational lower bounds ($\mathcal{L}$).

tions, the values from penalized quai-likelihood used for initialization and the computation times in seconds taken by different methods. Noncentering performed better than centering in this case with a shorter time to convergence, higher lower bound and a better estimate of the posterior standard deviation of β_{Age}. Partial noncentering further improved upon the results of noncentering with an improved estimate of the posterior standard deviation of β_0 and faster convergence. All the variational methods are again faster than MCMC by an order of magnitude.

4.6.5 Owl data

In this example we illustrate the use of the variational lower bound, a by-product of Algorithm 8, for model selection. For MCMC, on the other hand, it is not straightforward in general to get a good estimate of the marginal likelihood based on the MCMC output. It is also not always obvious how to apply standard model selection criteria like AIC and BIC to hierarchical models like GLMMs.

Roulin and Bersier (2007) analyzed the begging behaviour of nestling barn owls and looked at whether offspring beg for food at different intensities from the mother than father. They sampled $n = 27$ nests and counted the number of calls made by all offspring in the absence of parents. Half of the nests were given extra prey, and from the other half, prey were removed.

Measurements took place on two nights, and food treatment was swapped the second night. The number of measurements at each nest ranged from 4 to 52 with a total of 599. We use as covariates, sex of parent (Sex=1 if male, 0 if female), the time at which a parent arrived with a prey (t), and food treatment (Trt = 1 if 'satiated', 0 if 'deprived'). The number of nestlings per nest (broodsize, E) ranged from 1 to 7.

Zuur *et al.* (2009) modelled the number of calls at nest i for the jth observation as a Poisson distribution with mean μ_{ij} and used log transformed broodsize as an offset with nest as a random effect. The prime aim of their analysis was to find a sex effect and the largest model they considered was

$$\begin{aligned}\text{Model 1: } \log(\mu_{ij}) &= \log(E_{ij}) + \beta_0 + \beta_{\text{Sex}}\text{Sex}_{ij} + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij} \\ &\quad + \beta_{\text{Sex}\times\text{Trt}}\,\text{Sex}_{ij} \times \text{Trt}_{ij} + \beta_{\text{Sex}\times t}\,\text{Sex}_{ij} \times t_{ij} + u_i,\end{aligned}$$

where $\log(E_{ij})$ is an offset and $u_i \sim N(0, \sigma^2)$ for $i = 1, \ldots, 27$, $j = 1, \ldots, n_i$. At the recommendation of Zuur *et al.* (2009), we center t to reduce correlation of t with the intercept. Henceforth, we assume t_{ij} has been replaced by $t_{ij} - \text{mean}(t)$. In the first stage, we consider models 1 to 4 to determine if the two interaction terms should be retained. Models 2 to 4 are as follows:

$$\begin{aligned}\text{Model 2: } \log(\mu_{ij}) &= \log(E_{ij}) + \beta_0 + \beta_{\text{Sex}}\text{Sex}_{ij} + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij} \\ &\quad + \beta_{\text{Sex}\times\text{Trt}}\,\text{Sex}_{ij} \times \text{Trt}_{ij} + u_i,\end{aligned}$$

$$\begin{aligned}\text{Model 3: } \log(\mu_{ij}) &= \log(E_{ij}) + \beta_0 + \beta_{\text{Sex}}\text{Sex}_{ij} + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij} \\ &\quad + \beta_{\text{Sex}\times t}\,\text{Sex}_{ij} \times t_{ij} + u_i,\end{aligned}$$

$$\text{Model 4: } \log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_{\text{Sex}}\text{Sex}_{ij} + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij} + u_i.$$

From Table 4.5, the preferred model (with the highest lower bound) is model 4 where both interaction terms have been dropped from model 1.

Next, we consider models 5 to 7 where the main terms sex, food treatment and arrival time are each dropped in turn,

$$\text{Model 5: } \log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij} + u_i,$$

$$\text{Model 6: } \log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_{\text{Sex}}\text{Sex}_{ij} + u_i,$$

$$\text{Model 7: } \log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_t t_{ij} + \beta_{\text{Sex}}\text{Sex}_{ij} + u_i.$$

Table 4.5 indicates that model 5 is the preferred model where the term sex of the parent has been dropped from model 4. Now we consider dropping each of the terms food treatment and arrival time in turn or dropping the random effects u_i,

	NCP	CP	PNCP: W_i	
			fixed	updated
First stage:				
Model 1	−2544.6(0.2)	−2543.7(0.3)	−2543.6(0.4)	−2543.7(0.6)
Model 2	−2537.6(0.2)	−2536.6(0.3)	−2536.6(0.4)	−2536.6(0.5)
Model 3	−2540.2(0.2)	−2539.2(0.3)	−2539.2(0.3)	−2539.2(0.5)
Model 4	−2533.2(0.2)	−2532.1(0.3)	−2532.1(0.3)	−2532.1(0.4)
Second stage:				
Model 5	−2527.0(0.2)	−2525.5(0.2)	−2525.5(0.2)	−2525.4(0.3)
Model 6	−2628.3(0.2)	−2627.2(0.3)	−2627.1(0.3)	−2627.1(0.5)
Model 7	−2664.0(0.2)	−2662.9(0.2)	−2662.8(0.3)	−2662.8(0.4)
Third stage:				
Model 8	−2621.5(0.2)	−2620.0(0.2)	−2620.0(0.2)	−2620.0(0.3)
Model 9	−2660.4(0.2)	−2658.8(0.2)	−2658.8(0.2)	−2658.8(0.2)
Model 10	$-2689.4(< 0.05)$			
Final stage:				
Model 11	−2448.7 (1.1)	−2445.7(0.4)	−2445.8(0.3)	−2445.6(0.4)

Table 4.5: Owl data. Variational lower bounds for models 1 to 11 and computation time in brackets for the noncentered (NCP), centered (CP) and partially noncentered (PNCP) parametrizations.

Model 8: $\log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_{\text{Trt}}\text{Trt}_{ij} + u_i$,

Model 9: $\log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_t t_{ij} + u_i$,

Model 10: $\log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij}$.

Table 4.5 indicates that none of the main terms food treatment and arrival time as well as random effects should be dropped from model 5. Finally we consider adding a random slope for arrival time,

Model 11: $\log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + \beta_{\text{Trt}}\text{Trt}_{ij} + \beta_t t_{ij} + u_{1i} + u_{2i}t_{ij}$,

where $\left[\begin{smallmatrix} u_{1i} \\ u_{2i} \end{smallmatrix}\right] \sim N\left(0, \left[\begin{smallmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{21} & \sigma_{22}^2 \end{smallmatrix}\right]\right)$. From Table 4.5, the optimal model is model 11. This conclusion is similar to that of Zuur *et al.* (2009) and is the same regardless of which parametrization was used. It is thus sufficient to consider just the partially noncentered parametrization. The computation time taken by Algorithm 8 for each model fitting is very short and makes this a convenient way of carrying out model selection or for narrowing down the range of likely models. Further model comparisons can be performed using cross-validation or other approaches.

We present the estimated posterior means and standard deviations for the optimal model in Table 4.6. The marginal posterior distributions estimated by MCMC (solid line) and Algorithm 8 using partially noncentered

	PQL	NCP	CP	PNCP: W_i		MCMC
				fixed	updated	
β_0	0.60	0.53	0.51	0.51	0.51	0.50
	0.07	0.02	0.08	0.08	0.09	0.10
β_{Trt}	-0.55	-0.57	-0.57	-0.57	-0.57	-0.57
	0.08	0.03	0.03	0.03	0.03	0.04
β_t	-0.13	-0.15	-0.16	-0.16	-0.16	-0.16
	0.03	0.01	0.04	0.04	0.04	0.05
σ_{11}	0.24	0.44	0.46	0.45	0.46	0.47
	—	0.06	0.06	0.06	0.06	0.09
σ_{22}	0.11	0.22	0.23	0.22	0.23	0.23
	—	0.03	0.03	0.03	0.03	0.05
Time	0.4	1.1	0.4	0.3	0.4	255

Table 4.6: Owl data. Results showing initialization values from penalized quasi-likelihood (PQL), posterior means and standard deviations (respectively given by the first and second row of each variable) estimated by Algorithm 8 (using the noncentered (NCP), centered (CP) and partially noncentered (PNCP) parametrizations, and MCMC, computation times (seconds) and variational lower bounds ($\mathcal{L}$).

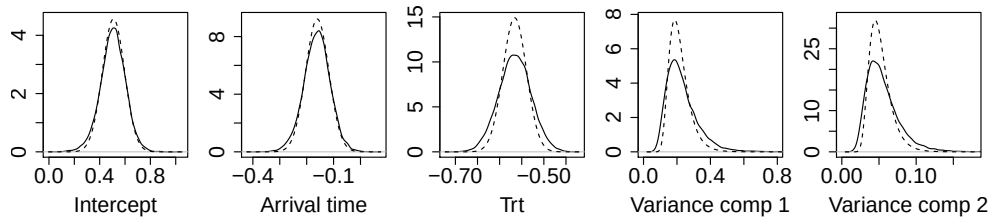

Figure 4.4: Owl data. Marginal posterior distributions for parameters in model 11 estimated by MCMC (solid line) and Algorithm 8 using partially noncentered parametrization where tuning parameters are updated (dashed line).

parametrization where tuning parameters are updated (dashed line) are shown in Figure 4.4. In this case, centering produced a better fit than noncentering and partial noncentering produced a fit that is close to that of centering. Updating the tuning parameters helped to improve the fit of the partially noncentered parametrization slightly and is closest to the MCMC fit. From the posterior density plots, there is good estimation of the posterior means by Algorithm 8 using partially noncentered parametrization with updated tuning parameters but there is still some underestimation of the posterior variance.

4.7 Conclusion

In this chapter, we have described a partially noncentered parametrization for GLMMs and compared the performance of different parametrizations using an algorithm called nonconjugate variational message passing. Focusing on Poisson and logistic mixed models, we applied our methods to the analysis of longitudinal data sets. For the logistic model, some parameter updates were not available in closed form and we used adaptive Gauss-Hermite quadrature to approximate the intractable integrals efficiently. Comparing the performance of Algorithm 8 under the partially noncentered parametrization with that of the centered and noncentered parametrizations, we observed that partial noncentering automatically tends towards the better of centering and noncentering so that it is not necessary to choose in advance between the centered and noncentered parametrizations. In many cases, the partially noncentered parametrization was able to improve upon the fit produced by the better of centering and noncentering to produce a fit that was closest to that of MCMC. In terms of computation time, the partially noncentered parametrization can also provide more rapid convergence when centering or noncentering is particularly slow. Very often, the lower bound attained by the partially noncentered parametrization is also higher than that of the centered and noncentered parametrizations giving a tighter lower bound to the log marginal likelihood. To some degree, the partially noncentered parametrization also alleviates the issue of underestimation of the posterior variance leading to some improvement in the estimation of the posterior variance particularly in the fixed effects which could be centered. Algorithm 8 under the partially noncentered parametrization thus offers itself as a fast, deterministic alternative to MCMC methods for fitting GLMMs with improved estimation compared to the centered and noncentered parametrizations. We also demonstrate that the variational lower bound produced as part of the computation in Algorithm 8 can be useful in model selection.

Chapter 5

A stochastic variational framework for fitting and diagnosing generalized linear mixed models

In Chapter 4, we described a partially noncentered parametrization for GLMMs and demonstrated how they can be fitted using nonconjugate variational message passing. Like other batch VB algorithms for models with observation specific latent variables, the nonconjugate variational message passing algorithm for GLMMs has to iterate between updating local variational parameters associated with individual observations and global variational parameters. For large data sets, this procedure becomes increasingly inefficient as local variational parameters associated with every unit have to be updated at every iteration. Generally, batch VB algorithms are also unsuitable in online settings where data arrive continuously as the algorithm can never complete one iteration. On the other hand, stochastic gradient optimization (Robbins and Monro, 1951) uses only a random subset of the data at each iteration to approximate the true gradient over the whole data so that computational cost is reduced significantly for large data sets (Bottou and Cun, 2005; Bottou and Bousquet, 2008). Hoffman *et al.* (2013) developed stochastic variational inference for conjugate-exponential family models by optimizing the VB objective function using stochastic gradient approximation.

In this chapter, we extend stochastic variational inference for conjugate-exponential family models to nonconjugate models and present a stochastic version of nonconjugate variational message passing for fitting GLMMs that is scalable to large data sets. This is achieved by combining updates in nonconjugate variational message passing with stochastic natural gradient optimization of the variational lower bound. One strong motivation

for the development of stochastic gradient optimization algorithms is their efficiency in terms of memory — because they process data in mini-batches, analysis of data sets which are so large that they cannot fit into memory can still be contemplated. We continue to use the partially noncentered parametrization for GLMMs introduced in Section 4.3 and focus on Poisson and logistic mixed models and their applications in longitudinal data analysis.

In addition, we show that diagnostics for prior-likelihood conflict, which are useful for Bayesian model criticism, can be obtained from nonconjugate variational message passing automatically, as an alternative to simulation-based, computationally intensive MCMC methods. Intuitively, the updates in variational message passing can be separated into "messages" coming from above and below a node in a hierarchical model and "mixed messages" indicate conflict. Our "mixed messages" diagnostics can be shown to approximate existing diagnostics in the statistical literature, namely, the conflict diagnostics of Marshall and Spiegelhalter (2007).

Finally, we demonstrate that for moderate-sized data sets, convergence can be accelerated by using the stochastic version of nonconjugate variational message passing in the initial stage of optimization before switching to the standard version. Some insights on step size optimization with respect to mini-batch sizes are provided.

This chapter is organized as follows. Section 5.1 provides some background. A stochastic version of the nonconjugate variational message passing algorithm is developed in Section 5.2. Section 5.3 describes how variational message passing facilitates automatic computation of diagnostics for prior-likelihood conflict. Section 5.4 considers examples including real and simulated data and Section 5.5 concludes.

The results presented in this chapter are covered in Tan and Nott (2013c), which has been submitted for publication.

5.1 Background

Recent developments in VB methodology have branched out to stochastic optimization, making VB a viable approach for handling large data sets. Hoffman *et al.* (2010) and Wang *et al.* (2011) developed online VB algorithms for latent Dirichlet allocation and the hierarchical Dirichlet process respectively using stochastic natural gradient optimization of the variational lower bound. Hoffman *et al.* (2013) generalized these methods to derive stochastic variational inference for conjugate-exponential family mod-

els and showed that stochastic variational inference converges faster than batch VB for large data sets. Paisley *et al.* (2012) proposed a stochastic optimization algorithm using control variates that allows direct maximization of the variational lower bound involving intractable integrals. Similar algorithms were considered by Ji *et al.* (2010) and Nott *et al.* (2012). Welling and Teh (2011) combined stochastic gradient optimization with Langevin dynamics for Bayesian learning from large data sets and Ahn *et al.* (2012) extended this algorithm to stochastic gradient Fisher scoring. Salimans and Knowles (2012) proposed a stochastic approximation algorithm that does not require analytic evaluation of integrals, extending the VB approach to any posterior that is available in closed form up to the proportionality constant. Hierarchical extensions of the basic approach allow the method to be made arbitrarily precise.

