Accurate Frequentist Generalised Linear Mixed Model Analysis via Expectation Propagation

by **James Yu**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy in Mathematics

under the supervision of **Prof. Matt Wand** and **Dr. Shev MacNamara**

University of Technology Sydney

Faculty of Science

April 22, 2021

Certificate of original authorship

I, **James Yu** declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy in Mathematics, in the School of Mathematical and Physical Sciences at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with the Australian Red Cross Blood Service.

This research is supported by an Australian Government Research Training Program.

Production Note:
Signature: Signature removed prior to publication.

Date: April 22, 2021

Acknowledgements

This thesis would not have been possible without the support of a number of individuals.

Many thanks to my mentor and supervisor Matt Wand. Thank you for your patience and kindness. This achievement would not have been possible without you. I could not have asked for a better supervisor.

I would also like to thank Stephen Wright. Thank you for your care and for the time you spent with me. I really appreciate it.

To my colleagues and friends at UTS, thank you for your help, constant support, for making me laugh and for listening to me.

Finally to Mum, Dad and Mario, I am eternally grateful to you for giving me the opportunities and experiences that have made me who I am. I dedicate this milestone to you.

List of papers/publications

The following list of paper and publications awards relate to the work presented in this thesis:

- Hall, P., Johnstone, I. M., Ormerod, J. T., Wand, M. P., & Yu, J. C. F. (2020). Fast and Accurate Binary Response Mixed Model Analysis via Expectation Propagation, Journal of the American Statistical Association, 115:532, 1902–1916, DOI: 10.1080/01621459.2019.1665529.
- Wand, M. P. & Yu, J. C. F. (2020). glmmEP: Generalized Linear Mixed Model Analysis via Expectation Propagation (Version 1.0-3.1). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/glmmEP/index.html
- Yu, J. C. F. (2019), "Fast and Accurate Frequentist Generalised Linear Mixed Model Analysis via Expectation Propagation", (2019) Enabling Algorithms Theme Symposium, University of Technology Sydney, Sydney, 13-14 June.
- Wand, M. P., & Yu, J. C. F. (2021) Density Estimation via Bayesian Inference Engines.
- First Place, 20th Annual J.B. Douglass Award, New South Wales Branch of Statistical Society of Australia, (2019).

Notation

In this chapter, important notations are introduced that we refer to throughout this thesis.

Acronyms

Table 1: Table with acronyms used in the thesis with their meanings.

Acronym	**Meaning**
AGHQ	Adaptive Gauss-Hermite quadrature
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BP	Best predictor
CDF	Cummulative density function
DAG	Directed acyclic graph
DC	Data cloning
EP	Expectation propagation
GHQ	Gauss-Hermite quadrature
GLMM	Generalised linear mixed models
KL	Kullback-Leiber
MCMC	Markov chain Monte Carlo
NM	Nelder-Mead
PDF	Probability density function
PQL	Penalised quasi-likelihood

Contents

Abstract

Generalised linear mixed models are a particularly powerful and well established statistical tool. Unlike linear mixed models, where the integrals arising in likelihood functions can be expressed in closed form, the likelihood functions expressed in generalised linear mixed models do not follow tractable solutions. Methods such as Gauss-Hermite quadrature and Laplace approximation are the standard approaches to overcome these integrals. Although Gauss-Hermite quadrature is accurate it is also slow, rendering it unsuitable for analyses with more than two or three random effects. Laplace approximations are the most feasible solution, however the approximate inference they provide in binary models is well known to be inaccurate. A less common approach is to use Bayesian ideas such as data cloning, however they involve a number of technicalities and as such are difficult to implement. Although expectation propagation is generally used in Bayesian settings, in this thesis we introduce a novel approach where we use it as frequentists to achieve high accuracy results with minimal computational cost for inference on generalised linear mixed models. We show our methodology can be used to solve one level probit models without the need for quadrature, providing consistent and accurate results. We explain how using quadrature we can also extend our method to logistic, Poisson and negative-binomial models. Additionally we show how these models can be extended to two level models and crossed random effects models for the probit case. Finally we present applications of our methodology on two real datasets, both with different technical challenges.

Chapter 1

Introduction and background

This chapter provides an introduction to the thesis as well as background information and theory required for the upcoming chapters. None of the work presented in this chapter is novel.

1.1 Introduction

In reality analysts do not always receive data that conforms to the properties required for the implementation of classical statistical procedures. Data often contains non-normally distributed response variables and heterogeneous variance. These variance structures may be explained by multilevel or hierarchical structures, where the heterogeneity stems from observations nested within larger groups of experimental units (Steenbergen & Jones, 2002[62]). As an example consider a dataset of results from an experiment that is repeated 100 times by 5 different people (there are a total of 500 replicates). Since each person will conduct the experiment slightly differently, modelling the data requires accounting for variability both within and across persons. This type of dataset is common in biological, medical and social sciences, and is becoming increasing prevalent in other fields. Secton 1.2 of Gelman & Hill (2007)[19] and the R package "mlmRev" (Bates, Maechler & Bolker, 2019[4]) provide real examples. Although it may be tempting to coerce data into classical statistical frameworks via tricks such as data transformations or ignoring the effect of data structures, doing so may violate key statistical assumptions and limit the scope of inference gained (Bolker, et al., 2009[8]). Subsequently, a more suitable approach of analysis is required.

Generalised linear mixed models (GLMMs) are a particularly powerful and well

established statistical tool. Like generalised linear models GLMMs can model non-normal responses, however they additionally account for nested data structures. GLMMs do this by allowing a population intercept and slope (known as fixed effects) as well as a unique intercept and slope for each experimental unit (known as random effects). These are the attributes that allow GLMMs to handle combinations of data structures and response distributions.

Unlike linear mixed models where the integrals required to solve likelihood functions can be obtained analytically, they do not have tractable solutions in GLMMs. Additionally, as the number of random effects increases they become more computationally intensive. In the frequentist setting, methods such as Gauss-Hermite quadrature (GHQ), penalised quasi-likelihood (PQL) and Laplace approximation are the standard approaches to overcome these integrals. PQL is one of the simplest and most widely used methods, available in a variety of computing software. However, it deals poorly with binary data where the standard deviations of random effects are large, often producing biased parameter estimates (Bolker, et al., 2009[8]). Additionally, rather than approximate a true likelihood, PQL approximates a quasi-likelihood, making likelihood based inference infeasible. Both GHQ and Laplace approximation are more accurate than PQL, with GHQ the most accurate of the three. However, the accuracy of GHQ comes at the price of speed. As the number of random effects increases GHQ slows considerably, rendering it unsuitable for analyses with more than two random effects. While Laplace approximations provide the most feasible solution, the approximate inference it provides in binary models is inaccurate (McCulloch, et al., 2008[41]), particularly when the number of observations per grouping unit are low.

Lele, et al. (2007)[35] present a reformulation of the typically Bayesian method Markov chain Monte Carlo, known as data cloning, which allows for the calculation of maximum likelihood estimates and confidence intervals. Although the proposed method is able to cover random effects, it also involves several difficult technical details regarding its fitting and implementation. Similar issues exist for the R package "MCMCglmm" (Hadfield, 2017[25]).

Perhaps a lesser known method, expectation propagation (EP) (Minka, 2001[43]) has underpinnings originating from computer science. A variety of software such as Stan (Stan Development Team, 2017[24]) and Infer.Net (Minka, et al., 2014[45]) exist and facilitate the implementation of such deterministic algorithms for fast inference. Although the speed of computation is increased over more traditional methods, algebraic overheads mean utilisation of this methodology is cumbersome. Minka (2005)[44] partially solves this issue by developing a message passing approach to EP, which allows the

algebra to be broken into fragments. Although the initial algebraic overhead is still large, once calculated, fragments can be reused via graphical structures without any additional algebra. Further numerical studies have shown EP to be more accurate than variational alternatives such as mean field variational Bayes and easily faster than Markov chain Monte Carlo. Specifically, Kim & Wand (2016)[31] provide computational studies which suggest EP becomes more accurate than mean field variational Bayes as sample size increases (Kim & Wand, 2017[32]).

Kim & Wand (2016)[31] demonstrate the form of EP for the simple statistical problem of Bayesian inference from independent and identically distributed observations from a normal distribution. This paper also provides the means to implement a message passing approach using factor graphs, which facilitates extension to larger models with minimal algebraic overheads. Kim & Wand (2017)[32] provide the structure of algorithms required for implementing EP in GLMM settings.

1.2 Thesis aim

Although many methods for GLMM analysis in a frequentist setting exist, they provide either poor accuracy or bad computational performance. Additionally, current applications of EP in statistics are primarily limited to Bayesian settings. This thesis aims to develop novel methodology for GLMM analysis in a frequentist setting which utilises EP for consistent and accurate inference, particularly in the case when low numbers of observations per group variable occur. We aim to build on Kim & Wand (2017)[32] by utilising message passing to streamline the computations from the factor graph. We aim to provide frameworks for one level probit, logistic, Poisson and negative binomial models, as well as two level and crossed random effects probit models. Finally we aim to create an R package and make the methodology available, with a demonstration on a dataset provided by the Australian Red Cross Blood Service.

1.3 Matrix storage and notation

For a vector $\boldsymbol{a}$ of length d, we denote the elements from i to j by $\boldsymbol{a}_{i:j}$. For clarity, consider

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad \text{then} \quad \boldsymbol{a}_{1:2} \equiv \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}.$$

To combat issues arising from storage of large matrices we now introduce concepts and notation analogous to Magnus & Neudecker (1999).[38] For a $d \times d$ matrix $\boldsymbol{A}$, $\text{vec}(\boldsymbol{A})$ returns a $d^2 \times 1$ vector, where the columns of $\boldsymbol{A}$ are stacked underneath each other in order from left to right. $\text{vech}(\boldsymbol{A})$ returns a $\frac{1}{2}d(d+1) \times 1$ vector, where the entries above the diagonal are removed and the remaining entries are stacked by column from left to right. $\text{vecbd}(\boldsymbol{A})$ returns a $\frac{1}{2}d(d-1) \times 1$ vector, where the entries below the diagonal are stacked by column from left to right. For example, if

$$\boldsymbol{A} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

then

$$\text{vec}(\boldsymbol{A}) \equiv \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}, \quad \text{vech}(\boldsymbol{A}) \equiv \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{22} \\ a_{32} \\ a_{33} \end{bmatrix} \quad \text{and} \quad \text{vecbd}(\boldsymbol{A}) \equiv \begin{bmatrix} a_{21} \\ a_{31} \\ a_{32} \end{bmatrix}. \tag{1.1}$$

For a vector $\boldsymbol{a}$ of length d^2, $\text{vec}^{-1}(\boldsymbol{a})$ returns a $d \times d$ matrix. The ith column corresponds to the entries

$$\boldsymbol{a}_{\big(d(i-1)+1\big):(di)}.$$

For a vector $\boldsymbol{a}^*$ of length $\frac{1}{2}d(d+1)$, $\text{vech}^{-1}(\boldsymbol{a}^*)$ returns a symmetric $d \times d$ matrix. The ith lower diagonal column and ith upper diagonal row correspond to the entries

$$\boldsymbol{a}^*_{\left(d(i-1)-\sum_{j=1}^{i-1}(j-1)+1\right):\left(di-\sum_{j=1}^{i}(j-1)\right)}.$$

As an example, where $\boldsymbol{a} = \text{vec}(\boldsymbol{A})$ and $\boldsymbol{a}^* = \text{vech}(\boldsymbol{A})$,

$$\text{vec}^{-1}(\boldsymbol{a}) \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \text{vech}^{-1}(\boldsymbol{a}^*) \equiv \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Note that when $\boldsymbol{A}$ is symmetric

$$\text{vech}^{-1}\big(\text{vech}(\boldsymbol{A})\big) = \boldsymbol{A}.$$

We denote the transpose of $\boldsymbol{A}$ by $\boldsymbol{A}^\top$, and assuming $\boldsymbol{A}$ is invertible we denote its inverse by $\boldsymbol{A}^{-1}$. The 3×3 diagonal matrix formed setting the upper and lower off diagonal entries of $\boldsymbol{A}$ to 0 is given by

$$\text{diag}(\boldsymbol{A}) \equiv \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix},$$

where $\boldsymbol{A}$ is defined in equation (1.1). The $d \times d$ identity matrix is denoted as $\boldsymbol{I}_d$, the $d \times 1$ vector with all entries equal to zero as $\boldsymbol{0}_d$, and the $d_1 \times d_2$ matrix with all entries equal to zero as $\boldsymbol{0}_{d_1 \times d_2}$.

The duplication matrix of order d is the unique $d^2 \times \frac{1}{2}d(d+1)$ matrix $\boldsymbol{D}_d$ of zeros and ones such that

$$\boldsymbol{D}_d \text{vech}(\mathbf{A}) = \text{vec}(\mathbf{A}),$$

where $\mathbf{A}$ is symmetric. The R function "`duplication.matrix()`" in the "matrixcalc" package (Novomestky, 2020[48]) allows for easy calculation of the duplication matrix for any matrix $\mathbf{A}$. The Moore-Penrose inverse of $\boldsymbol{D}_d$ is

$$\boldsymbol{D}_d^+ \equiv (\boldsymbol{D}_d^\top \boldsymbol{D}_d)^{-1} \boldsymbol{D}_d^\top.$$

The norm of a vector $\boldsymbol{a}$ is defined as

$$\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a}^\top \boldsymbol{a}}.$$

and for the same vector we define

$$\boldsymbol{a}^{\otimes k} \equiv \begin{cases} 1 & \text{for } k = 0 \\ \boldsymbol{a} & \text{for } k = 1 \\ \boldsymbol{a}\boldsymbol{a}^T & \text{for } k = 2 \end{cases}. \tag{1.2}$$

1.4 Notation for spaces

We denote spaces as follows:

- $\mathbb{R}$ Set of real numbers.
- $\mathbb{R}^d$ Real coordinate space of d dimensions.
- $\mathbb{R}_{\geq 0}$ Coordinate line with all real positive numbers (real numbers greater than or equal to zero).
- $\mathbb{R}_{>0}$ Coordinate line with all strictly positive real numbers (real numbers greater than zero).
- $\mathbb{Z}_{\geq 0}$ Coordinate line with all positive integers (integers greater than or equal to zero).

1.5 Exponential family theory and distributions

1.5.1 Exponential family theory

The exponential family is a set of probability distributions which can be written in a specified parameteric form, where different distributions arise from varying parameter values.

Definition 1. *Given a natural parameter vector $\boldsymbol{\eta}$, a vector $\boldsymbol{x}$ is from an exponential family of probability distributions if its probability distribution conditional on $\boldsymbol{\eta}$ can be expressed in the following form:*

$$f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\eta}) = \exp\big(\boldsymbol{\eta}^\top \boldsymbol{T}(\boldsymbol{x}) - A(\boldsymbol{\eta})\big)h(\boldsymbol{x}), \tag{1.3}$$

where $\boldsymbol{\eta}$ is the natural parameter vector, $\boldsymbol{T}(\boldsymbol{x})$ is the sufficient statistic, $A(\boldsymbol{\eta})$ is the log-partition function and $h(\boldsymbol{x})$ is the base measure.

Note that Definition 1 can be simplied for single parameter and univariate distributions.

We now clarify the previously arbitrary definitions of $\boldsymbol{\eta}$, $\boldsymbol{T}(\boldsymbol{x})$ and $A(\boldsymbol{\eta})$:

- $\boldsymbol{T}(\boldsymbol{x})$ is the sufficient statistic of the distribution, which is a function that given data $\boldsymbol{x}$ provides all the information required to describe the posterior distribution of the natural parameters $\boldsymbol{\eta}$.
- $\boldsymbol{\eta}$ are the parameters of each distribution written in their natural form. Canonical link functions take distribution parameters as an argument and return natural parameters. The natural parameter space H is the convex space given by the set of values
$$H = \{\boldsymbol{\eta} : A(\boldsymbol{\eta}) < \infty\}.$$
- $A(\boldsymbol{\eta})$ is the log of the normalising factor
$$A(\boldsymbol{\eta}) = \log\Big\{\int h(\boldsymbol{x}) \exp\big(\boldsymbol{\eta}^\top \boldsymbol{T}(\boldsymbol{x})\big) d\boldsymbol{x}\Big\},$$
which ensures that $f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\eta})$ is a probability density function. Additionaly, the moments of the sufficient statistc $\boldsymbol{T}(\boldsymbol{x})$ can be derived by differentiating $A(\boldsymbol{\eta})$. More explicitly, the mean and variance of the sufficient statistic are respectively given by the first and second derivatives of $A(\boldsymbol{\eta})$,
$$E\big(\boldsymbol{T}(\boldsymbol{x})\big) = \mathsf{D}_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^\top = (\nabla A)(\boldsymbol{\eta}) \quad \text{and} \quad \mathrm{Cov}\big(\boldsymbol{T}(\boldsymbol{x})\big) = \mathsf{D}_{\boldsymbol{\eta}}\big(\mathsf{D}_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^\top\big) = \mathsf{H}\big(A(\boldsymbol{\eta})\big) \tag{1.4}$$
where $\mathrm{Cov}\big(\boldsymbol{T}(\boldsymbol{x})\big)$ is the covariance matrix of $\boldsymbol{T}(\boldsymbol{x})$, $\mathsf{D}_{\boldsymbol{x}} f(\boldsymbol{x})$ is a $p \times d$ matrix with (i,j) entry equal to $df(\boldsymbol{x})_i / d\boldsymbol{x}_j$ for a $\mathbb{R}^p$-valued function f with argument $\in \mathbb{R}^d$, and $(\nabla A)(\boldsymbol{\eta})$ is a one-to-one function that returns a column vector of partial derivatives of $A(\boldsymbol{\eta})$ with respect to the entries of $\boldsymbol{\eta}$. A summary of $A(\boldsymbol{\eta})$ and $(\nabla A)(\boldsymbol{\eta})$ for exponential families is provided in Section 3.5 of Wainwright & Jordan (2008).[66]

Exponential families have extremely useful properties in statistical analysis. As such, they have been used in classical statistics for decades and more recently in machine learning settings. A major benefit of exponential families is conjugate priors, where posterior distributions are in the same probability distribution family as the prior probability distribution. This helps reduces computational complexity for Bayesian inference.

1.5.2 Probability distribution

We now present the distributions used in this thesis and their exponential family parameterisation.

1.5.2.1 Univariate normal distribution

Definition 2. *A scalar random variable x is from a univariate normal distribution with mean μ and variance $\sigma^2 > 0$ when its density function is*

$$f(x) = (2\pi\sigma^2)^{-1/2} \exp\Big(- \frac{(x-\mu)^2}{2\sigma^2}\Big),$$

which is written as $x \sim N(\mu, \sigma^2)$.

Definition 3. *A scalar random variable x is from a standard univariate normal distribution with $\mu = 0$ and $\sigma^2 = 1$ when its density function is*

$$\phi(x) = (2\pi)^{-1/2} \exp\Big(- \frac{x^2}{2}\Big),$$

which is written as $x \sim N(0, 1)$, and cumulative distribution function (CDF)

$$\Phi(x) \equiv \int_{-\infty}^{x} \phi(t)dt.$$

We set

$$\zeta(x) = \log\big(2\Phi(x)\big),$$

with first and second derivatives respectively

$$\zeta'(x) = \Phi(x)/\phi(x) \quad \text{and} \quad \zeta''(x) = -\zeta'(x)\big(x + \zeta'(x)\big).$$

The sufficient statistic and base measure for the univariate normal distribution are respectively defined as

$$\boldsymbol{T}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix} \quad \text{and} \quad h(x) = (2\pi)^{-1/2}.$$

The natural parameter vector and inverse mapping are respectively

$$\boldsymbol{\eta} \equiv \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \mu/\sigma^2 \\ -1/(2\sigma^2) \end{bmatrix} \quad \text{and} \quad \begin{matrix} \mu = -\eta_1/(2\eta_2) \\ \sigma^2 = -1/(2\eta_2) \end{matrix}. \tag{1.5}$$

The log-partition function is

$$A(\boldsymbol{\eta}) = -\eta_1^2/(4\eta_2) - \tfrac{1}{2}\log(-2\eta_2).$$

The domain of both $A(\boldsymbol{\eta})$ and $(\nabla A)(\boldsymbol{\eta})$ is $H = (\eta_1, \eta_2) : \eta_1 \in \mathbb{R},\ \eta_2 \in \mathbb{R}_{>0}$. The column vector of partial derivatives of $A(\boldsymbol{\eta})$ is given by

$$(\nabla A)(\boldsymbol{\eta}) = \begin{bmatrix} -\eta_1/(2\eta_2) \\ (\eta_1^2 - 2\eta_2)/(4\eta_2^2) \end{bmatrix}.$$

1.5.2.2 Multivariate normal distribution

Definition 4. *A $d \times 1$ vector of random variables $\boldsymbol{x}$ is from the multivariate normal distribution with $d \times 1$ mean vector $\boldsymbol{\mu}$ and $d \times d$ positive definite covariance matrix variance $\boldsymbol{\Sigma}$ when its density function is*

$$f(\boldsymbol{x}) = (2\pi)^{-d/2}\,|\boldsymbol{\Sigma}|^{-1/2} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big),$$

which is written as $\boldsymbol{x} \sim \boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Definition 5. *A $d \times 1$ vector of random variables $\boldsymbol{x}$ is from the standard multivariate normal distribution with $d \times 1$ mean vector $\mathbf{0}_d$ and $d \times d$ positive definite covariance matrix variance $\boldsymbol{I}_d$ when its density function is*

$$\phi_{\boldsymbol{I}}(\boldsymbol{x}) = (2\pi)^{-d/2} \exp\Big(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{x}\Big),$$

which is written as $\boldsymbol{x} \sim \boldsymbol{N}(\mathbf{0}_d, \boldsymbol{I}_d)$.

The sufficient statistic and base measure for the multivariate normal distribution are respectively defined as

$$\boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{x} \\ \text{vec}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix} \quad \text{and} \quad h(\boldsymbol{x}) = (2\pi)^{-d/2}.$$

The natural parameter vector and inverse mapping are respectively

$$\boldsymbol{\eta} \equiv \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\boldsymbol{D}_d^\top \text{vec}(\boldsymbol{\Sigma}^{-1}) \end{bmatrix} \quad \text{and} \quad \begin{matrix} \boldsymbol{\mu} = -\frac{1}{2}\boldsymbol{H}_2^{-1}\boldsymbol{\eta}_1 \\ \boldsymbol{\Sigma} = -\frac{1}{2}\boldsymbol{H}_2^{-1} \end{matrix}, \tag{1.6}$$

where

$$\boldsymbol{\eta}_2 \equiv \boldsymbol{D}_d^\top \text{vec}(\boldsymbol{H}_2) \quad \text{and} \quad \boldsymbol{H}_2 = \text{vec}^{-1}\big((\boldsymbol{D}_d^+)^\top \boldsymbol{\eta}_2\big). \tag{1.7}$$

The log-partition function is

$$A(\boldsymbol{\eta}) = -\frac{1}{4}\boldsymbol{\eta}_1^\top \boldsymbol{H}_2^{-1}\boldsymbol{\eta}_1 - \frac{1}{2}\log|-2\boldsymbol{H}_2|,$$

The domain of both $A(\boldsymbol{\eta})$ and $\nabla A(\boldsymbol{\eta})$ is $H = (\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) : \boldsymbol{\eta}_1 \in \mathbb{R}^d$, $\boldsymbol{\eta}_2 \in \mathbb{R}_{>0}^{\frac{1}{2}d(d\times 1)}$. The column vector of partial derivatives of $A(\boldsymbol{\eta})$ is

$$(\nabla A)(\boldsymbol{\eta}) = \mathsf{D}_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^\top.$$

1.5.2.3 Bernoulli distribution

Definition 6. *A scalar random variable x is from the Bernoulli distribution with probability of success p when its probability mass function is*

$$f(x) = p^x(1-p)^{1-x}, \quad x \in \{0, 1\}, \tag{1.8}$$

which is written as $x \sim$ Bernoulli(p).

The sufficient statistic and base measure for the Bernoulli distribution are respectively defined as

$$T(x) = x \quad \text{and} \quad h(x) = I(x \in \{0, 1\}).$$

Definition 7. *For a scalar input variable x*

$$\textit{logit}(x) \equiv \log\Big(\frac{x}{1-x}\Big) \quad \textit{and} \quad \textit{expit}(x) \equiv \frac{\exp(x)}{1+\exp(x)}. \tag{1.9}$$

The natural parameter vector and inverse mapping are respectively

$$\eta = \text{logit}(p) \quad \text{and} \quad p = \text{expit}(\eta). \tag{1.10}$$

The log-partition function is

$$A(\eta) = \log\big(1+\exp(\eta)\big).$$

The domain of $A(\eta)$ and $\nabla A(\eta)$ is $H = \eta : \eta \in \mathbb{R}$. The derivative of $A(\eta)$ is

$$(\nabla A)(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)}.$$

1.5.2.4 Poisson distribution

Definition 8. *A scalar random variable $x \in \mathbb{Z}_{\geq 0}$ is Poisson distributed with mean and variance λ when its density function is*

$$f(x) = \frac{\lambda^x \exp(-\lambda)}{\Gamma(x+1)},, \tag{1.11}$$

which is written as $x \sim$ Poisson(λ), where $\Gamma(x) = (x-1)!$.

The sufficient statistic and base measure for the Poisson distribution are respectively defined as

$$T(x) = x \quad \text{and} \quad h(x) = I(x \in \mathbb{Z}_{\geq 0})$$

The natural parameter vector and inverse mapping are respectively

$$\eta \equiv \log\lambda \quad \text{and} \quad \lambda = \exp(\eta). \tag{1.12}$$

The log-partition function is

$$A(\eta) = \exp(\eta).$$

The domain of both $A(\eta)$ and $\nabla A(\eta)$ is $H = \eta : \eta \in \mathbb{R}$. The derivative of $A(\eta)$ is

$$(\nabla A)(\eta) = \exp(\eta).$$

1.5.2.5 Negative binomial distribution

Definition 9. *A distributed scalar random variable $x \in \mathbb{Z}_{\geq 0}$ is from the negative binomial distribution with probability of success p and shape parameter $\kappa > 0$ when its density function is*

$$f(x) = \frac{\Gamma(x+\kappa)}{\Gamma(x+1)\Gamma(\kappa)}\Big(\frac{\mu}{\mu+\kappa}\Big)^{x}\Big(\frac{\kappa}{\mu+\kappa}\Big)^{\kappa}, \tag{1.13}$$

which is written as $x \sim NB(\mu, \kappa)$, where $\Gamma(x) = (x-1)!$ and $p = \mu/(\mu+\kappa)$.

The negative binomial distributions is closely related to the Poisson distribution, and begins to resemble it as the shape parameter κ increases to infinity. The sufficient statistic and base measure for the multivariate normal family are respectively defined as

$$T(x) = x \quad \text{and} \quad h(x) = \begin{pmatrix} x+\kappa-1 \\ x \end{pmatrix}.$$

The natural parameter vector and inverse mapping are respectively

$$\eta \equiv \log \mu \quad \text{and} \quad \mu = \exp(\eta). \tag{1.14}$$

The log-partition function is

$$A(\eta) = -\kappa \log\big(1 - \exp(\eta)\big).$$

The domain of both $A(\eta)$ and $\nabla A(\eta)$ is $H = \eta : \eta \in \mathbb{R}$. The derivative of $A(\eta)$ is

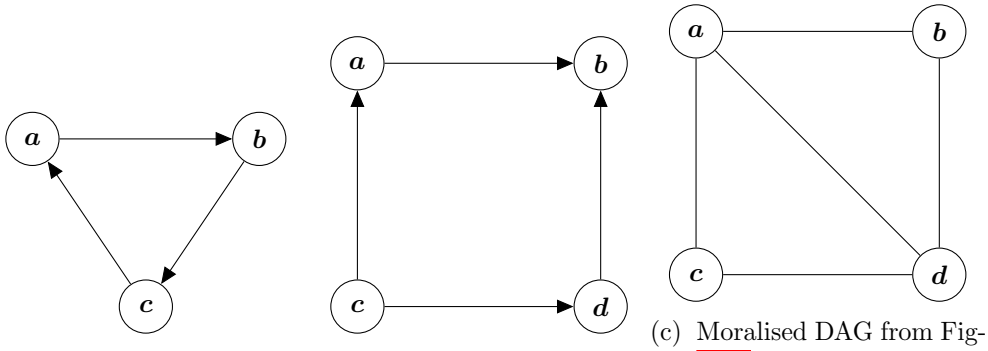
$$(\nabla A)(\eta) = \frac{\kappa \exp(\eta)}{1 - \exp(\eta)}.$$

1.6 Graph theory

Graph theory is vital in understanding message passing and helps to simplify what is an otherwise complex task. In this section we provide a brief review of the required graphical theory.

1.6.1 Directed acyclic graphs

Directed acyclic graphs are a graphical representation of a model consisting of nodes and edges. The nodes are used to represent vectors/matrices, while the lines between them known as edges, demonstrates the relationships between nodes. To be a directed acyclic graph, all edges must be directed and without cycles; that is, there cannot be any connected directed edge, where following the direction of said edge returns to the starting edge (see Figure 1.1b).



(a) A non DAG. (b) A DAG. (c) Moralised DAG from Figure 1.1b.

Figure 1.1: *Three basic graphical models. Figure 1.1b shows a directed acyclic graph (DAG); all edges are directed and without cycles. Conversely, Figure 1.1a is not a directed acyclic graph. Although it contains directed edges, it also contains a cycle. The moralised version of the directed acyclic graphs from Figure 1.1b is Figure 1.1c. No edges have a direction and the parent nodes* $\boldsymbol{a}$ *and* $\boldsymbol{d}$ *are linked.*

Moralisation of directed acyclic graphs help uncover the relationship between nodes. A parent node is the node connected away from the directed edge, while the node on the directed edge is the child node. Co-parents occur when a child node has two edges directed towards it. Specifically, the process of moralisation involves linking parents and co-parents. Consider the moralisation of Figure 1.1b; the child node $\boldsymbol{b}$ has co-parents $\boldsymbol{a}$ and $\boldsymbol{d}$. To moralise this graph, we link nodes $\boldsymbol{a}$ and $\boldsymbol{d}$ with an undirected edge. However, no further links are required for the parent node of $\boldsymbol{a}$ and $\boldsymbol{d}$ (node $\boldsymbol{c}$) since it does not have a co-parent. We then change all directed edges to undirected edges (Figure 1.1c).

The a node's Markov blanket consists of the nodes directly linked to it. Finding the Markov blanket is easy on a moralised graph and can lead to significant simplifications for large graphical models. For example, in our moralised graph, the Markov blanket of node $\boldsymbol{a}$ would be nodes $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$, where as the Markov blanket for node $\boldsymbol{c}$ would be nodes $\boldsymbol{a}$ and $\boldsymbol{d}$ (Pearl, 1988[53]).

1.6.2 Factor graphs

An extension of moralised graphs are factor graphs. Figure 1.2 demonstrates a factor graph corresponding to the model in equation (1.15),

$$h(x_1, x_2, x_3, x_4, x_5) = \sqrt{x_1 + x_2 x_5}\ \log(x_2 + x_3^2 x_4^7)^{1/2} \sin(x_3^2)(|x_5|^2 - 10), \qquad (1.15)$$

where the square nodes are functions in the equation known as factor nodes, and the circle nodes are arguments in the equation known as stochastic nodes. Neighbours of a

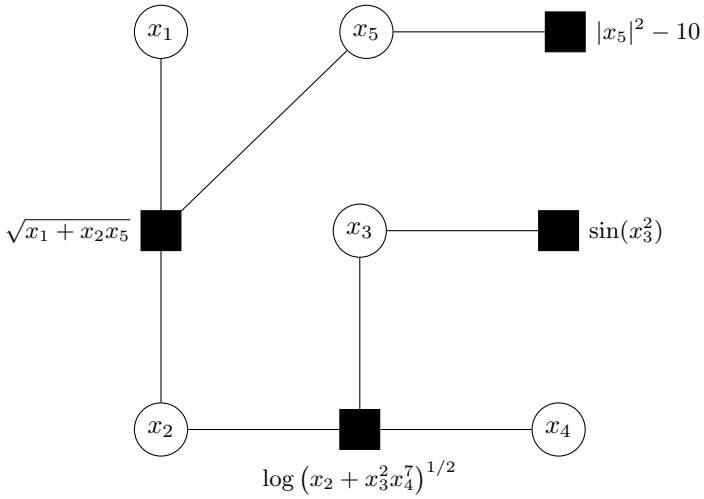


Figure 1.2: *The factor graph corresponding to the model presented in equation (1.15). The two types of nodes in the factor graph are shown here, where square nodes represent factor nodes and circular nodes represent stochastic nodes.*

node are the those nodes linked by an edge. For example, the factor node $\sqrt{x_1 + x_2 x_5}$ in Figure 1.2 is neighbours with stochastic nodes (x_1, x_2, x_5), where as the factor node $|x_5|^2 - 10$ only is neighbours with the stochastic node (x_5) (Rohde & Wand, 2015[58]).

1.7 Multilevel datasets

Often it is not possible or practical to collect data from a single experimental group with one source of variance. As such, real world datasets are often structured in a multilevel or hierarchical manner, whereby they have nested levels stemming from observations occuring within groups (Steenbergen & Jones, 2002[62]). For categorical variable A and B, A is nested in B if each category of A only occurs in one category of B. A variable is defined as a level if its values are a random sample from a wider population of values,

in which case the values it takes are called groups. In other words, groups of a level are a sample from a larger population of groups, where the observations have a distribution (McCulloch, Searle & Neuhaus, 2008[41]). Alternatively, a variable is a level if there is a functional difference between its values (Borenstein, et al., 2009[10]). Each level of the structure contains an unobserved residual component, which accounts for correlation within the structure.

Levels are a consequence of experimental design and data collection. We aim to clarify what constitutes a level by explaining an example. Before we continue with the example, we note variables can have two types of effect on the response. Fixed effects occur when a variable is deemed to have a constant functional effect on the response from a finite set of functional effects. Generally analysts are aware of such variables in studies. Random effects occur when variables are deemed to have differing functional effects from an infinite set of functional effects. They are assumed to be a sample of a random categorical variables from some distribution and are caused by unknown variables (McCulloch, Searle & Neuhaus, 2008[41]). It is often not immediately obvious when variables have fixed or random effects, with inference being a key motivating factor. Additionally there is some overlap between variables that are levels and cause random effects. We aim to clarify this in our example as well.

Consider a school that wants to compare the effect of time spent studying on pupils' exam scores for different subjects. Because class sizes are limited the school must collect pupils' exam scores from four classes (Raudenbush, 1993[57]). The structure of this data is called a one level dataset and is shown in Figure 1.3, where pupils are nested in classes. Time spent studying is a continuous variable and no assumption is made that the number of hours studied are random, so it is likely similar values will have a similar effect on exam score. With this in mind the effect of time spent studying on pupils' exam scores is fixed and it is not level in the dataset. Although the effect of subject on exam score is not continuous or random, even in the case where subjects are chosen at random, it is more than likely their effects are explainable and should be considered fixed effects. This is also evidence they are not a level in the study.

As mentioned earlier, inference is an important consideration when determining levels, as well as fixed and random effects. If inference from the study is restricted to the four classes in the study only, the classes represent the population of classes in the study. In other words, they have a finite set of fixed functional effects and are not a level in the dataset (they should be included as a fixed effect). Alternatively, if inference from the study is extended to the population of students in the school, it is reasonable to assume classes in the study are a random sample from the wider population of classes in the

school (McCulloch, Searle & Neuhaus, 2008[41]). These classes come from a probability distribution regarding their relationship with exam score. As such, in this case class is a level with four groups (one for each class). It should be clear that experimental design, data interpretation and inference all play a role in determining whether variables are a level and the type of effect they have on the repsonse.

In this thesis we cover two other types of multilevel datasets which we now explain briefly with examples. The natural extension of the one level dataset explained above is a two level dataset, where an additional level of nesting is added. Continuing the exam score example, consider exam scores measured on pupils of different classes from multiple schools in an area. Assuming inference is now extended to the population of pupils in all classes of schools in the area, it is reasonable to assume both schools and classes have random effects with their own distributions. This is a two level dataset where pupils are nested in classes nested in schools, as shown in Figure 1.4.

Alternatively, datasets with two levels can be crossed. For two arbitrary levels B and C are crossed if each observation nested from each groups in level C occur in different groups of level B. Following our exam score example, consider exam scores are measured on pupils of different classes in the same school, however the pupils of each class live in different areas. Assuming inference is extended to both classes and the areas where pupils live, a crossed dataset results as shown in Figure 1.5, where pupils are nested in classes and classes crossed with area. As before, both levels have random effects with their own distributions. Note, although it the crossed example has two levels, the first level is not nested in the second level. Instead, they both nest pupils and can be thought of as structurally at the same level in the data.

Because mutlilevel datasets are composed of non-homogeneous experimental units we must model them accordingly. Failing to do so can facilitate incorrect model interpretation. Linear mixed models provide a solution to modeling multilevel datasets composed of non-homogeneous experimental units. They are a form of regression model which gained popularity with the rise of computing power. These models improve on linear models by allowing an intercept and slope for the population effects (known as fixed effects) and a unique intercept and slope to account for the effect of each experimental unit (known as random effects).

Perhaps the clearest advantage of mixed models is when they are used for prediction (Gelman, 2006[18]). Mixed models allow for the estimation of group effects simultaneously with the effects of group-level predictors. This is not possible in fixed effects and ANOVA models, where the inferences are limited to the groups in the sample. With

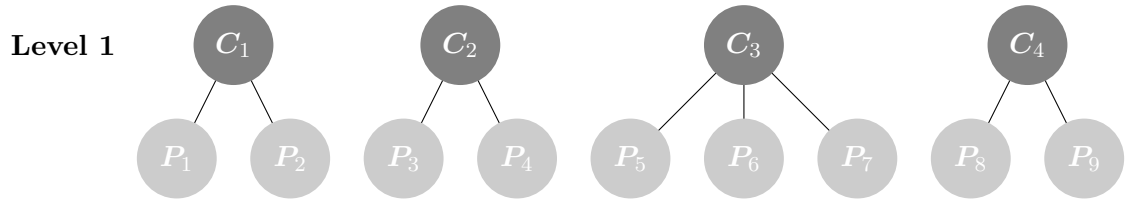


Figure 1.3: *Example of a two level dataset following the schools example. We denote* $\boldsymbol{P}_j$ *as the pupils in the classes and* $\boldsymbol{C}_i$ *the classes in the schools.*

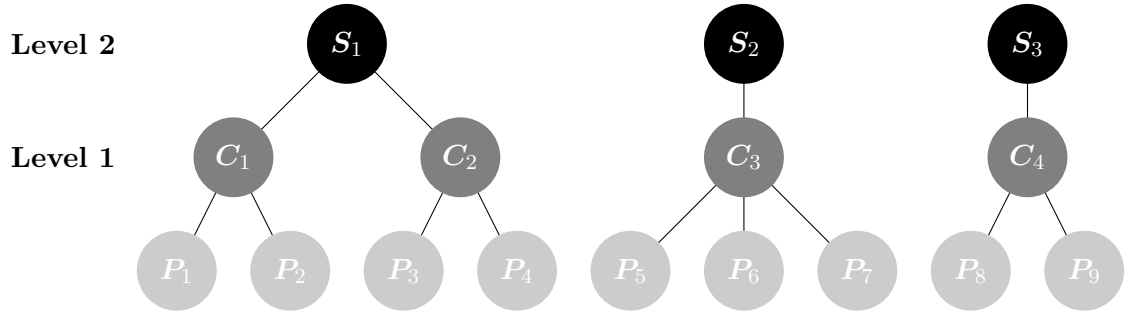


Figure 1.4: *Example of a two level dataset following the schools example. We denote* $\boldsymbol{P}_j$ *as the pupils in the classes,* $\boldsymbol{C}_i$ *the classes in the schools and* $\boldsymbol{S}_k$ *the schools.*

these traits in mind, it is clear that multilevel models provide a significant advantage over traditional regression techniques for prediction and data reduction.

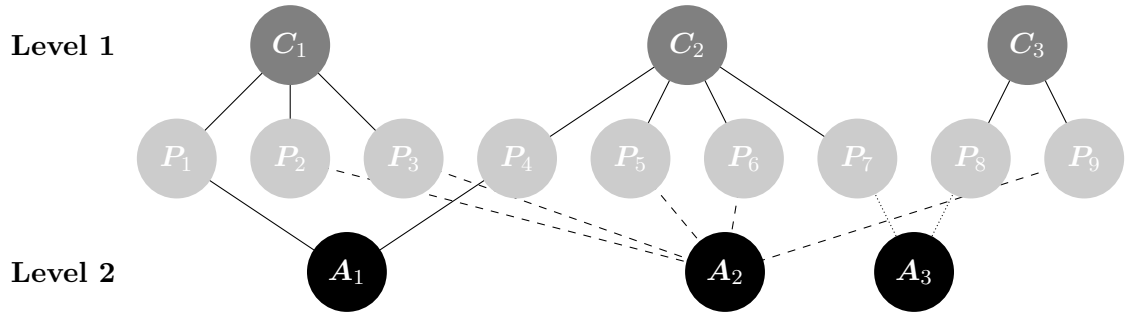


Figure 1.5: *Example of a crossed dataset following the schools example. We denote* $\boldsymbol{P}_j$ *as the pupils in the classes,* $\boldsymbol{C}_i$ *the classes in the schools and* $\boldsymbol{A}_k$ *the area.*

1.8 Generalised linear mixed models

Generalised linear mixed models are a powerful model that facilitate inference from multilevel datasets with non-normal response variables. Much like generalised linear models, a linear predictor η is used to incorporate information about the independent variables into the model. However, GLMMs include fixed and random effects in the linear predictor to account for error structures in the data, similar to linear mixed models. For GLMMs it is optional whether fixed intercepts and slopes are included in the linear predictor, however they must always include a random effect, via either an intercept or slope. The relationship between these model types is more clearly demonstrated in Table 1.1. The expected value of the response y conditional on the random effects is related to a linear predictor using a link function $g(.)$, where $F(.) = g^{-1}(.)$ is the inverse link function, also called the mean function. In other words, although the mean is not directly a linear combination of predictors, some function of the mean is. We now show general models for the three data structures explained in the previous subsection.

For a dataset with one level of nesting analogous to Figure 1.3, where level one has

Table 1.1: *The relationship between four types of linear models, where LM refers to linear models, GLM refers to generalised linear models, LMM refers to linear mixed models and GLMM refers to generalised linear mixed models.*

	Response is normally distributed	Response is not normally distributed
Fixed effects only	**LM**	**GLM**
Fixed and random effects	**LMM**	**GLMM**

m groups and n_i measurements in each group, assuming the effect of each group is normally distributed and the distribution of the response y_{ij} conditional on the random effects is from an exponential family f, the mixed model has the general form

$$y_{ij}|\boldsymbol{u}_i \overset{\text{ind.}}{\sim} f\big(F(y_{ij}|\boldsymbol{u}_i)\big), \quad \boldsymbol{u}_i \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}),$$
$$1 \leq i \leq m, \quad 1 \leq j \leq n_i. \tag{1.16}$$

The linear predictor is related to the expected value of the response by the link function F

$$g\big(E(y_{ij}|\boldsymbol{u}_i)\big) = \eta_{ij}, \quad \text{where} \quad E(y_{ij}|\boldsymbol{u}_i) = F(\eta_{ij}).$$

For the one level mixed model the linear predictor is

$$\eta_{ij} = \boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}},$$

where $\boldsymbol{x}_{ij}^{\mathbf{F}}$ is a $d^{\mathbf{F}} \times 1$ vector of predictors modelled as having fixed effects with coefficient vector $\boldsymbol{\beta}$, and $\boldsymbol{x}_{ij}^{\mathbf{R}}$ is a $d^{\mathbf{R}} \times 1$ vector of predictors modelled as having random effects with coefficient vectors $\boldsymbol{u}_i$. Note, when the first entry of $\boldsymbol{x}_{ij}^{\mathbf{F}}$ or $\boldsymbol{x}_{ij}^{\mathbf{R}}$ equals 1, it corresponds to including a fixed or random intercept respectively.

For a dataset with two levels of nesting analogous to Figure 1.4, consisting of m outer groups, n_i inner groups in each outer group, and o_{ij} measurements each in each inner group, assuming the random effects $\boldsymbol{u}_i^{\mathbf{L}1}$ of the inner group and $\boldsymbol{u}_{ij}^{\mathbf{L}2}$ of the outer group are independent and identically distributed as normal, and the distribution of the response y_{ijk} conditional on the random effects is from an exponential family f, the mixed model has the general form

$$y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2} \overset{\text{ind.}}{\sim} f\big(F(y_{ij}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})\big),$$
$$\boldsymbol{u}_i^{\mathbf{L}1} \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}1}}, \boldsymbol{\Sigma}^{\mathbf{L}1}) \quad \text{independently of} \quad \boldsymbol{u}_{ij}^{\mathbf{L}2} \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}2}}, \boldsymbol{\Sigma}^{\mathbf{L}2}),$$
$$1 \leq i \leq m, \quad 1 \leq j \leq n_i, \quad 1 \leq k \leq o_{ij}. \tag{1.17}$$

The linear predictor is related to the expected value of the response by the link function F

$$g\big(E(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})\big) = \eta_{ijk} \quad \text{where} \quad E(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}) = F(\eta_{ijk}).$$

For the two level mixed model the linear predictor is

$$\eta_{ijk} = \boldsymbol{\beta}^\top \boldsymbol{x}_{ijk}^{\mathbf{F}} + (\boldsymbol{u}_i^{\mathbf{L}1})^\top \boldsymbol{x}_{ijk}^{\mathbf{R}1} + (\boldsymbol{u}_{ij}^{\mathbf{L}2})^\top \boldsymbol{x}_{ijk}^{\mathbf{R}2},$$

where $\boldsymbol{x}_{ijk}^{\mathbf{F}}$ is a $d^{\mathbf{F}} \times 1$ vector of predictors, modelled as having fixed effects with coefficient vector $\boldsymbol{\beta}$, $\boldsymbol{x}_{ijk}^{\mathbf{R}1}$ is a $d^{\mathbf{R}1} \times 1$ vector of predictors modelled as having random effects from

the outer groups with coefficient vectors $\boldsymbol{u}_i^{\mathbf{L}1}$, and $\boldsymbol{x}_{ijk}^{\mathbf{R}2}$ is a $d^{\mathbf{R}2} \times 1$ vector of predictors modelled as having random effects from the inner groups with coefficient vectors $\boldsymbol{u}_{ij}^{\mathbf{L}2}$. Note, when the first entry of $\boldsymbol{x}_{ijk}^{\mathbf{F}}$, $\boldsymbol{x}_{ijk}^{\mathbf{R}1}$ or $\boldsymbol{x}_{ijk}^{\mathbf{R}2}$ equals 1, it corresponds respectively to including a fixed or random intercept for the outer or inner groups.

For a dataset with crossed levels of nesting analogous to Figure 1.5, consisting of two levels of m and m' groups, with observations indexed according to the pair $(i, i') \in \{1, \ldots, m\} \times \{1, \ldots, m'\}$, and $n_{ii'}$ observations in the (i, i') pair, assuming the random effects of each level denoted by $\boldsymbol{u}_i$ and $\boldsymbol{u}'_{i'}$ are independent and identically distributed as normal, and the distribution of the response $y_{ii'j}$ conditional on the random effects is from an exponential family f, the crossed mixed model has the general form

$$\begin{gathered} y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'} \overset{\text{ind.}}{\sim} f\big(F(y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'})\big), \\ \boldsymbol{u}_i \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}) \quad \text{independently of} \quad \boldsymbol{u}'_{i'} \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}'}}, \boldsymbol{\Sigma}'), \\ 1 \leq i \leq m, \quad 1 \leq i' \leq m', \quad 1 \leq j \leq n_{ii'}. \end{gathered} \tag{1.18}$$

The linear predictor is related to the expected value of the response by the link function F

$$g\big(E(y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'})\big) = \eta_{ii'j} \quad \text{where} \quad E(y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'}) = F(\eta_{ii'j}).$$

For the crossed mixed model the linear predictor is

$$\eta_{ii'j} = \boldsymbol{\beta}^\top \boldsymbol{x}_{ii'j}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ii'j}^{\mathbf{R}} + (\boldsymbol{u}'_{i'})^\top \boldsymbol{x}_{ii'j}^{\mathbf{R}'},$$

where $\boldsymbol{x}_{ii'j}^{\mathbf{F}}$ is a $d^{\mathbf{F}} \times 1$ vector of predictors, modelled as having fixed effects with coefficient vector $\boldsymbol{\beta}$, $\boldsymbol{x}_{ii'j}^{\mathbf{R}}$ is a $d^{\mathbf{R}} \times 1$ vector of predictors modelled as having random effects from the groups of the first level with coefficient vectors $\boldsymbol{u}_i$, and $\boldsymbol{x}_{ii'j}^{\mathbf{R}'}$ is a $d^{\mathbf{R}'} \times 1$ vector of predictors modelled as having random effects from the groups of the second level with coefficient vectors $\boldsymbol{u}'_{i'}$. Note, when the first entry of $\boldsymbol{x}_{ii'j}^{\mathbf{F}}$, $\boldsymbol{x}_{ii'j}^{\mathbf{R}}$ or $\boldsymbol{x}_{ii'j}^{\mathbf{R}'}$ equals 1, it corresponds respectively to including a fixed intercept, or a random intercept for the first or second groups.

Although other choices exist, selecting the link function F as the canonical link for the conditional response distribution f is a good default and provides a number of desirable properties. We now present distributions and link functions for binary and count response data on an arbitrary model structure. The distributions and links that follow can be applied to the random effects structures presented.

1.8.1 Binary response models

When dealing with binary response variables where $y \in \{0, 1\}$, it is common to assume the response is Bernoulli distributed where

$$y|u \sim \text{Bernoulli}\big(F(\eta)\big).$$

For binary outcomes, several popular inverse link functions exist, however we focus on two:

$$F = \begin{cases} \text{expit} & \text{logistic inverse link,} \\ \Phi & \text{probit inverse link} \end{cases}$$

where expit and Φ are defined in Section 1.5.2. The expit inverse link (inverse of the logit link) function is the most frequently used link when modelling binary data. It affords an elegant log-odds model interpretation as well as algebraic simplifications due to it being the canoncial link of the Bernoulli distribution. Although Probit links often facilitate more tractable solutions, their model coefficients do not have a direct interpretation in the same way that logit links do, and inference is restricted. As a side note, although special cases of grouped data with responses proportional to the number of observations per group exist, they are uncommon and as such are not discussed in this thesis.

1.8.2 Count response models

When modelling count variables where $y \in \mathbb{Z}_{\geq 0}$, two models are commonly used. The Poisson model assumes the response is from a Poisson distribution as

$$y|u \overset{\text{ind.}}{\sim} \text{Poisson}\big(F(\eta)\big),$$

where the inverse link is the canonical function $F = \exp$. As Poisson distributed variables are constrained to have an equal mean and variance, it is not appropriate for over-dispersed data where the variance is greater than the mean.

The negative binomial model is similar to the Poisson model but includes an additional parameter κ to account for over-dispersed data. It can be written as

$$y|u \overset{\text{ind.}}{\sim} \text{NB}\big(F(\eta), \kappa\big),$$

where as before the inverse link is the canonical function $F = \exp$. As the shape parameter $\kappa \to \infty$ the negative binomial distribution converges to the Poisson distribution.

1.9 Maximum likelihood

1.9.1 Likelihood functions

The likelihood of a model is the probability of observing the data given parameter inputs. It is a function of the parameters where random variables are fixed at the observed values. For a statistical model with probability density function $f(y)$ for data y and unknown parameters $\boldsymbol{\theta}$, the likelihood function is equivalent to the probability density function of the observed data y,

$$L(\boldsymbol{\theta}; y) = f(y; \boldsymbol{\theta}),$$

with a log-likelihood function given by (Collins, 2008[16])

$$\ell(\boldsymbol{\theta}; y) = \log L(\boldsymbol{\theta}).$$

In the case of m independent observations of y_i

$$\ell(\boldsymbol{\theta}; y) = \sum_{i=1}^{m} \log f(y_i; \boldsymbol{\theta}).$$

Note that by using the log-likelihood we sum over the independent observations as opposed to taking their product, which helps reduce numerical instability.

For the general one level model given in equation (1.16), the joint log-likelihood function for parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ is

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{i=1}^{m} \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \prod_{j=1}^{n_i} p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta}) p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) d\boldsymbol{u}_i,$$

where

$$p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) = f\big(F(y_{ij}|\boldsymbol{u}_i)\big) \quad \text{and} \quad p(\boldsymbol{u}_i;\boldsymbol{\Sigma}) = \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}}},\boldsymbol{\Sigma}).$$

For the general two level model given in equation (1.17), the log-likelihood for the parameters $(\boldsymbol{\beta},\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})$ is

$$\ell(\boldsymbol{\beta},\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2}) = \sum_{i=1}^{m} \log \int_{\mathbb{R}^{d^{\mathbf{R}1}+d^{\mathbf{R}2}}} \prod_{j=1}^{n_i}\prod_{k=1}^{o_{ij}} p\left(y_{ijk}\middle|\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix};\boldsymbol{\beta}\right) \times p\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2}\right) d\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix},$$

where

$$p\left(y_{ijk}\middle|\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix};\boldsymbol{\beta}\right) \equiv f\big(F(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\big)$$

and

$$p\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2}\right) \equiv \boldsymbol{N}\left(\begin{bmatrix}\boldsymbol{0}_{d^{\mathbf{R}1}}\\ \boldsymbol{0}_{d^{\mathbf{R}2}}\end{bmatrix},\begin{bmatrix}\boldsymbol{\Sigma}^{\mathbf{L}1} & \boldsymbol{0}_{d^{\mathbf{R}1}\times d^{\mathbf{R}2}}\\ \boldsymbol{0}_{d^{\mathbf{R}2}\times d^{\mathbf{R}1}} & \boldsymbol{\Sigma}^{\mathbf{L}2}\end{bmatrix}\right).$$

For the general crossed random effects model given in equation (1.18), the log-likelihood for the parameters $(\boldsymbol{\beta},\boldsymbol{\Sigma},\boldsymbol{\Sigma}')$ may be written as

$$\ell(\boldsymbol{\beta},\boldsymbol{\Sigma},\boldsymbol{\Sigma}') = \log \int_{\mathbb{R}^{md^{\mathbf{R}}+m'd^{\mathbf{R}'}}} \prod_{(i,i'):n_{ii'}>0}\prod_{j=1}^{n_{ii'}} p\left(y_{ii'j}\middle|\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix};\boldsymbol{\beta}\right) \times p\left(\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix};\boldsymbol{\Sigma},\boldsymbol{\Sigma}'\right) d\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix},$$

where

$$p\left(y_{ii'j}\middle|\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix};\boldsymbol{\beta}\right) \equiv f\big(F(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\big)$$

and

$$p\left(\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix};\boldsymbol{\Sigma},\boldsymbol{\Sigma}'\right) \equiv \boldsymbol{N}\left(\begin{bmatrix}\boldsymbol{0}_{md^{\mathbf{R}}}\\ \boldsymbol{0}_{m'd^{\mathbf{R}'}}\end{bmatrix},\begin{bmatrix}\boldsymbol{I}_m\otimes\boldsymbol{\Sigma} & \boldsymbol{0}_{md^{\mathbf{R}}\times m'd^{\mathbf{R}'}}\\ \boldsymbol{0}_{m'd^{\mathbf{R}'}\times md^{\mathbf{R}}} & \boldsymbol{I}_{m'}\otimes\boldsymbol{\Sigma}'\end{bmatrix}\right).$$

1.9.2 Maximum likelihood

Parameters at the peak of a likelihood function hypersurface can be interpreted as those that give the maximum probability of observing the data obtained under the fitted model. This process forms a frequentist probablistic framework called maximum likelihood estimation. By taking $L(\boldsymbol{\theta}|y)$ as a function of $\boldsymbol{\theta}$, the goal of maximum likelihood estimation is to return the parameters that maximise the $L(\boldsymbol{\theta}|y)$ (denoted by $\hat{\boldsymbol{\theta}}$).

Generally, maximum likelihood estimates can be solved by differentiating the likelihood function with respect to parameters and solving to find the root. The maxima of the likelihood and log-likelihood are equivalent, although as mentioned it is often preferable to work with the log-likelihood for several reasons. As such, the maximum likelihood estimates are given by

$$\frac{d\ell}{d\boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0}_d.$$

Maximum likelihood estimates are consistent and asymptotically normally distributed, with mean equal to $\boldsymbol{\theta}$ and variance equal to the inverse Fisher information (McCulloch, Searle & Neuhaus, 2008[41]),

$$\hat{\boldsymbol{\theta}} \sim \boldsymbol{N}\big(\boldsymbol{\theta}, \boldsymbol{I}(\hat{\boldsymbol{\theta}})^{-1}\big).$$

Two important issues arise from this. Firstly, the range of parameters being estimated must be considered. Specifically, in the case of variance parameters which occur only on the positive real number line, the search space must be constrained. Second, analytic solutions are not always tractable or practical to obtain. For models with parameters of varying dimensions derivative free options become more appealing.

1.10 Best prediction

GLMMs are often used for predicting values of random effects, despite the fact they are unobservable. The best predictor (BP) is the predictor with the lowest mean squared error. That is, for an arbitrary GLMM with data y_{ij}, the best predictor of the random effect $\boldsymbol{u}_i$ is

$$\mathrm{BP}(\boldsymbol{u}_i) \equiv \underset{\boldsymbol{u}_i \in \mathbb{R}}{\operatorname{argmin}}\ E\Big[\big(\boldsymbol{u}_i - \boldsymbol{u}\big)^2\Big].$$

It is straightforward to show the best predictor is the average of the random effect (Scott, Simonoff & Marx, 2013[60])

$$\text{BP}(\boldsymbol{u}_i) = E(\boldsymbol{u}_i|y_{ij}).$$

This can be calculated using the conditional specification

$$\text{BP}(\boldsymbol{u}_i) = \int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}_i \left(\frac{f(y_{ij}|\boldsymbol{u}_i) f(\boldsymbol{u}_i)}{\int_{\mathbb{R}^{d^{\mathbf{R}}}} f(\boldsymbol{y}_i|\boldsymbol{u}_i) f(\boldsymbol{u}_i)} \right) d\boldsymbol{u}_i.$$

McCulloch & Neuhaus (2012)[42] show prediction accuracy measured by mean square error is robust to moderate violations of the assumed random effects distribution. This suggests that for prediction, inference is relatively immune to variations of assumptions. Since it is difficult to confirm whether or not the assumptions of the random effects distribution are met, this attribute of best predictors is extremely useful.

A number of prediction schemes such as best linear predictors and best linear unbiased predictors. Best linear predictors are useful when the full PDF is not available as it only requires the first and second order moments. Unlike best predictors, restrictive assumptions are imposed in best linear predictors to minimise the mean squared error. Although these assumptions may result computation of results, they also ensure that best linear predictors never has a smaller mean squared error than best predictors. Best linear unbiased predictors are a subclass of best linear predictors, and as such their mean squared error never has a smaller mean squared error than best linear predictors. Sections 2 to 3 of Teunissen (2007).[63] provide a good overview of the differences between predictors and we refer interested readers there.

1.11 Current approximation methods

The integrals involved with calculating the likelihood surface and best predictors of GLMMs do not have tractable solutions. In the frequentist GLMM setting the two standard approaches to solving intractable integrals are Laplace approximations and Gauss-Hermite quadrature. We now provide a brief overview of each method.

1.11.1 Laplace approximation

Laplace approximations were established in 1774 and have since gone on to become a fundamental technique in mathematics and statistics. These approximations are in effect a second order Taylor series approximation around the maximum of the function to be approximated. This approximation allows the integral to be expressed as a Gaussian distribution with an analytic solution.

Consider an integrals of the form

$$I(x) = \int_a^b \exp\big(f(x)\big)dx,$$

where the function $f(x)$ has a maxima at x_0 such that $a < x_0 < b$ and $f''(x_0) < 0$. A second order Taylor expansion of $f(x)$ around x_0 is

$$f(x) \approx f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2}(x - x_0)^2 f''(x_0).$$

Since $f'(x_0) = 0$

$$f(x) = f(x_0) - \frac{1}{2}(x - x_0)^2\big(-f''(x_0)\big).$$

It can be seen that $\exp\big(f(x)\big)$ is expressible as a normal density function with mean x_0 and variance $-f''(x_0)^{-1}$. Given the assumptions imposed on $f(x)$

$$\int_a^b \exp\big(f(x)\big)dx \approx \exp\big(f(x_0)\big)\int_a^b \exp\left(\frac{(x - x_0)^2}{2f''(x_0)^{-1}}\right)dx.$$

When the integral on the right side of the previous equation is evaluated between $-\infty$ and ∞ it becomes a Gaussian integral, thus

$$\int_a^b \exp\big(f(x)\big)dx \approx \exp\big(f(x_0)\big)\sqrt{-2\pi f''(x_0)}.$$

For an arbitrary GLMM with log-likelihood function of parameters $\boldsymbol{\theta}$

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{m} \ell_i(\boldsymbol{\theta})$$

where

$$\ell_i(\boldsymbol{\theta}) = \int_{\mathbb{R}^{d^{\mathbf{R}}}} \prod_{j=1}^{n_i} f(y_{ij}|\boldsymbol{u}_i)f(\boldsymbol{u}_i)d\boldsymbol{u}_i,$$

the integral can be expressed as

$$\begin{aligned}\ell_i(\boldsymbol{\theta}) &= \int_{\mathbb{R}^d} \exp\left\{\sum_{j=1}^{n_i} \log\left(f(y_{ij}|\boldsymbol{u}_i)f(\boldsymbol{u}_i)\right)\right\} d\boldsymbol{u}_i \\ &= \int_{\mathbb{R}^d} \exp \tilde{f}(\boldsymbol{u}_i) d\boldsymbol{u}_i,\end{aligned}$$

where

$$\tilde{f}(\boldsymbol{u}_i) = \sum_{j=1}^{n_i} \log f(y_{ij}|\boldsymbol{u}_i)f(\boldsymbol{u}_i).$$

It can subsequently solved as shown above. Laplace approximation provides fast inference for GLMMs, performing well in cases where the number of observations of each group is large, i.e. large n_i. However, they are less accurate in cases where the number of observations of each group are low and the random effects have high variance. As such, it is worthwhile considering alternative approximation methods.

1.11.2 Gauss Hermite quadrature

Perhaps the simplest and most robust method of numerical integration, quadrature facilitates an approximate calculation of definite integrals. Gauss-Hermite quadrature (GHQ) approximates definite integrals from the normal family or with a log-quadratic factor by a weighted sum. For polynomials of degree $2m-1$ GHQ provides an exact result or less over the domain $[-1, 1]$ by focusing on selection of m optimal nodes x_i. Each node corresponds to the roots of an mth order Hermite polynomial $H_m(x)$ and is accompanied by a weight w_i defined by the Gauss-Hermite weight function (Liu & Pierce, 1994[36])

$$w_i = \frac{2^{m-1} m! \sqrt{\pi}}{m^2 \big(H_{m-1}(x_i)\big)^2}.$$

The nodes are symmetric about zero, where the range increases with m. The integral is then computed as the weighted sum of function values $f(x_i)$ at these points

$$\int_{-\infty}^{\infty} f(x)\phi(x)dx = \sum_{i=1}^{m} w_i f(x_i).$$

Note, that since Gaussian quadrature works over the domain $[-1, 1]$, a change of interval is required from $[a, b]$. Implementation of the Gaussian quadrature rule suggests the

integral can be approximated as follows

$$\int_a^b f(x)\phi(x)dx \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}\, x_i + \frac{a+b}{2}\right).$$

Adaptive quadrature follows a similar process to the one for traditional quadrature, however also implements an accuracy criterion based on the difference between two subintervals. If a large discrepancy exists between two intervals (i.e. there is a large amount of error between them), the subinterval is re-divided into two smaller subintervals and recalculated. It is this process which improves the accuracy of adaptive quadrature over traditional methods for poorly behaved functions. Additionally, the computational burden of this refinement process is reduced by implementing the accuracy criterion only when needed for smaller subintervals. The R function "`integrate()`" in the "stats" package allows for easy implementation of the adaptive quadrature via the Gauss-Kronrod method. Gauss-Kronrod quadrature extends Gaussian quadrature by adding $n + 1$ points to an n-point rule, such that the resulting rule is a polynomial of degree $2n + 1$. As such, a set of function evaluation points can be created, where the extra points (corresponding to the Kronrod extension) allow the computation of higher-order estimates, and the function values at the set points (corresponding to the Gaussian quadrature rule) provide lower-order estimates. The difference between these values forms the accuracy criterion previously discussed.

A major advantage of GHQ over other methods of quadrature is that once weights and nodes are calculated they can be stored to reduce computational costs of future calculations. For GLMMs the complexity of GHQ increases with the dimension of random effects. It additionally becomes inaccurate if the dimension of the random effects is greater than two due to limitations in the method used to factorise the high-dimensional integrals into a lower-dimensional one. As such it is not suitable for crossed models and models with more than two levels (Handayani, et al., 2017[27]).

1.11.3 Other methods

In addition to the previous a number of other methods exist. Penalised quasi-likelihood (PQL) (Breslow & Clayton, 1993[11]) is one of the most popular and high speed methods for handling the difficult integrals arising in GLMMs. PQL is useful in cases where we are missing information regarding the distribution of data and as such cannot obtain a full likelihood function, in which case a quasi-likelihood function can be used with a

penalty on the random effects component. The penalty ensures values of the random effects have a mean of zero. Laplace approximation can then be used to solve the intractable integral which arises. Although it provides fast inference and is robust to model misspecification it is not accurate and provides biased estimates when there is little data per group, such as in binary or low count data cases. More importantly, PQL calculates a quasi-likelihood rather than the true likelihood and as such may not be appropriate in situations where likelihood ratio testing is implemented.

Although traditional methods such as Markov chain Monte Carlo provide highly reliable estimates of intractable integrals for Bayesian models, there is little literature regarding implementation in a frequentist framework. Lele, et al. (2007)[35] present a reformulation of Markov chain Monte Carlo, known as data cloning, which allows maximum likelihood and confidence interval calculation. This method involves building a fully specified Bayesian model of the problem with uninformative priors and creating a large number of copies of the data which are assumed to be independent. The posterior is then calculated with the usual Markov chain Monte Carlo approach and the likelihood over the copies is used as the data. The mean of the resulting posterior distribution is equal to the maximum likelihood estimate and the number of copies times the variance of the posterior is equal to the variance of maximum likelihood estimate. Unlike GHQ and Laplace approximation, it easily extends to consider multiple random effects. However, this approach involves several difficult technical details. Additionally, although heuristic framework for data cloning exist, current software packages do not yet support the looping facility required for its fitting. Furthermore, it is not widely proven and still bares inherent issues of Markov chain Monte Carlo algorithms such as being computationally intensive.

1.12 Expectation propagation

Expectation propagation (EP) is a Bayesian iterative algorithm used for the computation of intractable posterior distributions. While Opper & Winther (2000)[52] first provided the scheme for Gaussian approximating families, Minka (2001)[43] provided the generalised scheme of the EP algorithm for all exponential families. EP is a reinterpretation of assumed density filtering (Opper, 1999[51]) such that the posterior approximation is no longer dependent on the order of data. Although this makes EP

more computationally intensive than assumed density filtering, EP is more accurate and consistent. Both algorithms are examples of mean field approximations (a common type of variational inference). These algorithms rely on approximating intractable posteriors by selecting approximate densities from tractable families by minimising a distance measure. This divergence measure and the choice of approximating density family differentiate methods of mean field approximations. The Kullback Leibler (KL) divergence is a popular distance measure used in mean field approximations, which we now explain.