Model checking is an important part of statistical analyses. In the Bayesian approach, assumptions are made about the sampling model and prior, and prior-likelihood conflict arises when the observed data are very unlikely under the prior model. Evans and Moshonov (2006) discussed how to assess whether there is prior-data conflict and Scheel *et al.* (2011) proposed a graphical diagnostic, the local critique plot, for identifying influential statistical modelling choices at the node level. See also Scheel *et al.* (2011) for a review of other methods in Bayesian model criticism. Marshall and Spiegelhalter (2007) proposed a diagnostic test for identifying divergent units in hierarchical models based on measuring the conflict between the likelihood of a parameter and its predictive prior given the remaining data. A simulation-based approach was adopted and diagnostic tests were carried out using MCMC. We show that the approach of Marshall and Spiegelhalter (2007) can be approximated in the variational message passing framework.

5.2 Stochastic variational inference for generalized linear mixed models

In this section, we develop stochastic variational inference for the GLMM specified in Section 4.2, focusing on Poisson and logistic mixed models and using the same priors as before. We consider the partially noncentered parametrization for GLMMs described in Section 4.3, which has been shown to be able to automatically determine a parametrization close to optimal. Recall that the set of unknown parameters θ in the GLMM consist of the

fixed effects β, the random effects covariance D and the partially noncentered random effects $\tilde{\alpha}_i$, $i = 1, \ldots, n$. Here, β and D can be regarded as "global" variables which are common across clusters while $\tilde{\alpha}_i$, $i = 1, \ldots, n$, can be thought of as "local" variables associated only with the individual units. In Section 4.4, we considered a variational approximation $q(\theta)$ to the joint posterior $p(\theta|y)$ of the form

$$q(\theta) = q(\beta)q(D)\prod_{i=1}^{n} q(\tilde{\alpha}_i),$$

where $q(\beta)$ is $N\left(\mu_\beta^q, \Sigma_\beta^q\right)$, $q(D)$ is $IW(\nu^q, S^q)$, and $q(\tilde{\alpha}_i)$ is $N\left(\mu_{\tilde{\alpha}_i}^q, \Sigma_{\tilde{\alpha}_i}^q\right)$, $i = 1, \ldots, n$. In the standard nonconjugate variational message passing algorithm for GLMMs (Algorithm 8), we iterate between updating the local variational parameters associated with $\tilde{\alpha}_i$ for each unit i, $i = 1, \ldots, n$, and re-estimating the global variational parameters associated with β and D. This can be inefficient for large data sets and impossible to accomplish for streaming data or data sets which are too massive to fit into memory.

Let λ_β, λ_D and $\lambda_{\tilde{\alpha}_i}$ denote the natural parameter vectors of $q(\beta)$, $q(D)$ and $q(\tilde{\alpha}_i)$ respectively for $i = 1, \ldots, n$. We have

$$\lambda_\beta = \begin{bmatrix} -\frac{1}{2} D_p^T \mathrm{vec}(\Sigma_\beta^{q\,-1}) \\ \Sigma_\beta^{q\,-1}\mu_\beta^q \end{bmatrix}, \; \lambda_D = \begin{bmatrix} -\frac{1}{2}\mathrm{vec}(S^q) \\ -\frac{\nu^q + r + 1}{2} \end{bmatrix}, \; \lambda_{\tilde{\alpha}_i} = \begin{bmatrix} -\frac{1}{2} D_r^T \mathrm{vec}(\Sigma_{\tilde{\alpha}_i}^{q\,-1}) \\ \Sigma_{\tilde{\alpha}_i}^{q\,-1}\mu_{\tilde{\alpha}_i}^q \end{bmatrix},$$

where D_p and D_r are defined in a similar manner as the matrix D_d in Section 4.4.1. In the stochastic version of nonconjugate variational message passing, we propose to randomly select a mini-batch, S, of units, of size $|S| \geq 1$ at each iteration and compute nonconjugate variational message passing updates for $\lambda_{\tilde{\alpha}_i}$, $i \in S$ repeatedly until convergence. Using these optimized local variational parameters, we then compute unbiased estimates of the natural gradients of $\mathcal{L}$ with respect to λ_β and λ_D and estimate λ_β and λ_D using stochastic gradient approximation. In other words, we use stochastic natural gradient ascent to find a setting of the global variational parameters that maximizes the lower bound, by considering the variational lower bound as a function of the global variational parameters with the local parameters optimized as a function of these global parameters. Similar approaches have been considered by Hoffman *et al.* (2010) for latent Dirichlet allocation, Wang *et al.* (2011) for the hierarchical Dirichlet process and Hoffman *et al.* (2013) for conjugate-exponential family models in general.

Next, we motivate and derive expressions of the natural gradient of

the variational lower bound under the assumptions made in nonconjugate variational message passing.

5.2.1 Natural gradient of the variational lower bound

The key idea in stochastic variational inference is to optimize $\mathcal{L}$ using stochastic gradient approximation (see Spall, 2003), where the gradients are computed based on mini-batches of data and represent unbiased estimates of the true gradients over the whole data set. Let us assume that $q(\theta)$ belongs to some parametric family with parameters λ and we write $q(\theta)$ as $q(\theta|\lambda)$. Hoffman *et al.* (2013) argued that in the optimization of $q(\theta|\lambda)$, the Euclidean metric might not be the best measure of distance between different parameter settings of λ. This is because a large change in λ might not be equivalent with a large change in the Kullback-Leibler divergence between $q(\theta|\lambda)$ and $p(\theta|y)$, which is what we are concerned with. They proposed using the natural gradient of $\mathcal{L}$ instead of the ordinary gradient in the stochastic optimization as the steepest direction of ascent is given by the natural gradient in a space where the dissimilarity between two probability distributions is measured in terms of the symmetrized Kullback-Leibler divergence (see Amari, 1998). Honkela *et al.* (2008) also showed that replacing the ordinary gradient in the conjugate gradient algorithm with the natural gradient can speed up variational learning. Therefore, we use the natural gradient instead of the ordinary gradient in the stochastic optimization.

In nonconjugate variational message passing, we assume that $q(\theta|\lambda)$ is factorized as $\prod_{i=1}^{m} q_i(\theta_i|\lambda_i)$ for some partition $\{\theta_1, \ldots, \theta_m\}$ of θ, and each q_i belongs to some exponential family, say,

$$q_i(\theta_i|\lambda_i) = \exp\{\lambda_i^T t_i(\theta_i) - h_i(\lambda_i)\},$$

where λ_i is the vector of natural parameters and $t_i(\cdot)$ are the sufficient statistics. Then $\lambda = \{\lambda_1, \ldots, \lambda_m\}$. Suppose $p(y, \theta) = \prod_a f_a(y, \theta)$ and $S_a = E_q\{\log f_a(y, \theta)\}$, where E_q denotes expectation with respect to $q(\theta|\lambda)$. From (4.13), the ordinary gradient of $\mathcal{L}$ with respect to λ_i is

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \lambda_i} - \mathcal{V}_i(\lambda_i)\lambda_i,$$

where the summation is over all factors in $N(\theta_i)$, the neighbourhood of θ_i in the factor graph of $p(y, \theta)$ and $\mathcal{V}_i(\lambda_i)$ denotes the variance-covariance matrix of $t_i(\theta_i)$. To obtain the natural gradient of $\mathcal{L}$ with respect to λ_i, we

premultiply $\frac{\partial \mathcal{L}}{\partial \lambda_i}$ with the inverse of the Fisher information matrix for the variational posterior $q_i(\theta_i|\lambda_i)$ (see, e.g. Honkela *et al.*, 2008; Hoffman *et al.*, 2013). The Fisher information matrix for $q_i(\theta_i|\lambda_i)$ is given by

$$\begin{aligned}
&E_q\left\{\frac{\partial \log q_i(\theta_i|\lambda_i)}{\partial \lambda_i}\left(\frac{\partial \log q_i(\theta_i|\lambda_i)}{\partial \lambda_i}\right)^T\right\} \\
&= E_q\left\{\left(t_i(\theta_i) - \frac{\partial h_i(\lambda_i)}{\partial \lambda_i}\right)\left(t_i(\theta_i) - \frac{\partial h_i(\lambda_i)}{\partial \lambda_i}\right)^T\right\} \\
&= \mathcal{V}_i(\lambda_i).
\end{aligned}$$

Provided $\mathcal{V}_i(\lambda_i)$ is invertible, the natural gradient $\nabla_{\lambda_i}\mathcal{L}$ is given by

$$\nabla_{\lambda_i}\mathcal{L} = \mathcal{V}_i(\lambda_i)^{-1}\sum_{a\in N(\theta_i)}\frac{\partial S_a}{\partial \lambda_i} - \lambda_i. \tag{5.1}$$

Note that the updates in nonconjugate variational message passing can be obtained by setting the natural gradient as zero.

Suppose each factor f_a in the neighbourhood of θ_i is conjugate to $q_i(\theta_i|\lambda_i)$, say,

$$f_a(y,\theta) = \exp\left\{g_a(y,\theta_{-i})^T t_i(\theta_i) - h_a(y,\theta_{-i})\right\},$$

where $\theta_{-i} = (\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_m)$. From (4.15), the natural gradient can be simplified as

$$\nabla_{\lambda_i}\mathcal{L} = \sum_{a\in N(\theta_i)} E_q\{g_a(y,\theta_{-i})\} - \lambda_i. \tag{5.2}$$

Note that $E_q\{g_a(y,\theta_{-i})\}$ does not depend on λ_i.

5.2.2 Stochastic nonconjugate variational message passing

Next, we present unbiased estimates of the natural gradients $\nabla_{\lambda_\beta}\mathcal{L}$ and $\nabla_{\lambda_D}\mathcal{L}$ obtained from a mini-batch S of randomly selected units. As before, we let $S_\beta = E_q\{\log p(\beta|\Sigma_\beta)\}$, $S_{\tilde{\alpha}_i} = E_q\{\log p(\tilde{\alpha}_i|\beta,D)\}$ and $S_{y_i} = E_q\{\log p(y_i|\beta,\tilde{\alpha}_i)\}$ for $i = 1,\ldots,n$. From (5.1), the natural gradient of $\mathcal{L}$ with respect to λ_β is

$$\nabla_{\lambda_\beta}\mathcal{L} = \mathcal{V}_\beta(\lambda_\beta)^{-1}\left\{\frac{\partial S_\beta}{\partial \lambda_\beta} + \sum_{i=1}^{n}\left(\frac{\partial S_{\tilde{\alpha}_i}}{\partial \lambda_\beta} + \frac{\partial S_{y_i}}{\partial \lambda_\beta}\right)\right\} - \lambda_\beta,$$

and an unbiased estimate of $\nabla_{\lambda_\beta}\mathcal{L}$ using the mini-batch S is

$$\hat{\nabla}_{\lambda_\beta}\mathcal{L} = \hat{\lambda}_\beta - \lambda_\beta, \tag{5.3}$$

where

$$\hat{\lambda}_\beta = \mathcal{V}_\beta(\lambda_\beta)^{-1}\left\{\frac{\partial S_\beta}{\partial \lambda_\beta} + \frac{n}{|S|}\sum_{i\in S}\left(\frac{\partial S_{\tilde{\alpha}_i}}{\partial \lambda_\beta} + \frac{\partial S_{y_i}}{\partial \lambda_\beta}\right)\right\}.$$

For $q(D)$, since the factors in the neighbourhood of D are all conjugate factors, we have from (5.2),

$$\nabla_{\lambda_D}\mathcal{L} = \begin{bmatrix} -\frac{1}{2}\text{vec}(S) \\ -\frac{\nu+r+1}{2} \end{bmatrix} + \sum_{i=1}^{n}\begin{bmatrix} -\frac{1}{2}B_i \\ -\frac{1}{2} \end{bmatrix} - \lambda_D,$$

where $B_i = \text{vec}\big[(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta)(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta)^T + \Sigma^q_{\tilde{\alpha}_i} + \tilde{W}_i\Sigma^q_\beta\tilde{W}_i^T\big]$. An unbiased estimate of $\nabla_{\lambda_D}\mathcal{L}$ using mini-batch S is

$$\hat{\nabla}_{\lambda_D}\mathcal{L} = \hat{\lambda}_D - \lambda_D, \tag{5.4}$$

where

$$\hat{\lambda}_D = \begin{bmatrix} -\frac{1}{2}\text{vec}(S) \\ -\frac{\nu+r+1}{2} \end{bmatrix} + \frac{n}{|S|}\sum_{i\in S}\begin{bmatrix} -\frac{1}{2}B_i \\ -\frac{1}{2} \end{bmatrix}.$$

When S is the entire data set, $\hat{\lambda}_\beta$ and $\hat{\lambda}_D$ are the updates of λ_β and λ_D in the standard nonconjugate variational message passing algorithm.

The stochastic version of nonconjugate variational message passing for fitting Poisson and logistic mixed models is presented in Algorithm 9. Refer to Section 4.3 for the definitions of the tuning parameters W_i and $\tilde{W}_i$ for $i = 1, \ldots, n$. Note that the definitions of F_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, n_i$, and G_i for $i = 1, \ldots, n$, given in Section 4.4.2 differs according to whether a Poisson or logistic mixed model is being fitted. In the case of logistic mixed models, adaptive Gauss-Hermite quadrature (Liu and Pierce, 1994) is required for the evaluation of F_{ij} and G_i. More details are given in Appendix D.

Algorithm 9: Stochastic nonconjugate variational message passing for GLMMs

Initialize variational parameters μ^q_β, Σ^q_β, ν^q, S^q, $\mu^q_{\tilde{\alpha}_i}$, $\Sigma^q_{\tilde{\alpha}_i}$ and the tuning parameters W_i for $i = 1, \ldots, n$.
For $t = 0, 1, 2, \ldots$,

1. Randomly select a subset S of $|S|$ units from the entire data set.

2. Update local variational parameters $\mu^q_{\tilde{\alpha}_i}$ and $\Sigma^q_{\tilde{\alpha}_i}$ for $i \in S$ repeatedly using the updates in nonconjugate variational message passing:

   - $\Sigma^q_{\tilde{\alpha}_i} \leftarrow \left(\nu^q S^{q-1} + \sum_{j=1}^{n_i} F_{ij} X^R_{ij} X^{R^T}_{ij}\right)^{-1}$,
   - $\mu^q_{\tilde{\alpha}_i} \leftarrow \mu^q_{\tilde{\alpha}_i} + \Sigma^q_{\tilde{\alpha}_i}\big\{ -\nu^q S^{q-1}(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i \mu^q_\beta) + X^{R^T}_i (y_i - G_i)\big\}$,

   until convergence is reached.