1.12.1 Kullback Leiber divergence and projection

The KL divergence is defined as a measure of distance between two density functions f_1 and f_2 such that

$$\mathrm{KL}(f_1 \| f_2) = \int_{\mathbb{R}^d} f_1(\boldsymbol{x}) \log\left(\frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})}\right) + f_2(\boldsymbol{x}) - f_1(\boldsymbol{x}) d\boldsymbol{x}, \tag{1.19}$$

where both densities are on $\mathbb{R}^d$. In other words, a KL divergence of zero indicates the two density functions are identical. The two rightmost terms of equation (1.19) are a correction factor for unnormalised densities, thus in the case of normalised densities the KL divergence can be simplified to

$$\mathrm{KL}(f_1 \| f_2) \equiv \int_{\mathbb{R}^d} f_1(\boldsymbol{x}) \log\left(\frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})}\right) d\boldsymbol{x}.$$

By Gibb's inequality it can be shown that $\mathrm{KL}(f_1 \| f_2) \geq 0$. It is also easy to show that KL divergence is an asymmetric measure where $\mathrm{KL}(f_1 \| f_2) \neq \mathrm{KL}(f_2 \| f_1)$.

The KL projection of the density function f onto a family Q is defined as the distribution q closest to f in the family of density functions Q,

$$\mathrm{proj}[f] \equiv \underset{q \in Q}{\operatorname{argmin}}\ \mathrm{KL}(f \| q). \tag{1.20}$$

This is a difficult problem to solve without placing any constraints on the family Q. However, by constraining Q to be an exponential family density function the problem is

simplified to moment-matching. Suppose

$$Q = \Big\{ q : q(\boldsymbol{x}) = \exp\big(\boldsymbol{T}(x)^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\big) h(x), \quad \boldsymbol{\eta} \in \boldsymbol{H} \Big\}.$$

Then projection of f onto q is solved by

$$\text{proj}[f] = \exp\big(\boldsymbol{T}(x)^\top \boldsymbol{\eta}^* - A(\boldsymbol{\eta}^*)\big) h(x),$$

where $\boldsymbol{\eta}^*$ is

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta} \in \boldsymbol{H}}{\text{argmin}} \bigg(A(\boldsymbol{\eta}) - \boldsymbol{\eta}^\top \int_{-\infty}^{\infty} \boldsymbol{T}(x) f(x) dx \bigg).$$

However, since we are working with the exponential family, it can be shown that the derivative vector of $A(\boldsymbol{\eta})$ equates to the expectation of the natural statistic and thus $\boldsymbol{\eta}^*$ is the solution to

$$\int_{-\infty}^{\infty} \boldsymbol{T}(x) f(x) dx = \int_{-\infty}^{\infty} \boldsymbol{T}(\boldsymbol{x}) \exp\big(\boldsymbol{T}(x)^\top \boldsymbol{\eta} - A(\boldsymbol{\eta}^*)\big) h(x) dx.$$

That is, $\boldsymbol{\eta}^*$ is chosen such that f and $\text{proj}[f]$ have the same natural statistic moments. When the approximating family is Gaussian, Result 1 of Kim & Wand (2016)[31] follows:

Result 1. *Let x be a non-degenerate random variable with density function f. The KL projection of a density function f onto the normal family, denoted by $proj_N[f]$, is the $N\big(\mu^*, (\sigma^2)^*\big)$ density function where*

$$\mu^* = E(x) \quad \textit{and} \quad (\sigma^2)^* = E(x^2) - \big(E(x)\big)^2.$$

Result 1 is easily extended to the multivariate normal distribution as in Result 2. That is, the projection is chosen to be the d-variate normal density function with the same mean vector and covariance as f (Kim & Wand, 2017[68]).

Result 2. *Let $\boldsymbol{x}$ be a non-degenerate random variable with density function f. The KL projection of a d-variate density function f onto the multivariate normal family, denoted by $proj_{\boldsymbol{N}}[f]$, is the $\boldsymbol{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ density function where*

$$\boldsymbol{\mu}^* = E(\boldsymbol{x}) \quad \textit{and} \quad \boldsymbol{\Sigma}^* = E(\boldsymbol{x}\boldsymbol{x}^\top) - E(\boldsymbol{x})E(\boldsymbol{x})^\top.$$

1.12.2 Mean field approximations

We now provide a brief overview of mean field approximations and how popular variational methods such as mean field variational Bayes and belief propagation are related to EP. For observed data $\boldsymbol{x}$ and parameter vector $\boldsymbol{\theta}$ consider approximations to the joint posterior density function $f(\boldsymbol{\theta}|\boldsymbol{x})$ of the form

$$f(\boldsymbol{\theta}|\boldsymbol{x}) \approx q^*(\boldsymbol{\theta}),$$

where $q^*(\boldsymbol{\theta}) = \prod_{i=1}^m q^*(\boldsymbol{\theta}_i)$ and $(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_m)$ is a partition of $\boldsymbol{\theta}$ and the $q^*(\boldsymbol{\theta}_i)$s are chosen to minimise the KL divergence of $f(\boldsymbol{\theta}|\boldsymbol{x})$ from a product density over the elements of the partition, that is

$$q^*(\boldsymbol{\theta}) = \underset{q \in Q}{\operatorname{argmin}} \ \mathrm{KL}\left(\prod_{i=1}^m q(\boldsymbol{\theta}_i) \| f(\boldsymbol{\theta}|\boldsymbol{x}) \right). \tag{1.21}$$

The optimisation problem in equation (1.21) corresponds to mean field variational Bayes. This leads to a mode seeking approximation, where $q^*(\boldsymbol{\theta})$ is selected by maximising the probability of being under $f(\boldsymbol{\theta}|\boldsymbol{x})$. As such, all samples from $q^*(\boldsymbol{\theta})$ will lie within a mode of $f(\boldsymbol{\theta}|\boldsymbol{x})$. Belief propagation is motivated by reversing the KL divergence that drives mean field variational Bayes, i.e.

$$q^*(\boldsymbol{\theta}) = \underset{q \in Q}{\operatorname{argmin}} \ \mathrm{KL}\left(f(\boldsymbol{\theta}|\boldsymbol{x}) \| \prod_{i=1}^m q(\boldsymbol{\theta}_i) \right). \tag{1.22}$$

By reversing the direction of KL divergence, $q^*(\boldsymbol{\theta})$ is selected by maximising the probability of $f(\boldsymbol{\theta}|\boldsymbol{x})$ being under $q(\boldsymbol{\theta}_i)$. As such, the optimal approximate distribution $q^*(\boldsymbol{\theta})$ will cover all modes of $f(\boldsymbol{\theta}|\boldsymbol{x})$. The difference in the approximation resulting from the different directions of KL divergence is illustrated in Figure 1.7. The mean seeking approximation that arises can be solved by a moment matching problem. However, this problem has the potential to become unboundedly complex. As mentioned before, one way to control the complexity of projections required is to restrict approximating factors to be in an exponential family. Both ADF and EP use the same direction of KL divergence as belief propagation, but constrain the approximating distributions to be from the exponential family to aid their implementation.

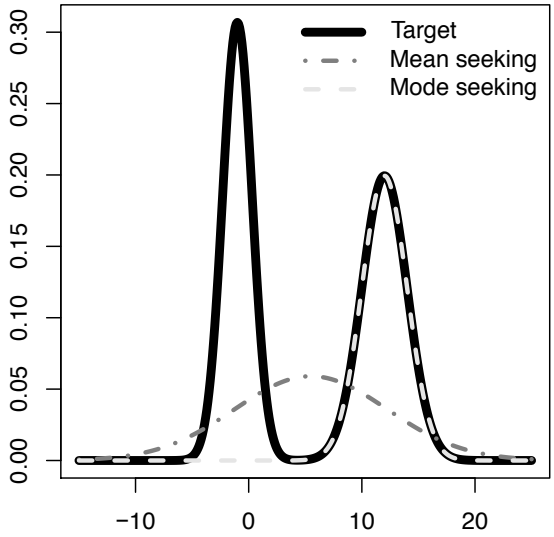


Figure 1.6: *Plot comparing density estimates using both directions of the KL divergence. The approximating density was Gaussian and the target density was a mixture of two Gaussians.*

1.12.3 Expectation propagation

We now explain the general EP schematic, re-iterating points previously made in this section. For m independent data points $\boldsymbol{x}_i$ where $i \in \{1, ..., m\}$ and latent variable $\boldsymbol{u}$ consider an intractable posterior

$$\begin{aligned} p(\boldsymbol{u}|\boldsymbol{x}) &= \frac{p_0(\boldsymbol{u}) \prod_{i=1}^m p(\boldsymbol{x}_i|\boldsymbol{u})}{\int p_0(\boldsymbol{u}) \prod_{i=1}^m p(\boldsymbol{x}_i|\boldsymbol{u}) d\boldsymbol{u}} \\ &= Z^{-1} p_0(\boldsymbol{u}) \prod_{i=1}^m p(\boldsymbol{x}_i|\boldsymbol{u}), \end{aligned} \tag{1.23}$$

where

$$p(\boldsymbol{x}_i|\boldsymbol{u}) = f_i(\boldsymbol{u}), \quad p_0(\boldsymbol{u}) = f_0(\boldsymbol{u}), \quad Z = \int p_0(\boldsymbol{u}) \prod_{i=1}^m p(\boldsymbol{x}_i|\boldsymbol{u}) d\boldsymbol{u},$$

and $f_0(\boldsymbol{u})$ is a member of the exponential family. Note the posterior in equation (1.23) is proportional to

$$p(\boldsymbol{u}|\boldsymbol{x}) \propto f_0(\boldsymbol{u}) \prod_{i=1}^m f_i(\boldsymbol{u}).$$

We wish to obtain a global approximation of the posterior with a density q selected

from the exponential family (denoted by Q)

$$q(\boldsymbol{u}) \propto \prod_{i=0}^{m} q_i(\boldsymbol{u}),$$

where $q_0(\boldsymbol{u}) = f_0(\boldsymbol{u})$. The optimal global approximation is the density which minimises the KL divergence to the posterior

$$q^*(\boldsymbol{u}) = \underset{q \in Q}{\operatorname{argmin}} \operatorname{KL}\big(p(\boldsymbol{u}|\boldsymbol{x}) \big\| q(\boldsymbol{u})\big).$$

Since the approximating density family is selected to be exponential, finding the optimal approximation reduces to a simple moment matching problem. This moment matching problem is not feasible over the whole posterior, so we instead conduct it on each site. The product of these site approximations is equivalent to the approximation over the whole posterior. As such, our goal is to find an approximation $q_i(\boldsymbol{u})$ which minimises the KL divergence over a site of the likelihood factor $p(\boldsymbol{x}_i|\boldsymbol{u})$. In other words, we require the KL projection of a site of our target distribution onto the approximating density family. However, as the target distribution is outside the exponential family space, we instead use a tractable hybrid distribution $h_i(\boldsymbol{u})$ that lies in between the target distribution and the approximating distribution, as shown in Figure 1.7. Each hybrid distribution (also

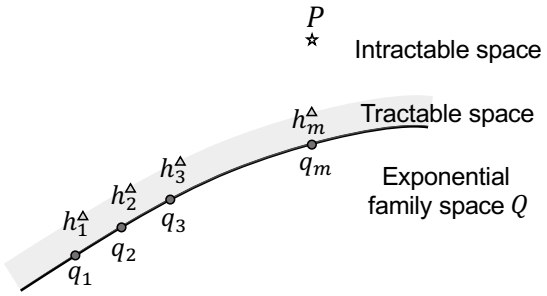


Figure 1.7: *Visualisation of the EP process (Barthelme, 2016[3]). The hybrid approximations are marked with triangles, the approximations are marked with cricles and the target density is marked with a star.*

called tilted distribution) is found by removing an approximating site from the global approximation (removing the ith site forms a cavity distribution denoted by $q_{-i}(\boldsymbol{u})$) and replacing it with the equivalent true likelihood site. Formally the ith hybrid is

$$h_i(\boldsymbol{u}) = f_i(\boldsymbol{u}) q_{-i}(\boldsymbol{u}),$$

where the ith cavity distribution $q_{-i}(\boldsymbol{u})$ is given by

$$q_{-i}(\boldsymbol{u}) = \prod_{i' \neq i}^{m} q_{i'}(\boldsymbol{u}).$$

We can now project this hybrid to the approximating family by computing the moments, which results in a new approximation of the whole posterior with features of the hybrid

$$q^{\text{new}}(\boldsymbol{u}) = \text{proj}[h_i(\boldsymbol{u})].$$

In the special case where the approximating family is Gaussian this problem simplifies to finding the mean, variance and normalising factor of the hybrid distribution. Using the new global approximation of the posterior we must now update the site $q_i(\boldsymbol{u})$ such that $q_i(\boldsymbol{u})q_{-i}(\boldsymbol{u})$ has the same moments as the hybrid distribution, i.e.

$$q_i^{\text{new}}(\boldsymbol{u}) = \frac{q^{\text{new}}(\boldsymbol{u})}{q_{-i}(\boldsymbol{u})}.$$

We hope that by iterating projections of the hybrid distributions onto the exponential family space and updating each $q_i(\boldsymbol{u})$, we find a stationary point which minimises the KL distance to the target density, as shown in Figure 1.7. At this point a global approximation to the posterior can be obtained by taking the product of the partitions,

$$q(\boldsymbol{u}) = \prod_{i=0}^{m} q_i(\boldsymbol{u}).$$

This process forms the EP algorithm, and is presented explicitly in Algorithm 1. By using natural parameters further simplification is available, where the optimal parameters are found by linear algebra.

Algorithm 1 *A generalised version of the EP algorithm.*

Initialise: by setting $q(\boldsymbol{u}) = \prod_{i=0}^{m} q_i(\boldsymbol{u})$, $q_0(\boldsymbol{u}) = f_0(\boldsymbol{u})$ and for $1 \leq i \leq m$, $q_i(\boldsymbol{u}) = 1$.
Cycle: Pick $i = 1, ..., m$:

Remove $q_i(\boldsymbol{u})$ from $q(\boldsymbol{u})$ to get the cavity distribution $q_{-i}(\boldsymbol{u})$,

$$q_{-i}(\boldsymbol{u}) \propto \prod_{i' \neq i}^{m} q_{i'}(\boldsymbol{u}). \tag{1.24}$$

Update the tilted distribution by replacing the approximating likelihood $q_i(\boldsymbol{u})$ by the exact one $f_i(\boldsymbol{u})$,

$$h_i(\boldsymbol{u}) = f_i(\boldsymbol{u}) q_{-i}(\boldsymbol{u}).$$

Project $h_i(\boldsymbol{u})$ back to the exponential family

$$q^{\text{new}}(\boldsymbol{u}) = \text{proj}\big[h_i(\boldsymbol{u})\big].$$

Update the approximate terms

$$q_i^{\text{new}}(\boldsymbol{u}) \propto \frac{q^{\text{new}}(\boldsymbol{u})}{q_{-i}(\boldsymbol{u})} \tag{1.25}$$

until all q_i converge.

After convergence is reached obtain a tractable approximation from

$$q(\boldsymbol{u}) = \prod_{i=0}^{m} q_i(\boldsymbol{u}).$$

1.12.4 Message passing

Minka (2005)[44] streamlines EP and other variational inference algorithms into a framework known as message passing, which allows significant algebraic and computational simplifications for large factor graphs. In this section we focus on the message passing approach for simple EP models and refer interested readers to Minka (2005)[44] for other variational methods.

A message is simply a function defined by a factor node that takes a subset of parental stochastic nodes as an input. The EP problem of obtaining a KL projection onto the required exponential family can be solved in terms of messages passed between neighbouring nodes of a relevant factor graph. Since the approximating family is constrained to be exponential, the messages can be expressed in natural parameters.

Let $f_i,\ 1 \leq i \leq m$ denote factor nodes and $\boldsymbol{u}$ denote a stochastic node that corresponds to a latent variable. For the models presented in this thesis, the neighbours of any ith factor node are the stochastic node $\boldsymbol{u}$. Kim & Wand (2016)[31] provide additional details for larger models. Given the latent variable $\boldsymbol{u}$ and observed data $\boldsymbol{x}$ the joint density function can be written as

$$p(\boldsymbol{u}, \boldsymbol{x}) = \prod_{i=1}^{m} f_i(\boldsymbol{u}).$$

For the simple factor graphs presented in this thesis, the message passing EP algorithm can be reduced to updating messages from the stochastic node to factor node as

$$m_{\boldsymbol{u}\to f_i}(\boldsymbol{u}) \longleftarrow \prod_{i' \neq i}^{m} m_{f_{i'}\to \boldsymbol{u}}(\boldsymbol{u}), \tag{1.26}$$

and updating the message from the factor to stochastic node as

$$m_{f_i\to \boldsymbol{u}}(\boldsymbol{u}) \longleftarrow \frac{\text{proj}\big[Z^{-1} m_{\boldsymbol{u}\to f_i}(\boldsymbol{u}) f_i(\boldsymbol{u})\big]}{m_{\boldsymbol{u}\to f_i}(\boldsymbol{u})}, \tag{1.27}$$

where Z is the normalising factor. In terms of the Algorithm 1, equation (1.26) for message passing EP corresponds to forming the cavity distribution in equation (1.24), and equation (1.27) corresponds updating the $q_i(\boldsymbol{u})$ approximation as in equation (1.25).

Once the messages have reached convergence, the KL optimal q-densities are obtained via

$$q(\boldsymbol{u}) \propto \prod_{i=1}^{m} m_{f_i\to \boldsymbol{u}}(\boldsymbol{u}).$$

This process is presented consisely in Algorithm 2. Kim & Wand (2016)[31] suggest using relative change in the approximate marginal log-likelihood as a stopping criterion. We refer readers there for further details.

Algorithm 2 *A generalised version of the message passing EP algorithm.*

Initialise: by setting $m_{f_i \to \boldsymbol{u}}(\boldsymbol{u}) = 1$, where $1 \leq i \leq m$.

Cycle: Pick $i = 1, ..., m$:

Get the cavity distribution,

$$m_{\boldsymbol{u} \to f_i}(\boldsymbol{u}) \longleftarrow \prod_{i' \neq i}^{m} m_{f_{i'} \to \boldsymbol{u}}(\boldsymbol{u}).$$

Project the tilted distribution onto the exponential family and update the approximate terms

$$m_{f_i \to \boldsymbol{u}}(\boldsymbol{u}) \longleftarrow \frac{\text{proj}\left[Z^{-1} m_{\boldsymbol{u} \to f_i}(\boldsymbol{u}) f_i(\boldsymbol{u})\right]}{m_{\boldsymbol{u} \to f_i}(\boldsymbol{u})}$$

until all $m_{f_i \to \boldsymbol{u}}$ converge.

After convergence is reached obtain a tractable approximation from

$$q(\boldsymbol{u}) = \prod_{i=1}^{m} m_{f_i \to \boldsymbol{u}}(\boldsymbol{u}).$$

1.13 Thesis structure

We have now explained the background information required and from this point on present novel research. Chapter 2 explores a simple model for the probit link where we estimate only the variance parameter σ^2 i.e. (the model only has a random intercept). Chapter 3 extends the work in Chapter 2 to the general case where there are multiple fixed and random effects. Chapter 4 applies the work of Chapters 2 and 3 to the logistic link function. Chapter 5 continues from Chapter 3 by applying the work from the previous chapters to count data with the Poisson link and the negative binomial link. Chapter 6 explores models with crossed random effects. Chapter 7 applies the methodology implemented in this thesis to two real datasets, one provided by the Australian Red Cross Blood Service. The thesis ends with a discussion and concluding remarks in Chapter 8.

Chapter 2

Expectation propagation for the simplest one level probit mixed model

In this chapter we develop methodology for frequentist inference of binary probit GLMMs for the simplest random intercepts only model. Our goal is to approximate the maximum likelihood of the parameter for variance between groups only (i.e. random intercepts only model) with 95% confidence intervals. Since this model assumes only one source of variance, a change in notation occurs from the covariance matrix Σ shown in Section 1.7, to the scalar variance σ^2. This model provides a good starting point for testing new methodology since only one dimensional integrals are required, as opposed to the integrals required for more complex models. Additionally, any difficulties caused by matrices are negated. We trial various methods of calculating confidence intervals and show how to obtain best predictors.

In this chapter, we assume a balanced dataset, where all m groups have the same n number of observations in them. For observed values of

$$y_{ij}; \quad 1 \leq i \leq m, \quad 1 \leq j \leq n,$$

where $y_{ij} \in \{0, 1\}$, the probit binary mixed model form is

$$y_{ij}|u_i \overset{\text{iid}}{\sim} \text{Bernoulli}\big(\Phi(u_i)\big), \qquad u_i \overset{\text{iid}}{\sim} N(0, \sigma^2), \tag{2.1}$$

where u_i is a scalar unobserved latent variable. The variance parameter likelihood can

be expressed as

$$\ell(\sigma^2) = \sum_{i=1}^{m} \ell_i(\sigma^2),$$

where

$$\ell_i(\sigma^2) \equiv \log \int_{-\infty}^{\infty} \prod_{j=1}^{n} \Phi\big((2y_{ij}-1)u_i\big)(2\pi\sigma^2)^{-1/2} \exp(-u_i^2/2\sigma^2)\, du_i, \tag{2.2}$$

and the best predictor of u_i is

$$\mathrm{BP}(u_i) = \frac{\int_{-\infty}^{\infty} u_i \prod_{j=1}^{n} \Phi\big((2y_{ij}-1)u_i\big) \exp(-u_i^2/2\sigma^2)\, du_i}{\int_{-\infty}^{\infty} \prod_{j=1}^{n} \Phi\big((2y_{ij}-1)u_i\big) \exp(-u_i^2/2\sigma^2)\, du_i}.$$

We denote the maximum likelihood estimate of σ^2 by

$$\widehat{\sigma^2} = \underset{\sigma^2}{\operatorname{argmax}}\ \ell(\sigma^2).$$

Calculation of the maximum likelihood estimator and best predictor are complicated by the intractable integral arising in equation (2.2). Each $\ell_i(\sigma^2)$ can approximated and summed to obtain the full log-likelihood. Traditionally these univariate integrals are solved using quadrature. We develop an EP scheme for estimating the likelihood surface by approximating each $\ell_i(\sigma^2)$ and summing them to obtain the full log-likelihood. We compare its performance to quadrature. Our aim is to show both methods provide reasonable and similar estimates.

Details of the quadrature approach to estimating the likelihood surface are given in Section 2.1. Our novel method using EP is explained in Section 2.2. Section 2.4 explores different ways to obtain the maximum likelihood estimate and confidence intervals for both our novel method and the quadrature approach. Details on obtaining best predictors for this model are given in Section 2.5, before comparing the approaches of likelihood estimation in a simulation study in Section 2.6.

2.1 Traditional quadrature likelihood approximation

Likelihood surface approximation by quadrature is easy to implement using software such as the R function `integrate()` in the "stats" package (R Core Team, 2019[56]) as discussed in Section 1.11.2. However, one must be careful to avoid issues with numerical instability. As such, we express the integrand in equation (2.2) as a function that attains

a maximum value of 1 over $u \in \mathbb{R}$, where the ith integral arising in each $\ell_i(\sigma^2)$ is

$$I_i(\sigma^2) \quad = \quad \exp\Big\{ \max_{u_0 \in \mathbb{R}} \big(h_i(u_0)\big) \Big\} \int_{-\infty}^{\infty} \exp\Big\{ h_i(u) - \max_{u_0 \in \mathbb{R}} \big(h_i(u_0)\big) \Big\}\, du$$

and

$$h_i(u) \equiv \sum_{j=1}^{n} \log \Phi\big((2y_{ij} - 1)u\big) - \frac{u^2}{2\sigma^2}.$$

The first and second derivatives of $h_i(u)$ are

$$h_i'(u) \quad = \quad \sum_{j=1}^{n} \zeta'\big((2y_{ij} - 1)u\big)(2y_{ij} - 1) - \frac{u}{\sigma^2}$$

and

$$h_i''(u) \quad = \quad \sum_{j=1}^{n} \zeta''\big((2y_{ij} - 1)u\big) - \frac{1}{\sigma^2}.$$

Assuming $\zeta''(x) < 0$ for all $x \in \mathbb{R}$, then $h_i''(u) < 0$ for all $u \in \mathbb{R}$. From this fact,

- h_i is a strictly concave function over $\mathbb{R}$
- $\exp\big(h_i(u)\big)$ is log-concave
- $\lim_{u \to -\infty} h_i'(u) = -\infty$ and $\lim_{u \to \infty} h_i'(u) = +\infty$.

As such, each $\ell_i(\sigma^2)$ of the likelihood is calculated as

$$\frac{1}{2}\log(2\pi\sigma^2) + \ell_i(\sigma^2) = h(u_{0i}) + \log \int_{-\infty}^{\infty} \exp\big(h_i(u_i) - h_i(u_{0i})\big)\, du_i,$$

where $h_i'(u_{0i}) = 0$ and the unique root u_{0i} is found using a bisection search, where the starting values are selected -1 and 1 to be for the lower and upper bounds respectively.

2.2 Expectation propagation likelihood approximation

We now propose a novel approach to likelihood approximation using EP (the log-likelihood is denoted by $\underset{\sim}{\ell}(\sigma^2)$) which is more amenable to cases involving higher dimensional integrals than traditional quadrature routines. As discussed in Section 1.12.3, the EP approximation is motivated by minimisation of a KL divergence criterion (see equation (1.19)), which selects an unnormalised normal density function to replace

each

$$\Phi\big((2y_{ij}-1)u_i\big), \quad 1 \leq j \leq n$$

in equation (2.2). Subsequently, the integrand is proportional to a product of univariate normal density functions which have explicit forms in probit case. Furthermore, the calculation of $\ell(\sigma^2)$ requires only fixed point iteration, so there is no need for any numerical integration. A major downfall of EP implementation is the high algebraic overhead. We aim to minimise this by using message passing similar to Kim & Wand (2017).[32]

Consider the family of unnormalised normal density functions written in exponential family form,

$$f_{\text{UN}}(x) = \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \end{bmatrix} \right\} \tag{2.3}$$

with natural parameters $\eta_0, \eta_1 \in \mathbb{R}$ and $\eta_2 < 0$. The goal of the EP problem is to find the optimal natural parameters, η_0^*, η_1^* and η_2^*, which minimise $\text{KL}(f_{\text{input}} \parallel f_{\text{UN}})$ where $f_{\text{input}} \in L_1$. This solution is referred to as the KL projection onto the family of unnormalised normal density functions, and is written as

$$\text{proj}_{\text{UN}}[f_{\text{input}}](x) = \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\}, \tag{2.4}$$

where

$$\boldsymbol{\eta}^* \equiv \begin{bmatrix} \eta_0^* \\ \eta_1^* \\ \eta_2^* \end{bmatrix},$$

$$(\eta_0^*, \eta_1^*, \eta_2^*) = \underset{(\eta_0,\eta_1,\eta_2)\in H}{\text{argmin}} \ \text{KL}(f_{\text{input}} \parallel f_{\text{UN}})$$

and H is the set of all allowable natural parameters. In the special case of KL projection onto the unnormalised univariate normal family this problem simplifies further to moment-matching, where $(\eta_0^*, \eta_1^*, \eta_2^*)$ is the unique vector that matches the zeroth, first and second order moments of f_{UN} and f_{input}. For the case of probit binary GLMMs,

EP requires repeated KL projections of the form

$$f_{\text{input}}(x) = \Phi(c_0 + c_1 x)\ \exp\left(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\right) \tag{2.5}$$

onto an unnormalised normal distribution, where $c_0 = 0$, $c_1 = 2y_{ij} - 1$, $x = u_i$, $\eta_1^{\text{input}} \in \mathbb{R}$ and $\eta_2^{\text{input}} < 0$. As such, we seek $\boldsymbol{\eta}^*$ such that

$$\int_{-\infty}^{\infty} x^k \Phi(c_0 + c_1 x)\ \exp\left\{ \begin{bmatrix} x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^{\text{input}} \right\} dx = \int_{-\infty}^{\infty} x^k \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} dx. \tag{2.6}$$

Consider Result 3.

Result 3. *For an unnormalised input function $f \in L_1$ such that $f \geq 0$ for all $x \in \mathbb{R}$, where $C_f \equiv \int_{\mathbb{R}} f(x)dx$, the projection onto the unnormalised normal family is*

$$proj_{UN}[f] = C_f proj_N[f/C_f](x),$$

where $proj_N$ is the projection onto the normal family.

By Result 3, obtaining the natural parameters $\boldsymbol{\eta}^*$ for projection onto the unnormalised normal family follows from obtaining the projection onto the normal family. More explicitly, the optimal natural parameters, η_1^* and η_2^*, are given according to the projection of the normalised function $f_{\text{input}}/C_{f_{\text{input}}}$ onto the normal family. We can subsequently use these optimal natural parameters to find the normalising natural parameter η_0^* via Result 4 and thus obtain the projection onto unnormalised normal family.

Result 4. *When the input density follows the form of equation (2.5), η_0^* is given by*

$$\begin{aligned} \eta_0^* &= \log(C_{f_{input}}) - A(\eta_1^*, \eta_2^*) - \frac{1}{2}\log(2\pi), \\ &= \log(C_{f_{input}}) + (\eta_1^*)^2/(4\eta_2^*) + \frac{1}{2}\log(-\eta_2^*/\pi). \end{aligned}$$

Thus to obtain the required projection, we first obtain the optimal natural parameters η_1^* and η_2^* to project onto the normal family as is presented in Result 5.

Result 5. *Given f follows the form of equation (2.5), the projection onto the univariate normal family is given by*

$$proj_N[f] = \exp\left(\boldsymbol{T}(x)^\top \boldsymbol{\eta}^*_{-1} - A(\boldsymbol{\eta}^*_{-1})\right)h(x)$$

where

$$\boldsymbol{\eta}^{input}_{-1} \equiv \begin{bmatrix} \eta^{input}_1 \\ \eta^{input}_2 \end{bmatrix}, \qquad \boldsymbol{\eta}^*_{-1} \equiv \begin{bmatrix} \eta^*_1 \\ \eta^*_2 \end{bmatrix} = k_{probit}\left(\boldsymbol{\eta}^{input}_{-1}; c_0, c_1\right),$$

$k_{probit}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1\right)$ *is defined in Definition 10 and* $\boldsymbol{T}(x)$ *and* $h(x)$ *follow from Section 1.5.2.1.*

Definition 10. *For primary scalar arguments $a_1 \in \mathbb{R}$ and $a_2 < 0$ and auxiliary scalar arguments $c_0, c_1 \in \mathbb{R}$, the function $k_{probit} : H \to H$ is given by*

$$k_{probit}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1\right) = \begin{bmatrix} r_5(a_1 + r_3 c_1) \\ r_5 a_2 \end{bmatrix},$$

with

$$r_1 = \sqrt{2(2 - c_1^2 a_2^{-1})}, \quad r_2 = (2c_0 - c_1 a_2^{-1} a_1) r_1^{-1}, \quad r_3 = 2\zeta'(r_2) r_1^{-1},$$
$$r_4 = -2\zeta''(r_2) r_1^{-2} \quad \textit{and} \quad r_5 = (a_2 + r_4 c_1^2)^{-1} a_2.$$

Using Result 5 we now obtain the normalising natural parameter η^*_0 to find the projection onto unnormalised normal family.

2.2.1 Projection onto the unnormalised normal family

Recalling the moment matching problem described by equation (2.6) and Result 4, the normalising factor can be shown to be

$$C_{f_{\text{input}}} = \int_{\mathbb{R}} f_{\text{input}}(x)\, dx = (2\pi)^{-1/2} \exp\left(A(\boldsymbol{\eta}^{\text{input}})\right)\, \Phi(r_2),$$

where r_2 is given in Definition 10 and $A(\boldsymbol{\eta})$ is defined in Section 1.5. By Result 4

$$\eta_0^* = \log\, \Phi(r_2^{\text{input}}) + \frac{1}{4}(\eta_1^*)^2/\eta_2 - \frac{1}{4}(\eta_1^{\text{input}})^2/\eta_2^{\text{input}} + \frac{1}{2}\log(\eta_2^*/\eta_2^{\text{input}}).$$

To obtain η_0^* we introduce the $c_{\text{probit}}(\boldsymbol{a}, \boldsymbol{b}; c_0, c_1)$ function in Definition 11, for $\boldsymbol{a} = [a_1\ a_2]^\top$ and $\boldsymbol{b} = [b_1\ b_2]^\top$.

Definition 11. *Consider first, primary scalar arguments a_1, a_2, b_1 and b_2, and auxiliary scalar arguments c_0 and c_1. Then the function $c_{probit} : H \to \mathbb{R}$ is given by*

$$c_{probit}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}; c_0, c_1\right) \equiv log\ \Phi(r_2) + \frac{1}{4}b_1^2/b_2 - \frac{1}{4}a_1^2/a_2 + \frac{1}{2}\log(b_2/a_2),$$

where r_2 follows from Definition 10.

To summarise, the projection of the input function onto the unnormalised normal family is obtained as in Result 6.

Result 6. *For an unnormalised input function of the form of equation (2.5),*

$$proj_{UN}[f_{input}] = \exp\left\{\begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \begin{bmatrix} \eta_0^* \\ \eta_1^* \\ \eta_2^* \end{bmatrix}\right\},$$

where

$$\eta_0^* = c_{probit}\left(\begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix}; c_0, c_1\right)$$

and

$$\begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix} = k_{probit}\left(\begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}; c_0, c_1\right).$$

Note the forms of these functions are useful in the next section where we show how to organise the projections required.