3. Update the global variational parameters μ^q_β, Σ^q_β, ν^q and S^q using

   - $\Sigma^q_\beta \leftarrow \Big[a_t\big\{\Sigma^{-1}_\beta + \frac{n}{|S|}\sum_{i\in S}\left(\nu^q \tilde{W}_i^T S^{q-1}\tilde{W}_i + \sum_{j=1}^{n_i} F_{ij} V_{ij} V_{ij}^T\right)\big\} + (1-a_t)\Sigma^{q\,-1}_\beta\Big]^{-1}$,
   - $\mu^q_\beta \leftarrow \mu^q_\beta + a_t \Sigma^q_\beta \Big[-\Sigma^{-1}_\beta \mu^q_\beta + \frac{n}{|S|}\sum_{i\in S}\big\{\nu^q \tilde{W}_i^T S^{q-1}(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta) + V_i^T(y_i - G_i)\big\}\Big]$,
   - $S^q \leftarrow (1-a_t)S^q + a_t\Big[S + \frac{n}{|S|}\sum_{i\in S}\big\{(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta)(\mu^q_{\tilde{\alpha}_i} - \tilde{W}_i\mu^q_\beta)^T + \Sigma^q_{\tilde{\alpha}_i} + \tilde{W}_i\Sigma^q_\beta\tilde{W}_i^T\big\}\Big]$,
   - $\nu^q \leftarrow (1-a_t)\nu^q + a_t(\nu + n)$.

The updates in step 2 of Algorithm 9 are from the nonconjugate variational message passing algorithm for GLMMs (Algorithm 8) while the stochastic approximation updates in step 3 can be derived from

$$\lambda^{(t)}_\beta = \lambda^{(t-1)}_\beta + a_t \hat{\nabla}_{\lambda_\beta}\mathcal{L}|_{\lambda_\beta = \lambda^{(t-1)}_\beta} \quad \text{and}$$
$$\lambda^{(t)}_D = \lambda^{(t-1)}_D + a_t \hat{\nabla}_{\lambda_D}\mathcal{L}|_{\lambda_D = \lambda^{(t-1)}_D}. \tag{5.5}$$

These stochastic approximation steps were introduced by Robbins and Monro (1951) for optimizing an objective function, which in our case is the lower bound $\mathcal{L}$, with local variational parameters optimized as a function of the global ones. Hoffman *et al.* (2013) note that the gradient of this function is the gradient of $\mathcal{L}$ with the local parameters fixed at their optimized values (see Hoffman *et al.*, 2013, equation (39)). The updates in (5.5) are similar to the update in step 2 of Algorithm 2, where a stochastic gradient approximation was also used. In this case, however, we are using the natural gradient of the variational lower bound instead of the usual gradient. Under certain regularity conditions (see Spall, 2003), the iterates will converge to a local maximum of the lower bound. In particular, the gain sequence a_t, $t \geq 0$ should satisfy the conditions in (2.10). See Section 2.5.2 for more discussion on the gain sequence a_t. Here, we consider

step sizes of the form $\frac{1}{(t+K)^\gamma}$ where $0.5 < \gamma \leq 1$ and $K \geq 0$ is a stability constant that helps to avoid unstable behaviour in the early iterations. In practice, choices of the step sizes can strongly influence the performance of the algorithm (Jank, 2006). As $\hat{\nabla}_{\lambda_\beta}\mathcal{L} = \hat{\lambda}_\beta - \lambda_\beta$ and $\hat{\nabla}_{\lambda_D}\mathcal{L} = \hat{\lambda}_D - \lambda_D$ from (5.3) and (5.4) respectively, we have from (5.5),

$$\lambda_\beta^{(t)} = (1 - a_t)\lambda_\beta^{(t-1)} + a_t\hat{\lambda}_\beta|_{\lambda_\beta=\lambda_\beta^{(t-1)}} \text{ and}$$
$$\lambda_D^{(t)} = (1 - a_t)\lambda_D^{(t-1)} + a_t\hat{\lambda}_D.$$

This implies that the t-iterate can be interpreted as a weighted average of the previous iterate and the nonconjugate variational message passing update estimated from mini-batch S. In fact, standard nonconjugate variational message passing can be recovered from Algorithm 9 if the update for the local parameters in step 2 is performed only once and $a_t = 1$ in step 3. This shows that nonconjugate variational message passing is a type of natural gradient method with step size 1 and other schedules are equivalent to damping. Previously, Sato (2001) showed that the VB algorithm was a type of natural gradient method and derived an online VB algorithm with a model selection mechanism for Gaussian mixture models using stochastic approximation.

Algorithm 9 is initialized using the fit to the generalized linear model considered in Section 4.3, obtained by pooling all the data and setting the random effects as zero. We set μ_β^q and Σ_β^q as estimates of the regression coefficients and their covariances respectively from the generalized linear model, $\nu^q = r$, $S^q = S$, $\mu_{\tilde{\alpha}_i}^q = \tilde{W}_i\mu_\beta^q$ and $\Sigma_{\tilde{\alpha}_i}^q = \hat{R}$, where $\hat{R}$ is as defined in (4.4). The tuning parameters $\{W_i\}$ were initialized by setting $D = \hat{R}$ and $\eta_i = X_i\mu_\beta^q$ for each $i = 1, \ldots, n$. Kass and Natarajan (2006) gave a justification of $\hat{R}$ being a reasonable guess for D in the absence of any other prior knowledge. Care should be taken in initializing the variational parameters as the nonconjugate variational message passing updates in step 2 are not guaranteed to converge. We used the initialization suggested above in all our examples and did not experience any convergence issues. The mean parameters of $\tilde{\alpha}_i$ were used to test for convergence in step 2 and we stop when $\frac{\left\|\mu_{\tilde{\alpha}_i}^{q\ (t)} - \mu_{\tilde{\alpha}_i}^{q\ (t-1)}\right\|}{\left\|\mu_{\tilde{\alpha}_i}^{q\ (t)}\right\|} < 0.01$ where $\|\cdot\|$ represents the Euclidean norm.

5.2.3 Switching from stochastic to standard version

Determining an appropriate stopping criterion for a stochastic approximation algorithm can be very challenging. Some commonly used stopping criteria include stopping when the relative change in parameter values or objective function is sufficiently small or when the gradient of the objective function is sufficiently close to zero (Spall, 2003). Such criteria do not provide any guarantees of the terminal iterate being close to the optimum, however, and may be satisfied by random chance. Booth *et al.* (1999) recommend applying such rules for several consecutive iterations to minimize chances of a premature stop. However, Jank (2006) gave an illustrative example to show that even this may not be enough of a safeguard. Moreover, stochastic approximation can become excruciatingly slow in later iterations due to the small step sizes.

Through our experimentations with moderate-sized data sets, we observe that gains made by Algorithm 9 are usually largest in the first few iterations. However, beyond a certain point, it can become slower than the standard version if the step sizes are too small or the iterates simply bounce around if the step sizes are still too big. We therefore suggest switching to the standard version when the stochastic version shows signs of slowing down. Using the lower bound both as a switching and stopping criterion, we propose switching from stochastic to standard nonconjugate variational message passing when the relative increase in the lower bound is less than 10^{-3} and terminating standard nonconjugate variational message passing when the absolute relative change in the lower bound is less than 10^{-6}. For large data sets or streaming data, it might be more practical to terminate Algorithm 9 beyond a certain period of available runtime.

For the examples in Section 5.4, the mini-batches in step 1 of Algorithm 9 were chosen by random-partitioning of the data set and the mini-batch sizes considered were such that different batches differ in size by at most one when n is not divisible by $|S|$. For greater efficiency, the lower bound is computed only after a complete sweep has been made through the data set. We replace t by $s_w + \frac{m}{M}$ in the step size where s_w indicates the number of sweeps that has been made through the data, M denotes the number of partitions of the data and $0 \leq m \leq M-1$ denotes the number of batches that has been analysed. It is possible to include an update of the tuning parameters W_i after each complete sweep. However, preliminary investigation did not suggest significant improvement in results when W_i is updated and hence, for the examples in Section 5.4, we did not update W_i

beyond the initialization.

5.3 Automatic diagnostics of prior-likelihood conflict as a by-product of variational message passing

Marshall and Spiegelhalter (2007) investigated a diagnostic test for identifying units that do not appear to be drawn from assumed underlying distributions based on measuring the conflict between likelihood of a parameter and its predictive prior given the remaining data. A simulation-based approach was adopted and tests were performed using MCMC. Here, we show that the approach of Marshall and Spiegelhalter (2007) can be approximated in the variational message passing framework and that variational message passing facilitates an automatic computation of diagnostics for prior-likelihood conflict, very useful for Bayesian model criticism. We focus on nonconjugate variational message passing for GLMMs.

First, we review briefly the diagnostic test proposed by Marshall and Spiegelhalter (2007). In the context of GLMMs with a partially noncentered parametrization, the parameter of interest for identifying divergent units is $\tilde{\alpha}_i$, $i = 1, \ldots, n$. For $\tilde{\alpha}_i$, Marshall and Spiegelhalter (2007) suggest generating a predictive prior replicate $\tilde{\alpha}_i^{\text{rep}} \sim p(\tilde{\alpha}_i|y_{-i})$ where y_{-i} denotes the observed data y with unit i left out and

$$p(\tilde{\alpha}_i|y_{-i}) = \int p(\tilde{\alpha}_i|\beta, D)p(\beta, D|y_{-i})\, d\beta dD. \tag{5.6}$$

In the simulation approach, β^{rep}, D^{rep} would be generated from $p(\beta, D|y_{-i})$ using MCMC followed by simulation of $\tilde{\alpha}_i^{\text{rep}}|\beta^{\text{rep}}, D^{\text{rep}}$. This is compared with a likelihood replicate $\tilde{\alpha}_i^{\text{fix}} \sim p(\tilde{\alpha}_i|y_i)$ generated using only data from the unit y_i being tested and a non-informative prior, $p(\tilde{\alpha}_i)$, for $\tilde{\alpha}_i$ since $p(\tilde{\alpha}_i|y_i) \propto p(y_i|\tilde{\alpha}_i)p(\tilde{\alpha}_i)$. These prior and likelihood replications represent two independent sources of evidence about $\tilde{\alpha}_i$ and conflict between them suggests discrepancies in the model. The above discussion ignores nuisance parameters. In our case, we need to regard β as a nuisance parameter. As $p(\tilde{\alpha}_i|y_i) \propto p(\tilde{\alpha}_i) \int p(y_i|\beta, \tilde{\alpha}_i)p(\beta|\tilde{\alpha}_i)\, d\beta$ and β is not estimable from individual unit i, Marshall and Spiegelhalter (2007)[p. 420] recommend generating $\tilde{\alpha}_i^{\text{fix}}$ from $f(\alpha_i|y)$ where

$$f(\alpha_i|y) \propto p(\tilde{\alpha}_i) \int p(y_i|\tilde{\alpha}_i, \beta)p(\beta|y_{-i})\, d\beta.$$

Note that the two replications $\tilde{\alpha}_i^{\text{rep}}$ and $\tilde{\alpha}_i^{\text{fix}}$ are no longer entirely independent as y_{-i} will slightly influence $\tilde{\alpha}_i^{\text{fix}}$ through β. To compare the prior and likelihood replicates, Marshall and Spiegelhalter (2007) considered $\tilde{\alpha}_i^{\text{diff}} = \tilde{\alpha}_i^{\text{rep}} - \tilde{\alpha}_i^{\text{fix}}$ and calculated a conflict p-value

$$p_{i,\text{con}}^L = P(\tilde{\alpha}_i^{\text{diff}} \leq 0|y)$$

as the proportion of times simulated values of $\tilde{\alpha}_i^{\text{diff}}$ are less than or equal to zero for scalar $\tilde{\alpha}_i$. Depending on the context, the upper tail area $p_{i,\text{con}}^U = 1 - p_{i,\text{con}}^L$ or the 2-sided p-value $2 \times \min(p_{i,\text{con}}^L, p_{i,\text{con}}^U)$ may be of interest instead. If $\tilde{\alpha}_i^{\text{diff}}$ is not a scalar, $\text{E}(\tilde{\alpha}_i^{\text{diff}}|y)^T\text{Cov}(\tilde{\alpha}_i^{\text{diff}}|y)^{-1}\text{E}(\tilde{\alpha}_i^{\text{diff}}|y)$ can be used as a standardized discrepancy measure. An alternative to this cross-validatory approach is to simulate $\tilde{\alpha}_i^{\text{rep}}|\beta^{\text{rep}}, D^{\text{rep}}$ using β^{rep}, D^{rep} generated from $p(\beta, D|y)$ without leaving out y_i. This introduces only mild conservatism as y_i influences $\tilde{\alpha}_i^{\text{rep}}$ through β and D (Marshall and Spiegelhalter, 2007).

From (4.14), the nonconjugate variational message passing update for $\lambda_{\tilde{\alpha}_i}$ is given by

$$\begin{aligned}
&\mathcal{V}_{\tilde{\alpha}_i}(\lambda_{\tilde{\alpha}_i})^{-1}\left(\frac{\partial S_{\tilde{\alpha}_i}}{\partial \lambda_{\tilde{\alpha}_i}} + \frac{\partial S_{y_i}}{\partial \lambda_{\tilde{\alpha}_i}}\right) \\
&\quad = \begin{bmatrix} -\frac{\nu^q}{2} D_r^T \text{vec}(S^{q-1}) \\ \nu^q S^{q-1} \tilde{W}_i \mu_\beta^q \end{bmatrix} + \begin{bmatrix} -\frac{1}{2} D_r^T \text{vec}(\sum_{j=1}^{n_i} F_{ij} X_{ij}^R X_{ij}^{R^T}) \\ (\sum_{j=1}^{n_i} F_{ij} X_{ij}^R X_{ij}^{R^T}) \mu_{\tilde{\alpha}_i}^q + X_i^{R^T}(y_i - G_i) \end{bmatrix}.
\end{aligned}$$

The first term can be considered as a message from the prior $p(\tilde{\alpha}_i|\beta, D)$ and the second term a message from the likelihood of unit y_i, $p(y_i|\tilde{\alpha}_i, \beta)$. We argue below that the first message from the prior can be interpreted as natural parameter of a Gaussian approximation to $p(\tilde{\alpha}_i|y_{-i})$ while the second message from the likelihood can be interpreted as natural parameter of a Gaussian approximation to $f(\tilde{\alpha}_i|y)$. Let $\Sigma_{\text{lik}} = (\sum_{j=1}^{n_i} F_{ij} X_{ij}^R X_{ij}^{R^T})^{-1}$ and $\mu_{\text{lik}} = \mu_{\tilde{\alpha}_i}^q + \Sigma_{\text{lik}} X_i^{R^T}(y_i - G_i)$. This would imply that $\tilde{\alpha}_i^{\text{rep}} \sim N(\tilde{W}_i \mu_\beta^q, \frac{1}{\nu^q} S^q)$ and $\tilde{\alpha}_i^{\text{fix}} \sim N(\mu_{\text{lik}}, \Sigma_{\text{lik}})$ so that $\tilde{\alpha}_i^{\text{diff}} \sim N(\tilde{W}_i \mu_\beta^q - \mu_{\text{lik}}, \frac{1}{\nu^q} S^q + \Sigma_{\text{lik}})$, assuming $\tilde{\alpha}_i^{\text{rep}}$ and $\tilde{\alpha}_i^{\text{fix}}$ are considered independent. Since these messages are computed in the nonconjugate variational message passing algorithm, conflict p-values can be calculated easily at convergence for identification of divergent units.

For moderate to large data sets, the difference between $p(\beta, D|y_{-i})$ and $p(\beta, D|y)$ is small and we approximate $p(\beta, D|y_{-i})$ in (5.6) by the varia-

tional posterior $q(\beta)q(D)$. This combined with Jensen's inequality gives

$$\begin{aligned}\log p(\tilde{\alpha}_i|y_{-i}) &\approx \log E_{-\tilde{\alpha}_i}\{p(\tilde{\alpha}_i|\beta, D)\} \\ &\geq E_{-\tilde{\alpha}_i}\{\log p(\tilde{\alpha}_i|\beta, D)\}.\end{aligned}$$

Approximating $p(\tilde{\alpha}_i|y_{-i})$ by $\exp[E_{-\tilde{\alpha}_i}\{\log p(\tilde{\alpha}_i|\beta, D)\}]$, we then have $\tilde{\alpha}_i^{\text{rep}} \sim N(\tilde{W}_i\mu_\beta^q, \frac{1}{\nu^q}S^q)$. On the other hand, the total message gives us the natural parameter of $q(\tilde{\alpha}_i)$ which is an approximation of $p(\tilde{\alpha}_i|y)$. If we think of $p(\tilde{\alpha}_i|y_{-i})$ as the "prior" to be updated when y_i becomes available, we have

$$p(\tilde{\alpha}_i|y) \propto p(\tilde{\alpha}_i|y_{-i})p(y_i|\tilde{\alpha}_i, y_{-i}),$$

which implies that

$$\frac{p(\tilde{\alpha}_i|y)}{p(\tilde{\alpha}_i|y_{-i})} \propto p(y_i|\tilde{\alpha}_i, y_{-i}).$$

Interpreting the first message as a Gaussian approximation to $p(\tilde{\alpha}_i|y_{-i})$ and the sum of the two messages as a Gaussian approximation to $p(\tilde{\alpha}_i|y)$, the ratio of these two normal distributions gives an approximation (up to a proportionality constant) of $p(y_i|\tilde{\alpha}_i, y_{-i})$. As a function of $\tilde{\alpha}_i$, the ratio of the two normal distributions is proportional to

$$\frac{\exp\{-\frac{1}{2}(\tilde{\alpha}_i - \mu_{\tilde{\alpha}_i}^q)^T {\Sigma_{\tilde{\alpha}_i}^q}^{-1}(\tilde{\alpha}_i - \mu_{\tilde{\alpha}_i}^q)\}}{\exp\{-\frac{1}{2}(\tilde{\alpha}_i - \tilde{W}_i\mu_\beta^q)^T\nu^q S^{q-1}(\tilde{\alpha}_i - \tilde{W}_i\mu_\beta^q)\}},$$

which gives a normal distribution with natural parameters

$$\begin{bmatrix} -\frac{1}{2}D_r^T\text{vec}({\Sigma_{\tilde{\alpha}_i}^q}^{-1} - \nu^q S^{q-1}) \\ {\Sigma_{\tilde{\alpha}_i}^q}^{-1}\mu_{\tilde{\alpha}_i}^q - \nu^q S^{q-1}\tilde{W}_i\mu_\beta^q \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}D_r^T\text{vec}(\Sigma_{\text{lik}}^{-1}) \\ \Sigma_{\text{lik}}^{-1}\mu_{\text{lik}} \end{bmatrix},$$

precisely that given by the second message. As

$$p(y_i|\tilde{\alpha}_i, y_{-i}) = \int p(y_i|\beta, \tilde{\alpha}_i)p(\beta|\tilde{\alpha}_i, y_{-i})\, d\beta$$

and $p(\beta|\tilde{\alpha}_i, y_{-i})$ is close to $p(\beta|y_{-i})$ when the number of clusters is large, the second message can be considered as giving the natural parameter of a Gaussian approximation to $f(\tilde{\alpha}_i|y)$ if we assume a uniform prior for $p(\tilde{\alpha}_i)$. Finally, even though $\tilde{\alpha}_i^{\text{rep}}$ and $\tilde{\alpha}_i^{\text{fix}}$ are not entirely independent, for large data sets, the dependence between $\tilde{\alpha}_i^{\text{rep}}$ and $\tilde{\alpha}_i^{\text{fix}}$ will be increasingly weak as the number of clusters increases.

For large data sets, automatic computation of diagnostics for prior-likelihood conflict can be an attractive alternative to the simulation-based

approach using MCMC methods. While the approximations made in our derivation are crude, the diagnostics can be computed automatically in the nonconjugate variational message passing algorithm and is a handy screening tool. Clusters flagged as divergent can be studied more closely and possibly conflict p-values recomputed by Monte Carlo. The arguments above generalize to detecting conflict for other parameters of the model also.

5.4 Examples

In Section 5.4.1, we use the Bristol infirmary inquiry data to compare the conflict p-values computed using the nonconjugate variational message passing algorithm with those obtained using the cross-validatory approach of Marshall and Spiegelhalter (2007). In Sections 5.4.2 and 5.4.3, we apply the stochastic version of nonconjugate variational message passing to a real data set and a simulated data set respectively, in the initial stage of optimization before switching to the standard version. In all the examples, the partially noncentered parametrization was used and we consider a $N(0, 1000)$ prior for β. We also experimented with various settings of K and γ. The Muscatine coronary risk factor study data set and the skin cancer prevention study data set can be found at `http://www.biostat.harvard.edu/~fitzmaur/ala2e/`. All code was written in the R language and run on a dual processor Windows PC 3.30 GHz workstation.

5.4.1 Bristol infirmary inquiry data

In 1998, a public inquiry was set up to look into the management of children receiving complex cardiac surgical services at the Bristol Royal Infirmary from 1984 to 1995. The outcomes of paediatric cardiac surgical services at Bristol, UK, relative to other specialist centres was a key issue. We consider a subset of the data presented to the Inquiry recorded by Hospital Episode Statistics on the mortality rates in open surgeries for 12 hospitals including Bristol (hospital 1), for children under 1 year old, from 1991 to 1995. This data can be found in Marshall and Spiegelhalter (2007) Table 1. Spiegelhalter *et al.* (2002a) and Marshall and Spiegelhalter (2007) modelled this data using a logistic GLMM. Although the number of clusters is small in this example whereas our methodology is motivated by applications to large data sets, this example is interesting as a benchmark data set in

the literature for calculating prior-likelihood conflict diagnostics from the nonconjugate variational message passing algorithm.

Let $Y_i = \sum_{j=1}^{n_i} y_{ij}$ represent the number of deaths at hospital i, $i = 1, \ldots, 12$. We have $y_{ij} \sim \text{Bernoulli}(\pi_i)$ where $y_{ij} = 1$ if patient j at hospital i died and 0 otherwise. Let

$$\text{logit}(\pi_i) = \beta + u_i \quad \text{where} \quad u_i \sim N(0, D).$$

To assess the accuracy of the approximate conflict p-values obtained from the standard nonconjugate variational message passing algorithm, we use the cross-validatory conflict p-values obtained using the simulation-based approach of Marshall and Spiegelhalter (2007) as a "gold-standard" and compute these for comparison. In the cross-validatory approach, each hospital i is removed in turn from the analysis, and the parameters $\beta^{\text{rep}}, D^{\text{rep}}|y_{-i}$ are generated using MCMC followed by a simulated $\pi_i^{\text{rep}}|\beta^{\text{rep}}, D^{\text{rep}}$. Assuming a Jeffrey's prior for π_i, a π_i^{fix} is then simulated from $\text{Beta}(Y_i + 0.5, n_i - Y_i + 0.5)$. Excess mortality is of concern and the upper-tail area is used as a 1-sided p-value so that $p_{i,\text{con}} = P(\pi_i^{\text{rep}} \geq \pi_i^{\text{fix}})$. 100 000 simulations were used in calculating the cross-validatory conflict p-values. Fitting via MCMC was performed in WinBUGS (Lunn *et al.*, 2000) through `R` by using `R2WinBUGS` (Sturtz *et al.*, 2005) as an interface. Two chains were run simultaneously to assess convergence, each with 51,000 iterations, and the first 1000 iterations were discarded in each chain as burn-in. The MCMC algorithm was initialized using the fit from penalized quasi-likelihood and the same priors were used in MCMC and nonconjugate variational message passing. The total time taken for updating in WinBUGS is 372 seconds while non-conjugate variational message passing took 6 seconds in CPU time. There are some difficulties in comparing nonconjugate variational message passing and MCMC in this way as the time taken for the variational algorithm to converge depends on the initialization, stopping rule and the rate of convergence is problem-dependent. The updating time for MCMC is also problem-dependent and depends on the length of burn-in and number of sampling iterations.

The cross-validatory conflict p-values computed using MCMC ($p_{i,\text{con}}^{\text{CV}}$) and conflict p-values estimated using nonconjugate variational message passing ($p_{i,\text{con}}^{\text{NCVMP}}$) for all hospitals are shown in Figure 5.1. The plot in Figure 5.1 indicates very good agreement between the two sets of p-values. To reflect the importance of good agreement at the extremes, Marshall and

hospital	$p_{i,\text{con}}^{\text{CV}}$	$p_{i,\text{con}}^{\text{NCVMP}}$
1	0.001	0.004
2	0.438	0.448
3	0.935	0.934
4	0.126	0.131
5	0.297	0.305
6	0.721	0.730
7	0.737	0.751
8	0.663	0.672
9	0.437	0.450
10	0.375	0.386
11	0.764	0.771
12	0.718	0.733

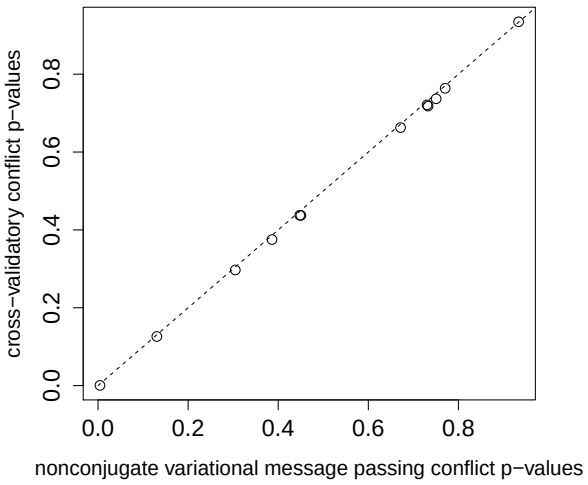

Figure 5.1: Bristol infirmary inquiry data. Cross-validatory conflict p-values ($p_{i,\text{con}}^{\text{CV}}$) and approximate conflict p-values from nonconjugate variational message passing ($p_{i,\text{con}}^{\text{NCVMP}}$).

Spiegelhalter (2007) computed the relative agreement between p-values as

$$\left| \frac{\Phi^{-1}(p_{i,\text{con}}^{\text{CV}}) - \Phi^{-1}(p_{i,\text{con}}^{\text{NCVMP}})}{\Phi^{-1}(p_{i,\text{con}}^{\text{NCVMP}})} \right| \times 100\%,$$

where Φ^{-1} denotes the inverse cumulative distribution function of the standard normal. The relative error between $p_{i,\text{con}}^{\text{CV}}$ and $p_{i,\text{con}}^{\text{NCVMP}}$ is 9% which is close to the relative error of 7% between cross-validatory and full data conflict p-values reported in Marshall and Spiegelhalter (2007). For moderate to large data sets, the variational message passing approach will be an extremely attractive alternative to computationally intensive MCMC methods for obtaining prior-likelihood conflict diagnostics.

5.4.2 Muscatine coronary risk factor study

A total of 4856 children took part in the Muscatine coronary risk factor study (Woolson and Clarke, 1984), which was undertaken to examine the development and persistence of risk factors for coronary disease in children. Over the period 1977–1981, weight and height data were collected biennially from five cohorts of children, aged 5–7, 7–9, 9–11, 11–13 and 13–15 at the beginning of the study. The data is incomplete with less than 40% of the children surveyed on all three occasions. In previous analyses, some authors treated this data as potentially missing not at random (e.g. Zhou *et al.*, 2010) while others assumed the data are missing at random (Fitzmau-

rice *et al.*, 1994; Kenward and Molenberghs, 1998). We assume the data are missing at random and focus on computational comparisons between standard and stochastic nonconjugate variational message passing. The binary response, y_{ij}, is an indicator of whether the ith child is obese at the jth occasion. For the ith child, we consider the covariates, $\text{gender}_i = 1$ if female, 0 if male and age_{ij} = midpoint of age cohort at jth occasion -12. Fitzmaurice *et al.* (2004) modelled the marginal probability of obesity as a logistic function of gender and linear and quadratic age. We consider the following logistic random intercept model,

$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_1 \text{gender}_i + \beta_2 \text{age}_{ij} + \beta_3 \text{age}_{ij}^2 + u_i,$$

where $u_i \sim N(0, \sigma^2)$ for $i = 1, \ldots, 4856$, $1 \leq j \leq 3$. The standard nonconjugate variational message passing algorithm took 345 seconds to converge for this moderately large data set. The performance of stochastic nonconjugate variational message passing was investigated using different mini-batch sizes and various parameter settings for the step sizes. We considered $|S| \in \{1, 50, 99, 242\}$ where the mini-batch sizes were chosen to correspond to the online setting and approximately 1%, 2% and 5% of $n = 4856$. We let the stability constant K take values 0, 1 and 5 and γ be 0.5, 0.75 or 1. In the online setting $|S| = 1$, we considered larger stability constants, $K \in \{250, 500, 1000\}$. For each mini-batch size and parameter setting for the step-size, we perform five runs of the stochastic nonconjugate variational message passing switching to the standard version each time the relative increment in the lower bound after a complete sweep through the data is less than 10^{-3}. The average time taken for the algorithm to converge in each case is shown in Figure 5.2. The solid lines, dashed lines and dot-dashed lines correspond to $\gamma = 1$, 0.75 and 0.5 respectively. The best parameter settings and average time to convergence for each mini-batch size are summarized in Table 5.1.

From these results, we observed that as the mini-batch size increases, smaller values of γ and K, that is, a slower rate of decrease in step-size and

$\lvert S\rvert$	1	50	99	242
K	250	1	0	0
γ	1	1	0.75	0.5
time	233	133	116	149

Table 5.1: Coronary risk factor study. Best parameter settings and average time to convergence (in seconds) for different mini-batch sizes.