2.2.2 Message passing formulation

We now express the EP approximation of $\ell(\sigma^2)$ using message passing updates as presented in Kim & Wand (2017).[68] Note each $\ell_i(\sigma^2)$ can be written as

$$\ell_i(\sigma^2) = \log \int_{-\infty}^{\infty} \left(\prod_{j=1}^{n} p(y_{ij}|u_i) \right) p(u_i; \sigma^2) du_i, \tag{2.7}$$

where

$$p(y_{ij}|u_i) \equiv \Phi\big((2y_{ij} - 1)u_i\big) \quad \text{and} \quad p(u_i; \sigma^2) \equiv (2\pi\sigma^2)^{-1/2} \exp\big(-u_i^2/(2\sigma^2)\big)$$

are respectively the conditional density function of each response given its random effect and the density function of the random effect. The alternate expression

$$p(u_i; \sigma^2) = \exp\left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^\top \boldsymbol{\eta}_{\sigma^2} \right\}, \quad \text{where} \quad \boldsymbol{\eta}_{\sigma^2} \equiv \begin{bmatrix} -\frac{1}{2}\log(2\pi\sigma^2) \\ 0 \\ -1/(2\sigma^2) \end{bmatrix} \tag{2.8}$$

is more amenable to the message passing approach to follow and is also worth noting. Using factor graphs (discussed in Section 1.6.2), Figure 2.1 visualises the dependence structure of the product in equation (2.7), where the circular stochastic node corresponds to the random vector u_i and solid squares indicate the $n+1$ factor nodes. The dependence of each factor node on the stochastic node u_i is demonstrated through the edges.

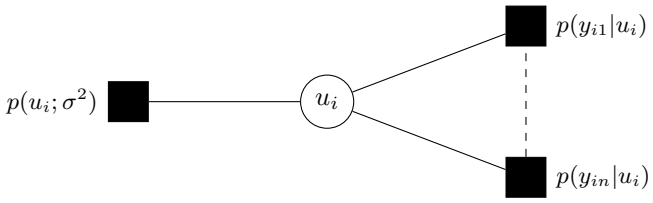


Figure 2.1: *Factor graph representation of the product structure of the integrand in equation (2.7).*

With this factor graph in place implementing EP is analogous to the Bayesian approach in Minka (2005).[44] As previously explained, the EP approximation of $\ell_i(\sigma^2)$ is motivated by minimization of a KL divergence and imposition of exponential family

constraints. Consider that

$$\underset{\sim}{p}(y_{ij}|u_i) = \exp\left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^\top \boldsymbol{\eta}_{ij} \right\}, \quad 1 \leq j \leq n$$

is initialised such that u_i is an unnormalised normal density function. Then for each $j = 1, ..., n$, the $\boldsymbol{\eta}_{ij}$ update requires minimisation of

$$\mathrm{KL}\left(p(y_{ij}|u_i) \left(\prod_{j' \neq j}^{n} \underset{\sim}{p}(y_{ij'}|u_i) \right) p(u_i; \sigma^2) \middle\| \left(\prod_{j'=1}^{n} \underset{\sim}{p}(y_{ij'}|u_i) \right) p(u_i; \sigma^2) \right) \tag{2.9}$$

as functions of u_i. Thus we can use Result 6 to perform updates until convergence of $\boldsymbol{\eta}_{ij}$s. Using Figure 2.1, EP can be compartmentalised by the notion of message passing as shown in Section 4.1 of Minka (2005).[44] First let us define messages passed from the factor $p(y_{ij}|u_i)$ to u_i as

$$m_{p(y_{ij}|u_i) \to u_i}(u_i) \equiv \underset{\sim}{p}(y_{ij}|u_i).$$

Messages from the factor $p(y_{ij}|u_i)$ to the stochastic node u_i are updated by equation (60) of Minka (2005),[44] where $\alpha = 1$ and $s' = 1$ (since the KL divergence we are working with is unnormalised). Simplifications given we have one stochastic node result in the following expression for equation (2.9),

$$m_{p(y_{ij}|u_i) \to u_i}(u_i) \longleftarrow \frac{\mathrm{proj}_{\mathrm{UN}}\left[m_{u_i \to p(y_{ij}|u_i)}(u_i)\ p(y_{ij}|u_i) \right](u_i)}{m_{u_i \to p(y_{ij}|u_i)}(u_i)}, \quad 1 \leq j \leq n. \tag{2.10}$$

Similarly the update of message passed from $p\left(u_i; \sigma^2\right)$ to u_i is

$$m_{p(u_i;\sigma^2) \to u_i}(u_i) \longleftarrow \frac{\mathrm{proj}_{\mathrm{UN}}\left[m_{u_i \to p(u_i;\sigma^2)}(u_i)\ p(u_i; \sigma^2) \right](u_i)}{m_{u_i \to p(u_i;\sigma^2)}(u_i)}. \tag{2.11}$$

By equation (54) of Minka (2005)[44] the updates of stochastic node to factor messages are

$$m_{u_i \to p(y_{ij}|u_i)}(u_i) = m_{p(u_i;\sigma^2) \to u_i}(u_i) \prod_{j' \neq j}^{n} m_{p(y_{ij}|u_i) \to u_i}(u_i), \quad 1 \leq j \leq n \tag{2.12}$$

and

$$m_{u_i \to p(u_i;\sigma^2)}(u_i) = \prod_{j=1}^{n} m_{p(y_{ij}|u_i) \to u_i}(u_i). \tag{2.13}$$

We now seek any algebraic simplifications of the key messages. Recall that $p(u_i;\sigma^2)$ can be written using natural parameters in the form of equation (2.8) and that the unnormalised normal density constraint is enforced on equations (2.10) and (2.12). Then

$$m_{u_i\to p(u_i;\sigma^2)}(u_i) = \exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{u_i\to p(u_i;\sigma^2)}\right\}. \tag{2.14}$$

Substituting the above forms into equation (2.11) leads to

$$\begin{aligned} m_{p(u_i;\sigma^2)\to u_i}(u_i) &\longleftarrow \frac{\text{proj}_{\text{UN}}\left[\exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{u_i\to p(u_i;\sigma^2)}\right\}\exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{\sigma^2}\right\}\right]}{\exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{u_i\to p(u_i;\sigma^2)}\right\}} \\ &= \frac{\exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{u_i\to p(u_i;\sigma^2)}\right\}\exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{\sigma^2}\right\}}{\exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{u_i\to p(u_i;\sigma^2)}\right\}} \\ &= \exp\left\{\begin{bmatrix}1\\u_i\\u_i^2\end{bmatrix}^\top \boldsymbol{\eta}_{\sigma^2}\right\}. \end{aligned}$$

This implies that the message $m_{p(u_i;\sigma^2)\to u_i}(u_i) = p(u_i;\sigma^2)$ is constant throughout the message passing updates. As such, we now set

$$\boldsymbol{\eta}_{p(u_i;\sigma^2)\to u_i} \longleftarrow \boldsymbol{\eta}_{\sigma^2}. \tag{2.15}$$

For convenience,

$$\boldsymbol{\eta}^{\otimes} \equiv \boldsymbol{\eta}_{u_i \to p(y_{ij}|u_i)}.$$

Recall that from equation (2.14)

$$m_{u_i \to p(y_{ij}|u_i)}(u_i) = \exp\left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^{\top} \boldsymbol{\eta}^{\otimes} \right\} = \exp(\eta_0^{\otimes}) \exp\left(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2\right).$$

Substituting this into equation (2.10) leads to

$$\begin{aligned} m_{p(y_{ij}|u_i) \to u_i}(u_i) &\longleftarrow \frac{\text{proj}_{\text{UN}}\Big[\exp(\eta_0^{\otimes}) \exp(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2) \Phi\big((2y_{ij} - 1)u_i\big) \Big]}{\exp(\eta_0^{\otimes}) \exp(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2)} \\ &= \frac{\exp(\eta_0^{\otimes}) \text{proj}_{\text{UN}}\Big[\Phi\big((2y_{ij} - 1)u_i\big) \exp(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2) \Big]}{\exp(\eta_0^{\otimes}) \exp(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2)} \\ &= \frac{\text{proj}_{\text{UN}}\Big[\Phi(c_0 + c_{1_{ij}} u_i) \exp(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2) \Big]}{\exp(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2)}, \end{aligned}$$

where $c_0 = 0$ and $c_{1_{ij}} \equiv 2y_{ij} - 1$. By utilising Result 6,

$$m_{p(y_{ij}|u_i) \to u_i}(u_i) \longleftarrow \exp\left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^{\top} \boldsymbol{\eta}_{p(y_{ij}|u_i) \to u_i} \right\}, \tag{2.16}$$

where the linear and quadratic coefficient updates are

$$\left(\boldsymbol{\eta}_{p(y_{ij}|u_i) \to u_i}\right)_{1:2} \longleftarrow k_{\text{probit}}\left(\boldsymbol{\eta}_{1:2}^{\otimes}; c_0, c_{1_{ij}}\right) - \boldsymbol{\eta}_{1:2}^{\otimes} \tag{2.17}$$

and the constant coefficient update is

$$\left(\boldsymbol{\eta}_{p(y_{ij}|u_i) \to u_i}\right)_0 \longleftarrow c_{\text{probit}}\Big(\boldsymbol{\eta}_{1:2}^{\otimes}, \left(\boldsymbol{\eta}_{p(y_{ij}|u_i) \to u_i}\right)_{1:2} + \boldsymbol{\eta}_{1:2}^{\otimes}; c_0, c_{1_{ij}}\Big). \tag{2.18}$$

Using the simplification of equation (2.10) and (2.11), equation (2.12) can be shown to

be

$$m_{u_i \to p(y_{ij}|u_i)}(u_i) \longleftarrow \exp\left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^\top \boldsymbol{\eta}_{u_i \to p(y_{ij}|u_i)} \right\}, \tag{2.19}$$

where

$$\boldsymbol{\eta}_{u_i \to p(y_{ij}|u_i)} \longleftarrow \boldsymbol{\eta}_{p(u_i;\sigma^2) \to u_i} + \sum_{j' \neq j} \boldsymbol{\eta}_{p(y_{ij'}|u_i) \to u_i}.$$

The scheme of the message passing approach to EP can be summarised as in Section 6 of Minka (2005):[44]

1. Initialise all factor to stochastic node messages.
2. Cycle until convergence of all factor to stochastic node message:

   For each factor:

   (a) Compute the messages passed to the factor via equation (2.14) or equation (2.19).

   (b) Compute the messages passed from the factor via equation (2.15) or equation (2.16).

The EP approximation to each log-likelihood $\underset{\sim}{\ell}_i(\sigma^2)$ component is given by

$$\underset{\sim}{\ell}_i(\sigma^2) = \log \int_{\mathbb{R}} \left(\prod_{j=1}^{n} m_{p(y_{ij}|u_i) \to u_i}(u_i) \right) m_{p(u_i;\sigma^2) \to u_i}(u_i) du_i. \tag{2.20}$$

The success of EP depends on each of the messages in equation (2.20) being an unnormalised normal density. This allows results in a closed form solution to the

integral as follows:

$$\begin{aligned}
&\int_{\mathbb{R}} \left(\prod_{j=1}^{n} m_{p(y_{ij}|u_i)\to u_i}(u_i) \right) m_{p(u_i;\sigma^2)\to u_i}(u_i) du_i \\
&\quad = \int_{\mathbb{R}} \prod_{j=1}^{n} \exp \left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^\top \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i} \right\} \exp \left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^\top \boldsymbol{\eta}_{p(u_i|\sigma^2)\to u_i} \right\} du_i \\
&\quad = (2\pi)^{1/2} \exp \left\{ \left(\boldsymbol{\eta}_{\sigma^2} + \sum_{j=1}^{n} \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i} \right)_0 + A\left(\left(\boldsymbol{\eta}_{\sigma^2} + \sum_{j=1}^{n} \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i} \right)_{1:2} \right) \right\}.
\end{aligned}$$

The full algorithm for the approximation of $\ell(\sigma^2)$ using EP is provided in Algorithm 3.

Algorithm 3 *Explicit expression of the algorithm used for the message passing approach to EP in the random intercepts only model.*

Inputs: y_{ij}, $\boldsymbol{x}_{ij}^{\mathbf{R}}$, $1 \leq i \leq m$, $1 \leq j \leq n$; $\boldsymbol{\Sigma}(d^{\mathbf{R}} \times d^{\mathbf{R}}$, is symmetric and positive definite).
Set constants: $c_0 \longleftarrow 0$, $c_{1_{ij}} \longleftarrow 2y_{ij} - 1$; $1 \leq i \leq m$, $1 \leq j \leq n$,

$$\boldsymbol{\eta}_{p(u_i;\sigma^2)\to u_i} \longleftarrow \boldsymbol{\eta}_{\sigma^2} \equiv \begin{bmatrix} -\frac{1}{2}\log(2\pi\sigma^2) \\ 0 \\ 1/(2\sigma^2) \end{bmatrix}, \quad 1 \leq i \leq m. \tag{2.21}$$

For $i = 1, ..., m$:

Initialise: $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i} \in \mathbb{R}, \quad 1 \leq j \leq n$ as per equation (2.25).

Cycle:

$$\text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big) \longleftarrow \sum_{j=1}^{n} \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$$

For $j = 1, ..., n$:

$$\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)} \longleftarrow \boldsymbol{\eta}_{p(u_i;\sigma^2)\to u_i} + \text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big) - \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$$

$$\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_{1:2} \longleftarrow k_{\text{probit}}\Big(\big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}; c_0, c_{1_{ij}}\Big) - \big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}$$

until all natural parameter vectors converge.

For $j = 1, ..., n$:

$$\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_0 \longleftarrow c_{\text{probit}}\Big(\big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}, \big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_{1:2} + \big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}; c_0, c_{1_{ij}}\Big).$$

$$\text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big) \longleftarrow \sum_{j=1}^{n} \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$$

Output: The full approximate log-likelihood is given by

$$\underset{\sim}{\ell}(\sigma^2) = (m/2)\log(2\pi) + \sum_{i=1}^{m}\Bigg\{\Big(\boldsymbol{\eta}_{\sigma^2} + \text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)\Big)_0 + A\Big\{\Big(\boldsymbol{\eta}_{\sigma^2} + \text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)\Big)_{1:2}\Big\}\Bigg\},$$

where $A(\boldsymbol{\eta})$ is defined in equation (1.5) and $\boldsymbol{\eta}_{\sigma^2}$ follows from equation (2.21).

2.2.3 Starting values for Algorithm 3

The EP message passing algorithm proposed relies on good starting values for convergence. We now derive starting values for $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$ using a Taylor series expansion. Note that

$$\log p(y_{ij}|u_i) = \zeta(a_{ij}) - \log(2),$$

where $a_{ij} \equiv (2y_{ij}-1)u_i$ and ζ is defined in Section 1.5.2.1. Let $\widehat{u}_i$ be a Laplace approximation to u_i. Now consider the following Taylor series expansion of the data dependent component of $\ell(\sigma^2)$:

$$\begin{aligned}
\zeta(a_{ij}) &= \zeta\big(\widehat{a}_{ij} + (u_i - \widehat{u}_i)(2y_{ij}-1)\big) \\
&= \zeta(\widehat{a}_{ij}) + (u_i - \widehat{u}_i)(2y_{ij}-1)\zeta'(\widehat{a}_{ij}) + \frac{1}{2}\big((u_i - \widehat{u}_i)(2y_{ij}-1)\big)^2\zeta''(\widehat{a}_{ij}) + ... \\
&= \begin{bmatrix} 1 \\ u_i - \widehat{u}_i \\ (u_i - \widehat{u}_i)^2 \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + ...,
\end{aligned}$$

where $\widehat{a}_{ij} \equiv (2y_{ij}-1)\widehat{u}_i$ and

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} \zeta(\widehat{a}_{ij}) \\ (2y_{ij}-1)\zeta'(\widehat{a}_{ij}) \\ \frac{1}{2}\zeta''(\widehat{a}_{ij}) \end{bmatrix}.$$

It follows that the quadratic approximation to $\log\ p(y_{ij}|u_i)$ based on Taylor expansion about $\widehat{u}_i$ is $\log\ \breve{p}(y_{ij}|u_i)$ where

$$\breve{p}(y_{ij}|u_i) \equiv \exp\left\{ \begin{bmatrix} 1 \\ u_i - \widehat{u}_i \\ (u_i - \widehat{u}_i)^2 \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} \right\}. \tag{2.22}$$

The starting value recommendation for $\eta_{p(y_{ij}|u_i)}$ is based on replacement of $p(y_{ij}|u_i)$ by $\breve{p}(y_{ij}|u_i)$ in equation (2.15):

$$m_{\breve{p}(y_{ij}|u_i)\to u_i}(u_i) \longleftarrow \frac{\text{proj}\big[m_{u_i\to\breve{p}(y_{ij}|u_i)}(u_i)\breve{p}(y_{ij}|u_i)\big](u_i)}{m_{u_i\to\breve{p}(y_{ij}|u_i)}(u_i)} = \breve{p}(y_{ij}|u_i). \tag{2.23}$$

Note that in this case, since $\breve{p}(y_{ij}|u_i)$ is already univariate normal the projection is superfluous. The starting values for $\eta_{p(y_{ij}|u_i)\to u_i}$ that arises from this substitution is

$$\exp\left\{\begin{bmatrix}1\\ u_i\\ u_i^2\end{bmatrix}^\top \eta^{\text{start}}_{p(y_{ij}|u_i)\to u_i}\right\} = \exp\left\{\begin{bmatrix}1\\ u_i-\widehat{u}_i\\ (u_i-\widehat{u}_i)^2\end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij}\right\}. \tag{2.24}$$

By matching coefficients of like terms it is easy to show

$$\eta^{\text{start}}_{p(y_{ij}|u_i)\to u_i} = \begin{bmatrix}\eta_0^{\text{start}}\\ (2y_{ij}-1)\zeta'(\widehat{a}_{ij})-\zeta''(\widehat{a}_{ij})\widehat{u}_i\\ \frac{1}{2}\zeta''(\widehat{a}_{ij})\end{bmatrix} \tag{2.25}$$

where

$$\eta_0^{\text{start}} = \zeta(\widehat{a}_{ij}) - (2y_{ij}-1)\zeta'(\widehat{a}_{ij})\widehat{u}_i + \frac{1}{2}\zeta''(\widehat{a}_{ij})\widehat{u}_i^2.$$

Note that in Algorithm 3 η_0^{start} is not used in the cycle loop and thus can be set to any arbitrary number without affecting the algorithm. A good choice for $\widehat{u}_i$ is Laplace approximation. For the R computing environment, the function `glmer()` of the package "lme4" (Bates, et al., 2018[5]) provides fast Laplace approximation-based predictions for the u_i.

2.3 Evaluation of the estimates

Using R software, we implemented and compared the quadrature and EP approach to calculating $\ell(\sigma^2)$ and $\underset{\sim}{\ell}'(\sigma^2)$ respectively. Figure 2.2 shows estimates of the likelihood surface for both methods. It demonstrates that they provide reasonable estimates of the true value of σ^2 and that any differences between them are extremely small. Having shown equivalence of the methods with regard to estimating the likelihood function, we now turn to methodology for estimation of its maximum with 95% confidence intervals.

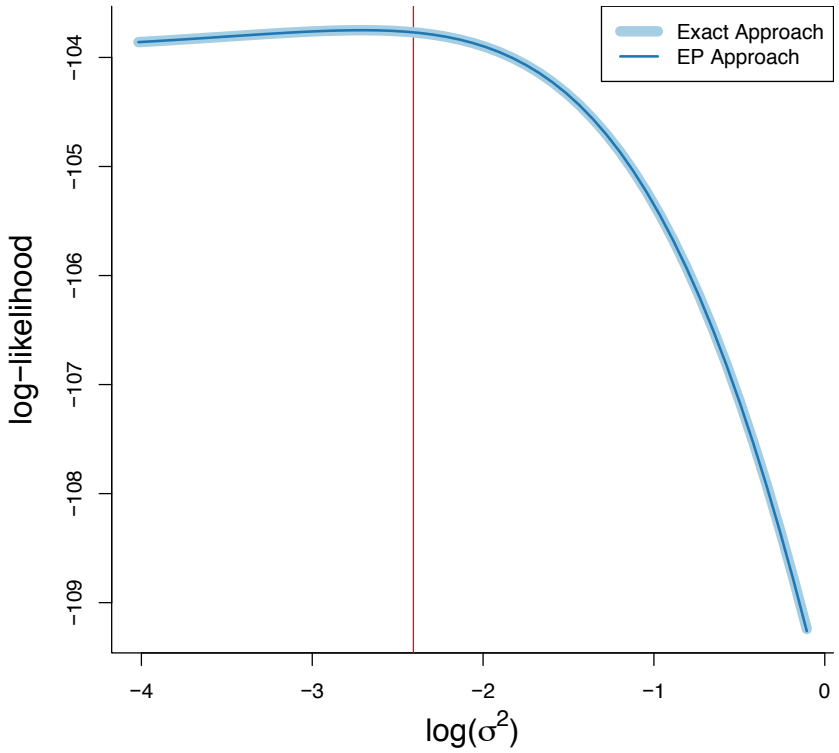


Figure 2.2: *A comparison plot of the log-likelihood surface over the parameter* σ^2 *for probit models calculated using univariate quadrature and EP. The true* σ^2 *value is represented by the red line on the log scale and other lines follow the legend. The data generated had a 30 groups with 5 responses per group and the true value of* σ^2 *was* 0.09.

2.4 Computing point estimates and confidence intervals

The maximum likelihood estimator of the probit mixed model parameter σ^2 for the quadrature and EP approach are respectively given by $\widehat{\sigma}^2 = \underset{\sigma^2}{\text{argmax}}\ \ell(\sigma^2)$ and $\widehat{\underset{\sim}{\sigma}}^2 = \underset{\sigma^2}{\text{argmax}}\ \underset{\sim}{\ell}(\sigma^2)$. To find their stationary points we require the first derivative of the likelihood functions $\ell'(\sigma^2)$ and $\underset{\sim}{\ell}'(\sigma^2)$. Calculation of the second derivative (denoted by $\ell''(\sigma^2)$ and $\underset{\sim}{\ell}''(\sigma^2)$) facilitates calculation of confidence intervals. Using the second derivatives, we now show how to obtain confidence intervals.

2.4.1 Confidence interval estimation

The overarching schematic for calculation of confidence intervals for both approximate and exact approaches to estimating the σ^2 parameter are analogous so we show it only for the exact case. Since the variance parameter σ^2 is constrained to be a positive number it is more appropriate to work with the parameter in the transformed space

$$\omega \equiv \log(\sigma) = \frac{1}{2}\log(\sigma^2) = g(\sigma^2).$$

Given the maximum likelihood estimator is asymptotically normally distributed, the transfomed parameter estimate is

$$\widehat{\omega} \sim N\left(g(\sigma^2_{\text{true}}), \frac{1}{\boldsymbol{I}\big(g(\widehat{\sigma^2})\big)}\right). \tag{2.26}$$

Since

$$\boldsymbol{I}\big(g(\widehat{\sigma^2})\big) = \frac{\boldsymbol{I}(\widehat{\sigma^2})}{\big(g'(\widehat{\sigma^2})\big)^2} \tag{2.27}$$

and $g'(\sigma^2) = (2\sigma^2)^{-1}$, it follows

$$\widehat{\omega} \sim N\left(\omega_{\text{true}}, \frac{1}{(2\widehat{\sigma^2})^2\big(-\ell''(\widehat{\sigma^2})\big)}\right).$$

Thus for a 95% confidence interval we expect that

$$0.95 \approx P\left(\widehat{\omega} - \frac{1.96}{\sqrt{(2\widehat{\sigma^2})^2\big(-\ell''(\widehat{\sigma^2})\big)}} < \omega_{\text{true}} < \widehat{\omega} + \frac{1.96}{\sqrt{(2\widehat{\sigma^2})^2\big(-\ell''(\widehat{\sigma^2})\big)}}\right).$$

Setting

$$\omega_{\text{low}} = \frac{1}{2}\log(\widehat{\sigma^2}) - \frac{1.96}{\sqrt{(2\widehat{\sigma^2})^2\big(-\ell''(\widehat{\sigma^2})\big)}} \quad \text{and} \quad \omega_{\text{upp}} = \frac{1}{2}\log(\widehat{\sigma^2}) + \frac{1.96}{\sqrt{(2\widehat{\sigma^2})^2\big(-\ell''(\widehat{\sigma^2})\big)}},$$

the lower and upper 95% confidence intervals for parameter σ^2 are given by

$$\sigma^2_{\text{low}} = \exp(2\omega_{\text{low}}) \quad \text{and} \quad \sigma^2_{\text{upp}} = \exp(2\omega_{\text{upp}}).$$

2.4.2 Derivative approximation

For the quadrature and EP approach to obtaining the σ^2 log-likelihood, there are multiple methods by which the first and second derivatives can be obtained. These solutions are either based on analytical or quasi-Newtown solutions to the derivatives required.

Quasi-Newton methods negate the need to analytically compute the first and second derivatives and can return the Hessian matrix for higher dimensional cases. We implement both Nelder-Mead (NM) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms for optimisation via the R function `optim()` in the "stats" package (R Core Team, 2019[56]). We can use unconstrained optimisation on log transformed σ^2, then back transform to calculate confidence intervals. First we use the more robust NM search to roughly locate the maxima, then refine it via a BFGS search. Using the BFGS method for optimisation at the maxima via the `optim()` function has the additional benefit of returning the second derivative. We initialise the NM algorithm using an estimate for the optimal value $\widehat{\sigma^2}$ via Laplace approximation, which is provided by the R function `glmer()` in the package "lme4" (Bates, et al., 2018).[5] Implementation in R of optimisation for both quadrature and EP approaches involves using a simple wrapper function around the parameter to be optimised, such that optimisation is on the unconstrained space.

The analytical solution may be available at a lower computational cost compared to quasi-Newtown approaches, however are without doubt more algebraically intensive. First, for the exact likelihood we let

$$J_{si}(\sigma^2) \equiv (2\pi\sigma^2)^{-1/2} \int_{-\infty}^{\infty} u_i^s \prod_{j=1}^{n} \Phi\big((2y_{ij}-1)u_i\big) \exp(-u_i^2/2\sigma^2) du_i,$$

where $s \in \{0,1,2,3,4,5,6\}$, such that

$$\ell(\sigma^2) = \sum_{i=1}^{m} \log\big(J_{0i}(\sigma^2)\big).$$

Then the first and second derivatives of the likelihood function can be shown to be,

$$\ell'(\sigma^2) = \frac{1}{2\sigma^4}\left(\sum_{i=1}^{m} \frac{J_{2i}(\sigma^2)}{J_{0i}(\sigma^2)} - m\sigma^2\right) \tag{2.28}$$

and

$$\ell''(\sigma^2) = \frac{1}{4\sigma^8}\left\{\sum_{i=1}^{m}\left\{\frac{J_{4i}(\sigma^2) - 4\sigma^2 J_{2i}(\sigma^2)}{J_{0i}(\sigma^2)} - \left(\frac{J_{2i}(\sigma^2)}{J_{0i}(\sigma^2)}\right)^2\right\} + 2m\sigma^4\right\}. \tag{2.29}$$

In the quadrature case, each $J_{si}(\sigma^2)$ is computed individually before being summed to compute the relevant derivative likelihood. To improve numerical stability we re-implement the work in Section 2.1 to find the maximum value of the integral within each $J_{si}(\sigma^2)$ and limit the range of values each integral can obtain to between 0 and 1, i.e.

$$J_{si}(\sigma^2) \equiv (2\pi\sigma^2)^{-1/2}\int_{-\infty}^{\infty} u_i^s \exp\big(h(u_i) - h(u_{0i})\big)du_i \exp\big(h(u_{0i})\big), \tag{2.30}$$

where $h(x)$ and u_{0i} are defined as in Section 2.1. Note that the $\exp\big(h(u_{0i})\big)$ term becomes superfluous since all $J_{si}(\sigma^2)$ are calculated as ratios of other $J_{si}(\sigma^2)$. We use a bisection search over the surface of $\ell'(\sigma^2)$ to find the minimum and then compute the confidence intervals as explained in Section 2.4.1. Figure 2.3 shows plots of the likelihood and its first and second derivatives obtained via adaptive Gauss-Hermite quadrature.

EP approximation of the maximum likelihood estimate $\widehat{\underset{\sim}{\sigma^2}}$ benefits from letting

$$\underset{\sim}{J}_{si}(\sigma^2) \equiv (2\pi\sigma^2)^{-1/2}\int_{-\infty}^{\infty} u_i^s \underset{\sim}{w}_i(u_i)\exp\big(-u_i^2/(2\sigma^2)\big)du_i,$$

where

$$\underset{\sim}{w}_i(u) \equiv \exp\left\{\begin{bmatrix}1\\ u\\ u^2\end{bmatrix}^{\top} \mathrm{SUM}\big(\eta_{p(y_i|u_i)\to u_i}\big)\right\}$$

such that

$$\underset{\sim}{\ell}(\sigma^2) = \sum_{i=1}^{m}\log\big(\underset{\sim}{J}_{0i}(\sigma^2)\big).$$

Since $\underset{\sim}{w}_i(u)$ is proportional to a normal density function closed form solutions for the integrals arising in $\underset{\sim}{\ell}'(\sigma^2)$ and $\underset{\sim}{\ell}''(\sigma^2)$ exist as before. There are two different approaches for obtaining EP approximations of $\underset{\sim}{J}_{si}(\sigma^2)$, which we now discuss.

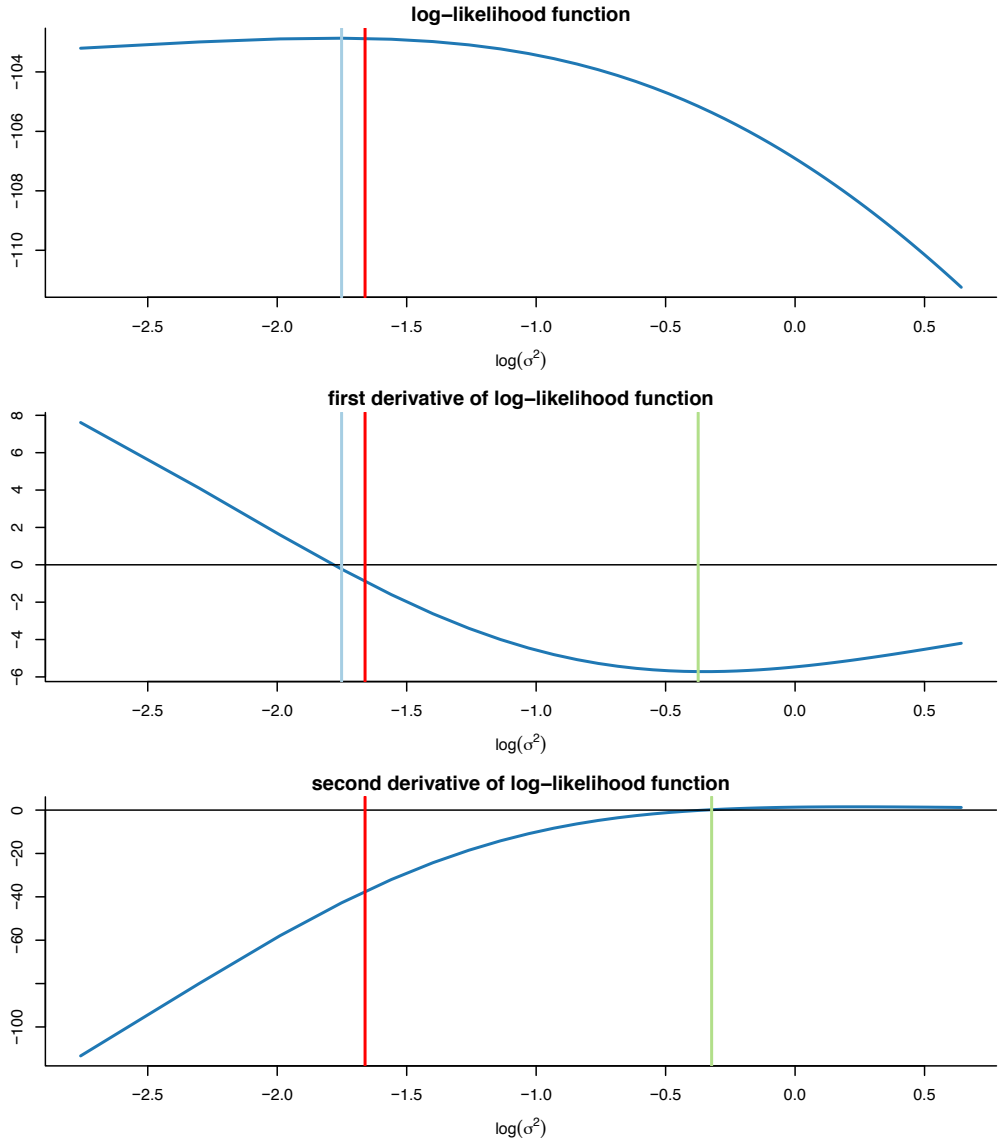


Figure 2.3: *Plot of the $\ell(\sigma^2)$ and its first two derivatives solved via adaptive Gauss-Hermite quadrature. The red line represents the true $\sigma^2 = 0.19$, the light blue line is the point where the first derivative cuts the x axis and the green line is analogous to the light blue line for the second derivative. The number of groups was 30 with 5 responses per group.*

2.4.2.1 Expectation propagation analytical approach I

The first approach involves directly approximating the $\underset{\sim}{J}_{si}(\sigma^2)$ via the natural parameters passed in Algorithm 3. Consider that,

$$\underset{\sim}{J}_{si}(\sigma^2) = \int_{-\infty}^{\infty} u^s \exp\left\{ \begin{bmatrix} 1 \\ u \\ u^2 \end{bmatrix}^\top \boldsymbol{\eta}_i^{\boxplus} \right\} du,$$

where

$$\boldsymbol{\eta}_i^{\boxplus} = \begin{bmatrix} \eta_{0i}^{\boxplus} \\ \eta_{1i}^{\boxplus} \\ \eta_{2i}^{\boxplus} \end{bmatrix} \equiv \boldsymbol{\eta}_{\sigma^2} + \text{SUM}\big(\boldsymbol{\eta}_{p(y_i|u_i)\to u_i}\big) \tag{2.31}$$

and $\boldsymbol{\eta}_{\sigma^2}$ is defined in equation (2.8). Since u_i is a normal random variable with natural parameters $\eta_{1i}^{\boxplus}$ and $\eta_{2i}^{\boxplus}$, we can write

$$\underset{\sim}{J}_{si}(\sigma^2) = (2\pi)^{-1/2} \exp\left(\eta_{0i}^{\boxplus} u + A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\right) E(u^s),$$

where

$$E\left(u^s\right) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} u^s \exp\left(\eta_{1i}^{\boxplus} u + \eta_{2i}^{\boxplus} u^2 - A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\right) du.$$

The integrals required for these expressions can be solved with closed form solutions. Consider Result 7.

Result 7. *If x is a normal random variable with natural parameters η_1 and η_2, then*

$$\begin{aligned}
E(x) &= -\frac{\eta_1}{2\eta_2}, \\
E(x^2) &= \frac{\eta_1^2}{4\eta_2^2} - \frac{1}{2\eta_2}, \\
E(x^3) &= \frac{3\eta_1}{4\eta_2^2} - \frac{\eta_1^3}{8\eta_2^3}, \\
E(x^4) &= \frac{\eta_1^4}{16\eta_2^4} - \frac{3\eta_1^2}{4\eta_2^3} + \frac{3}{4\eta_2^2}, \\
E(x^5) &= \frac{5\eta_1^3}{8\eta_2^4} - \frac{\eta_1^5}{32\eta_2^5} - \frac{15\eta_1}{8\eta_2^3},
\end{aligned}$$

and

$$E(x^6) = \frac{\eta_1^6}{64\eta_2^6} - \frac{15\eta_1^4}{32\eta_2^5} + \frac{45\eta_1^2}{16\eta_2^4} - \frac{15}{8\eta_2^3}.$$

By Result 7,

$$\begin{aligned}
\underset{\sim}{J}_{0i}(\sigma^2) &= (2\pi)^{-1/2} \exp\left(\eta_{0i}^{\boxplus} + A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\right), \\
\underset{\sim}{J}_{1i}(\sigma^2) &= (2\pi)^{-1/2} \exp\left(\eta_{0i}^{\boxplus} + A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\right)\left(-\eta_{1i}^{\boxplus}/(2\eta_{2i}^{\boxplus})\right), \\
\underset{\sim}{J}_{2i}(\sigma^2) &= (2\pi)^{-1/2} \exp\left(\eta_{0i}^{\boxplus} + A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\right)\left((\eta_{1i}^{\boxplus})^2 - 2\eta_{2i}^{\boxplus}\right)/\left(4(\eta_{2i}^{\boxplus})^2\right)
\end{aligned}$$

and

$$\underset{\sim}{J}_{4i}(\sigma^2) = (2\pi)^{-1/2} \exp\left(\eta_{0i}^{\boxplus} + A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\right)\left\{(\eta_{1i}^{\boxplus})^4 + 12\eta_{2i}^{\boxplus}\left(\eta_{2i}^{\boxplus} - (\eta_{1i}^{\boxplus})^2\right)\right\}/\left(16(\eta_{2i}^{\boxplus})^4\right).$$

To aid with simplification of the following calculations we introduce Result 8.

Result 8. *When $s = \{1, 2, 4\}$, ratios of the form $\frac{\underset{\sim}{J}_{si}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)}$ can be simplified to*

$$\begin{aligned}
\frac{\underset{\sim}{J}_{1i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} &= -\eta_{1i}^{\boxplus}/(2\eta_{2i}^{\boxplus}), \\
\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} &= \left((\eta_{1i}^{\boxplus})^2 - 2\eta_{2i}^{\boxplus}\right)/\left(4(\eta_{2i}^{\boxplus})^2\right), \\
\frac{\underset{\sim}{J}_{4i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} &= \left\{(\eta_{1i}^{\boxplus})^4 + 12\eta_{2i}^{\boxplus}\left(\eta_{2i}^{\boxplus} - (\eta_{1i}^{\boxplus})^2\right)\right\}/\left(16(\eta_{2i}^{\boxplus})^4\right).
\end{aligned}$$

Recalling the form of the exact expression from equation (2.28), we can implement an EP approximation approach to the first derivative by replacing each J_{si} with $\underset{\sim}{J}_{si}$. This leads to

$$\underset{\sim}{\ell}'(\sigma^2) = \frac{1}{2\sigma^4}\left(\sum_{i=1}^{m}\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} - m\sigma^2\right).$$

Analogously, the EP approximation approach to the second derivative can be obtained by replacing each J_{si} by $\underset{\sim}{J}_{si}$ in the $\ell''(\sigma^2)$ expression given in equation (2.29), such that

$$\underset{\sim}{\ell}''(\sigma^2) = \frac{1}{4\sigma^8}\left\{\sum_{i=1}^{m}\left\{\frac{\underset{\sim}{J}_{4i}(\sigma^2) - 4\sigma^2\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} - \left(\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)}\right)^2\right\} + 2m\sigma^4\right\}.$$

In summary, at the end of Algorithm 2.2.3 when the natural parameters have converged, they can be used to calculate the derivatives required. Algorithm 4 aims to clarify this process.

Algorithm 4 *Alteration of Algorithm 3 to calculate the log-likelihood of σ^2 and its first and second derivatives via EP analytical approach I.*

After convergence is reached of Algorithm 3, we can obtain $\underset{\sim}{\ell}'(\sigma^2)$ and $\underset{\sim}{\ell}''(\sigma^2)$ by,

$$\underset{\sim}{\ell}'(\sigma^2) = \frac{1}{2\sigma^4}\left(\sum_{i=1}^{m}\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} - m\sigma^2\right),$$

$$\underset{\sim}{\ell}''(\sigma^2) = \frac{1}{4\sigma^8}\left\{\sum_{i=1}^{m}\left\{\frac{\underset{\sim}{J}_{4i}(\sigma^2) - 4\sigma^2\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} - \left(\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)}\right)^2\right\} + 2m\sigma^4\right\}$$

where the ratios of $\underset{\sim}{J}_{si}$ are defined in Result 8.

2.4.2.2 Expectation propagation analytical approach II

The second approach involves approximating each $\underset{\sim}{\ell}'_i(\sigma^2)$ and $\underset{\sim}{\ell}''_i(\sigma^2)$ by finding the required $\underset{\sim}{J}_{si}(\sigma^2)$ and then summing them to obtain the first and second derivatives of the likelihood. As opposed to the previous approach where we project $f_{\text{input}}(x)$ (as defined in equation (2.5)) onto the normal family, here we project it onto what we refer to as the "power normal" family, which has the general form of

$$p(x) \propto x^s(2\pi\sigma^2)^{-1/2}\exp\big(-(x-\mu)^2/(2\sigma^2)\big), \quad s \in \{0, 2, 4\}.$$

As before, closed form solutions to the integrals required exist, however the algebra neccessary becomes increasingly difficult as s increases. Consider then, the following family of unnormalised power normal density functions written in exponential family form

$$f_{\text{UN}}(x) = \exp\left\{ \begin{bmatrix} x^s \\ x^{s+1} \\ x^{s+2} \end{bmatrix}^\top \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \end{bmatrix} \right\}, \tag{2.32}$$

with natural parameters $\eta_0, \eta_1 \in \mathbb{R}$ and $\eta_2 < 0$. Then as before, we wish to find

$$\text{proj}_{\text{UN}}[f_{\text{input}}](x) = \exp\left\{ \begin{bmatrix} x^s \\ x^{s+1} \\ x^{s+2} \end{bmatrix}^\top \boldsymbol{\eta}^* \right\}, \tag{2.33}$$

where

$$\boldsymbol{\eta}^* \equiv \begin{bmatrix} \eta_0^* \\ \eta_1^* \\ \eta_2^* \end{bmatrix}$$

and

$$(\eta_0^*, \eta_1^*, \eta_2^*) = \underset{(\eta_0,\eta_1,\eta_2)\in H}{\text{argmin}} \ \text{KL}(f_{\text{input}} \parallel f_{\text{UN}}).$$

For the first and second derivatives of probit binary GLMMs, EP requires repeated projection of the form given in equation (2.5) onto an unnormalised power normal distribution where $\eta_1^{\text{input}} \in \mathbb{R}$ and $\eta_2^{\text{input}} < 0$. Consider Result 9.

Result 9. *For an unnormalised input function $f \in L_1$ such that $f(x) \geq 0$ for all $x \in \mathbb{R}$ where $C_f \equiv \int_{\mathbb{R}} f(x)dx$, the projection onto the unnormalised power normal family is*

$$proj_{UPN}[f] = C_f proj_{PN}[f/C_f].$$

where $proj_{PN}[.]$ is the projection onto the power normal family and $proj_{UPN}[.]$ is the projection onto the unnormalised power normal family.

As before, the optimal natural parameters η_1^* and η_2^* are given according to the projection of the normalised function f/C_f onto the power normal family. We can subsequently use these optimal natural parameters to find the normalising natural

parameter η_0^* via Result 4 and thus obtain the projection onto unnormalised power normal family.

Result 10. *When the input density follows the form of equation (2.5), projections onto the power normal family η_0^* is given by*

$$\eta_0^* = \log(C_f) - A_s(\eta_1^*, \eta_2^*) + log\ h_s(x),$$

where

$$A_s(\eta_1, \eta_2) = \frac{-\eta_1^2}{4\eta_2} - \frac{1}{2}\log(-2\eta_2) + \log\big(E(x^s)\big), \quad h_s(x) = \frac{x^s}{\sqrt{2\pi}}$$

and $E(x^s)$ is given by Result 7.

Thus to obtain the required projection, we first obtain the optimal natural parameters η_1^* and η_2^* to project onto the power normal family. To do this, first consider the normalised approximation of the input density to be of the form

$$f_{\text{input}}(x) = x^s(2\pi\sigma^2)^{-1/2}\exp\big(-(x-\mu)^2/(2\sigma^2)\big)\big(E(x^s)\big)^{-1},$$

where

$$E(x^s) = \int_{-\infty}^{\infty} x^s(2\pi\sigma^2)^{-1/2}\exp\big(-(x-\mu)^2/(2\sigma^2)\big)dx$$

is used to normalise the unnormalised approximation of $f_{\text{input}}(x)$ from the power normal family. Next note that using matrix notation and converting to natural parameters,

$$f_{\text{input}}(x) = x^s \exp\left\{ \begin{bmatrix} x \\ x^2 \end{bmatrix}^\top \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} - \left\{ -\frac{1}{2}\log(-2\eta_2) + \log\big(E(x^s)\big) - \frac{\eta_1^2}{4\eta_2} \right\} \right\} (2\pi)^{-1/2}.$$

As in the proof of Result 5, this can be expressed as

$$f_{\text{input}}(x) \quad = \quad \exp\big(\boldsymbol{T}(x)^\top \boldsymbol{\eta}_{-1} - A_s(\boldsymbol{\eta}_{-1})\big)h_s(x),$$

where

$$\boldsymbol{T}(x) \equiv \begin{bmatrix} x \\ x^2 \end{bmatrix}, \quad h_s(x) \equiv \frac{x^s}{\sqrt{2\pi}}, \quad \boldsymbol{\eta}_{-1} \equiv \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \equiv \begin{bmatrix} \mu/\sigma^2 \\ -1/\left(2\sigma^2\right) \end{bmatrix} \tag{2.34}$$

and

$$A_s(\boldsymbol{\eta}_{-1}) \quad = \quad \frac{-\eta_1^2}{4\eta_2} - \frac{1}{2}\log(-2\eta_2) + \log\big\{E(x^s)\big\}.$$

Consider the following relationship which defines the gradient of the likelihood function at $\boldsymbol{\eta}_{-1}$:

$$\nabla f(\boldsymbol{\eta}_{-1}) \quad \equiv \quad (\nabla A_s)(\boldsymbol{\eta}_{-1}) - \boldsymbol{\tau}. \tag{2.35}$$

Note that when $\nabla f(\boldsymbol{\eta}_{-1}) = 0$ we have

$$\boldsymbol{\eta}_{-1} \equiv (\nabla A_s)^{-1}(\boldsymbol{\tau})$$

where

$$\boldsymbol{\tau} = \begin{bmatrix} \mathcal{M}_{s+1}/\mathcal{M}_s \\ \mathcal{M}_{s+2}/\mathcal{M}_s \end{bmatrix}, \qquad (\nabla A_s)(\boldsymbol{\eta}_{-1}) \quad = \quad \begin{bmatrix} \frac{\partial}{\partial \eta_1} A_s(\boldsymbol{\eta}_{-1}) \\ \frac{\partial}{\partial \eta_2} A_s(\boldsymbol{\eta}_{-1}) \end{bmatrix}$$

and $\mathcal{M}_s$ is defined as per Section 2.7.2 of the appendix. To find the optimal natural parameters for the projection required we must minimise the function $(\nabla A_s)(\boldsymbol{\eta}) - \boldsymbol{\tau}$ such that we obtain the $\boldsymbol{\eta}^*_{-1}$ where it is 0. We do this using a Newton-Raphson search, initialised at a rough estimate of the value that minimises the sum of squares deviation from the root of the inverse-defining equation,

$$\boldsymbol{\eta}^*_{-1} = \text{argmin} \sum \left((\nabla A_s)(\boldsymbol{\eta}_{-1}) - \boldsymbol{\tau} \right)^2,$$

found via a Nelder-Mead search initialised at $(\nabla A_s)(\boldsymbol{\eta})$ for the normal family. We can accomplish the Nelder-Mead step using the same `optim()` function in R as in the previous section. To match our previous notation, consider Definition 12.

Definition 12. *Define the function*

$$k_{probitPower}(\boldsymbol{a}; c_0, c_1, s) = (\nabla A)^{-1}(\boldsymbol{\tau}, s), \tag{2.36}$$

where

$$\boldsymbol{\tau} = \begin{bmatrix} \mathcal{M}_{s+1}(c_0, c_1, \boldsymbol{a})/\mathcal{M}_s(c_0, c_1, \boldsymbol{a}) \\ \mathcal{M}_{s+2}(c_0, c_1, \boldsymbol{a})/\mathcal{M}_s(c_0, c_1, \boldsymbol{a}) \end{bmatrix},$$

$(\nabla A)^{-1}(\boldsymbol{\tau}, s)$ *is the function that returns* $\boldsymbol{\eta}_{-1}$ *when* $\nabla f(\boldsymbol{\eta}_{-1}) = 0$ *(as per equation (2.35)) and* $\mathcal{M}_{s+k}(c_0, c_1, \boldsymbol{a})$ *is*

$$\begin{aligned}
\mathcal{M}_0(c_0, c_1, \boldsymbol{a}) &= \mathcal{W}_0(r_8, r_9), \\
\mathcal{M}_1(c_0, c_1, \boldsymbol{a}) &= a_1\mathcal{W}_0(r_8, r_9) + c_1 r_9^{-1}\mathcal{W}_1(r_8, r_9), \\
\mathcal{M}_2(c_0, c_1, \boldsymbol{a}) &= a_1^2\mathcal{W}_0(r_8, r_9) + 2a_1c_1r_9^{-1}\mathcal{W}_1(r_8, r_9) - 2a_2\mathcal{W}_2(r_8, r_9), \\
\mathcal{M}_3(c_0, c_1, \boldsymbol{a}) &= a_1^3\mathcal{W}_0(r_8, r_9) + 3a_1^2c_1r_9^{-1}\mathcal{W}_1(r_8, r_9) - 6a_1a_2\mathcal{W}_2(r_8, r_9) \\
&\quad -2c_1r_9^{-1}a_2\mathcal{W}_3(r_8, r_9), \\
\mathcal{M}_4(c_0, c_1, \boldsymbol{a}) &= a_1^4\mathcal{W}_0(r_8, r_9) + 4a_1^3c_1r_9^{-1}\mathcal{W}_1(r_8, r_9) - 12a_1^2a_2\mathcal{W}_2(r_8, r_9) \\
&\quad -8a_1a_2c_1r_9^{-1}\mathcal{W}_3(r_8, r_9) + 4a_2^2\mathcal{W}_4(r_8, r_9), \\
\mathcal{M}_5(c_0, c_1, \boldsymbol{a}) &= a_1^5\mathcal{W}_0(r_8, r_9) + 5a_1^4c_1r_9^{-1}\mathcal{W}_1(r_8, r_9) - 20a_1^3a_2\mathcal{W}_2(r_8, r_9) \\
&\quad -20a_1^2a_2c_1r_9^{-1}\mathcal{W}_3(r_8, r_9) + 20a_1a_2^2\mathcal{W}_4(r_8, r_9) + 4a_2^2c_1r_9^{-1}\mathcal{W}_5(r_8, r_9), \\
\mathcal{M}_6(c_0, c_1, \boldsymbol{a}) &= a_1^6\mathcal{W}_0(r_8, r_9) + 6a_1^5c_1r_9^{-1}\mathcal{W}_1(r_8, r_9) - 30a_1^4a_2\mathcal{W}_2(r_8, r_9) \\
&\quad -40a_1^3a_2c_1r_9^{-1}\mathcal{W}_3(r_8, r_9) + 60a_1^2a_2^2\mathcal{W}_4(r_8, r_9) \\
&\quad +24a_1a_2^2c_1r_9^{-1}\mathcal{W}_5(r_8, r_9) - 8a_2^3\mathcal{W}_6(r_8, r_9),
\end{aligned}$$

for $\mathcal{W}_{s+k}(a, b)$ *defined as in equation (2.47) and (2.43),* $\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$, $r_1 = \sqrt{2(2 - c_1^2a_2^{-1})}$, $r_2 = (2c_0 - c_1a_2^{-1}a_1)r_1^{-1}$, $r_8 = 2r_2r_1$ *and* $r_9 = c_1(-2a_2)^{-1/2}$.

Note that the derivation of Definition 12 follows that of Definition 5 and as such we do not provide details and instead refer interested readers to Appendix 2.7.2. Each $\mathcal{M}_s(c_0, c_1, \boldsymbol{a})$ is found by simple algebra in an analogous manner to those required for Definition 10. The first and second derivatives required for both search methods of $(\nabla A)^{-1}(\boldsymbol{\tau}, s)$ are provided in Appendix 2.7.4. As before, to obtain the required projection, we present Result 11.

Result 11. *Given f follows the form of equation (2.5), the projection onto the power normal family is given by*

$$proj_{PN}[f_{input}] = \exp\left(\boldsymbol{T}(x)^\top \boldsymbol{\eta}^*_{-1} - A_s(\boldsymbol{\eta}^*_{-1})\right) h_s(x)$$

where

$$\boldsymbol{\eta}^*_{-1} = k_{probitPower}\left(\boldsymbol{\eta}^{input}; c_0, c_1, s\right), \qquad \boldsymbol{\eta}^{input}_{-1} \equiv \begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}^*_{-1} \equiv \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix},$$

and $\boldsymbol{T}(x)$ and $h_s(x)$ follow from equation (2.34).

We now require a function to find the normalising natural parameter η_0^* required for the projection onto unnormalised power normal family. Following analogous arguments used for the projection onto the unnormalised normal family, we define

$$c_{\text{probitPower}}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}; c_0, c_1, s\right) = \log\, \mathcal{M}_s\left(c_0, c_1, \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}\right) - A_s\left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\right) - \log\, h_s(x),$$

where $\log\left(\mathcal{M}_s(c_0, c_1, \boldsymbol{a})\right)$, $A_s(\boldsymbol{b})$ and $\log\, h_s(x)$ are defined as per the previous definitions. We present Algorithm 5, which requires three loops over the main portion of the algorithm to calculate the each $\underset{\sim}{J}_{si}(\sigma^2)$, as opposed to the previous method. Additionally note that each $\underset{\sim}{J}_{si}(\sigma^2)$ is found in each loop and then used to find the first and second derivative of the likelihood.

Algorithm 5 *Algorithm to calculate the log-likelihood of σ^2 and its first and second derivatives via EP analytical approach II.*

Set constants: $c_0 \leftarrow 0; \quad c_{1_{ij}} \leftarrow 2y_{ij} - 1; \quad 1 \leq i \leq m, \quad 1 \leq j \leq n.$

$$\boldsymbol{\eta}_{p(u_i;\sigma^2)\to u_i} \leftarrow \boldsymbol{\eta}_{\sigma^2} \equiv \begin{bmatrix} -\frac{1}{2}\log(2\pi\sigma^2) \\ 0 \\ 1/(2\sigma^2) \end{bmatrix}, \quad 1 \leq i \leq m.$$

For $s = 0, 2, 4$:

For $i = 1, ..., m$:

Initialise: $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i} \in \mathbb{R}, \quad 1 \leq j \leq n$ as per the equation (2.25).

Cycle:

$\text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big) \leftarrow \sum_{j=1}^{n} \big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)$

For $j = 1, ..., n$:

$$\begin{aligned} \boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)} &\leftarrow \boldsymbol{\eta}_{\sigma^2} + \text{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big) - \boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i} \\ \big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_{1:2} &\leftarrow k_{\text{probitPower}}\Big(\big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}; c_0, c_{1_{ij}}, s\Big) \\ &\quad - \big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2} \end{aligned}$$

until all natural parameter vectors converge.

For $j = 1, ..., n$:

$$\begin{aligned} \big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_0 &\leftarrow c_{\text{probitPower}}\Big(\big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}, \big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_{1:2} \\ &\quad + \big(\boldsymbol{\eta}_{u_i\to p(y_{ij}|u_i)}\big)_{1:2}; c_0, c_{1_{ij}}, s\Big) \\ \boldsymbol{\eta}_i^{\boxplus} &\equiv \boldsymbol{\eta}_{\sigma^2} + \text{SUM}\big(\boldsymbol{\eta}_{p(y_i|u_i)\to u_i}\big) \\ \underset{\sim}{J}_{0i}(\sigma^2) &= (2\pi)^{-1/2}\exp\big(\eta_{0i}^{\boxplus} + A(\eta_{1i}^{\boxplus}, \eta_{2i}^{\boxplus})\big), \\ \text{If } s = 2, \quad \underset{\sim}{J}_{2i}(\sigma^2) &= \underset{\sim}{J}_{0i}(\sigma^2)\big((\eta_{1i}^{\boxplus})^2 - 2\eta_{2i}^{\boxplus}\big)/\big(4(\eta_{2i}^{\boxplus})^2\big), \\ \text{If } s = 4, \quad \underset{\sim}{J}_{4i}(\sigma^2) &= \underset{\sim}{J}_{0i}(\sigma^2)\Big\{(\eta_{1i}^{\boxplus})^4 + 12\eta_{2i}^{\boxplus}\big(\eta_{2i}^{\boxplus} - (\eta_{1i}^{\boxplus})^2\big)\Big\}/\big(16(\eta_{2i}^{\boxplus})^4\big). \end{aligned}$$

where the ratios of $\underset{\sim}{J}_{si}$ are defined in Result 8.

After convergence is reached.

$$\begin{aligned} \underset{\sim}{\ell}'(\sigma^2) &= \frac{1}{2\sigma^4}\left(\sum_{i=1}^{m}\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} - m\sigma^2\right), \\ \underset{\sim}{\ell}''(\sigma^2) &= \frac{1}{4\sigma^8}\left\{\sum_{i=1}^{m}\left\{\frac{\underset{\sim}{J}_{4i}(\sigma^2) - 4\sigma^2\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)} - \left(\frac{\underset{\sim}{J}_{2i}(\sigma^2)}{\underset{\sim}{J}_{0i}(\sigma^2)}\right)^2\right\} + 2m\sigma^4\right\}. \end{aligned}$$

2.5 Best predictor

In addition to estimating σ^2, we also wish to predict the random effect u_i for each group. First, consider that in the exact case

$$\begin{aligned} \text{BP}(u_i) &= E(u_i|\boldsymbol{y}_i) \\ &= \int_{-\infty}^{\infty} u_i p(u_i|\boldsymbol{y}_i;\sigma^2)du_i \\ &= \frac{\int_{-\infty}^{\infty} u_i p(\boldsymbol{y}_i|u_i)p(u_i;\sigma^2)du_i}{\int_{-\infty}^{\infty} p(\boldsymbol{y}_i|u_i)p(u_i;\sigma^2)du_i} = \frac{J_{1i}(\sigma^2)}{J_{0i}(\sigma^2)}, \end{aligned} \tag{2.37}$$

where $\boldsymbol{y}_i = (y_{i1}, ..., y_{in})$ and $J_{si}(\sigma^2)$ for $s \in \{0, 1, 2\}$ follows from equation (2.28). Equation (9.7) of McCulloch, Searle & Neuhaus (2008)[41] suggests that $E_{\boldsymbol{y}_i}\big(\text{Var}(u_i|\boldsymbol{y}_i)\big)$ provides a reasonable approximation of prediction errors i.e. $\text{Var}\big(\text{BP}(u_i) - u_i\big)$, where

$$\text{Var}(u_i|\boldsymbol{y}_i) = \frac{J_{2i}(\sigma^2)}{J_{0i}(\sigma^2)} - \left(\frac{J_{1i}(\sigma^2)}{J_{0i}(\sigma^2)}\right)^2.$$

However, the approximation is hindered by the expectation over the distribution of the $\boldsymbol{y}_i$ vector. For the EP case, the best predictor of u_i can be obtained via products of Algorithm 3. Let

$$\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i \equiv \boldsymbol{\eta}_\sigma^2 + \text{SUM}(\boldsymbol{\eta}_{p(\boldsymbol{y}_i|u_i)\to u_i}) = \begin{bmatrix} \widehat{\underset{\sim}{\eta}}_{i1} \\ \widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2} \end{bmatrix}, \tag{2.38}$$

where $\widehat{\underset{\sim}{\eta}}_{i1}$ corresponds to the first entry of $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i$, $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2}$ corresponds to the remaining entries and $\boldsymbol{\eta}_\sigma^2$ and $\text{SUM}(\boldsymbol{\eta}_{p(\boldsymbol{y}_i|u_i;\boldsymbol{\beta})\to u_i})$ are as previously defined in Algorithm 3 with $\sigma^2 = \widehat{\underset{\sim}{\sigma^2}}$. Note that Algorithm 3 involves using

$$\exp\left\{ \begin{bmatrix} 1 \\ u_i \\ u_i^2 \end{bmatrix}^\top \widehat{\underset{\sim}{\boldsymbol{\eta}}}_i \right\} \quad \text{to replace} \quad p(\boldsymbol{y}_i|u_i)p(u_i;\sigma^2).$$

We can approximate $\text{BP}(u_i) = E(\widehat{\underset{\sim}{u}}_i)$, where $\widehat{\underset{\sim}{u}}_i$ is univariate normal with natural parameters $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i$. Thus,

$$\text{BP}(u_i) = -\widehat{\underset{\sim}{\eta}}_{i1}/2\widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2}.$$

As before, the approximation of prediction errors given by $E_{\boldsymbol{y}_i}\big(\text{Var}(u_i|\boldsymbol{y}_i)\big)$, where

$$\underset{\sim}{\text{Var}}(u_i|\boldsymbol{y}) = -1/2\widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2},$$

is hindered by the expectation over the distribution over the $\boldsymbol{y}_i$ vector.

2.6 Simulation Study

A simulation study comparing four methods of obtaining maximum likelihood estimates and confidence intervals in Section 2.4 was run in the R computing environment (R Core Team, 2019[56]) on a MacBook Air laptop with two 2.2 gigahertz processors and 8 gigabytes of random access memory. The four methods are:

- Adaptive Gauss-Hermite quadrature using quasi-Newton optimisation and second derivative calculation.
- EP using quasi-Newton optimisation and second derivative calculation.
- EP using analytical approach I to first and second derivative calculation.
- EP using analytical approach II to first and second derivative calculation.

Preliminary studies of datasets of $m = 20$ groups with $n = 2$ observations per group showed a lack of stability for EP analytical approach II. It was deemed such instability was caused by poor starting values in the bisection search. Although it is possible to increase the number of iterations in the bisection and get convergence on a root, for speed starting values were based on the results of the quadrature approach using quasi-Newtown methods. Additionally, poor behaviour of the likelihood function for low sample sizes may have caused problems during the optimisation of its derivative.

With this in mind our study consisted of 100 repetitions of $\widehat{\sigma}$ estimates for each of the estimation methods for each set of simulated data. The true σ value of the dataset was either $\sigma_{\text{true}} = 0.25, 0.50$ or 1.00. Each of these true values were used to generate test datasets with $m = 50, 250$ and 1250 groups. The number of observations per group was fixed at $n = 10$. Due to the presence of multiple iterative loops, computational speed was severely compromised. Since the code used implements a variety of R packages which make calls to various low level languages, we omit a comparison of run speed and focus purely on the accuracy and stability of the methods.

A random sample of 10 confidence intervals for each method are compared visually in Figure 2.5, showing although there is generally parity between the methods, approach I to EP returned the tightest intervals by a small margin.

A comparison between the estimated $\widehat{\sigma}$ of each method and the σ_{true} for each dataset was conducted using Wilcoxon-tests. The p-values from this study are presented with

boxplots of log estimate error in Figure 2.4, where the error ϵ is given by

$$\epsilon = \frac{\|\widehat{\sigma} - \sigma_{\text{true}}\|}{\widehat{\sigma}}. \tag{2.39}$$

For each of the methods tested and each of the σ_{true} values, the variance decreased as the number of groups in the dataset increased. For the dataset where $\sigma_{\text{true}} = 0.5$, there was no statistical difference between the estimate and true parameter regardless of the number of groups in the dataset. For the dataset with $\sigma_{\text{true}} = 1$ and $m = 1250$ groups, the variance of the estimates for the exact case using `optim()` and approach I to EP increased, while the other methods did not. For datasets where $\sigma_{\text{true}} = 0.25$, there was no statistical difference for group sizes less than $m = 1250$. The discrepancy at $m = 1250$ is most likely due to the reduction of variance in datasets with a high number of groups which lead to an oversensitive test statistic.

Given the algebraic simplifications afforded by using quasi-Newton methods to facilitate derivative calculation in addition to the stability problems of other methods, there is little evidence it is worthwhile to continue with Approach I and Approach II to EP. Furthermore these approaches are difficult to implement for the general case of GLMMs where variates and random effects can be any dimension.