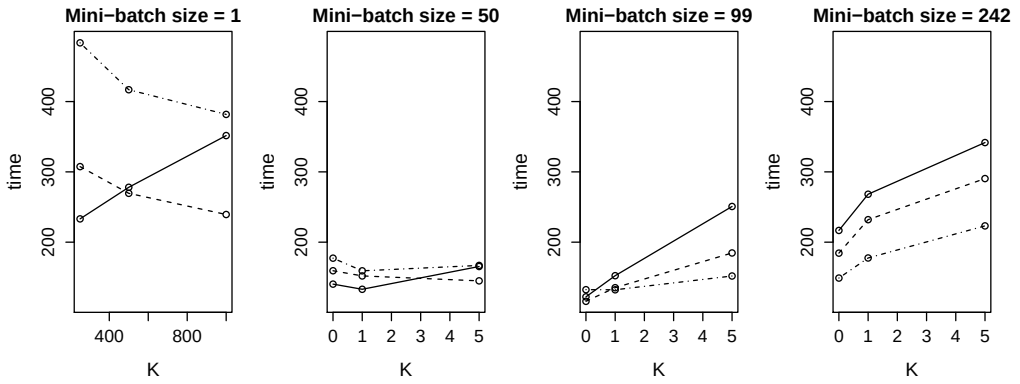

Figure 5.2: Coronary risk factor study. Plot of average time to convergence against the stability constant K for different mini-batch sizes. The solid, dashed and dot-dashed lines correspond to $\gamma = 1$, 0.75 and 0.5 respectively.

larger step-sizes lead to faster convergence. However, a significantly larger stability constant and smaller step sizes are required in the online setting to prevent unstable behaviour in the early iterations. The mini-batch size of 50 (approximately 1% of n) performed well across a wide range of step-sizes with the average time to convergence ranging from 133 to 167 seconds. The shortest average time to convergence is 116 seconds for the mini-batch of size 99 with $K = 0$ and $\gamma = 0.75$. This is a third of the computation time required to perform standard nonconjugate variational message passing. Figure 5.3 tracks the average lower bound attained at the end of each sweep through the data for the different batch sizes corresponding to the best parameter settings listed in Table 5.1. Only the first ten sweeps are shown. This figure shows that with appropriately chosen step-sizes, the stochastic version of nonconjugate variational message passing is able to make much bigger gains than the standard version particularly in the first few sweeps. Thus, even for moderate-sized data sets, significant gains can be made by making use of stochastic nonconjugate variational message passing in the initial stage of optimization.

5.4.3 Skin cancer prevention study

In a clinical trial conducted to test the effectiveness of beta-carotene in preventing non-melanoma skin cancer (Greenberg *et al.*, 1989), 1805 high risk patients were randomly assigned to receive either a placebo or 50 mg of beta-carotene per day for five years. Subjects were biopsied once a year to ascertain the number of new skin cancers since the last examination. The response y_{ij} is a count of the number of new skin cancers in year j for the ith subject. Covariate information for the ith subject include age_i, the age

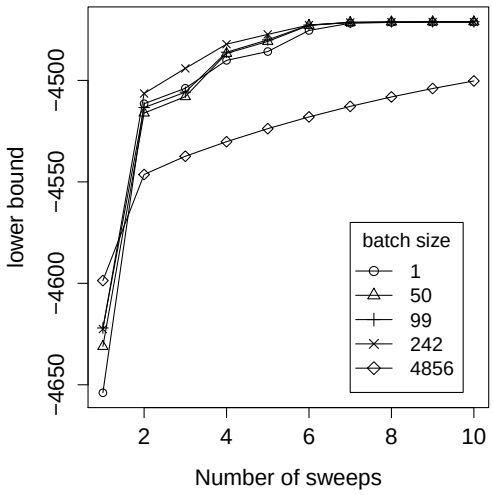

Figure 5.3: Coronary risk factor study. Plot of average lower bound against number of sweeps through entire data set for different batch sizes under the best parameter settings.

in years at the beginning of the study, $\text{gender}_i = 1$ if male and 0 if female, exposure_i, a count of the number of previous skin cancers, $\text{skin}_i = 1$ if skin has burns and 0 otherwise, $\text{treatment}_i = 1$ if the ith subject receives beta-carotene and 0 if placebo and year_{ij}, the year of follow-up. We consider $n = 1683$ subjects with complete covariate information. Using conditional Akaike information to perform model selection, Donohue *et al.* (2011) fitted different Poisson GLMMs to this data and arrived at the model

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \text{age}_i + \beta_2 \text{skin}_i + \beta_3 \text{gender}_i + \beta_4 \text{exposure}_i + u_i,$$

where $u_i \sim N(0, \sigma^2)$ for $i = 1, \ldots, 1683$, $1 \leq j \leq 5$. The treatment and year effects did not prove to be significant in their analyses. Using this model, we investigate the performance of standard and stochastic nonconjugate variational message passing algorithms. As this data set is small, preliminary investigation shows that the time to convergence of the standard and stochastic nonconjugate variational message passing algorithms are close and stochastic nonconjugate variational message passing did not provide significant gains over the standard version. We thus simulated a data set comprising of $n = 1683 \times 6 = 10098$ subjects by using the posterior means of the unknown parameters from the standard nonconjugate variational message passing fit to the original data set. Thus, we replicate the design matrices for each cluster 6 times. For this simulated data, standard nonconjugate variational message passing took 118 seconds to converge.

We considered mini-batch sizes corresponding to the online setting and

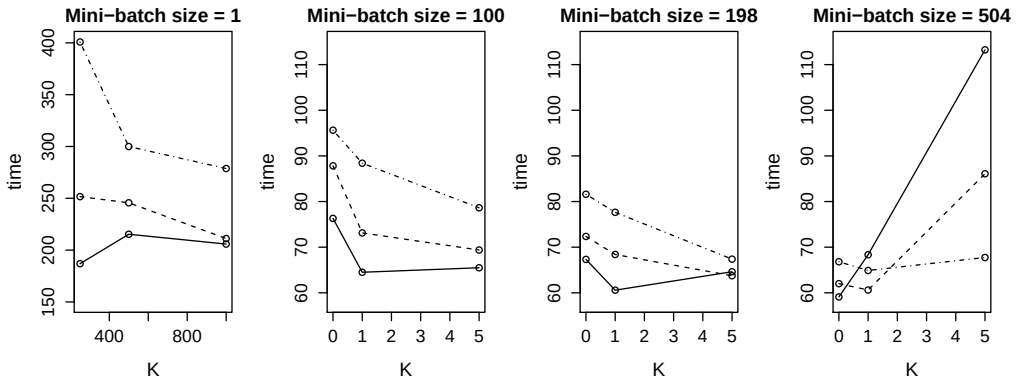

Figure 5.4: Skin cancer study. Plot of average time to convergence against the stability constant K for different mini-batch sizes. The solid, dashed and dot-dashed lines correspond to $\gamma = 1, 0.75$ and 0.5 respectively.

approximately 1%, 2% and 5% of $n = 10098$, that is, $|S| \in \{1, 100, 198, 504\}$. We let γ be 0.5, 0.75 or 1 and the stability constant K take values 0, 1 and 5 for $|S| \in \{100, 198, 504\}$ and values 250, 500, 1000 for $|S| = 1$. For each mini-batch size and parameter setting for the step-size, we did five runs of the stochastic nonconjugate variational message passing, switching to standard nonconjugate variational message passing each time the relative increment in the lower bound after a complete sweep through the data is less than 10^{-3}. The average time taken for the algorithm to converge in each case is shown in Figure 5.4. The solid lines, dashed lines and dot-dashed lines correspond to $\gamma = 1, 0.75$ and 0.5 respectively. The best parameter settings and average time to convergence for each mini-batch size are summarized in Table 5.2.

As in the example in Section 5.4.2, larger stability constants are preferred when $|S| = 1$. For this simulated data, a higher rate of decrease in step-size is desirable with $\gamma = 1$ yielding the best performance across different mini-batch sizes. Larger batch sizes also seem to lead to faster convergence. Figure 5.5 compares the rate of convergence of standard and stochastic nonconjugate variational message passing for one of the runs where $|S| = 504$, $K = 0$ and $\gamma = 1$. The variational lower bound $\mathcal{L}$ is -23617.3 at convergence and we have plotted $\log(-23617 - \mathcal{L})$ against time.

$\lvert S\rvert$	1	100	198	504
K	250	1	1	0
γ	1	1	1	1
time	187	65	61	59

Table 5.2: Skin cancer study. Best parameter settings and average time to convergence (in seconds) for different mini-batch sizes.

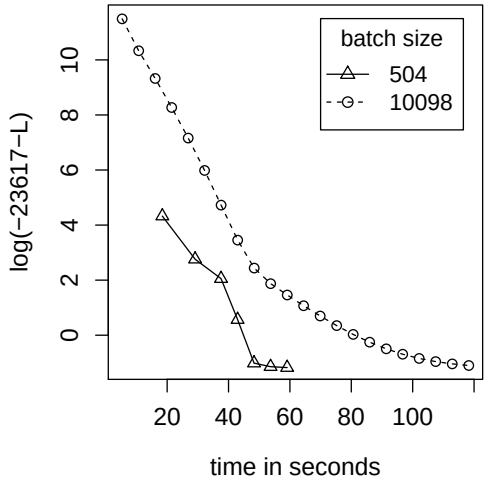

Figure 5.5: Plot of $\log(-23617 - \mathcal{L})$ against time for the mini-batch of size 504, $K = 0$ and $\gamma = 1$.

Stochastic nonconjugate variational message passing took just 7 sweeps to converge in 59 seconds while the standard version took 22 sweeps and converged in 118 seconds. This represents a reduction in computation time by a factor of 2.

5.5 Conclusion

In this chapter, we have extended stochastic variational inference to nonconjugate models and derived a stochastic version of the nonconjugate variational message passing algorithm, scalable to large data sets. The data sets that we have considered were only of moderate size. Nevertheless, by applying the stochastic version of the nonconjugate variational message passing algorithm in the first few iterations, the time to convergence for these data sets can be reduced by half or more. The stochastic version seems computationally preferable once the number of clusters is more than several thousand. We would imagine the gain to be bigger for larger data sets and more work remains to be done in that aspect. Experimentation with various settings of K and γ suggest that γ close to 1 and a large stability constant K is preferred in the online setting while mini-batches larger in size perform better with larger step-sizes. Comparison of the conflict p-values obtained from the nonconjugate variational message passing algorithm with those computed using the approach of Marshall and Spiegelhalter (2007) suggest very good agreement. For large data sets, the variational message passing approach will be an extremely attractive alternative to computationally intensive MCMC methods in obtaining prior-likelihood diagnostics.

Chapter 6

Conclusions and future work

This thesis has developed fast variational algorithms for the fitting of some very flexible models, namely, the MHR model, the MLMM and the GLMM. In the case of the MHR model and the GLMM, the advantages of using variational approximation methods as compared to MCMC methods are illustrated in model fitting and model choice. We show that variational approximation provides good point estimates and excellent predictive inference with computation time reduced by as much as an order of magnitude.

The MHR model extends mixture of regression models by allowing the mixture components to be heteroscedastic. However, the variance parameters in the model cannot be optimized in closed form and we have developed an approximate method for dealing with these parameters that is computationally efficient. For the MLMM, we have developed a variational greedy algorithm which is fully automated and capable of performing parameter estimation and model selection at the same time. This greedy approach avoids some of the difficulties associated with the EM algorithm such as dependency on initialization and overfitting. The nonconjugate variational message passing algorithm (Knowles and Minka, 2011) extends variational message passing to nonconjugate models and has greatly expanded the scope of models which can be fitted using VB. Closed form updates are now possible even for models without conjugate priors, such as the Poisson GLMM. We have extended the applications of nonconjugate variational message passing to the multivariate case and demonstrated how it can be used to fit Poisson and logistic models with very good results.

We have shown empirically that reparametrization of the MLMM using hierarchical centering, in cases where there is weak identifiability of certain model parameters, can lead to improved convergence in the variational algorithm, both in terms of reduced computation time as well as better clustering results. Some theoretical support was provided for this

observation. In addition, we have investigated the performance of different parametrizations such as the centered, noncentered and partially noncentered parametrizations in the context of variational approximations for GLMMs. Partially noncentered parametrizations were found to be able to adapt to the quantity of information in the data and determine automatically a parametrization close to optimal. Very often, partial noncentering was also able to accelerate convergence and produce more accurate posterior approximations than centering or noncentering. These favourable properties suggest using the partially noncentered parametrization as the default parametrization since it is not possible to tell in advance which of centering or noncentering performs better without using both.

Finally, we have explored how stochastic approximation can be combined with variational methods to improve the accuracy of the posterior approximations or to make variational inference a viable approach for large data sets. For the MHR model, we have proposed using stochastic gradient approximation to optimize the variational lower bound after first integrating out the latent mixture components indicators. An improved gradient estimate was proposed and the idea of perturbing existing means and variances helped to keep the optimization low-dimensional. The idea of stochastic gradient approximation was revisited when we developed the stochastic version of nonconjugate variational message passing. By using unbiased gradient estimates computed from mini-batches of data, the variational lower bound can be optimized as a function of the global variables using stochastic gradient approximation, provided the local variational parameters have been optimized as a function of these global parameters. This idea allows nonconjugate variational message passing to be applied to very large data sets as data can now be processed in mini-batches. While we have only applied this methodology to data sets of moderate sizes, the results are encouraging, suggesting that greater gains in computational efficiency can be expected for larger data sets.

We discuss below some possible extensions of our work and future research directions.

Partially noncentered parametrizations. The amount of centering is controlled by the tuning matrix W_i. While we have attempted to infer the form of W_i from the simple linear mixed model, it might be helpful to investigate in greater depth how W_i can be specified for optimal performance as well as to perform some analysis about its properties. The parameter expanded VB method of Qi and Jaakkola (2006) is in some ways very sim-

ilar to partially noncentered parametrizations and a deeper understanding of the relationship between these two methods might generate new ideas in speeding up variational algorithms. Papaspiliopoulos *et al.* (2007) discussed reparametrization techniques for constructing effective MCMC algorithms for a wide range of models such as spatial GLMMs, diffusion stochastic volatility models and hidden Markov models. It would be interesting to investigate the performance of partially noncentered parametrizations for such models in the context of variational approximations.

Nonconjugate variational message passing. For the MHR model, we have developed an approximate method for dealing with the variance parameters in the model which cannot be optimized in closed form. Since a normal distribution has been assumed for the variance parameters, it might be possible to optimize these parameters using nonconjugate variational message passing. We have demonstrated that nonconjugate variational message passing is a type of natural gradient method and the combination of stochastic gradient approximation with nonconjugate variational message passing opens up many possibilities. Further study on the optimization of the step size sequence or the development of adaptive step size sequence may be helpful in bringing about greater speed ups in the algorithm.

Stochastic approximations. Recent development in VB methodology have branched out to stochastic optimization which has enabled limitations in VB such as the reliance on analytic solutions to integrals and conjugacy in the posterior to be overcome, making VB a viable approach for handling large data sets (e.g. Hoffman *et al.*, 2013). The ideas developed here can be extended to other models. For the logistic mixed model, there remains significant underestimation of the random effects standard deviation when it is fitted using nonconjugate variational message passing. This could be due to the assumed factorized posterior. Salimans and Knowles (2012) discussed how such independence assumptions can be relaxed using stochastic approximation as well as the use of mixture of standard distributions as the approximating variational marginal posteriors. Application of these methods to logistic mixed models might help to improve the approximations of the posterior distributions of the random effects standard deviations. It would also be interesting to explore using stochastic approximation methods to construct online VB algorithms in applications where model estimation needs to be performed as data accumulates, for instance, in the modelling of infectious diseases where control strategies need to adapt quickly to the progress of an epidemic Jewell *et al.* (2009).

Bibliography

Ahn, S., Korattikara, A. and Welling, M. (2012). Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning* (eds. J. Langford and J. Pineau), 1591–1598. Omnipress, Madison, WI.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10, 251–276.

Andrieu, C. and Thoms, A. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18, 343–373.

Armagan, A. and Dunson, D. (2011). Sparse variational analysis of linear mixed models for large data sets. *Statistics and Probability Letters*, 81, 1056–1062.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M. and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25, 25–29.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence* (eds. K. Laskey and H. Prade), 21–30. Morgan Kaufmann, San Francisco, CA.

—— (2000). A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems 12* (eds. S. A. Solla, T. K. Leen and K.-R. Müller), 209–215. MIT Press, Cambridge, MA.

Biernacki, C., Celeux, G. and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate

Gaussian mixture models. *Computational Statistics and Data Analysis*, 41, 561–575.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer, New York.

Bishop, C. M. and Svensén, M. (2003). Bayesian hierarchical mixtures of experts. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence* (eds. C. Meek and U. Kjærulff), 57–64. Morgan Kaufmann, San Francisco, CA.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1, 121–144.

Blocker, A. W. (2011). Fast Rcpp implementation of Gauss-Hermite quadrature. R package "fastGHQuad" version 0.1-1. Available at `http://cran.r-project.org/`.

Booth, J. G., Casella, G. and Hobert, J. P. (2008). Clustering using objective functions and stochastic search. *Journal of the Royal Statistical Society: Series B*, 70, 119–139.

Booth, J. G. and Hobert, J. P. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society: Series B*, 61, 265–285.

Bottou, L. and Bousquet, O. (2008). The trade-offs of large scale learning. In *Advances in Neural Information Processing Systems 20* (eds. J.C. Platt, D. Koller, Y. Singer and S. Roweis), 161–168. Neural Information Processing Systems, La Jolla, CA.

Bottou, L. and Cun, Y. L. (2005). On-line learning for very large data sets. *Applied stochastic models in business and industry*, 21, 137–151.

Boughton, W. (2004). The Australian water balance model. *Environmental Modelling and Software*, 19, 943–956.

Braun, M. and McAuliffe, J. (2010). Variational inference for large-scale models of discrete choice. *Journal of the American Statistical Association*, 105, 324–335.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88, 9–25.

Brown, P. and Zhou, L. (2010). MCMC for generalized linear mixed models with glmmBUGS. *The R Journal*, 2, 13–16.

Browne, W. J. and Draper, D. (2006). A comparison of Bayesian and likelihood-based methods for fitting multilevel models. *Bayesian Analysis*, 1, 473–550.

Cai, B. and Dunson, D. B. (2008). Bayesian variable selection in generalized linear mixed models. *Random Effect and Latent Variable Model Selection* (eds. D. B. Dunson), 192, 63–91. Springer, New York.

Celeux, G., Martin O. and Lavergne C. (2005). Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments. *Statistical Modelling*, 5, 243–267.

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

Christensen, O. F., Roberts, G. O. and Sköld, M. (2006). Robust Markov chain Monte Carlo methods for spatial generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 15, 1–17.

Coke, G. and Tsao, M. (2010). Random effects mixture models for clustering electrical load series. *Journal of Time Series Analysis*, 31, 451–464.

Constantinopoulos, C. and Likas, A. (2007). Unsupervised learning of Gaussian mixtures based on variational component splitting. *IEEE Transactions on Neural Networks*, 18, 745–755.

Corduneanu, A., and Bishop, C. M. (2001). Variational Bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics 2001* (eds. T. Jaakkola and T. Richardson), 27–34, Morgan Kaufmann, San Francisco, CA.

De Backer, M., De Vroey, C., Lesaffre, E., Scheys, I., and De Keyser, P. (1998). Twelve weeks of continuous oral therapy for toenail onychomycosis caused by dermatophytes: A double-blind comparative trial of terbinafine 250 mg/day versus itraconazole 200 mg/day. *Journal of the American Academy of Dermatology*, 38, 57–63.

De Freitas, N., Højen-Sørensen, P., Jordan, M. I. and Russell, S. (2001). Variational MCMC. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (eds. J. Breese and D. Koller), 120–127. Morgan Kaufmann, San Francisco, CA.

De Iorio, M., Müller, P., Rosner, G. L. and MacEachern, S. N. (2004). An ANOVA model for dependent random measures. *Journal of the American Statistical Association*, 99, 205–215.

Delyon, B. and Juditsky, A. (1993). Accelerated stochastic approximation. *SIAM Journal on Optimization*, 3, 868–881.

Donohue, M. C., Overholser, R., Xu, R. and Vaida, F. (2011). Conditional Akaike information under generalized linear and proportional hazards mixed models. *Biometrika*, 98, 685–700.

Dunson, D. B., Pillai, N. and Park, J.-H. (2007). Bayesian density regression. *Journal of the Royal Statistical Society: Series B*, 69, 163–183.

Evans, M. and Moshonov, H. (2006). Checking for prior-data conflict. *Bayesian Analysis*, 4, 893–914.

Faes, C., Ormerod, J. T. and Wand, M. P. (2011). Variational Bayesian inference for parametric and nonparametric regression with missing data. *Journal of the American Statistical Association*, 106, 959–971.

Finley, A. O., Banerjee, S. and McRoberts, R. E. (2008). A Bayesian approach to multi-source forest area estimation. *Environmental and Ecological Statistics*, 15, 241–258.

Fitzmaurice, G. M., Laird, N. M. and Lipsitz, S. R. (1994). Analysing incomplete longitudinal binary responses: a likelihood-based approach. *Biometrics*, 50, 601–612.

Fitzmaurice, G. M., Laird, N. M. and Ware, J. H. (2004). *Applied Longitudinal Analysis*. Wiley, New Jersey.

Fong, Y., Rue, H. and Wakefield, J. (2010). Bayesian inference for generalised linear mixed models. *Biostatistics*, 11, 397–412.

Frühwirth-Schnatter, S. (2004). Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques. *The Econometrics Journal*, 7, 143–167.

Gelfand, A. E., Sahu, S. K. and Carlin, B. P. (1995). Efficient parametrisations for normal linear mixed models. *Biometrika*, 82, 479–488.

—— (1996). Efficient parametrizations for generalized linear mixed models. In *Bayesian Statistics 5* (eds. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith), 165–180. Clarendon Press, Oxford.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2004). *Bayesian Data Analysis*, 2nd ed. Chapman and Hall/CRC, Boca Raton, FL.

Geweke, J. and Amisano, G. (2010). Comparing and evaluating Bayesian predictive distributions of asset returns. *International Journal of Forecasting*, 26, 216–230.

Geweke, J. and Keane, M. (2007). Smoothly mixing regressions. *Journal of Econometrics*, 138, 252–291.

Ghahramani, Z. and Beal, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems 12* (eds. S. A. Solla, T. K. Leen and K.-R. Müller), 449–455. MIT Press, Cambridge, MA.

—— (2001). Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems 13* (eds. T. K. Leen, T. G. Dietterich and V. Tresp), 507–513. MIT Press, Cambridge, MA.

Greenberg, E. R., Baron, J. A., Stevens, M. M., Stukel, T. A., Mandel, J. S., Spencer, S. K., Elias, P. M., Lowe, N., Nierenberg, D. N., Bayrd G. and Vance, J. C. (1989). The skin cancer prevention study: design of a clinical trial of beta-carotene among persons at high risk for nonmelanoma skin cancer. *Controlled Clinical Trials*, 10, 153–166.

Griffin, J. E. and Steel, M. F. J. (2006). Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101, 179–194.

Hoffman, M. D., Blei, D. M. and Bach, F. (2010). Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 23* (eds. J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel and A. Culotta), 856–864. Neural Information Processing Systems, La Jolla, CA.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14, 1303–1347.

Honkela, A., Tornio, M., Raiko, T. and Karhunen, J. (2008). Natural conjugate gradient in variational inference. In *Neural Information Processing* (eds. M. Ishikawa, K. Doya, H. Miyamoto and T. Yamakawa), 305–314. Springer-Verlag, Berlin.

Honkela, A. and Valpola, H. (2003). On-line variational Bayesian learning. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, 803–808.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2, 193–218.

Ideker, T., Thorsson, V., Ranish, J. A., Christmas, R., Buhler, J., Eng, J. K., Bumgarner, R., Goodlett, D. R., Aebersold, R. and Hood, L. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292, 929–934.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10, 25–37.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3, 79–87.

Jank, W. (2006). Implementing and diagnosing the stochastic approximation EM algorithm. *Journal of Computational and Graphical Statistics*, 15, 803–829.

Jewell, C. P., Kypraios, T., Neal, P. and Roberts, G. O. (2009). Bayesian analysis for emerging infectious diseases. *Bayesian Analysis*, 4, 465–496.

Ji, C., Shen, H. and West, M. (2010). Bounded approximations for marginal likelihoods. Available at `http://ftp.stat.duke.edu/WorkingPapers/10-05.pdf`.

Jiang, W. and Tanner, M. (1999). Hierarchical mixtures-of-experts for exponential family regression models: approximation and maximum likelihood estimation. *The Annals of Statistics*, 27, 987–1011.

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

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, 181–214.

Kass, R. E. and Natarajan, R. (2006). A default conjugate prior for variance components in generalized linear mixed models (Comment on article by Browne and Draper). *Bayesian Analysis*, 1, 535–542.

Kenward, M. G. and Molenberghs, G. (1998). Likelihood based frequentist inference when data are missing at random. *Statistical Science*, 13, 236–247.

Kesten, H. (1958). Accelerated stochastic approximation. *The Annals of Mathematical Statistics*, 29, 41–59.

Knowles, D. A. and Minka, T. P. (2011). Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems 24* (eds. J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. Pereira and K. Q. Weinberger), 1701–1709. Neural Information Processing Systems, La Jolla, CA.

Li, F., Villani, M. and Kohn, R. (2010). Flexible modeling of conditional distributions using smooth mixtures of asymmetric student t densities. *Journal of Statistical Planning and Inference*, 140, 3638–3654.

—— (2011). Modeling conditional densities using finite smooth mixtures. In *Mixtures: Estimation and Applications* (eds. K. L. Mengersen, C. P. Robert and D. M. Titterington). Wiley, Chichester, UK.

Liu, Q. and Pierce, D. A. (1994). A note on Gauss-Hermite quadrature. *Biometrika*, 81, 624–629.

Liu, J. S. and Wu, Y. N. (1999). Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94, 1264–1274.

Luan, Y. and Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, 19, 474–482.

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS - a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, 10, 325–337.

MacEachern, S. N. (1999). Dependent nonparametric processes. In *ASA Proceedings of the Section on Bayesian Statistical Science*, 50–55. American Statistical Association, Alexandria, VA.

Magnus, J. R. and Neudecker, H. (1988). Matrix differential calculus with applications in statistics and econometrics. Wiley, Chichester, UK.

Marshall, E. C. and Spiegelhalter, D. J. (2007). Identifying outliers in Bayesian hierarchical models: a simulation-based approach. *Bayesian Analysis*, 2, 409-444.

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

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

McLachlan, G. and Peel, D. (2000). *Finite mixture models*. Wiley, New York.

McLachlan, G. J., Do, K. A. and Ambroise, C. (2004). *Analyzing microarray gene expression data*. Wiley, New York.

Meng, X. L. (1994). On the rate of convergence of the ECM algorithm. *The Annals of Statistics*, 22, 326–339.

Meng, X. L. and van Dyk, D. A. (1997). The EM algorithm - an old folk song sung to a fast new tune (with discussion). *Journal of the Royal Statistical Society: Series B*, 59, 511–567.

—— (1999). Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, 86, 301–320.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11, 125–139.

Ng, S. K., McLachlan, G. J., Wang, K., Ben-Tovim Jones, L. and Ng, S.-W. (2006). A mixture model with random-effects components for clustering correlated gene-expression profiles. *Bioinformatics*, 22, 1745–1752.

Norets, A. (2010). Approximation of conditional densities by smooth mixtures of regressions. *The Annals of Statistics*, 38, 1733–1766.

Nott, D. J., Tan, S. L., Villani, M. and Kohn, R. (2012). Regression density estimation with variational methods and stochastic approximation. *Journal of Computational and Graphical Statistics*, 21, 797–820.

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: a tutorial. *Reliability Engineering and System Safety*, 91, 1290–1300.

O'Hagan, A. and Forster, J. (2004). *Kendall's Advanced Theory of Statistics, Vol. 2B: Bayesian Inference*, 2nd ed. Arnold, London.

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

—— (2012). Gaussian variational approximate inference for generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 21, 2–17.

Overstall, A. M. and Forster, J. J. (2010). Default Bayesian model determination methods for generalised linear mixed models. *Computational Statistics and Data Analysis*, 54, 3269–3288.

Paisley, J., Blei, D. M. and Jordan, M. I. (2012). Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning* (eds. J. Langford and J. Pineau), 1367–1374. Omnipress, Madison, WI.

Papaspiliopoulos, O., Roberts, G. O. and Sköld, M. (2003). Non-centered parametrizations for hierarchical models and data augmentation. In *Bayesian Statistics 7* (eds. J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, M. West), 307–326. Oxford University Press, New York.

—— (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 22, 59–73.

Paquet, U., Winther, O. and Opper, M. (2009). Perturbation corrections in approximate inference: mixture modelling applications. *Journal of Machine Learning Research*, 10, 935–976.

Parisi, G. (1988). *Statistical Field Theory*. Addison-Wesley, Redwood City, California.

Peng, F., Jacobs, R. A. and Tanner, M. A. (1996). Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition. *Journal of the American Statistical Association*, 91, 953–960.

Pepelyshev, A. (2010). The role of the nugget term in the Gaussian process method. In *mODa 9-Advances in Model-Oriented Design and Analysis* (eds. A. Giovagnoli, A. C. Atkinson, B. Torsney and C. May), 149–156. Springer, New York.

Pesaran, M. H. and Timmermann, A. (2002). Market timing and return prediction under model instability. *Journal of Empirical Finance*, 9, 495–510.

Qi, Y. and Jaakkola, T. S. (2006). Parameter expanded variational Bayesian methods. In *Advances in Neural Information Processing Systems 19* (eds. B. Schölkopf, J. Platt and T. Hofmann), 1097–1104. MIT Press, Cambridge.

Raudenbush, S. W., Yang, M. L. and Yosef, M. (2000) Maximum likelihood for generalized linear models with nested random effects via high-order, multivariate Laplace approximation. *Journal of Computational and Graphical Statistics*, 9, 141–157.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society: Series B*, 59, 731–792.

Rijmen, F. and Vomlel, J. (2008). Assessing the performance of variational methods for mixed logistic regression models. *Journal of Statistical Computation and Simulation*, 78, 765–779.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22, 400–407.

Roos, M. and Held, L. (2011). Sensitivity analysis in Bayesian generalized linear mixed models for binary data. *Bayesian Analysis*, 6, 259–278.

Roulin, A. and Bersier, L. F. (2007). Nestling barn owls beg more intensely in the presence of their mother than in the presence of their father. *Animal Behaviour*, 74, 1099–1106.

Sahu, S. K. and Roberts, G. O. (1999). On convergence of the EM algorithm and the Gibbs sampler. *Statistics and Computing*, 9, 55–64.

Salimans, T. and Knowles, D. A. (2012). Fixed-form variational posterior approximation through stochastic linear regression. Available at arXiv:1206.6679.

Sato, M. (2001). Online model selection based on the variational Bayes. *Neural Computation*, 13, 1649–1681.

Saul, L. K. and Jordan, M. I. (1998). A mean field learning algorithm for unsupervised neural networks. In *Learning in graphical models* (eds. M. I. Jordan), 541–554. Kluwer Academic, Boston.