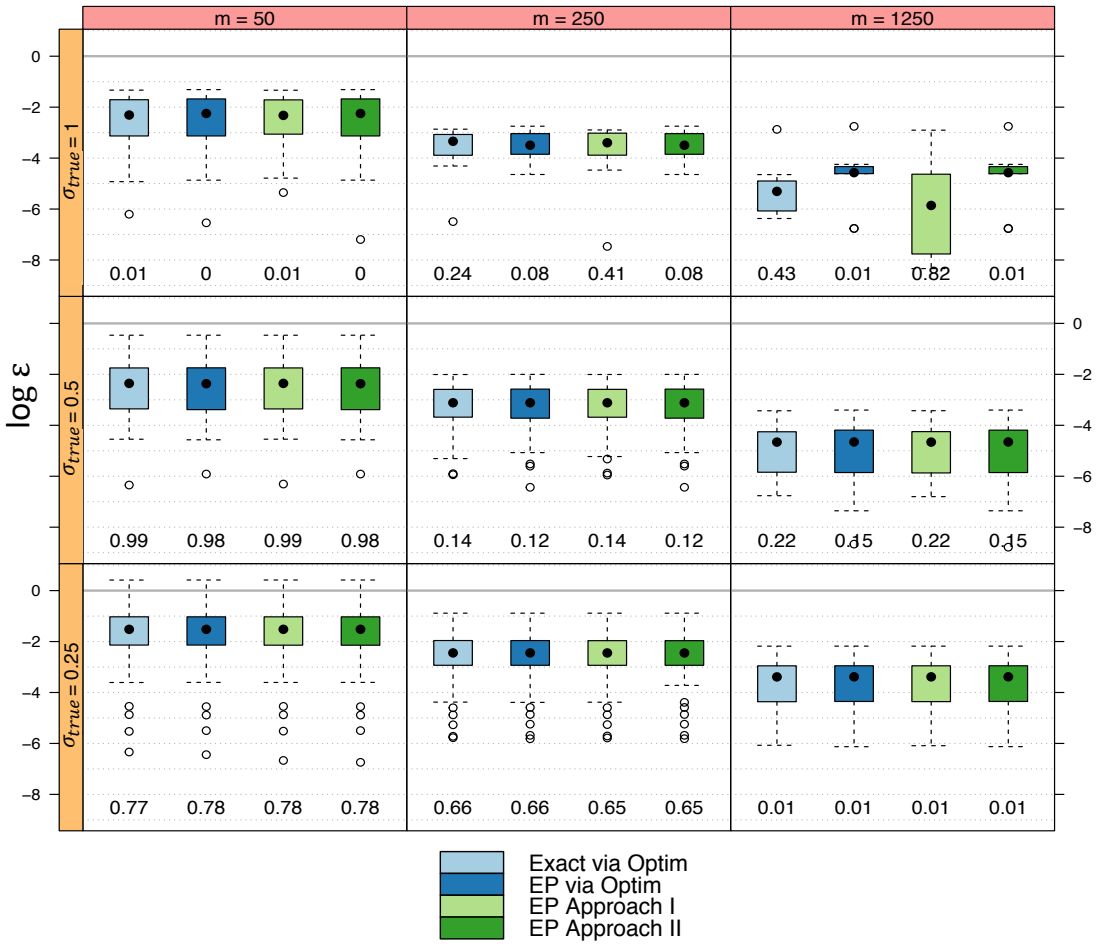


Figure 2.4: *Panel plot with error boxplots for the four methods of approximation discussed in this section. P-values from a Wilcoxon comparison test are shown under each boxplot. The legend shows which method producted each boxplot. The calculation of error is given in equation (2.39).*

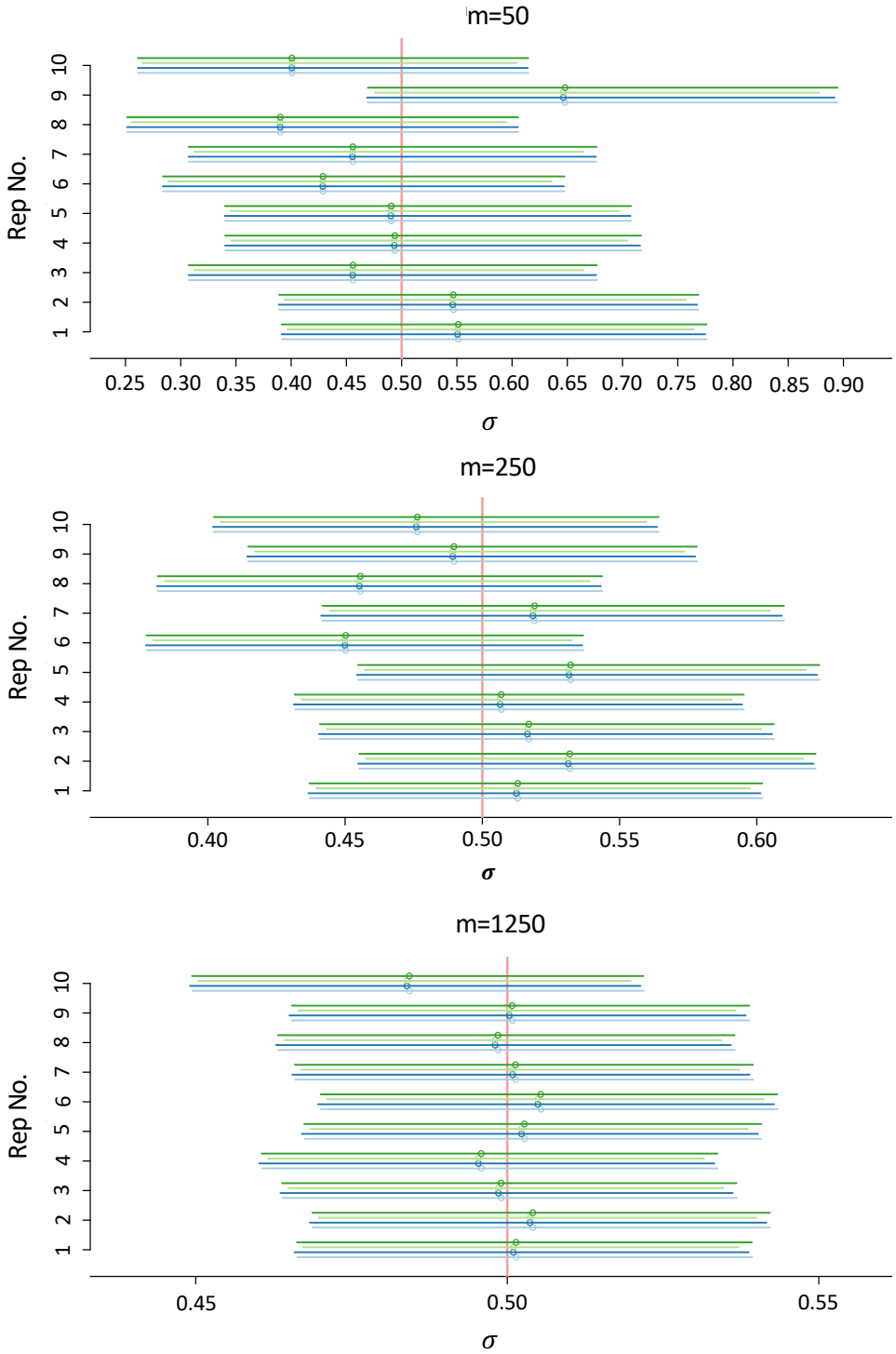


Figure 2.5: *Plot displaying 10 randomly selected 95% confidence intervals of each method across various group sizes and* $\sigma_{true} = 0.5$*, as shown by the red line. The lines correspond with the methods of this section as shown in the legend of Figure 2.4.*

2.7 Appendix

2.7.1 Proof of Result 3

Consider an unnormalised normal univariate density function

$$f_{\text{UN}}(x;\boldsymbol{\eta}) = \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta} \right\}.$$

Then the KL divergence of $f_{\text{UN}}(x;\boldsymbol{\eta})$ from f_{input} is

$$\begin{aligned} \text{KL}\big(f_{\text{input}} \parallel f_{\text{UN}}\big) &= \int_{\mathbb{R}} f_{\text{input}}(x) \log\big(f_{\text{input}}(x)/f_{\text{UN}}(x;\boldsymbol{\eta})\big) + f_{\text{UN}}(x;\boldsymbol{\eta}) - f_{\text{input}}(x)dx \\ &= \mathcal{K}(\boldsymbol{\eta}) + \text{const}, \end{aligned}$$

where

$$\mathcal{K}(\boldsymbol{\eta}) \equiv (2\pi)^{d/2} \exp\big(\boldsymbol{\eta}_0 + A(\boldsymbol{\eta}_{1:2})\big) - \begin{bmatrix} \int_{\mathbb{R}} f_{\text{input}}(x)dx \\ \int_{\mathbb{R}} x f_{\text{input}}(x)dx \\ \int_{\mathbb{R}} x^2 f_{\text{input}}(x)dx \end{bmatrix}^\top \boldsymbol{\eta},$$

and "const" represents all the terms not dependent on $\boldsymbol{\eta}$. Note that the derivative vector of $\mathcal{K}(\boldsymbol{\eta})$ is

$$\mathsf{D}\mathcal{K}(\boldsymbol{\eta}) = (2\pi)^{d/2} \exp\big(\boldsymbol{\eta}_0 + A(\boldsymbol{\eta}_{1:2})\big) \begin{bmatrix} 1 \\ \mathsf{D}A(\boldsymbol{\eta}_{1:2})^\top \end{bmatrix}^\top - \begin{bmatrix} \int_{\mathbb{R}} f_{\text{input}}(x)dx \\ \int_{\mathbb{R}} x f_{\text{input}}(x)dx \\ \int_{\mathbb{R}} x^2 f_{\text{input}}(x)dx \end{bmatrix}^\top.$$

Setting $\mathsf{D}\mathcal{K}(\boldsymbol{\eta})^\top = 0$ to minimise the $\text{KL}\big(f_{\text{input}} \parallel f_{\text{UN}}\big)$,

$$(2\pi)^{d/2} \exp\big(\boldsymbol{\eta}_0 + A(\boldsymbol{\eta}_{1:2})\big) \begin{bmatrix} 1 \\ \nabla A(\boldsymbol{\eta}_{1:2}) \end{bmatrix}^\top = \begin{bmatrix} \int_{\mathbb{R}} f_{\text{input}}(x)dx \\ \int_{\mathbb{R}} x f_{\text{input}}(x)dx \\ \int_{\mathbb{R}} x^2 f_{\text{input}}(x)dx \end{bmatrix}^\top$$

where $\nabla A(\boldsymbol{\eta}_{1:2}) \equiv \mathsf{D}A(\boldsymbol{\eta}_{1:2})^\top$ is the gradient vector of $A(\boldsymbol{\eta}_{1:2})$. It is then easy to show

$$\boldsymbol{\eta}_0^* = \log(C_{f_{\text{input}}}) - A(\boldsymbol{\eta}_{1:2}^*) - \frac{d}{2}\log(2\pi),$$

where

$$\boldsymbol{\eta}_{1:2}^* = (\nabla A)^{-1} \left\{ \begin{bmatrix} \int_{\mathbb{R}} x f_{\text{input}}(x)/C_{f_{\text{input}}} dx \\ \int_{\mathbb{R}} x^2 f_{\text{input}}(x)/C_{f_{\text{input}}} dx \end{bmatrix} \right\} \tag{2.40}$$

with existence and uniqueness of $(\nabla A)^{-1}$ being guaranteed by Proposition 3.2 of Wainwright & Jordan (2008),[66] and $C_{f_{\text{input}}} \equiv \int_{\mathbb{R}} f_{\text{input}}(x)dx$. The Hessian matrix of $\mathcal{K}(\boldsymbol{\eta})$ is

$$\begin{aligned} \mathsf{H}\mathcal{K}(\boldsymbol{\eta}) &= (2\pi)^{d/2} \exp\left(\boldsymbol{\eta}_0 + A(\boldsymbol{\eta}_{1:2})\right) \\ &\quad \times \left\{ \begin{bmatrix} 1 \\ \nabla A(\boldsymbol{\eta}_{1:2}) \end{bmatrix} \begin{bmatrix} 1 \\ \nabla A(\boldsymbol{\eta}_{1:2}) \end{bmatrix}^\top + \begin{bmatrix} 0 & 0 \\ 0 & \mathsf{H}A(\boldsymbol{\eta}_{1:2}) \end{bmatrix} \right\}. \end{aligned}$$

By Proposition 3.1 of Wainwright & Jordan (2008),[66] A is strictly convex on its domain and thus $\mathsf{H}A(\boldsymbol{\eta}_{1:2})$ is positive definite. Thus, $\mathsf{H}\mathcal{K}(\boldsymbol{\eta})$ is positive definite for all $\boldsymbol{\eta}$ and so equation (2.40) is the unique maximiser of $\text{KL}\big(f_{\text{input}} \parallel f_{\text{UN}}\big)$. Therefore

$$\text{proj}_{\text{UN}}[f_{\text{input}}](x) = \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\},$$

where $\boldsymbol{\eta}^*$ is as previously defined. However, $\boldsymbol{\eta}^*_{1:2}$ is the same natural parameter vector that arises via projection of $f_{\text{input}}/C_{f_{\text{input}}}$ onto the family of univariate normal density functions, thus

$$\text{proj}_{\text{N}}[f_{\text{input}}/C_{f_{\text{input}}}](x) = \exp\left\{ \begin{bmatrix} x \\ x^2 \end{bmatrix} \boldsymbol{\eta}^*_{1:2} - A(\boldsymbol{\eta}^*_{1:2}) \right\} (2\pi)^{-d/2}$$

which immediately leads to Result 3.

2.7.2 Proof of Result 5

We now detail closed form solutions to the integrals arising in the log-likelihood calculations of univariate GLMMs and its derivatives.

Lemma 1. *For the following integrals of the form* $\mathcal{W}_s(a,b) = \int_{-\infty}^{\infty} x^s \Phi(a+bx)\phi(x)dx$, *where* $s \in \{0,1,2,3,4,5,6\}$ *and* $a, b \in \mathbb{R}$, *the following closed form solutions exist:*

$$\mathcal{W}_0(a,b) = \int_{-\infty}^{\infty} \Phi(a+bx)\phi(x)dx = \Phi\left(\frac{a}{\sqrt{b^2+1}}\right), \tag{2.41}$$

$$\mathcal{W}_1(a,b) = \int_{-\infty}^{\infty} x\Phi(a+bx)\phi(x)dx = \frac{b}{\sqrt{b^2+1}}\phi\left(\frac{a}{\sqrt{b^2+1}}\right), \tag{2.42}$$

$$\mathcal{W}_2(a,b) = \int_{-\infty}^{\infty} x^2\Phi(a+bx)\phi(x)dx = \Phi\left(\frac{a}{\sqrt{b^2+1}}\right) - \frac{ab^2}{\sqrt{(b^2+1)^3}}\phi\left(\frac{a}{\sqrt{b^2+1}}\right), \tag{2.43}$$

$$\mathcal{W}_3(a,b) = \int_{-\infty}^{\infty} x^3\Phi(a+bx)\phi(x)dx = \frac{|b|\left(2b^4+(a^2+5)b^2+3\right)}{\sqrt{(b^2+1)^5}} \times\phi\left(\frac{a}{\sqrt{b^2+1}}\right), \tag{2.44}$$

$$\mathcal{W}_4(a,b) = \int_{-\infty}^{\infty} x^4\Phi(a+bx)\phi(x)dx = 3\Phi\left(\frac{a}{\sqrt{b^2+1}}\right) - \frac{ab^2\left(3b^4+(a^2+9)b^2+6\right)}{\sqrt{(b^2+1)^7}}\phi\left(\frac{a}{\sqrt{b^2+1}}\right), \tag{2.45}$$

$$\mathcal{W}_5(a,b) = \int_{-\infty}^{\infty} x^5\Phi(a+bx)\phi(x)dx = (b^2+1)^{-9/2}\Big(15+10(5+a^2)b^2 +(63+14a^2+a^4)b^4+4(9+a^2)b^6+8b^8\Big)\phi\left(\frac{a}{\sqrt{b^2+1}}\right), \tag{2.46}$$

$$\mathcal{W}_6(a,b) = \int_{-\infty}^{\infty} x^6\Phi(a+bx)\phi(x)dx = 15\Phi\left(\frac{a}{\sqrt{b^2+1}}\right) - (b^2+1)^{-11/2} \times\Big\{ab^2\Big(45+15(9+a^2)b^2+(150+20a^2+a^4)b^4+5(15+a^2)b^6 +15b^8\Big)\Big\}\phi\left(\frac{a}{\sqrt{b^2+1}}\right). \tag{2.47}$$

2.7.2.1 Proof of Lemma 1

We next provide the derivation of $\mathcal{W}_s(a,b)$ where $s=3$. $s \in \{0,1,2,4,5,6\}$ are solved analogously and hence their working out is omitted. For the integral

$$\mathcal{W}_s(a,b) = \int_{-\infty}^{\infty} x^3\Phi(a+bx)\phi(x)dx,$$

consider the following conjecture:

$$\phi^{(3)}(x) \equiv (3x-x^3)\phi(x). \tag{2.48}$$

Proof: First note the derivatives

$$\begin{aligned}\phi'(x) &= -x\phi(x),\\ \phi''(x) &= (x^2-1)\phi(x),\\ \phi'''(x) &= (3x-x^3)\phi(x).\end{aligned}$$

Using simple algebraic manipulations

$$\begin{aligned}x^3\phi(x) &= -(3x-x^3)\phi(x)+3x\phi(x)\\ &= -\phi'''(x)-3\phi'(x).\end{aligned}$$

Thus

$$\begin{aligned}\int_{-\infty}^{\infty} x^3\Phi(a+bx)\phi(x)dx &= \int_{-\infty}^{\infty}\Phi(a+bx)\phi'''(x)dx-3\int_{-\infty}^{\infty}\Phi(a+bx)\phi'(x)dx\\ &= -T_A-3T_B, \qquad (2.49)\end{aligned}$$

where

$$T_A \equiv \int_{\mathbb{R}}\phi'''(x)\Phi(a+bx)dx \quad \text{and} \quad T_B \equiv \int_{\mathbb{R}}\phi'(x)\Phi(a+bx)dx.$$

Now

$$\begin{aligned}T_A &= \int_{\mathbb{R}}\phi'''(x)\Phi(a+bx)dx\\ &= -b\int_{\mathbb{R}}\phi''(x)\phi(a+bx)dx\\ &= -\int_{\infty}^{\infty}\phi^{(2)}(x)\frac{1}{\frac{1}{b}}\phi^{(0)}\left(\frac{x-(-a/b)}{(1/b)}\right)dx\\ &= -\int_{\infty}^{\infty}\phi^{(2)}(x)\phi^{(0)}_{\frac{1}{b}}\big(x-(-a/b)\big)dx.\end{aligned}$$

Then by C.1.12 of Wand & Jones (1995)[67] it can be written

$$\begin{aligned}&= -(-1)^2\phi^{(2)}_{\sqrt{1+\frac{1}{b^2}}}\big(-(-a/b)\big)\\ &= -\left(\frac{b^2}{1+b^2}\right)^{3/2}\phi^{(2)}\left(\frac{a}{\sqrt{1+b^2}}\right).\end{aligned}$$

Using the form of $\phi^{(2)}$ where x is given in the proof of the conjecture

$$\begin{aligned}&= -\frac{|b|^3}{\sqrt{(1+b^2)^3}}\left(\frac{a^2}{1+b^2}-1\right)\phi\left(\frac{a}{\sqrt{(1+b^2)}}\right)\\ &= -\frac{|b|^3}{\sqrt{(1+b^2)^3}}\left(\frac{a^2-(1+b^2)}{\sqrt{(1+b^2)^2}}-1\right)\phi\left(\frac{a}{\sqrt{(1+b^2)}}\right)\\ &= -\frac{|b|^3(a^2-b^2-1)}{\sqrt{(1+b^2)^5}}\phi\left(\frac{a}{\sqrt{(1+b^2)}}\right).\end{aligned}$$

The T_B part is calculated analogously to the first

$$\begin{aligned} T_B &= \int_{-\infty}^{\infty} \phi'(x)\Phi(a+bx)dx \\ &= -b\int_{-\infty}^{\infty} \phi^{(0)}(x)\phi^{(0)}(a+bx)dx \\ &= -\int_{-\infty}^{\infty} \phi^{(0)}(x)\phi^{(0)}_{\frac{1}{b}}\big(x-(-a/b)\big)dx \\ &= -\left(1+\frac{1}{b^2}\right)^{-1/2}\phi\left(\frac{a}{\sqrt{(b^2+1)}}\right) \\ &= -\frac{|b|}{\sqrt{1+b^2}}\phi\left(\frac{a}{\sqrt{b^2+1}}\right). \end{aligned}$$

Substituting T_A and T_B into equation (2.49) leads directly to the required result. Two interesting patterns worth noting occur:

- The cumulative density function Φ occurs only for even values of s.
- The algebraic expression of the integral (equation (2.49) for the $s = 3$ case) takes a negative value when s is odd and positive when s is even.

Each moment of the target density shown in equation (2.5) can be obtained as

$$\mathcal{M}_k \equiv \int_{-\infty}^{\infty} x^k\Phi(c_0+c_1x)(2\pi)^{-1/2}\exp\left\{\begin{bmatrix}x\\x^2\end{bmatrix}^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\right\}dx\exp\big(A(\boldsymbol{\eta})\big)(2\pi)^{1/2}.$$

Taking the inverse of natural parameters as per equation 1.5 yields,

$$\mathcal{M}_k = \int_{-\infty}^{\infty} x^k\Phi(c_0+c_1x)\ (2\pi)^{-1/2}\exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}dx\exp\big(A(\boldsymbol{\eta})\big)(2\pi)^{1/2}\sigma.$$

Let $u = \frac{x-\mu}{\sigma}$, then $x = \mu+\sigma u$ and $dx = \sigma du$, then

$$\mathcal{M}_k = \int_{-\infty}^{\infty} (\mu+\sigma u)^k\ \Phi(c_0+c_1\mu+c_1\sigma u)\ \phi(u)du\ Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta}) + 1/2\log(2\pi) + \log(\sigma)\big)$. Using Lemma 1, it is easy to find each of the kth moments

$$\begin{aligned} Z_1^{-1}\mathcal{M}_0 &= \Phi(r_2), \\ Z_1^{-1}\mathcal{M}_1 &= \mu\Phi(r_2) + (2c_1\sigma^2r_1^{-1})\phi(r_2), \\ Z_1^{-1}\mathcal{M}_2 &= \mu^2\Phi(r_2) + (4c_1\sigma^2\mu r_1^{-1})\phi(r_2) + \sigma^2\Phi(r_2) - (4r_2c_1^2\sigma^4r_1^{-2})\phi(r_2), \end{aligned}$$

for $r_2 = 2(c_0+c_1\mu)r_1^{-1}$ and $r_1 = 2\sqrt{(c_1\sigma)^2+1}$. The optimal mean and variance parameters follow respectively,

$$\mu^* = E(x) = \frac{\mathcal{M}_1}{\mathcal{M}_0} = \mu + (2c_1\sigma^2r_1^{-1})\zeta'(r_2)$$

and

$$\begin{aligned}(\sigma^*)^2 = E(x^2) - E(x)^2 &= \frac{\mathcal{M}_2}{\mathcal{M}_0} - \left(\frac{\mathcal{M}_1}{\mathcal{M}_0}\right)^2 \\ &= \sigma^2 - (4c_1^2\sigma^4 r_1^{-2})\big(r_2 - \zeta'(r_2)\big)\zeta'(r_2).\end{aligned}$$

By converting to natural parameters and using matrix notation we arrive at the required result. Stable computation of $\zeta'(r_2)$ and $\zeta''(r_2)$ in R is provided by the function `zeta()` in the package "sn" (Azzalini, 2016[1]), where the derivatives are controlled by the argument "k".

2.7.3 Proof of Result 7

We next provide the derivation of $E(x^s)$ for $s = 2$. Note $s \in \{0, 1, 3, 4, 5, 6\}$ are solved analogously and hence their working out is omitted. First note, for each s the integral

$$\int_{-\infty}^{\infty} x^s \phi(x) dx$$

takes the following values:

$$\int_{-\infty}^{\infty} x^0 \phi(x) dx = 1, \quad \int_{-\infty}^{\infty} x^1 \phi(x) dx = 0, \quad \int_{-\infty}^{\infty} x^2 \phi(x) dx = 1,$$

$$\int_{-\infty}^{\infty} x^3 \phi(x) dx = 0, \quad \int_{-\infty}^{\infty} x^4 \phi(x) dx = 3, \quad \int_{-\infty}^{\infty} x^5 \phi(x) dx = 0,$$

$$\int_{-\infty}^{\infty} x^6 \phi(x) dx = 15.$$

With this in mind consider

$$\begin{aligned}E(u^2) &= \int_{-\infty}^{\infty} u^2 \exp\big(T(u)^\top \eta - A(\eta)\big)(2\pi)^{-1/2} du \\ &= \int_{-\infty}^{\infty} u^2 \frac{1}{\sigma}\phi\Big(\frac{u-\mu}{\sigma}\Big) du.\end{aligned}$$

Then employing the change of variables $z = \frac{u-\mu}{\sigma} \implies u = \mu + \sigma z$, it can be written

$$\begin{aligned}E(u^2) &= \int_{-\infty}^{\infty} (\mu + \sigma z)^2 \phi(z) dz \\ &= \mu^2 \int_{-\infty}^{\infty} \phi(z) dz + 2\mu\sigma \int_{-\infty}^{\infty} z\phi(z) dz + \sigma^2 \int_{-\infty}^{\infty} z^2 \phi(z) dz.\end{aligned}$$

Noting $\int_{-\infty}^{\infty} z\phi(z) dz$ integrates to zero while $\int_{-\infty}^{\infty} \phi(z) dz$ and $\int_{-\infty}^{\infty} z^2 \phi(z) dz$ integrates to 1 leads to the required result.

2.7.4 Details on finding the inverse function gradient map log-partition function for each s

The following section of appendix aims to more explicitly explain the function $(\nabla A)^{-1}(\boldsymbol{\tau}, s)$ and the calculations it involves for each s. We only provide a full explanation for the $s = 0$ case since the other cases are analogous.

2.7.4.1 The $s = 0$ case

Firstly note that

$$E(x^0) = 1.$$

Following from the previous section, $E(x^0)$ can be used to normalise the approximation of the input density

$$f_{\text{UN}}(x) = (2\pi\sigma^2)^{-1/2} \exp\Big(-(x-\mu)^2/(2\sigma^2)\Big).$$

Next note that using the standard change of variables and natural parameters

$$\begin{aligned} f_{\text{UN}}(x) &= (2\pi)^{-1/2} \exp\left\{ \begin{bmatrix} x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}_{-1} - \Big(-\eta_1^2/(4\eta_2) - 1/2\log(-2\eta_2)\Big) \right\} \\ &= \exp\Big(\boldsymbol{T}(x)^\top \boldsymbol{\eta}_{-1} - A(\boldsymbol{\eta}_{-1})\Big) h(x), \end{aligned}$$

where

$$h(x) \equiv (2\pi)^{-1/2}, \qquad A(\boldsymbol{\eta}_{-1}) = -\eta_1^2/(4\eta_2) - 1/2\log(-2\eta_2).$$

The derivative vector of the log-partition function can be shown to be

$$(\nabla A)(\boldsymbol{\eta}_{-1}) = \begin{bmatrix} -\frac{\eta_1}{2\eta_2} \\ \big(\frac{\eta_1}{2\eta_2}\big)^2 - \frac{1}{2\eta_2} \end{bmatrix}.$$

We can now use the derivative vector of the log-partition to find an estimate of the minimum sum of squares deviation from the root of the inverse-defining equation using a Nelder-Mead search implemented through the `optim()` function in R. We then obtain the Hessian of the derivative vector of the log-partition function

$$\mathsf{H}\big(A(\boldsymbol{\eta}_{-1})\big) = \begin{bmatrix} -\frac{1}{2\eta_2} & \frac{\eta_1}{2\eta_2^2} \\ \frac{\eta_1}{2\eta_2^2} & \frac{\eta_2 - \eta_1^2}{2\eta_2^3} \end{bmatrix}$$

such that we can implement a Newton Raphson search to further refine the approximation of $\boldsymbol{\eta}^*$ using the minimum calculated through our Nelder-Mead search as a starting value.

2.7.4.2 The $s = 2$ case

Following calculations analogous to the $s = 0$ case it is easy to show:

$$(\nabla A)(\boldsymbol{\eta}_{-1}) \quad = \quad \begin{bmatrix} \frac{\eta_1(6\eta_2-\eta_1^2)}{2\eta_2(\eta_1^2-2\eta_2)} \\ \frac{12\eta_1^2\eta_2-12\eta_2^2-\eta_1^4}{4\eta_2^2(2\eta_2-\eta_1^2)} \end{bmatrix}$$

and

$$\mathsf{H}\left(A(\boldsymbol{\eta}_{-1})\right) = \begin{bmatrix} \frac{-\eta_1^4-12\eta_2^2}{2\eta_2(\eta_1^2-2\eta_2)^2} & \frac{\eta_1(12\eta_2^2-4\eta_1^2\eta_2+\eta_1^4)}{2\eta_2^2(2\eta_2-\eta_1^2)^2}, \\ \frac{\eta_1(12\eta_2^2-4\eta_1^2\eta_2+\eta_1^4)}{2\eta_2^2(2\eta_2-\eta_1^2)^2} & \frac{12\eta_2^3-24\eta_1^2\eta_2^2+9\eta_1^4\eta_2-\eta_1^6}{2\eta_2^3(2\eta_2-\eta_1^2)^2} \end{bmatrix}.$$

2.7.4.3 The $s = 4$ case

As in the first derivative case, the calculations required are analogous to the $s = 0$ case, and so

$$(\nabla A)(\boldsymbol{\eta}_{-1}) \quad = \quad \begin{bmatrix} \frac{4\eta_1^3-24\eta_2\eta_1}{12\eta_2^2-12\eta_2\eta_1^2+\eta_1^4} - \frac{\eta_1}{2\eta_2} \\ \frac{180\eta_1^2\eta_2^2-120\eta_2^3-30\eta_1^4\eta_2+\eta_1^6}{4\eta_2^2(12\eta_2^2-12\eta_1^2\eta_2+\eta_1^4)} \end{bmatrix}$$

and

$$\mathsf{H}(A(\boldsymbol{\eta}_{-1})) =$$
$$\begin{bmatrix} \frac{16\eta_2\eta_1^6-\eta_1^8-120\eta_2^2\eta_1^4-720\eta_2^4}{2\eta_2(12\eta_2^2-12\eta_1^2\eta_2+\eta_1^4)^2} & \frac{\eta_1(720\eta_2^4-480\eta_1^2\eta_2^3+216\eta_1^4\eta_2^2-24\eta_1^6\eta_2+\eta_1^8)}{2\eta_2^2(12\eta_2^2-12\eta_1^2\eta_2+\eta_1^4)^2} \\ \frac{\eta_1(720\eta_2^4-480\eta_1^2\eta_2^3+216\eta_1^4\eta_2^2-24\eta_1^6\eta_2+\eta_1^8)}{2\eta_2^2(12\eta_2^2-12\eta_1^2\eta_2+\eta_1^4)^2} & \frac{720\eta_2^5-2160\eta_1^2\eta_2^4+1560\eta_1^4\eta_2^3-384\eta_1^6\eta_2^2+33\eta_1^8\eta_2-\eta_1^{10}}{2\eta_2^3(12\eta_2^2-12\eta_1^2\eta_2+\eta_1^4)^2} \end{bmatrix}$$

Chapter 3

Expectation propagation for general one level probit mixed models

With knowledge from Chapter 2 on the random intercepts only model, we now extend our model to the more general case of GLMMs discussed in Section 1.7 that allow for any number of fixed and random effects. We aim to find an approximation to the maximum likelihood of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ with 95% confidence intervals. Our model is specified as in equation (3.1)

$$y_{ij}|\boldsymbol{u}_i \overset{\text{ind.}}{\sim} \text{Bernoulli}\left(\Phi\left(\boldsymbol{\beta}^\top \boldsymbol{x}^{\mathbf{F}}_{ij} + \boldsymbol{u}_i^\top \boldsymbol{x}^{\mathbf{R}}_{ij}\right)\right), \quad \boldsymbol{u}_i \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}),$$
$$1 \leq i \leq m, \quad 1 \leq j \leq n_i, \tag{3.1}$$

where the notation follows from the general one level model presented in Section 1.8. The log-likelihood can be expressed as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{i=1}^{m} \ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}),$$

where

$$\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \Phi\Big((2y_{ij} - 1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}^{\mathbf{F}}_{ij} + \boldsymbol{u}_i^\top \boldsymbol{x}^{\mathbf{R}}_{ij}\big)\Big) \right\} \tag{3.2}$$
$$\times\, |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \right) d\boldsymbol{u}_i$$

The content of this chapter is published as: Hall, P., Johnstone, I. M., Ormerod, J. T., Wand, M. P., & Yu, J. C. F. (2020). Fast and Accurate Binary Response Mixed Model Analysis via Expectation Propagation, Journal of the American Statistical Association, 115:532, 1902–1916, DOI: 10.1080/01621459.2019.1665529

and the best predictor of $\boldsymbol{u}_i$ is

$$\mathrm{BP}(\boldsymbol{u}_i) \equiv \frac{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}\left\{\prod_{j=1}^{n_i} \Phi\Big((2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big)\right\} \exp\big(-\tfrac{1}{2}\boldsymbol{u}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{u}\big) d\boldsymbol{u}}{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{\prod_{j=1}^{n_i} \Phi\Big((2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big)\right\} \exp\big(-\tfrac{1}{2}\boldsymbol{u}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{u}\big) d\boldsymbol{u}}. \quad (3.3)$$

We are now left with the more complex problem of intractable $d^{\mathbf{R}}$-dimensional integrals. As mentioned in Section 2.1, although Gauss-Hermite quadrature can be used to obtain high accuracy results for models with one random effect (i.e $d^{\mathbf{R}} = 1$), it is often not feasible computationally for the case where more than two or three random effects exist. It is in this case that the benefits of approximate inference, specifically via EP, become more obvious. Additionally, the maximisation problem of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ is complicated over a changing dimensional space, so we utilise the quasi-Newton approach discussed in the Chapter 2.

The structure of this chapter follows that of the previous, first explaining the schematic of likelihood approximation using EP in Section 3.1, before discussing computation of point estimates and confidence intervals in Section 3.2. The calculation of best predictors for the random effects is given in Section 3.3 before the results of our simulation studies are presented in Section 3.4.

3.1 Expectation propagation likelihood approximation

We now show how EP can be used to approximate the likelihood by updating and summing the natural parameter updates. Details of the required projections and the message passing formulation to compartmentalise the algebra are provided. As in the univariate case, we can approximate each $\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ and sum them to obtain the full log-likelihood. EP is motivated by the selection of an unnormalised multivariate normal density function that minimises the KL divergence criterion to replace each

$$\Phi\Big((2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big), \quad 1 \leq j \leq n_i$$

in equation (3.2). Doing so for the probit case constrains the integrand to be a product of multivariate normal density functions with closed form solutions.

Consider the full KL divergence presented in equation (1.19) and the following family of unnormalised multivariate normal density functions written in exponential family

form

$$f_{\text{UN}}(\boldsymbol{x}) = \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \begin{bmatrix} \eta_0 \\ \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} \right\}, \tag{3.4}$$

with natural parameters $\eta_0 \in \mathbb{R}$, $\boldsymbol{\eta}_1$ (a $d \times 1$ vector) and $\boldsymbol{\eta}_2$ (a $\frac{1}{2}d(d+1)$ vector). In the higher dimensional general case of GLMM, the goal of the EP problem is to find the optimal natural parameters (denoted by $\eta_0^*, \boldsymbol{\eta}_1^*$ and $\boldsymbol{\eta}_2^*$) which minimise $\text{KL}(f_{\text{input}} \parallel f_{\text{UN}})$, where $f_{\text{input}} \in L_1(\mathbb{R}^d)$. This solution is referred to as the KL projection onto the family of multivariate normal density functions and is written as

$$\text{proj}_{\boldsymbol{N}}[f_{\text{input}}](\boldsymbol{x}) = \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta}^* \right\}, \tag{3.5}$$

where

$$\boldsymbol{\eta}^* \equiv \begin{bmatrix} \eta_0^* \\ \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}$$

and

$$(\eta_0^*, \boldsymbol{\eta}_1^*, \boldsymbol{\eta}_2^*) = \underset{(\eta_0, \boldsymbol{\eta}_1, \boldsymbol{\eta}_2) \in H}{\text{argmin}} \ \text{KL}(f_{\text{input}} \parallel f_{\text{UN}}).$$

In the special case of KL projection onto the unnormalised multivariate normal family, this problem simplifies further to moment-matching, where $(\eta_0^*, \boldsymbol{\eta}_1^*, \boldsymbol{\eta}_2^*)$ is the unique vector that matches the zeroth, first and second order moments of f_{UN} and f_{input}.

For probit binary GLMMs, EP requires repeated projection of

$$f_{\text{input}}(\boldsymbol{x}) = \Phi(c_0 + \mathbf{c_1}^\top \boldsymbol{x}) \exp\left((\boldsymbol{\eta}_1^{\text{input}})^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2^{\text{input}} \boldsymbol{x} \right) \tag{3.6}$$

onto an unnormalised multivariate normal distribution, $c_0 = (2y_{ij} - 1)\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}}$, $\boldsymbol{c}_1 =$

$(2y_{ij}-1)\boldsymbol{x}_{ij}^{\mathbf{R}}$ and $\boldsymbol{x}=\boldsymbol{u}_i$. With this in mind, we seek $\boldsymbol{\eta}^*$ such that

$$\int_{\mathbb{R}^d}\boldsymbol{x}^{\otimes k}\Phi(c_0+\boldsymbol{c}_1^\top\boldsymbol{x})\exp\left\{\begin{bmatrix}\boldsymbol{x}\\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top)\end{bmatrix}^\top\boldsymbol{\eta}^{\text{input}}\right\}d\boldsymbol{x}$$

$$=\int_{\mathbb{R}^d}\boldsymbol{x}^{\otimes k}\exp\left\{\begin{bmatrix}1\\ \boldsymbol{x}\\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top)\end{bmatrix}^\top\boldsymbol{\eta}^*\right\}d\boldsymbol{x}, \quad (3.7)$$

where $k\in\{0,1,2\}$.