Scharl, T., Grün, B. and Leisch, F. (2010). Mixtures of regression models for time course gene expression data: evaluation of initialization and random effects. *Bioinformatics*, 26, 370–377.

Scheel, I., Green, P. J. and Rougier, J. C. (2011). A graphical diagnostic for identifying influential model choices in Bayesian hierarchical models. *Scandinavian Journal of Statistics*, 38, 529–550.

Smyth, G. K. (1989). Generalized linear models with varying dispersion. *Journal of the Royal Statistical Society: Series B*, 51, 47–60.

Spall, J. C. (2003). Introduction to stochastic search and optimization: estimation, simulation and control. Wiley, New Jersey.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9, 3273–3297.

Spiegelhalter, D. J., Aylin, P., Best, N. G., Evans, S. J. W. and Murray, G. D. (2002a). Commissioned analysis of surgical performance using routine data: lessons from the Bristol inquiry. *Journal of the Royal Statistical Society: Series A*, 165, 191–231.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and Van der Linde, A. (2002b). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society: Series B*, 64, 583–616.

Sturtz, S., Ligges, U., and Gelman, A. (2005). R2WinBUGS: A package for running WinBUGS from R. *Journal of Statistical Software*, 12, 1–16.

Tan, L. S. L. and Nott, D. J. (2013a). Variational approximation for mixtures of linear mixed models. *Journal of Computational and Graphical Statistics*. Advance online publication. doi: `10.1080/10618600.2012.761138`

—— (2013b). Variational inference for generalized linear mixed models using partially noncentered parametrizations. *Statistical Science*, 28, 168–188.

—— (2013c). A stochastic variational framework for fitting and diagnosing generalized linear mixed models. Available at arXiv:1208.4949.

Thall, P. F. and Vail, S. C. (1990). Some covariance models for longitudinal count data with overdispersion. *Biometrics*, 46, 657–671.

Ueda, N. and Ghahramani, Z. (2002). Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15, 1223–1241.

Vehtari, A. and Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14, 2439–2468.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*, 4th ed. Springer, New York.

Verbeek, J. J., Vlassis, N. and Kröse, B. (2003). Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15, 469–485.

Villani, M., Kohn, R. and Giordani, P. (2009). Regression density estimation using smooth adaptive Gaussian mixtures. *Journal of Econometrics*, 153, 155–173.

Wand, M. P. (2002). Vector differential calculus in statistics. *The American Statistician*, 56, 55–62.

—— (2013). Fully simplified multivariate normal updates in non-conjugate variational message passing. Available at `http://www.uow.edu.au/~mwand/fsupap.pdf`.

Wand, M. P., Omerod, J. T., Padoan, S. A. and Frührwirth, R. (2011). Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6, 847–900.

Wang, C., Paisley, J. and Blei, D. M. (2011). Online variational inference for the hierarchical Dirichlet process. *Journal of Machine Learning Research - Proceedings Track, Vol. 15: Fourteenth International Conference on Artificial Intelligence and Statistics* (eds. G. Gordon, D. Dunson and M. Dudk), 752–760.

Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*

(eds. R. G. Cowell and Z, Ghahramani), 373–380. Society for Artificial Intelligence and Statistics.

Waterhouse, S., MacKay, D. and Robinson, T. (1996). Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems 8* (eds. D. S. Touretzky, M. C. Mozer and M. E. Hasselmo), 351–357. MIT Press, Cambridge, MA.

Wedel, M. (2002). Concomitant variables in finite mixture models. *Statistica Neerlandica*, 56, 362–375.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning* (eds. L. Getoor and T. Scheffer), 681–688. Omnipress, Madison, WI.

West, M. (1985). Generalized linear models: outlier accommodation, scale parameters and prior distributions. In *Bayesian Statistics 2* (eds. J. M. Bernardo, M. H. Degroot, D. V. Lindley and A. F. M. Smith), 531–538. North-Holland, Amsterdam.

Winn, J. and Bishop, C. M. (2005). Variational message passing. *Journal of Machine Learning Research*, 6, 661–694.

Wood, S. A., Jiang, W., and Tanner, M. A. (2002). Bayesian mixture of splines for spatially adaptive nonparametric regression. *Biometrika*, 89, 513–528.

Wood, S. A., Kohn, R., Cottet, R., Jiang, W. and Tanner, M. (2008). Locally adaptive nonparametric binary regression. *Journal of Computational and Graphical Statistics*, 17, 352–372.

Woolson, R. F. and Clarke, W. R. (1984). Analysis of categorical incomplete longitudinal data. *Journal of the Royal Statistical Society: Series A*, 147, 87–99.

Wu, B., McGrory, C. A. and Pettitt, A. N. (2012). A new variational Bayesian algorithm with application to human mobility pattern modeling. *Statistics and Computing*, 22, 185–203.

Yeung, K. Y., Medvedovic, M. and Bumgarner, R. E. (2003). Clustering gene-expression data with repeated measurements. *Genome Biology*, 4, R34.

Yu, Y. and Meng, X. L. (2011). To center or not to center: that is not the question - An ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 20, 531–570.

Yu, D. and Yau, K. K. W. (2012). Conditional Akaike information criterion for generalized linear mixed models. *Computational Statistics and Data Analysis*, 56, 629–644.

Zhao, Y., Staudenmayer, J., Coull, B. A. and Wand, M. P. (2006). General design Bayesian generalized linear mixed models. *Statistical Science*, 21, 35–51.

Zhou, Y., Little, R. J. A. and Kalbfleisch, J. D. (2010). Block-conditional missing at random models for missing data. *Statistical Science*, 25, 517-532.

Zhu, H. T. and Lee, S. Y. (2002). Analysis of generalized linear mixed models via a stochastic approximation algorithm with Markov chain Monte Carlo method. *Statistics and Computing*, 12, 175–183.

Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A. and Smith, G. M. (2009). *Mixed effects models and extensions in ecology with R*. Springer, New York.

Appendix A

Derivation of variational lower bound for Algorithm 1

From (2.3), the variational lower bound on $\sup_\gamma \log p(\gamma)p(y|\gamma)$ can be written as

$$E_q\{\log p(y,\theta)\} - E_q\{\log q(\theta_{-\gamma})\}, \tag{A.1}$$

where $E_q(\cdot)$ denotes expectation with respect to the variational approximation q. To evaluate the lower bound, we use the two lemmas below which we state without proof.

Lemma A.1. Suppose $p_1(x) = N(\mu_1, \Sigma_1)$ and $p_2(x) = N(\mu_2, \Sigma_2)$ where x is a p-dimensional vector, then $\int p_2(x) \log p_1(x)\, dx = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(\mu_2-\mu_1)^T\Sigma_1^{-1}(\mu_2-\mu_1) - \frac{1}{2}\mathrm{tr}(\Sigma_1^{-1}\Sigma_2)$.

Lemma A.2. Suppose $p(\tau) = N(\mu, \Sigma)$. Then

(a) $\int (y - x^T\tau)^2 p(\tau)\, d\tau = (y - x^T\mu)^2 + x^T\Sigma x$,

(b) $\int \exp(-x^T\tau)p(\tau)\, d\tau = \exp(\frac{1}{2}x^T\Sigma x - x^T\mu)$.

Consider the first term in (A.1). Write $z_{ij} = I(\delta_i = j)$ where $I(\cdot)$ denotes the indicator function. We have

$$\begin{aligned}\log p(y,\theta) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}\big\{\log p(y_i|\delta_i = j, \beta_j, \alpha_j) + \log p_{ij}(\gamma)\big\} \\ + \sum_{j=1}^{k}\big\{\log p(\beta_j) + \log p(\alpha_j)\big\} + \log p(\gamma).\end{aligned}$$

Taking expectations with respect to q, we have

$$\begin{aligned} E_q\big\{\log p(y,\theta)\big\} = \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\Big\{ &-\tfrac{1}{2}\log(2\pi) - \tfrac{1}{2}w_{ij}\exp\big(\tfrac{1}{2}u_i^T\Sigma_{\alpha_j}^q u_i - u_i^T\mu_{\alpha_j}^q\big) \\ &- \tfrac{1}{2}u_i^T\mu_{\alpha j}^q + \log p_{ij}(\mu_\gamma^q)\Big\} + \sum_{j=1}^{k}\Big\{ -\tfrac{p}{2}\log 2\pi - \tfrac{1}{2}\log|\Sigma_{\beta j}^0| \\ &- \tfrac{1}{2}\mathrm{tr}({\Sigma_{\beta j}^0}^{-1}\Sigma_{\beta j}^q) - \tfrac{1}{2}(\mu_{\beta j}^q - \mu_{\beta j}^0)^T{\Sigma_{\beta j}^0}^{-1}(\mu_{\beta j}^q - \mu_{\beta j}^0) \\ &- \tfrac{m}{2}\log 2\pi - \tfrac{1}{2}(\mu_{\alpha j}^q - \mu_{\alpha j}^0)^T{\Sigma_{\alpha j}^0}^{-1}(\mu_{\alpha j}^q - \mu_{\alpha j}^0) \\ &- \tfrac{1}{2}\log|\Sigma_{\alpha j}^0| - \tfrac{1}{2}\mathrm{tr}({\Sigma_{\alpha j}^0}^{-1}\Sigma_{\alpha j}^q)\Big\} + \log p(\mu_\gamma^q), \end{aligned} \tag{A.2}$$

where $w_{ij} = (y_i - x_i^T\mu_{\beta_j}^q)^2 + x_i^T\Sigma_{\beta_j}^q x_i$ and $p(\mu_\gamma^q)$ denotes the prior distribution for γ evaluated at μ_γ^q. In evaluating the expectation for the likelihood term, we have used the independence of β_j and α_j in the variational posterior.

Turning to the second term in (A.1), we have

$$\begin{aligned} E_q\{\log q(\theta_{-\gamma})\} &= \sum_{j=1}^{k}\big[E_q\{\log q(\beta_j)\} + E_q\{\log q(\alpha_j)\}\big] + \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\log q_{ij} \\ &= \sum_{j=1}^{k}\big(-\tfrac{p}{2}\log 2\pi - \tfrac{1}{2}\log|\Sigma_{\beta j}^q| - \tfrac{p}{2} - \tfrac{q}{2}\log 2\pi - \tfrac{1}{2}\log|\Sigma_{\alpha j}^q| \\ &\quad - \tfrac{m}{2}\big) + \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\log q_{ij}, \end{aligned} \tag{A.3}$$

and putting (A.2) and (A.3) together gives the lower bound in (2.4).

Appendix B

Derivation of variational lower bound for Algorithm 3

The variational lower bound is given by $\mathcal{L} = E_q\{\log p(y,\theta)\} - E_q\{\log q(\theta_{-\gamma})\}$. Consider the first term, $E_q\{\log p(y,\theta)\}$. Let $z_{ij} = I(\delta_i = j)$ where $I(\cdot)$ denotes the indicator function. We have

$$\begin{aligned}\log p(y,\theta) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}\Big\{ &\log p(y_i|\delta_i = j, \beta_j, a_i, b_j, \Sigma_{ij}) + \log p(a_i|\sigma^2_{a_j}) \\ &+ \log p_{ij}(\gamma)\Big\} + \sum_{j=1}^{k}\Big\{ \log p(\beta_j) + \log p(b_j|\sigma^2_{b_j}) + \log p(\sigma^2_{a_j}) \\ &+ \log p(\sigma^2_{b_j}) + \sum_{l=1}^{g}\log p(\sigma^2_{jl})\Big\} + \log p(\delta).\end{aligned}$$

Taking expectations with respect to q, we have

$$
\begin{aligned}
E_q\{\log p(y,\theta)\} = \sum_{i=1}^{n}\sum_{j=1}^{k} q_{ij}\Big[& -\tfrac{n_i}{2}\log(2\pi) - \sum_{l=1}^{g}\tfrac{\kappa_{il}}{2}\big\{\log\lambda_{jl}^q - \psi(\alpha_{jl}^q)\big\} \\
& - \tfrac{1}{2}\xi_{ij}^T {\Sigma_{ij}^q}^{-1}\xi_{ij} - \tfrac{1}{2}\mathrm{tr}({\Sigma_{ij}^q}^{-1}\Lambda_{ij}) - \tfrac{s_1}{2}\big\{\log\lambda_{a_j}^q - \psi(\alpha_{a_j}^q)\big\} \\
& - \tfrac{s_1}{2}\log(2\pi) - \tfrac{\alpha_{a_j}^q}{2\lambda_{a_j}^q}\big\{{\mu_{a_i}^q}^T\mu_{a_i}^q + \mathrm{tr}(\Sigma_{a_i}^q)\big\} + \log p_{ij}(\mu_\gamma^q)\Big] \\
& + \sum_{j=1}^{k}\Big[-\tfrac{p}{2}\log(2\pi) - \tfrac{1}{2}\big\{{\mu_{\beta_j}^q}^T\Sigma_{\beta_j}^{-1}\mu_{\beta_j}^q + \mathrm{tr}(\Sigma_{\beta_j}^{-1}\Sigma_{\beta_j}^q)\big\} \\
& - \tfrac{1}{2}\log|\Sigma_{\beta_j}| - \tfrac{s_2}{2}\log(2\pi) - \tfrac{s_2}{2}\big\{\log\lambda_{b_j}^q - \psi(\alpha_{b_j}^q)\big\} \\
& - \tfrac{\alpha_{b_j}^q}{2\lambda_{b_j}^q}\big\{{\mu_{b_j}^q}^T\mu_{b_j}^q + \mathrm{tr}(\Sigma_{b_j}^q)\big\} + \alpha_{a_j}\log\lambda_{a_j} - \log\Gamma(\alpha_{a_j}) \\
& - (\alpha_{a_j}+1)\big\{\log\lambda_{a_j}^q - \psi(\alpha_{a_j}^q)\big\} - \tfrac{\lambda_{a_j}\alpha_{a_j}^q}{\lambda_{a_j}^q} + \alpha_{b_j}\log\lambda_{b_j} \\
& - \log\Gamma(\alpha_{b_j}) - \tfrac{\lambda_{b_j}\alpha_{b_j}^q}{\lambda_{b_j}^q} - (\alpha_{b_j}+1)\big\{\log\lambda_{b_j}^q - \psi(\alpha_{b_j}^q)\big\}\Big] \\
& + \sum_{j=1}^{k}\sum_{l=1}^{g}\Big[\alpha_{jl}\log\lambda_{jl} - (\alpha_{jl}+1)\big\{\log\lambda_{jl}^q - \psi(\alpha_{jl}^q)\big\} \\
& - \log\Gamma(\alpha_{jl}) - \tfrac{\lambda_{jl}\alpha_{jl}^q}{\lambda_{jl}^q}\Big] + \log p(\mu_\gamma^q).
\end{aligned}
$$

Here $p(\mu_\gamma^q)$ denotes the prior distribution for γ evaluated at μ_γ^q, $\xi_{ij} = y_i - X_i\mu_{\beta_j}^q - W_i\mu_{a_i}^q - V_i\mu_{b_j}^q$, ${\Sigma_{ij}^q}^{-1} = \mathrm{blockdiag}\left(\frac{\alpha_{j1}^q}{\lambda_{j1}^q}I_{\kappa_{i1}}, \ldots, \frac{\alpha_{jg}^q}{\lambda_{jg}^q}I_{\kappa_{ig}}\right)$ and $\Lambda_{ij} = X_i\Sigma_{\beta_j}^q X_i^T + W_i\Sigma_{a_i}^q W_i^T + V_i\Sigma_{b_j}^q V_i^T$.

For the second term, $E_q\{\log q(\theta_{-\gamma})\}$, we have

$$
\begin{aligned}
E_q\{\log q(\theta_{-\gamma})\} = \sum_{j=1}^{k} \Big[& E_q\{\log q(\beta_j)\} + E_q\{\log q(b_j)\} + E_q\{\log q(\sigma^2_{b_j})\} \\
& + E_q \log q(\sigma^2_{a_j}) \Big] + \sum_{i=1}^{n} \Big\{ E_q\{\log q(a_i)\} + E_q\{\log q(\delta_i)\} \Big\} \\
& + \sum_{j=1}^{k} \sum_{l=1}^{g} E_q\{\log q(\sigma^2_{jl})\} \\
= \sum_{j=1}^{k} \Big[& - \tfrac{p}{2} \log(2\pi) - \tfrac{p}{2} - \tfrac{1}{2} \log |\Sigma^q_{\beta_j}| - \tfrac{s_2}{2} \log(2\pi) - \tfrac{s_2}{2} \\
& - \tfrac{1}{2} \log |\Sigma^q_{b_j}| + (\alpha^q_{b_j} + 1)\psi(\alpha^q_{b_j}) - \log \lambda^q_{b_j} - \log \Gamma(\alpha^q_{b_j}) \\
& - \alpha^q_{b_j} + (\alpha^q_{a_j} + 1)\psi(\alpha^q_{a_j}) - \log \lambda^q_{a_j} - \log \Gamma(\alpha^q_{a_j}) - \alpha^q_{a_j} \Big] \\
& + \sum_{i=1}^{n} \Big[- \tfrac{s_1}{2} \log(2\pi) - \tfrac{s_1}{2} - \tfrac{1}{2} \log |\Sigma^q_{a_i}| + \sum_{j=1}^{k} q_{ij} \log q_{ij} \Big] \\
& + \sum_{j=1}^{k} \sum_{l=1}^{g} \big\{ (\alpha^q_{jl} + 1)\psi(\alpha^q_{jl}) - \log \lambda^q_{jl} - \log \Gamma(\alpha^q_{jl}) - \alpha^q_{jl} \big\}.
\end{aligned}
$$

Putting the expressions for $E_q\{\log p(y, \theta)\}$ and $E_q\{\log q(\theta_{-\gamma})\}$ together gives the lower bound for Algorithm 3 in (3.3).

Appendix C

Derivation of variational lower bound for Algorithm 8

From (1.2), (4.6) and (4.18), the variational lower bound for Algorithm 8 is given by

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{n} S_{y_i} + \sum_{i=1}^{n} S_{\tilde{\alpha}_i} + S_\beta + E_q\{\log p(D|\nu, S)\} - E_q\{\log q(\beta)\} \\ &- \sum_{i}^{n} E_q\{\log q(\tilde{\alpha}_i)\} - E_q\{\log q(D)\}.\end{aligned}$$

To evaluate the terms in the lower bound, we use Lemma A.1 and Lemma C.1 stated below:

Lemma C.1. Suppose $p(D) = IW(\nu, S)$ where D is a symmetric, positive definite $r \times r$ matrix, then $\int p(D) \log |D| \, dD = \log |S| - \sum_{l=1}^{r} \psi\left(\frac{\nu-l+1}{2}\right) - r \log 2$ and $\int p(D) D^{-1} \, dD = \nu S^{-1}$.

Using these two lemmas, we can compute most of the terms in the lower bound:

$$\begin{aligned}S_\beta &= \int q(\beta) \log p(\beta|\Sigma_\beta) \, d\beta \\ &= -\tfrac{p}{2} \log(2\pi) - \tfrac{1}{2} \log |\Sigma_\beta| - \tfrac{1}{2} \mu_\beta^{q\,T} \Sigma_\beta^{-1} \mu_\beta^q - \tfrac{1}{2} \mathrm{tr}(\Sigma_\beta^{-1} \Sigma_\beta^q),\end{aligned}$$

$$\begin{aligned}S_{\tilde{\alpha}_i} &= \int q(\beta) q(D) q(\tilde{\alpha}_i) \log p(\tilde{\alpha}_i|\beta, D) \, d\beta \, dD \, d\tilde{\alpha}_i \\ &= -\tfrac{r}{2} \log(2\pi) - \tfrac{1}{2} \left\{ \log |S^q| - \sum_{l=1}^{r} \psi\left(\tfrac{\nu^q-l+1}{2}\right) - r \log 2 \right\} \\ &\quad -\tfrac{\nu^q}{2} \left[(\mu_{\tilde{\alpha}_i}^q - \tilde{W}_i \mu_\beta^q)^T S^{q-1} (\mu_{\tilde{\alpha}_i}^q - \tilde{W}_i \mu_\beta^q) + \mathrm{tr}\{S^{q-1} (\Sigma_{\tilde{\alpha}_i}^q + \tilde{W}_i \Sigma_\beta^q \tilde{W}_i^T)\} \right],\end{aligned}$$

$$\begin{aligned}E_q\{\log p(D|\nu, S)\} &= \int q(D) \log p(D|\nu, S) \, dD \\ &= -\tfrac{\nu^q}{2} \mathrm{tr}(S^{q\,-1} S) - \tfrac{r(r-1)}{4} \log(\pi) - \sum_{l=1}^{r} \log \Gamma\left(\tfrac{\nu+1-l}{2}\right) \\ &\quad -\tfrac{\nu+r+1}{2} \left\{ \log |S^q| - \sum_{l=1}^{r} \psi\left(\tfrac{\nu^q-l+1}{2}\right) - r \log 2 \right\} \\ &\quad +\tfrac{\nu}{2} \log |S| - \tfrac{\nu r}{2} \log 2,\end{aligned}$$

$$
\begin{aligned}
E_q\{\log q(\beta)\} &= \int q(\beta)\log q(\beta)\,d\beta \\
&= -\tfrac{p}{2}\log(2\pi) - \tfrac{1}{2}\log|\Sigma_\beta^q| - \tfrac{p}{2},
\end{aligned}
$$

$$
\begin{aligned}
E_q\{\log q(\tilde{\alpha}_i)\} &= \int q(\tilde{\alpha}_i)\log q(\tilde{\alpha}_i)\,d\tilde{\alpha}_i \\
&= -\tfrac{r}{2}\log(2\pi) - \tfrac{1}{2}\log|\Sigma_{\tilde{\alpha}_i}^q| - \tfrac{r}{2},
\end{aligned}
$$

$$
\begin{aligned}
E_q\{\log q(D)\} &= \int q(D)\log q(D)\,dD \\
&= -\tfrac{\nu^q r}{2}\log 2 - \tfrac{r(r-1)}{4}\log\pi - \textstyle\sum_{l=1}^{r}\log\Gamma\left(\tfrac{\nu^q+1-l}{2}\right) + \tfrac{\nu^q}{2}\log|S^q| \\
&\quad -\tfrac{\nu^q+r+1}{2}\left\{\log|S^q| - \textstyle\sum_{l=1}^{r}\psi\left(\tfrac{\nu^q-l+1}{2}\right) - r\log 2\right\} - \tfrac{\nu^q r}{2}.
\end{aligned}
$$

The only term left to evaluate is

$$
S_{y_i} = \int q(\beta)q(\tilde{\alpha}_i)\log p(y_i|\beta,\tilde{\alpha}_i)\,d\beta\,d\tilde{\alpha}_i.
$$

For Poisson responses with the log link function [see (4.8)],

$$
S_{y_i} = y_i^T\{\log(E_i) + V_i\mu_\beta^q + X_i^R\mu_{\tilde{\alpha}_i}^q\} - E_i^T\kappa_i - 1_{n_i}^T\log(y_i!),
$$

where $\kappa_i = \exp\{V_i\mu_\beta^q + X_i^R\mu_{\tilde{\alpha}_i}^q + \frac{1}{2}\mathrm{diag}(V_i\Sigma_\beta^q V_i^T + X_i^R\Sigma_{\tilde{\alpha}_i}^q X_i^{R^T})\}$. As for Bernoulli responses with the logit link function [see (4.9)], recall that

$$
B^{(r)}(\mu,\sigma) = \int_{-\infty}^{\infty} b^{(r)}(\sigma x+\mu)\phi(x;0,1)\,dx,
$$

where $b^{(r)}(x)$ denotes the rth derivative of $b(x) = \log\{1+\exp(x)\}$ with respect to x. Therefore, we have

$$
\begin{aligned}
E_q\big[\log\{1+\exp(V_{ij}^T\beta + X_{ij}^{R^T}\tilde{\alpha}_i)\}\big] &= E_q\{b(V_{ij}^T\beta + X_{ij}^{R^T}\tilde{\alpha}_i)\} \\
&= \int_{-\infty}^{\infty} b(\sigma_{ij}^q x + \mu_{ij}^q)\phi(x;0,1)\,dx \\
&= B^{(0)}(\mu_{ij}^q,\sigma_{ij}^q),
\end{aligned}
$$

where $\mu_{ij}^q = V_{ij}^T\mu_\beta^q + X_{ij}^{R^T}\mu_{\tilde{\alpha}_i}^q$, $\sigma_{ij}^q = \sqrt{V_{ij}^T\Sigma_\beta^q V_{ij} + X_{ij}^{R^T}\Sigma_{\tilde{\alpha}_i}^q X_{ij}^R}$ for each $i = 1,\ldots,n$, $j = 1,\ldots,n_i$. Hence,

$$
S_{y_i} = y_i^T(V_i\mu_\beta^q + X_i^R\mu_{\tilde{\alpha}_i}^q) - \sum_{j=1}^{n_i} B^{(0)}(\mu_{ij}^q,\sigma_{ij}^q),
$$

where $B^{(0)}(\mu_{ij}^q,\sigma_{ij}^q)$ is evaluated using adaptive Gauss-Hermite quadrature

(see Appendix D). The variational lower bound is thus given by

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{n} S_{y_i} + \tfrac{1}{2}\sum_{i=1}^{n} \log|\Sigma^q_{\tilde{\alpha}_i}| + \tfrac{1}{2}\log|\Sigma_\beta^{-1}\Sigma^q_\beta| - \tfrac{1}{2}\mathrm{tr}(\Sigma_\beta^{-1}\Sigma^q_\beta) - \tfrac{1}{2}\mu_\beta^{qT}\Sigma_\beta^{-1}\mu^q_\beta \\ &- \tfrac{\nu^q}{2}\log|S^q| + \tfrac{\nu}{2}\log|S| - \sum_{l=1}^{r}\log\Gamma\left(\tfrac{\nu^q+1-l}{2}\right) + \sum_{l=1}^{r}\log\Gamma\left(\tfrac{\nu+1-l}{2}\right) \\ &+ \tfrac{p+nr}{2} + \tfrac{nr}{2}\log 2.\end{aligned}$$

Note that this expression is valid only after each of the parameter updates has been made in Algorithm 8.

Appendix D

Gauss-Hermite quadrature

To evaluate the variational lower bound and gradients in Algorithm 8 for the logistic mixed model, we compute $B^{(r)}(\mu_{ij}^q, \sigma_{ij}^q)$ for each $i = 1, \ldots, n$, $j = 1, \ldots, n_i$ and $r = 0, 1, 2$ using adaptive Gauss-Hermite quadrature (Liu and Pierce, 1994). Ormerod and Wand (2012) has considered a similar approach. In Gauss-Hermite quadrature, integrals of the form $\int_{-\infty}^{\infty} f(x)\mathrm{e}^{-x^2}\, dx$ are approximated by $\sum_{k=1}^{m} w_k f(x_k)$ where m is the number of quadrature points, the nodes x_k are zeros of the mth order Hermite polynomial and w_k are suitably corresponding weights. This approximation is exact for polynomials of degree $2m-1$ or less. For low-order quadrature to be effective, some transformation is usually required so that the integrand is sampled in a suitable range. Following the procedure recommended by Liu and Pierce (1994), we rewrite $B^{(r)}(\mu_{ij}^q, \sigma_{ij}^q)$ as

$$\begin{aligned}
&\int_{-\infty}^{\infty} \frac{b^{(r)}(\sigma_{ij}^q x + \mu_{ij}^q)\phi(x; 0, 1)}{\phi(x; \hat{\mu}_{ij}^q, \hat{\sigma}_{ij}^q)} \phi(x; \hat{\mu}_{ij}^q, \hat{\sigma}_{ij}^q) dx \\
&= \sqrt{2}\hat{\sigma}_{ij}^q \int_{-\infty}^{\infty} \left[\exp(x^2) b^{(r)}\{\sigma_{ij}^q(\hat{\mu}_{ij}^q + \sqrt{2}\hat{\sigma}_{ij}^q x\} + \mu_{ij}^q)\phi(\hat{\mu}_{ij}^q + \sqrt{2}\hat{\sigma}_{ij}^q x; 0, 1) \right] \\
&\qquad \cdot \exp(-x^2)\, dx,
\end{aligned}$$

which can be approximated using Gauss-Hermite quadrature by

$$\sqrt{2}\hat{\sigma}_{ij}^q \sum_{k=1}^{m} w_k \exp(x_k^2) b^{(r)}\{\sigma_{ij}^q(\hat{\mu}_{ij}^q + \sqrt{2}\hat{\sigma}_{ij}^q x_k) + \mu_{ij}^q\}\phi(\hat{\mu}_{ij}^q + \sqrt{2}\hat{\sigma}_{ij}^q x_k; 0, 1).$$

For the integrand to be sampled in an appropriate region, we take $\hat{\mu}_{ij}^q$ to be the mode of the integrand and $\hat{\sigma}_{ij}^q$ to be the standard deviation of the

normal density approximating the integrand at the mode, so that

$$\hat{\mu}_{ij}^q = \arg\max_x \left\{ b^{(r)}(\sigma_{ij}^q x + \mu_{ij}^q)\phi(x;0,1) \right\},$$
$$\hat{\sigma}_{ij}^q = \left[-\frac{d^2}{dx^2} \log \left\{ b^{(r)}(\sigma_{ij}^q x + \mu_{ij}^q)\phi(x;0,1) \right\} \Big|_{x=\hat{\mu}_{ij}^q} \right]^{-\frac{1}{2}},$$

for $j = 1, \ldots, n_i$ and $i = 1, \ldots, n$. For computational efficiency, we evaluate $\hat{\mu}_{ij}^q$ and $\hat{\sigma}_{ij}^q$, $i = 1, \ldots, n$, $j = 1, \ldots, n_i$, for the case $r = 1$ only once in each cycle of updates and use these values for $r = 0, 2$. No significant loss of accuracy was observed in doing this. We implement adaptive Gauss-Hermite quadrature in `R` using the `R` package `fastGHQuad` (Blocker, 2011). The quadrature nodes and weights can be obtained via the function `gaussHermiteData()` and the function `aghQuad()` approximates integrals using the method of Liu and Pierce (1994). We used 10 quadrature points in all the examples.