Now consider the multivariate extension of Result 3.

Result 12. *For an unnormalised input function $f\in L_1(\mathbb{R}^d)$ such that $f\geq 0$ for all $x\in\mathbb{R}$ where $C_f\equiv\int_{\mathbb{R}^d}f(\boldsymbol{x})d\boldsymbol{x}$, the projection onto the unnormalised multivariate normal family is*

$$proj_{UN}[f](\boldsymbol{x})=C_f proj_{\boldsymbol{N}}[f/C_f](\boldsymbol{x}),$$

where $proj_{\boldsymbol{N}}$ is the projection onto the multivariate normal family.

Obtaining the natural parameters $\boldsymbol{\eta}^*$ for projection onto the unnormalised multivariate normal family follows from obtaining the projection onto the multivariate normal family. More explicitly, the optimal natural parameters $\boldsymbol{\eta}_1^*$ and $\boldsymbol{\eta}_2^*$, are given according to the projection of the normalised function $f_{\text{input}}/C_{f_{\text{input}}}$ onto the multivariate normal family. We can subsequently use these optimal natural parameters to find the normalising natural parameter $\boldsymbol{\eta}_0^*$ via Result 13 and thus obtain the projection onto unnormalised normal family.

Result 13. *When f_{input} density follows the form of equation (2.5), η_0^* is given by*

$$\eta_0^*=\log(C_{f_{input}})-A(\boldsymbol{\eta}_1^*,\boldsymbol{\eta}_2^*)-\frac{d}{2}\log(2\pi),$$

where the log-partition function is as defined in Section 1.5.2.2.

Thus to obtain the required projection, we first obtain the optimal natural parameters $\boldsymbol{\eta}_1^*$ and $\boldsymbol{\eta}_2^*$ to project onto the multivariate normal family as is presented in Result 14.

Result 14. *Given f_{input} follows the form of equation (3.6), the projection onto the multivariate normal family is given by*

$$proj_{\boldsymbol{N}}[f_{input}] = \exp\left(\boldsymbol{T}(\boldsymbol{x})^\top \boldsymbol{\eta}^*_{-1} - A(\boldsymbol{\eta}^*_{-1})\right) h(\boldsymbol{x}),$$

where

$$\boldsymbol{\eta}^*_{-1} = K_{probit}(\boldsymbol{\eta}^{input}_{-1}; c_0, \boldsymbol{c}_1), \qquad \boldsymbol{\eta}^{input}_{-1} \equiv \begin{bmatrix} \boldsymbol{\eta}^{input}_1 \\ \boldsymbol{\eta}^{input}_2 \end{bmatrix}, \qquad \boldsymbol{\eta}^*_{-1} \equiv \begin{bmatrix} \boldsymbol{\eta}^*_1 \\ \boldsymbol{\eta}^*_2 \end{bmatrix},$$

$K_{probit}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1\right)$ *is defined in Definition 13 and $\boldsymbol{T}(\boldsymbol{x})$ and $h(\boldsymbol{x})$ follow from Section 1.5.2.2.*

Definition 13. *For primary arguments $\boldsymbol{a}_1$ $(d \times 1)$ and $\boldsymbol{a}_2$ $\left(\frac{1}{2}d(d+1) \times 1\right)$ such that $vec^{-1}(-\boldsymbol{D}_d^{+\top}\boldsymbol{a}_2)$ is symmetric and positive definite, and auxiliary arguments $c_0 \in \mathbb{R}$ and $\boldsymbol{c}_1$ $(d \times 1)$, the function $K_{probit} : H \to H$ is given by*

$$K_{probit}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1\right) \equiv \begin{bmatrix} \mathbf{R}_5^\top(\boldsymbol{a}_1 + r_3\boldsymbol{c}_1) \\ \boldsymbol{D}_d^\top vec(\mathbf{R}_5^\top \mathbf{A}_2) \end{bmatrix}, \tag{3.8}$$

where

$$\boldsymbol{A}_2 \equiv vec^{-1}(\boldsymbol{D}_d^{+\top}\boldsymbol{a}_2), \quad r_1 \equiv \sqrt{2(2 - \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{c}_1)}, \quad r_2 \equiv (2c_0 - \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1)r_1^{-1}$$
$$r_3 \equiv 2\zeta'(r_2)r_1^{-1}, \quad r_4 \equiv -2\zeta''(r_2)r_1^{-2} \quad \text{and} \quad \mathbf{R}_5 \equiv (\boldsymbol{A}_2 + r_4\boldsymbol{c}_1\boldsymbol{c}_1^\top)^{-1}\mathbf{A}_2.$$

The proof of Definition 13 is given in Appendix 3.5.1. Using Result 14 we now obtain the normalising natural parameter η_0^* to find the projection onto unnormalised normal family.

3.1.1 Projection onto the unnormalised multivariate normal family

Recall the moment matching problem described by equation (3.7) and Result 13. Then, as in the univariate case we require

$$C_{f_{\text{input}}} = \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x})d\boldsymbol{x} = (2\pi)^{d/2}\exp\big(A(\boldsymbol{\eta}^{\text{input}})\big)\Phi(r_2),$$

where r_2 is given in Definition (13) and $A(\boldsymbol{\eta})$ is defined in Section 1.5.2.2. Analogous to the argument given in the univariate case we then get

$$\boldsymbol{\eta}_0^* = \log\ \Phi(r_2)+\frac{1}{4}(\boldsymbol{\eta}_1^*)^\top(\boldsymbol{H}_2^*)^{-1}\boldsymbol{\eta}_1^*-\frac{1}{4}\big(\boldsymbol{\eta}_1^{\text{input}}\big)^\top\big(\boldsymbol{H}_2^{\text{input}}\big)^{-1}\boldsymbol{\eta}_1^{\text{input}}+\frac{1}{2}\log\big(|\boldsymbol{H}_2^*|/|\boldsymbol{H}_2^{\text{input}}|\big).$$

Definition 14. *Consider first, primary arguments $\boldsymbol{a}_1$ and $\boldsymbol{b}_1$ and auxiliary argument $\boldsymbol{c}_1$ where all three are $d\times 1$. Next consider arguments $\boldsymbol{a}_2$ and $\boldsymbol{b}_2$ which are all $\big(\frac{1}{2}d(d+1)\times 1\big)$ such that both $vec^{-1}\Big(-\big(\boldsymbol{D}_d^+\big)^\top\boldsymbol{a}_2\Big)$ and $vec^{-1}\Big(-\big(\boldsymbol{D}_d^+\big)^\top\boldsymbol{b}_2\Big)$ are symmetric and positive definite. Finally note auxiliary argument $c_0\in\mathbb{R}$. Then the function $C_{probit}: H\times H\to\mathbb{R}$ is given by*

$$C_{probit}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix},\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix};c_0,\boldsymbol{c}_1\right)\equiv\log\Phi(r_2)+\frac{1}{4}\boldsymbol{b}_1^\top\boldsymbol{B}_2^{-1}\boldsymbol{b}_1-\frac{1}{4}\boldsymbol{a}_1^\top\boldsymbol{A}_2^{-1}\boldsymbol{a}_1 + \frac{1}{2}\log\big(|\boldsymbol{B}_2|/|\boldsymbol{A}_2|\big),$$

where $\boldsymbol{A}_2\equiv vec^{-1}\Big(\big(\boldsymbol{D}_d^+\big)^\top\boldsymbol{a}_2\Big)$, $\boldsymbol{B}_2\equiv vec^{-1}\Big(\big(\boldsymbol{D}_d^+\big)^\top\boldsymbol{b}_2\Big)$ and r_2 follows from Definition 13.

In summary, the calculations required obtain the KL projection of the input function onto the unnormalised multivariate normal family are given by Result 15.

Result 15. *For an unnormalised input function of the form of equation (3.6) then*

$$proj_{UN}\left[f_{input}\right](x) = \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{x} \\ vech\left(\boldsymbol{x}\boldsymbol{x}^\top\right) \end{bmatrix}^\top \begin{bmatrix} \eta_0^* \\ \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}\right\},$$

where

$$\begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix} = K_{probit}\left(\begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}; c_0, \mathbf{c}_1\right) \quad and \quad \eta_0^* = C_{probit}\left(\begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}; c_0, \mathbf{c}_1\right).$$

We now show how this result can be used in a message passing framework to minimise the algebra required.

3.1.2 Message passing formulation

Note the components of the sum of the log-likelihood function $\ell_i\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}\right)$ can be written as

$$\ell_i\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}\right) \quad = \quad \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left(\prod_{j=1}^{n_i} p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta})\right) p(\boldsymbol{u}_i; \boldsymbol{\Sigma})\, d\boldsymbol{u}_i, \tag{3.9}$$

where

$$p(y_{ij}|\boldsymbol{u}_i, \boldsymbol{\beta}) \equiv \Phi\Big((2y_{ij} - 1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big)$$

and

$$p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) \equiv |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{u}_i\right)$$

are respectively the conditional density functions of each response given its random effect and the density function of that random effect. Also note the alternate matrix expression

$$p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) \equiv \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \boldsymbol{\eta}_{\boldsymbol{\Sigma}}\right\}, \quad \text{where} \quad \boldsymbol{\eta}_{\boldsymbol{\Sigma}} \equiv \begin{bmatrix} -\frac{1}{2}\log\left|2\pi\boldsymbol{\Sigma}\right| \\ 0_{d^{\mathbf{R}}} \\ -\frac{1}{2}\boldsymbol{D}_{d^{\mathbf{R}}}^\top \text{vec}(\boldsymbol{\Sigma}^{-1}) \end{bmatrix}. \tag{3.10}$$

The structure of the product of equation (3.9) can be represented as Figure 3.1, where circular stochastic nodes correspond to the random vector $\boldsymbol{u}_i$, solid squares indicate the $n_i + 1$ factor nodes and the dependencies of the factor nodes on the stochastic node $\boldsymbol{u}_i$ is demonstrated through the edges.

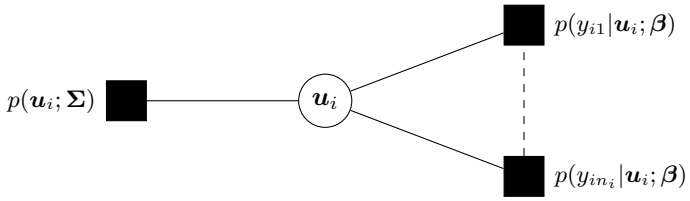


Figure 3.1: *Factor graph representation of the product structure of the integrand in equation (3.9).*

We now proceed in a manner analogous to Section 2.2.2 for the univariate case, using the Bayesian approach of Minka (2005).[44] The EP approximation of $\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ involves projection onto the unnormalised multivariate normal family. Suppose that

$$p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) = \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \boldsymbol{\eta}_{ij} \right\}, \quad 1 \leq j \leq n_i,$$

is initialised to be from the family of unnormalised multivariate normal density functions in $\boldsymbol{u}_i$. Then, for each $j = 1, ..., n_i$, the $\boldsymbol{\eta}_{ij}$ update involves minimisation of

$$\text{KL}\left(p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) \left(\prod_{j' \neq j}^{n_i} p(y_{ij'}|\boldsymbol{u}_i;\boldsymbol{\beta}) \right) p(\boldsymbol{u}_i;\boldsymbol{\Sigma}) \middle\| \left(\prod_{j'=1}^{n_i} p(y_{ij'}|\boldsymbol{u}_i;\boldsymbol{\beta}) \right) p(\boldsymbol{u}_i;\boldsymbol{\Sigma}) \right) \quad (3.11)$$

as a function of $\boldsymbol{u}_i$. This can be achieved using Result 15 to update $\boldsymbol{\eta}_{ij}$ in an iterative procedure until it converges.

Using message passing, we compartmentalise the otherwise cumbersome algebra via a simple extension of the steps involved in the univariate case. As such we do not repeat all of the steps again, instead providing the main key equations, which mirror equation (54) and (83) of Minka (2005).[44] Since derivation of the simplification required for the structure of the message passing scheme is a simple extension of the univariate case, we do not repeat it. The KL divergence of equation (3.11) can be re-expressed, such that

the messages from the factor $p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ to stochastic node $\boldsymbol{u}_i$ are updated according to

$$m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow \frac{\text{proj}_{\text{UN}}\left[m_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i)p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\right](\boldsymbol{u}_i)}{m_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i)}, \quad 1 \leq j \leq n_i \tag{3.12}$$

and the update of message passed from $p(\boldsymbol{u}_i;\boldsymbol{\Sigma})$ to $\boldsymbol{u}_i$ is

$$m_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow \frac{\text{proj}_{\text{UN}}\left[m_{\boldsymbol{u}_i\to p(\boldsymbol{u}_i;\boldsymbol{\Sigma})}(\boldsymbol{u}_i)p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\right](\boldsymbol{u}_i)}{m_{\boldsymbol{u}_i\to p(\boldsymbol{u}_i;\boldsymbol{\Sigma})}(\boldsymbol{u}_i)}. \tag{3.13}$$

Similarly, the updates of stochastic node to factor messages are

$$m_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i) = m_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i}(\boldsymbol{u}_i)\prod_{j'\neq j}^{n_i} m_{p(y_{ij'}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i), \quad 1 \leq j \leq n_i \tag{3.14}$$

and

$$m_{\boldsymbol{u}_i\to p(\boldsymbol{u}_i;\boldsymbol{\Sigma})}(\boldsymbol{u}_i) = \prod_{j=1}^{n_i} m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i). \tag{3.15}$$

We now seek any algebraic simplifications of the key messages, particularly use of natural parameters. Recall that $p(\boldsymbol{u}_i;\boldsymbol{\Sigma})$ can be written using natural parameters in the form of equation (3.10) and that the unnormalised normal density constraint is enforced on equations (3.12) and (3.14). Then

$$m_{\boldsymbol{u}_i\to p(\boldsymbol{u}_i;\boldsymbol{\Sigma})}(\boldsymbol{u}_i) = \exp\left\{\begin{bmatrix}1\\ \boldsymbol{u}_i\\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\end{bmatrix}^\top \boldsymbol{\eta}_{\boldsymbol{u}_i\to p(\boldsymbol{u}_i;\boldsymbol{\Sigma})}\right\}. \tag{3.16}$$

Substituting the above forms into equation (3.13) leads to

$$m_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow p(\boldsymbol{u}_i;\boldsymbol{\Sigma}) = \exp\left\{\begin{bmatrix}1\\ \boldsymbol{u}_i\\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i)^\top\end{bmatrix}^\top \boldsymbol{\eta}_{\boldsymbol{\Sigma}}\right\}.$$

This implies the message $m_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i}(\boldsymbol{u}_i)$ is constant throughout the message passing updates. As such, we now set

$$\boldsymbol{\eta}_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i} \longleftarrow \boldsymbol{\eta}_{\boldsymbol{\Sigma}}. \tag{3.17}$$

For convenience, we denote the natural parameter vector

$$\boldsymbol{\eta}^{\otimes} \equiv \boldsymbol{\eta}_{\boldsymbol{u}_i \to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}.$$

Recall that from equation (3.16)

$$\begin{aligned} m_{\boldsymbol{u}_i \to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i) &= \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \boldsymbol{\eta}^{\otimes} \right\} \\ &= \exp(\eta_0^{\otimes}) \exp\left\{ \boldsymbol{u}_i^\top \boldsymbol{\eta}_1^{\otimes} + \left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top \boldsymbol{\eta}_2^{\otimes} \right\}. \end{aligned}$$

Substituting this into equation (3.12) and following simplifications analogous to the univariate case lead to

$$m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow \frac{\text{proj}_{\boldsymbol{UN}}\left[\Phi(c_0 + \boldsymbol{c}_{1_{ij}}^\top \boldsymbol{u}_i) \exp\left\{ \boldsymbol{u}_i^\top \boldsymbol{\eta}_1^{\otimes} + \left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top \boldsymbol{\eta}_2^{\otimes} \right\}\right]}{\exp\left\{ \boldsymbol{u}_i^\top \boldsymbol{\eta}_1^{\otimes} + \left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top \boldsymbol{\eta}_2^{\otimes} \right\}},$$

where $c_0 = (2y_{ij} - 1)\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{c}_{1_{ij}} \equiv (2y_{ij} - 1)\boldsymbol{x}_{ij}^{\mathbf{R}}$. Utilising Result 15,

$$m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i)^\top \end{bmatrix}^\top \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i} \right\}, \tag{3.18}$$

where the linear and quadratic coefficient updates are

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2} \longleftarrow K_{\text{probit}}\left(\boldsymbol{\eta}_{1:2}^{\otimes}; c_0, \boldsymbol{c}_{1_{ij}}\right) - \boldsymbol{\eta}_{1:2}^{\otimes} \tag{3.19}$$

and the constant coefficient update is

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_0 \longleftarrow C_{\text{probit}}\left(\boldsymbol{\eta}_{1:2}^{\otimes}, \left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2} + \boldsymbol{\eta}_{1:2}^{\otimes}; c_0, \boldsymbol{c}_{1_{ij}}\right).$$

Using the simplification of equation (3.12) and (3.13), equation (3.14) can be shown to be

$$m_{\boldsymbol{u}_i \to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i) \longleftarrow \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \boldsymbol{\eta}_{\boldsymbol{u}_i \to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})} \right\}, \tag{3.20}$$

where

$$\boldsymbol{\eta}_{\boldsymbol{u}_i \to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i} + \sum_{j'\neq j} \boldsymbol{\eta}_{p(y_{ij'}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}.$$

The EP approximation of each component $\underset{\sim}{\ell}_i(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ of the log-likelihood sum is given by

$$\underset{\sim}{\ell}_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \log \int_{\mathbb{R}^{d\mathbf{R}}} \left(\prod_{j=1}^{n_i} m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \right) m_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) d\boldsymbol{u}_i. \tag{3.21}$$

The success of EP depends on each of the messages in equation (3.21) being an unnormalised multivariate normal density function and that a closed from solution to the integral as follows:

$$\begin{aligned}
&\int_{\mathbb{R}^{d\mathbf{R}}} \left(\prod_{j=1}^{n_i} m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \right) m_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) d\boldsymbol{u}_i \\
&= \int_{\mathbb{R}^{d\mathbf{R}}} \prod_{j=1}^{n_i} \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i} \right\} \\
&\qquad \times \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \boldsymbol{\eta}_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i} \right\} d\boldsymbol{u}_i \\
&= (2\pi)^{d\mathbf{R}/2} \exp\left\{ \left(\boldsymbol{\eta}_{\boldsymbol{\Sigma}} + \sum_{j=1}^{n_i} \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i} \right)_0 + A\left(\left(\boldsymbol{\eta}_{\boldsymbol{\Sigma}} + \sum_{j=1}^{n_i} \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i} \right)_{1:2} \right) \right\}.
\end{aligned}$$

The full algorithm for the approximation of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ using EP is provided in Algorithm 6.

Algorithm 6 *Explicit form of algorithm used for the message passing approach to EP*

Inputs: y_{ij}, $\boldsymbol{x}_{ij}^{\mathbf{F}}$, $\boldsymbol{x}_{ij}^{\mathbf{R}}$, $1 \leq i \leq m$, $1 \leq j \leq n_i$; $\boldsymbol{\beta}(d^{\mathbf{F}} \times 1)$, $\boldsymbol{\Sigma}(d^{\mathbf{R}} \times d^{\mathbf{R}}$, are symmetric and positive definite).

Set constants: $c_{0_{ij}} \longleftarrow (2y_{ij} - 1)(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}})$, $c_{1_{ij}} \longleftarrow (2y_{ij} - 1)\boldsymbol{x}_{ij}^{\mathbf{R}}$, $1 \leq i \leq m$, $1 \leq j \leq n_i$.

$$\boldsymbol{\eta}_{p(\boldsymbol{u}_i;\boldsymbol{\Sigma})\to\boldsymbol{u}_i} \longleftarrow \boldsymbol{\eta}_{\boldsymbol{\Sigma}} \equiv \begin{bmatrix} -\frac{1}{2}\log|2\pi\boldsymbol{\Sigma}| \\ \mathbf{0}_{d^{\mathbf{R}}} \\ -\frac{1}{2}\boldsymbol{D}_{d^{\mathbf{R}}}^\top \mathrm{vec}(\boldsymbol{\Sigma}^{-1}) \end{bmatrix}, \quad 1 \leq i \leq m.$$

For $i = 1, ..., m$:

Initialise: $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$, $1 \leq j \leq n_i$ as per the equation (3.24).

Cycle:

$$\mathrm{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big) \longleftarrow \sum_{j=1}^{n_i} \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$$

For $j = 1, ..., n_i$:

$$\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{u}_i,\boldsymbol{\Sigma})\to\boldsymbol{u}_i} + \mathrm{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big) - \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$$

$$(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i})_{1:2} \longleftarrow K_{\mathrm{probit}}\Big(\big(\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}\big)_{1:2}; c_{0_{ij}}, c_{1_{ij}}\Big) - \big(\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}\big)_{1:2}$$

until convergence of all natural parameters vectors.

For $j = 1, ..., n_i$:

$$(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i})_0 \longleftarrow C_{\mathrm{probit}}\Big(\big(\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}\big)_{1:2}, \big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)_{1:2} + \big(\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}\big)_{1:2}; c_{0_{ij}}, c_{1_{ij}}\Big)$$

$$\mathrm{SUM}\big(\boldsymbol{\eta}_{p(\boldsymbol{y}_i|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big) \longleftarrow \sum_{j=1}^{n_i} \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}.$$

Output: The full approximate log-likelihood is given by

$$\underset{\sim}{\ell}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \frac{md^{\mathbf{R}}}{2}\log(2\pi) + \sum_{i=1}^{m}\Bigg\{\Big(\boldsymbol{\eta}_{\boldsymbol{\Sigma}} + \mathrm{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)\Big)_0 + A\Big\{\Big(\boldsymbol{\eta}_{\boldsymbol{\Sigma}} + \mathrm{SUM}\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)\Big)_{1:2}\Big\}\Bigg\}$$

where, $A(\boldsymbol{\eta})$ is defined as in equation (1.7) and $\boldsymbol{\eta}_{\boldsymbol{\Sigma}}$ follows from equation (3.10).

3.1.3 Starting values for Algorithm 6

The EP message passing algorithm proposed relies on good starting values for convergence. We now derive starting values for $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ using a Taylor series expansion. Note that

$$\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) = \zeta(a_{ij}) - \log(2), \quad \text{where} \quad a_{ij} \equiv (2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)$$

and ζ is defined as before. Let $\widehat{\boldsymbol{u}}_i$ be a Laplace approximation to $\boldsymbol{u}_i$. Now consider the following Taylor series expansion of the data dependent component of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$:

$$\begin{aligned}
\zeta(a_{ij}) &= \zeta\big(\widehat{a}_{ij} + (2y_{ij}-1)(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big) \\
&= \zeta(\widehat{a}_{ij}) + (2y_{ij}-1)\zeta'(\widehat{a}_{ij})(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}} + \frac{1}{2}\big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)^2 \zeta''(\widehat{a}_{ij}) + ... \\
&= \begin{bmatrix} 1 \\ \boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i \\ \operatorname{vech}\big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top\big) \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + ...,
\end{aligned}$$

where $\widehat{a}_{ij} \equiv (2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \widehat{\boldsymbol{u}}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)$ and

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} \zeta(\widehat{a}_{ij}) \\ (2y_{ij}-1)\zeta'(\widehat{a}_{ij})\boldsymbol{x}_{ij}^{\mathbf{R}} \\ \frac{1}{2}\zeta''(\widehat{a}_{ij})\boldsymbol{D}_{d^{\mathbf{R}}}^\top \operatorname{vec}\big(\boldsymbol{x}_{ij}^{\mathbf{R}}(\boldsymbol{x}_{ij}^{\mathbf{R}})^\top\big) \end{bmatrix}.$$

It follows that the quadratic approximation to $\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ based on Taylor expansion about $\widehat{\boldsymbol{u}}_i$ is $\log \breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ where

$$\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) \equiv \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i \\ \operatorname{vech}\big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top\big) \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} \right\}. \tag{3.22}$$

The starting value recommendation for $\eta_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ is based on replacement of $p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ by $\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ in equation (3.17):

$$m_{\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow \frac{\operatorname{proj}\big[m_{\boldsymbol{u}_i\to\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i)\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\big](\boldsymbol{u}_i)}{m_{\boldsymbol{u}_i\to\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})}(\boldsymbol{u}_i)} = \breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}). \tag{3.23}$$

Note that in this case, since $\breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ is already univariate normal the projection is superfluous. The starting values for $\eta_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ that arises from this substitution is

$$\exp\left\{\begin{bmatrix}1\\ \boldsymbol{u}_i\\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\end{bmatrix}^\top \eta^{\text{start}}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right\} = \exp\left\{\begin{bmatrix}1\\ \boldsymbol{u}_i-\widehat{\boldsymbol{u}}_i\\ \text{vech}\big((\boldsymbol{u}_i-\widehat{\boldsymbol{u}}_i)(\boldsymbol{u}_i-\widehat{\boldsymbol{u}}_i)^\top\big)\end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij}\right\}.$$

By matching coefficients of like terms it is easy to show

$$\eta^{\text{start}}_{p(y_{ij}|u_i)\to u_i} = \begin{bmatrix}\eta_0^{\text{start}}\\ (2y_{ij}-1)\zeta'(\widehat{a}_{ij})\boldsymbol{x}^{\mathbf{R}}_{ij} - \zeta''(\widehat{a}_{ij})\boldsymbol{x}^{\mathbf{R}}_{ij}(\boldsymbol{x}^{\mathbf{R}}_{ij})^\top\widehat{\boldsymbol{u}}_i\\ \frac{1}{2}\zeta''(\widehat{a}_{ij})\boldsymbol{D}_{d^{\mathbf{R}}}^\top\text{vec}\big(\boldsymbol{x}^{\mathbf{R}}_{ij}(\boldsymbol{x}^{\mathbf{R}}_{ij})^\top\big)\end{bmatrix}, \tag{3.24}$$

where

$$\eta_0^{\text{start}} = \zeta(\widehat{a}_{ij}) - (2y_{ij}-1)\zeta'(\widehat{a}_{ij})(\boldsymbol{x}^{\mathbf{R}}_{ij})^\top\widehat{\boldsymbol{u}}_i + \frac{1}{2}\zeta''(\widehat{a}_{ij})\big((\boldsymbol{x}^{\mathbf{R}}_{ij})^\top\widehat{\boldsymbol{u}}_i\big)^2.$$

In Algorithm 3 η_0^{start} is not used in the cycle loop and thus can be set to any arbitrary number without affecting the algorithm. We use Laplace approximation to estimate $\widehat{\boldsymbol{u}}_i$. For the R computing environment, the function `glmer()` of the package "lme4" (Bolker, et al., 2018[5]) provides fast Laplace approximation-based predictions for the $\boldsymbol{u}_i$.

3.2 Computation of point estimates and confidence intervals

We now address the computations required to find point estimates and confidence intervals for the parameters $\boldsymbol{\beta}, \boldsymbol{\Sigma}$. Although we still use the R function `optim()` in the "stats" package (R Core Team, 2019[56]) the higher dimensionality introduces new problems, particularly when optimising the variance parameter $\boldsymbol{\Sigma}$ since it is constrained to be symmetric and positive definite. Before conducting any optimisation, we must ensure the search occurs over the cone of symmetric positive definite $d^{\mathbf{R}} \times d^{\mathbf{R}}$ matrices, which can be accomplished by re-parameterising the $\boldsymbol{\Sigma}$ matrix (Bateman & Pinheiro, 2000[55]). For the general case where d-random effects are involved, the following procedure is recommended:

1. Before conducting any optimisation, convert $\boldsymbol{\Sigma}$ to the unconstrained space $\boldsymbol{\theta}$. When:

(a) $d = 1$:

$$\boldsymbol{\theta} \equiv \frac{1}{2}\log(\boldsymbol{\Sigma}).$$

(b) $d > 1$:

i. Obtain the spectral decomposition of $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \boldsymbol{u}_{\boldsymbol{\Sigma}}\boldsymbol{\lambda}_{\boldsymbol{\Sigma}}\, \boldsymbol{u}_{\boldsymbol{\Sigma}}^{\top},$$

where $\boldsymbol{\lambda}_{\boldsymbol{\Sigma}}$ is the diagonal matrix of eigenvalues and $\boldsymbol{u}_{\boldsymbol{\Sigma}}$ is the orthogonal matrix of matching eigenvectors.

ii. Calculate the matrix logarithm of $\boldsymbol{\Sigma}$ using the spectral decomposition,

$$\log(\boldsymbol{\Sigma}) = \boldsymbol{u}_{\boldsymbol{\Sigma}}\log(\boldsymbol{\lambda}_{\boldsymbol{\Sigma}})\boldsymbol{u}_{\boldsymbol{\Sigma}}^{\top}.$$

iii. Use the logarithm $\boldsymbol{\Sigma}$ to aquire $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} \equiv \text{vech}\left(\frac{1}{2}\log(\boldsymbol{\Sigma})\right).$$

2. Now use a quasi-Newton optimisation method to obtain the maximum likelihood approximation

$$(\widehat{\underset{\sim}{\boldsymbol{\beta}}}, \widehat{\underset{\sim}{\boldsymbol{\theta}}}) = \underset{\theta \in \mathbb{R}^{d(d+1)/2}}{\text{argmax}}\, \underset{\sim}{\ell}(\boldsymbol{\beta}, \boldsymbol{\theta}).$$

We suggest conducting an initial search via the Nelder-Mead method, with refinements by BFGS algorithm. Both can be implemented via the `optim()` R function in the "stats" package (R Core Team, 2019[56]).

3. Since it is well established that working with log(standard deviation) and $\tanh^{-1}$(correlation) parameters for confidence interval contruction results in better asymptotic normality than their non-transformed counterparts, convert $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}})$ to $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}})$. For:

(a) $d = 1$, no conversion is required since there is no correlaton parameter and the variance parameter is already log transformed. For consistancy with the multidimensional setting, note the following notation change:

$$\widehat{\boldsymbol{\omega}} = \widehat{\boldsymbol{\theta}}.$$

(b) $d > 1$:

i. Obtain the spectral decomposition of $\text{vech}^{-1}(\widehat{\boldsymbol{\theta}})$,

$$\text{vech}^{-1}(\widehat{\boldsymbol{\theta}}) = \boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}\, \boldsymbol{\lambda}_{\widehat{\boldsymbol{\theta}}}\, \boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}^{\top},$$

where $\boldsymbol{\lambda}_{\widehat{\boldsymbol{\theta}}}$ is the diagonal matrix of eigenvalues and $\boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}$ is the orthogonal matrix of matching eigenvectors.

ii. Then using the spectral decomposition find exponent of $\boldsymbol{\lambda}_{\widehat{\boldsymbol{\theta}}}$,

$$\widehat{\boldsymbol{\Sigma}} = \boldsymbol{u}_{\widehat{\boldsymbol{\theta}}} \ \ \exp(2\boldsymbol{\lambda}_{\widehat{\boldsymbol{\theta}}})\boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}^{\top}.$$

iii. Convert to $\widehat{\boldsymbol{\omega}}$

$$\widehat{\boldsymbol{\omega}} = \begin{bmatrix} \frac{1}{2}\log\big(\mathrm{diag}(\widehat{\boldsymbol{\Sigma}})\big) \\ \tanh^{-1}\Big\{\mathrm{vecbd}(\widehat{\boldsymbol{\Sigma}})/\sqrt{\mathrm{vecbd}\big(\mathrm{diag}(\widehat{\boldsymbol{\Sigma}})\mathrm{diag}(\widehat{\boldsymbol{\Sigma}})^{\top}\big)}\ \Big\} \end{bmatrix}.$$

4. Find the Hessian matrix at the maximum $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}})$ (denoted by $H\ell(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}})$) using the BFGS quasi-Newton method, which returns the hessian when implemented via the R function `optim()` in the "stats" package (R Core Team, 2019[56]). Although we need values of $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}})$ to be returned in the Hessian, the constraints on these parameters mean the Hessian should be calculated on the $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}})$ space. The conversion can be obtained as follows, for:

(a) $d = 1$:

$$\widehat{\boldsymbol{\theta}} = \exp(2\widehat{\boldsymbol{\omega}}).$$

(b) $d > 1$, form the $d \times d$ symmetric matrix $\widehat{\boldsymbol{\Sigma}}$:

i. Let $\widehat{\boldsymbol{\omega}}_1$ denote the first d entries of $\widehat{\boldsymbol{\omega}}$ and $\widehat{\boldsymbol{\omega}}_2$ denote the remaining $\frac{1}{2}d(d-1)$ entries of $\widehat{\boldsymbol{\omega}}$. Set $\mathrm{diag}(\widehat{\boldsymbol{\Sigma}}) = \exp(2\widehat{\boldsymbol{\omega}}_1)$. Obtain the below-diagonal entries of $\widehat{\boldsymbol{\Sigma}}$ so that

$$\mathrm{vecbd}(\widehat{\boldsymbol{\Sigma}}) = \tanh(\widehat{\boldsymbol{\omega}}_2) \odot \mathrm{vecbd}\big(\exp(\widehat{\boldsymbol{\omega}}_1)\exp(\widehat{\boldsymbol{\omega}}_1)^{\top}\big)$$

holds. Obtain the above-diagonal entries of $\widehat{\boldsymbol{\Sigma}}$, and reconstruct $\widehat{\boldsymbol{\Sigma}}$ by symmetry.

ii. Find the spectral decomposition of $\widehat{\boldsymbol{\Sigma}}$,

$$\widehat{\boldsymbol{\Sigma}} = \boldsymbol{u}_{\widehat{\boldsymbol{\Sigma}}}\mathrm{diag}(\boldsymbol{\lambda}_{\widehat{\boldsymbol{\Sigma}}})\boldsymbol{u}_{\widehat{\boldsymbol{\Sigma}}}^{\top}.$$

iii. Using the spectral decomposition make the conversion to $\boldsymbol{\theta}$,

$$\widehat{\boldsymbol{\theta}} = \mathrm{vech}\Big\{\frac{1}{2}\boldsymbol{u}_{\widehat{\boldsymbol{\Sigma}}}\mathrm{diag}\big(\log(\boldsymbol{\lambda}_{\widehat{\boldsymbol{\Sigma}}})\big)\boldsymbol{u}_{\widehat{\boldsymbol{\Sigma}}}^{\top}\Big\}.$$

5. Form $100(1-\alpha)\%$ confidence intervals for the entries of $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}})$ using

$$\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\omega}} \end{bmatrix} \pm \Phi^{-1}\left(1-\frac{1}{2}\alpha\right)\sqrt{-\text{diag}\Big\{\big(\boldsymbol{H}\ \ell(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}})\big)^{-1}\Big\}}.$$

6. Back transform the confidence interval limits for the $\widehat{\boldsymbol{\omega}}$ component, to correspond to the standard deviation and correlation parameters as follows:

$$\begin{bmatrix} \sqrt{\text{diag}(\widehat{\boldsymbol{\Sigma}})} \\ \text{vecbd}(\widehat{\boldsymbol{\Sigma}}) / \sqrt{\text{vecbd}\big(\text{diag}(\widehat{\boldsymbol{\Sigma}})\text{diag}(\widehat{\boldsymbol{\Sigma}})^\top\big)} \end{bmatrix}.$$

3.3 Best predictor

In addition to estimating $(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, we also wish to predict the random effects $\boldsymbol{u}_i$ for each group. For the binary mixed model via EP, the best predictor of $\boldsymbol{u}_i$ for the multivariate case is found in a similar manner to the univariate case,

$$\begin{aligned} \text{BP}(\boldsymbol{u}_i) &= E(\boldsymbol{u}_i|\boldsymbol{y}_i) \\ &= \int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}_i p(\boldsymbol{u}_i|\boldsymbol{y}_i; \boldsymbol{\beta}, \boldsymbol{\Sigma}) d\boldsymbol{u}_i \\ &= \int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}_i \left\{ \frac{p(\boldsymbol{y}_i|\boldsymbol{u}_i; \boldsymbol{\beta}) p(\boldsymbol{u}_i; \boldsymbol{\Sigma})}{\int_{\mathbb{R}^{d^{\mathbf{R}}}} p(\boldsymbol{y}_i|\boldsymbol{u}_i; \boldsymbol{\beta}) p(\boldsymbol{u}_i; \boldsymbol{\Sigma})} \right\} d\boldsymbol{u}_i, \end{aligned} \tag{3.25}$$

where $\boldsymbol{y}_i = (y_{i1}, ..., y_{in_i})$. We now show that by products of Algorithm 6 facilitate the empirical best predictions of the $\boldsymbol{u}_i$. Let

$$\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i \equiv \boldsymbol{\eta}_{\boldsymbol{\Sigma}} + \text{SUM}\big(\boldsymbol{\eta}_{p(\boldsymbol{y}_i|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big) = \begin{bmatrix} \widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i1} \\ \widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2} \end{bmatrix}, \tag{3.26}$$

where $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i1}$ corresponds to the first $d^{\mathbf{R}}$ entries of $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i$, $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2}$ corresponds to the remaining $d^{\mathbf{R}}$ entries of $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i$ and $\boldsymbol{\eta}_{\boldsymbol{\Sigma}}$ and $\text{SUM}\big(\boldsymbol{\eta}_{p(\boldsymbol{y}_i|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)$ are as previously defined in Algorithm 6 with $(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \big(\widehat{\underset{\sim}{\boldsymbol{\beta}}}, \widehat{\underset{\sim}{\boldsymbol{\Sigma}}}\big)$. Note that Algorithm 6 involves using

$$\exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top) \end{bmatrix}^\top \widehat{\underset{\sim}{\boldsymbol{\eta}}}_i \right\} \quad \text{to replace} \quad p(\boldsymbol{y}_i|\boldsymbol{u}_i; \boldsymbol{\beta}) p(\boldsymbol{u}_i; \boldsymbol{\Sigma}).$$

Subsequently as in the univariate case, we can approximate $\text{BP}(\boldsymbol{u}_i) = E(\widehat{\underset{\sim}{\boldsymbol{u}}}_i)$, where $\widehat{\underset{\sim}{\boldsymbol{u}}}_i$ is multivariate normal with natural parameter $\widehat{\underset{\sim}{\boldsymbol{\eta}}}_i$. Thus,

$$\text{BP}(\boldsymbol{u}_i) = -\frac{1}{2}\left(\text{vec}^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2}\right)\right)^{-1} \widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i1}.$$

Although $\text{Cov}\big(\text{BP}(\boldsymbol{u}_i) - \boldsymbol{u}_i\big)$ is well approximated with $E_{\boldsymbol{y}_i}\big(\text{Cov}(\boldsymbol{u}_i|\boldsymbol{y}_i)\big)$ (McCulloch, Searle & Neuhaus, 2008[41]), where

$$\underset{\sim}{\text{Cov}}(\boldsymbol{u}_i|\boldsymbol{y}) = -\frac{1}{2}\left(\text{vec}^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \widehat{\underset{\sim}{\boldsymbol{\eta}}}_{i2}\right)\right)^{-1},$$

the approximation is hindered by the expectation over the distribution over the $\boldsymbol{y}_i$ vector.

3.4 Simulation study

Three simulation studies were conducted, each for 1000 replicates on two computers:

- Computer 1 - MacBook Air laptop with two 2.2 gigahertz processors and 8 gigabytes of random access memory.
- Computer 2 - University of Technology Sydney Interactive High Performance Computing facility Jupiter node with eight 3.6 gigahertz processors and 32 gigabytes random access memory.

We will refer to these computers by name from here on. For each EP algorithm we set our error tolerance value to 10^{-5} and use a maximum of 100 Nelder-Mead search iterations during the optimisation process.

3.4.1 Comparison of maximum likelihood estimates for univariate random effects

The first simulation study was repeated on datasets simulated according to equation (3.1) with true parameter values,

$$\boldsymbol{\beta}_{\text{true}} = [0, 1]^\top \quad \text{and} \quad \boldsymbol{\Sigma}_{\text{true}} = \sigma^2_{\text{true}} = 1. \tag{3.27}$$

Each dataset had $m = 100$ groups and $n_i = 2$ observations in each group. The $\boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{x}_{ij}^{\mathbf{R}}$ vectors were respectively of the form

$$\boldsymbol{x}_{ij}^{\mathbf{F}} = [1, x_{1_{ij}}]^\top \quad \text{and} \quad \boldsymbol{x}_{ij}^{\mathbf{R}} = 1,$$

where $x_{1_{ij}}$ was generated independently from a uniform distribution on the unit interval. The model described was fit using each of the following methods:

1. Adaptive Gauss-Hermite quadrature using 100 points of evaluation, implemented via the function `glmer()` in the R package "lme4" (Bates, et al., 2018[5]).
2. Laplace approximation implemented via the function `glmer()` in the R package "lme4" (Bates, et al., 2018[5]).
3. EP as described in this Section.
4. Data cloning as used by the R package "dclone" (Solymos, 2010[61]) with 10 clones.

We first compare point estimates and confidence intervals produced by adaptive Gauss-Hermite quadrature against Laplace approximation, EP and data cloning. The first row of Figure 3.3 shows Laplace approximation results in poor statistical inference for all parameters estimated, with empirical coverage less than 95% level. The empirical coverage of the variance parameter by Laplace approximation is particularly low at 81.1%. The second row of Figure 3.3 shows EP produces very similar results to adaptive Gauss-Hermite quadrature for both the empirical coverage and confidence intervals of fixed effects parameters. It also shows EP is slightly conservative with coverage 97.5% for the variance parameter compared to 94.2% that of adaptive Gauss-Hermite quadrature. Data cloning produced similar empirical coverage values to EP (95.7%, 95.1% and 97.4% for β_0, β_1 and σ respectively). Out of the methods tested, data cloning and EP were the only methods with empirical coverage greater than or equal to 95% across all parameters.

Figure 3.2 compares estimated mean squared error and mean squared error of prediction for the four approaches, where the latter is given by the mean squared error of five randomly selected u_i random intercepts. We also include t-based 95% confidence intervals to provide an indication of the variability of simulation-based mean squared error estimation. The plots show EP performs well in comparison with adaptive Gauss-Hermite quadrature for maximum likelihood estimation and best prediction, and generally improves upon Laplace approximation and data cloning. Although we are aware that a comparison of time is obscured by factors such as language of implementation,

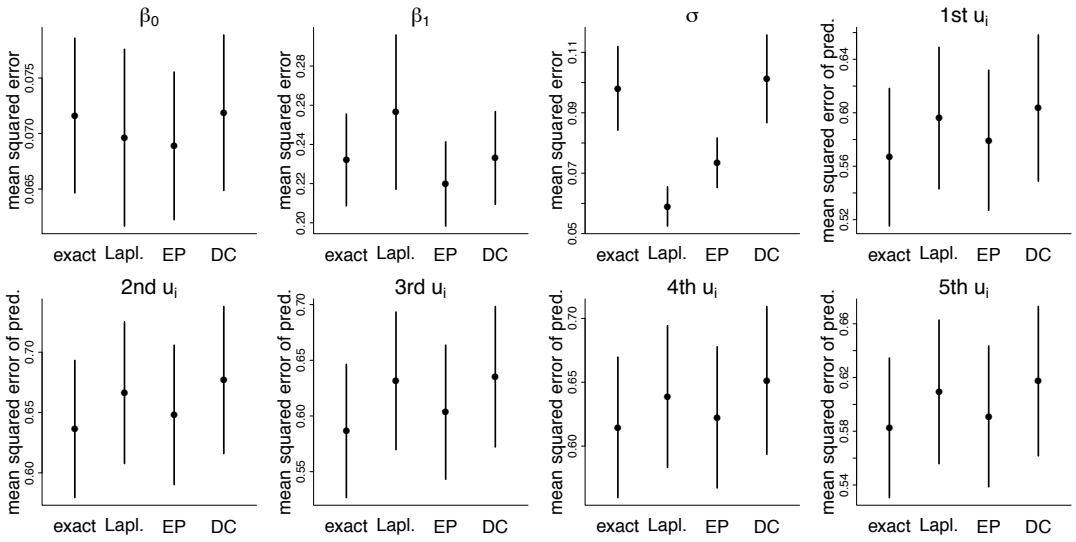


Figure 3.2: *Summary of confidence interval coverage for the simulation study with true parameter values from equation (3.29). The horizontal lines are the EP-based confidence intervals for 50 randomly chosen replications of the simulation study, the solid circular points indicate the corresponding point estimates and the vertical lines indicate true parameter values. The percentage given in the top right-hand corner of each panel is the empirical coverage over all 1000 replications.*

the average time for the EP routine over the 1000 replications completed on Computer 1 was 0.1960 seconds, whilst data cloning and adaptive Gauss-Hermite took an average of 143 and 16.1 seconds respectively. Laplace was also extremely time efficient, taking on average 0.158 seconds. It is worthing noting construction of confidence intervals occupies a large amount of time for the quadrature approach as it is not optimised with the `glmer()` function.

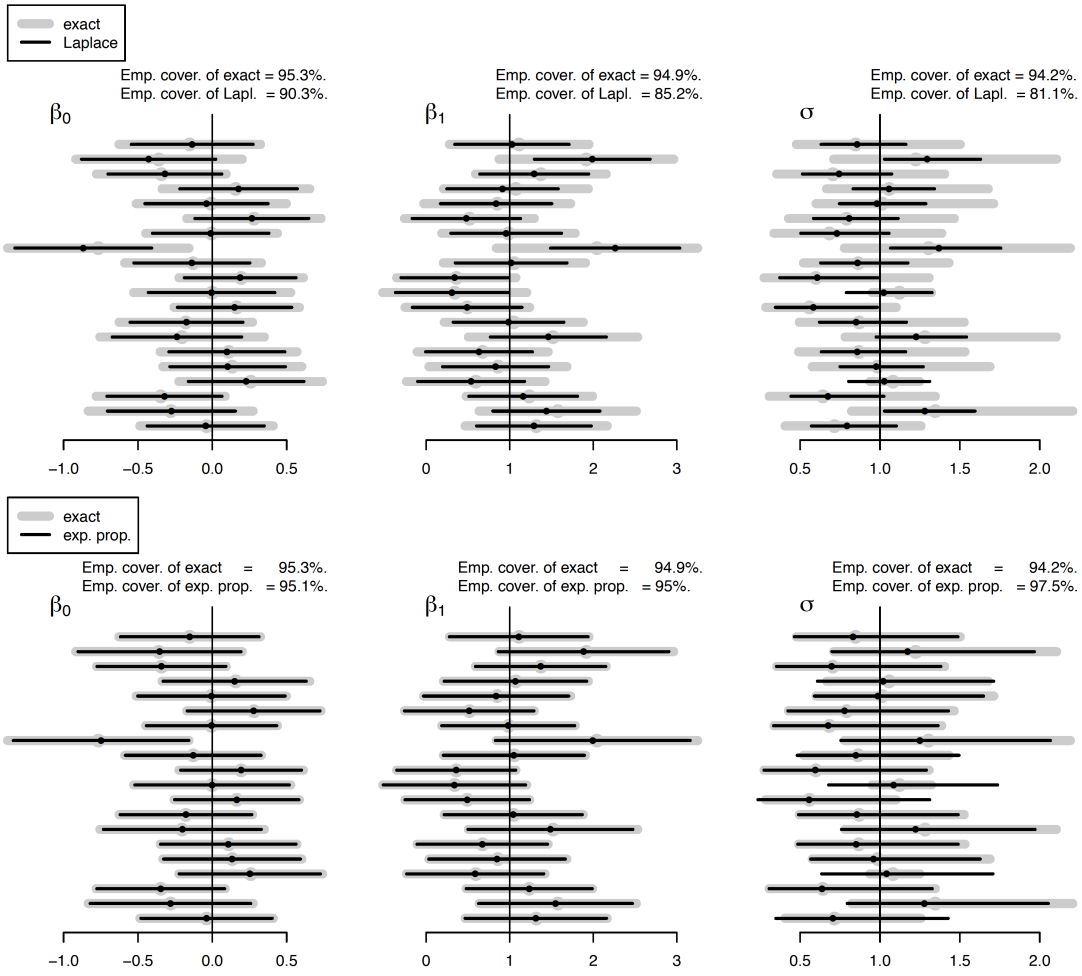


Figure 3.3: *Comparison of point estimates and 95% confidence intervals for the simulation study with true parameter values given by equation (3.27). We display 20 randomly chosen replications of the simulation study described. The first row of panels compares exact maximum likelihood with Laplace approximation while the second row compares exact maximum likelihood with expectation propagation approximation. The vertical lines indicate true parameter values and the percentages displayed at the top of each panel are empirical coverages over all 1000 replications for each method involved in the comparison.*

3.4.2 Maximum likelihood estimates for bivariate random effects

Data for the second simulation study was simulated according to equation (3.1) with true parameter values

$$\boldsymbol{\beta}_{\text{true}} = [0.37, 0.93, -0.46, 0.08, -1.34, 1.09]^\top \quad \text{and} \quad \boldsymbol{\Sigma}_{\text{true}} = \begin{bmatrix} 0.53 & -0.36 \\ -0.36 & 0.92 \end{bmatrix}. \tag{3.28}$$

We fixed the number of groups in the data at $m = 250$ and the number of measurements in the ith group was a randomly generated integer between 20 and 30 on a uniform distribution. The $\boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{x}_{ij}^{\mathbf{R}}$ vectors were of the form

$$\boldsymbol{x}_{ij}^{\mathbf{F}} = [1, x_{1_{ij}}, x_{2_{ij}}, x_{3_{ij}}, x_{4_{ij}}, x_{5_{ij}}]^\top \quad \text{and} \quad \boldsymbol{x}_{ij}^{\mathbf{R}} = [1, x_{1_{ij}}]^\top,$$

where each $x_{k_{ij}}$ was generated independently from a uniform distribution on the unit interval.

Figure 3.4 displays point estimates and corresponding 95% confidence intervals for each interpretable model parameters, for 50 randomly chosen replications. As before, the empirical coverage values based on all 1000 replications are in the top right-hand corner of each panel. For five of the nine parameters the empirical coverage values of EP is greater then the expected 95% empirical coverage. However, even the random intercept variance parameter with the lowest coverage of 94.2% is within 1% of the expected coverage. By implementing the required EP algorithm in the low level language Fortran 77 we maintain high speed inference despite the higher samples and increased complexity of the model. On Computer 1, the average computing time over the 1000 replications of the simulation study was 18 seconds, the upper quartile was 20 seconds and the maximum was 34 seconds.

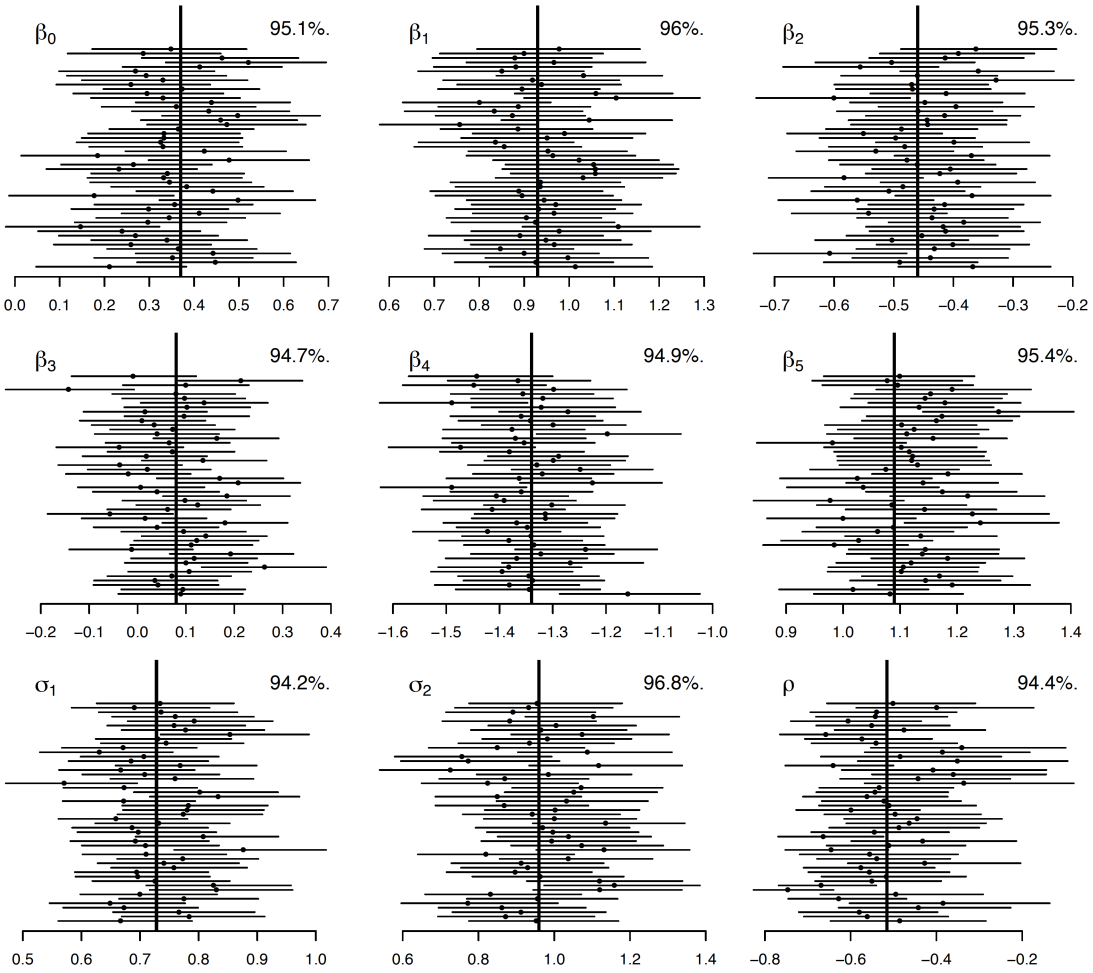


Figure 3.4: *Summary of confidence interval coverage for the simulation study with true parameter values from equation (3.28). The horizontal lines are the EP-based confidence intervals for 50 randomly chosen replications of the simulation study, the solid circular points indicate the corresponding point estimates and the vertical lines indicate true parameter values. The percentage given in the top right-hand corner of each panel is the empirical coverage over all 1000 replications.*

3.4.3 Maximum likelihood estimates for trivariate random effects

The third simulation study was repeated 1000 times, where datasets were simulated according to (3.29) with true parameter values

$$\boldsymbol{\beta}_{\text{true}} = [0.37, 0.93, -0.46, 0.08, -1.34, 1.09]^\top \quad \text{and} \quad \boldsymbol{\Sigma}_{\text{true}} = \begin{bmatrix} 0.53 & -0.36 & -0.11 \\ -0.36 & 0.92 & 0.08 \\ -0.11 & 0.08 & 0.74 \end{bmatrix}. \tag{3.29}$$

Each dataset had $m = 250$ groups, and each ith group had a randomly generated number of observations, between 20 and 30 on a uniform distribution. The $\boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{x}_{ij}^{\mathbf{R}}$ vectors were of the form

$$\boldsymbol{x}_{ij}^{\mathbf{F}} = [1, x_{1_{ij}}, x_{2_{ij}}, x_{3_{ij}}, x_{4_{ij}}, x_{5_{ij}}]^\top \quad \text{and} \quad \boldsymbol{x}_{ij}^{\mathbf{R}} = [1, x_{1_{ij}}, x_{2_{ij}}]^\top,$$

where each $x_{k_{ij}}$ was generated independently from a uniform distribution on the unit interval.

The resulting estimates and 95% confidence intervals for each interpretable model parameter of the study are presented in Figure 3.5, where the numbers in the upper-right hand corner of each panel are the empirical coverage values based on all 1000 replicates. Only 25 randomly chosen replicates are shown in each of the panels for ease of viewing. Across all 12 parameters estimated, the empirical coverage values showed excellent accuracy. While three of the fixed effects parameters had less than 95% empirical coverage, three of the random effects parameters had more than 96% empirical coverage. This component of the study was run on Computer 2 and thus we do not provide computational times.

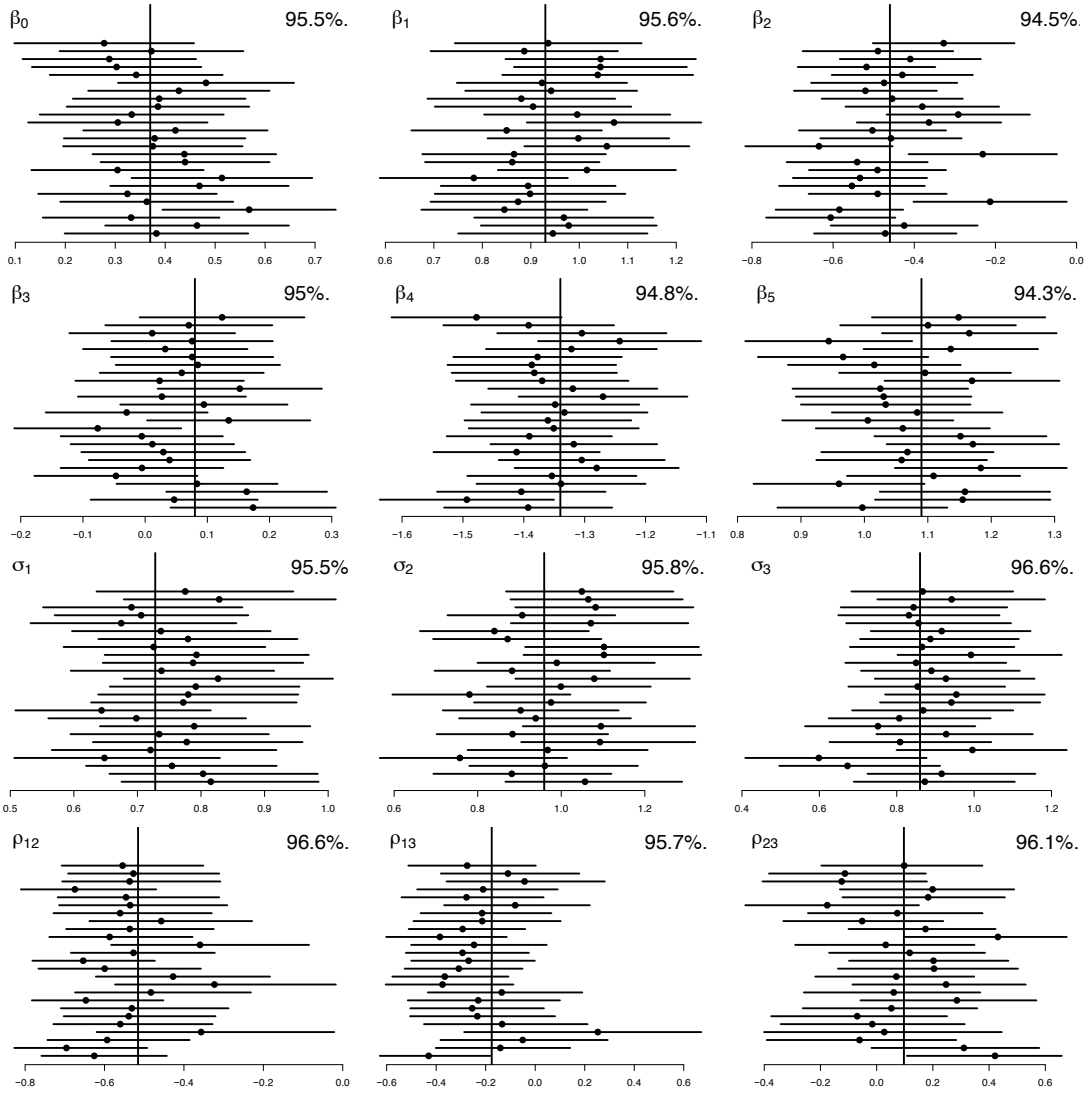


Figure 3.5: *Summary of confidence interval coverage for the simulation study with true parameter values from equation (3.29). The horizontal lines are the EP-based confidence intervals for 25 randomly chosen replications of the simulation study, the solid circular points indicate the corresponding point estimates and the vertical lines indicate true parameter values. The percentage given in the top right-hand corner of each panel is the empirical coverage over all 1000 replications.*

3.5 Appendix

3.5.1 Proof of Definition 13

To facilitate the computation of intractable integrals that arise in each of the moments, we now introduce Lemmas 2 and 3. For $\boldsymbol{x} \in \mathbb{R}^d$, define

$$\phi_{\boldsymbol{\Sigma}}(\boldsymbol{x}) \equiv (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\Big(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\Big).$$

Lemma 2. *For any function $g : \mathbb{R} \to \mathbb{R}$ and $d \times 1$ vectors $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_3$, the following is true:*

$$\int_{\mathbb{R}^d} g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} = \int_{-\infty}^{\infty} g\big(\|\boldsymbol{\alpha}_1\|x\big)\phi(x)dx, \tag{3.30}$$

$$\int_{\mathbb{R}^d} g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_2^\top \boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} = \big(\boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_2\big)/\|\boldsymbol{\alpha}_1\| \int_{-\infty}^{\infty} x g\big(\|\boldsymbol{\alpha}_1\|x\big)\phi(x)dx \tag{3.31}$$

and

$$\int_{\mathbb{R}^d} g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_2^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_3^\top \boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} = \big(\boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_3\big)\int_{-\infty}^{\infty} g\big(\|\boldsymbol{\alpha}_1\|x\big)\phi(x)dx + \big(\boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_2\big)\big(\boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_3\big)/\|\boldsymbol{\alpha}_1\|^2 \int_{-\infty}^{\infty} (x^2-1) g\big(\|\boldsymbol{\alpha}_1\|x\big)\phi(x)dx. \tag{3.32}$$

3.5.1.1 Proof of Lemma 2

Note that the integrals on the left-hand side in equations (3.30), (3.31) and (3.32) are respectively

$$E\Big(g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\Big), \quad E\Big(g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_2^\top \boldsymbol{x}\big)\Big) \quad \text{and} \quad E\Big(g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_2^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_3^\top \boldsymbol{x}\big)\Big),$$

where

$$\boldsymbol{x} \sim \boldsymbol{N}\big(\boldsymbol{0}_d, \boldsymbol{I}_d\big).$$

We now present simplification of the integral in equation (3.32). Simplification of the integrals in equation (3.30) and (3.31) are analogous. Make the change of variables

$$\boldsymbol{s} \equiv \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \boldsymbol{A}\boldsymbol{x}, \quad \text{where} \quad \boldsymbol{A} \equiv \begin{bmatrix} \boldsymbol{\alpha}_1^\top \\ \boldsymbol{\alpha}_2^\top \\ \boldsymbol{\alpha}_3^\top \end{bmatrix}$$

so that

$$E\Big(g\big(\boldsymbol{\alpha}_1^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_2^\top \boldsymbol{x}\big)\big(\boldsymbol{\alpha}_3^\top \boldsymbol{x}\big)\Big) = E\big(g(s_1)s_2s_3\big), \quad \text{where} \quad \boldsymbol{s} \sim \boldsymbol{N}\big(\boldsymbol{0}_3, \boldsymbol{A}\boldsymbol{A}^\top\big).$$

We then note that,

$$\begin{aligned} E\big(g(s_1)s_2s_3\big) &= \int_{-\infty}^{\infty} g(s_1)\bigg(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s_2s_3p(s_2s_3|s_1)ds_2ds_3\bigg)p(s_1)ds_1 \\ &= \int_{-\infty}^{\infty} g(s_1)\bigg(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Big(\mathrm{Cov}(s_2,s_3|s_1)+E(s_2|s_1)E(s_3|s_1)\Big)ds_2ds_3\bigg) \\ &\quad \times p(s_1)ds_1 \end{aligned}$$

and

$$\begin{aligned} \begin{bmatrix} s_2 \\ s_3 \end{bmatrix}\bigg|s_1 \;\sim\; \boldsymbol{N}\Bigg(&\big(s_1/\|\boldsymbol{\alpha}_1\|^2\big)\begin{bmatrix}\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_2 \\ \boldsymbol{\alpha}_3^\top\boldsymbol{\alpha}_2\end{bmatrix}, \\ &\begin{bmatrix}\|\boldsymbol{\alpha}_2\|^2 & \boldsymbol{\alpha}_2^\top\boldsymbol{\alpha}_3 \\ \boldsymbol{\alpha}_2^\top\boldsymbol{\alpha}_3 & \|\boldsymbol{\alpha}_3\|^2\end{bmatrix} - \big(1/\|\boldsymbol{\alpha}_1\|^2\big)\begin{bmatrix}\big(\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_2\big)^2 & \big(\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_2\big)\big(\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_3\big) \\ \big(\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_2\big)\big(\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_3\big) & \big(\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_3\big)^2\end{bmatrix}\Bigg). \end{aligned}$$

Simple algebraic manipulations can be used to arrive at equation 3.32.

Lemma 3. *For integrals of the forms listed below, corresponding closed form solutions exist:*

$$\int_{\mathbb{R}^d}\Phi(a+\boldsymbol{b}^\top\boldsymbol{x})\phi_I(\boldsymbol{x})d\boldsymbol{x} = \Phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg), \tag{3.33}$$

$$\int_{\mathbb{R}^d}\boldsymbol{x}\Phi(a+\boldsymbol{b}^\top\boldsymbol{x})\phi_I(\boldsymbol{x})d\boldsymbol{x} = \frac{\boldsymbol{b}}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg), \tag{3.34}$$

$$\begin{aligned}\int_{\mathbb{R}^d}\boldsymbol{x}\boldsymbol{x}^\top\Phi(a+\boldsymbol{b}^\top\boldsymbol{x})\phi_I(\boldsymbol{x})d\boldsymbol{x} \\ = \Phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg)\boldsymbol{I}_d - \frac{a\boldsymbol{b}\boldsymbol{b}^\top}{\sqrt{(\boldsymbol{b}^\top\boldsymbol{b}+1)^3}}\phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg),\end{aligned} \tag{3.35}$$

where $a\in\mathbb{R}$ and $\boldsymbol{b}$ is $d\times 1$ vector.

3.5.1.2 Proof of Lemma 3

To arrive at equation (3.34), let $\boldsymbol{e}_i$ be a $d\times 1$ vector with its ith entry equal to 1 and zeroes elsewhere. Then, using Lemma 2 the ith entry of the right-hand side of equation (3.34) is

$$\begin{aligned}\int_{\mathbb{R}^d}(\boldsymbol{e}_i^\top\boldsymbol{x})\Phi\big(a+\boldsymbol{b}^\top\boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} &= \frac{\boldsymbol{e}_i^\top\boldsymbol{b}}{\|\boldsymbol{b}\|}\int_{-\infty}^{\infty}x\Phi\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx \\ &= -\frac{\boldsymbol{e}_i^\top\boldsymbol{b}}{\|\boldsymbol{b}\|}\int_{-\infty}^{\infty}\Phi\big(a+\|\boldsymbol{b}\|x\big)\phi'(x)dx \\ &= \boldsymbol{e}_i^\top\boldsymbol{b}\int_{-\infty}^{\infty}\phi\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx,\end{aligned} \tag{3.36}$$

where equation (3.36) follows via integration by parts. The components of the integrand in equation (3.36) can be expressed as

$$(2\pi)^{-1}\exp\Big\{-\frac{1}{2}\big((a+\|\boldsymbol{b}\|x)^2+x^2\big)\Big\}=\phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg)\phi\bigg(\frac{x+a\|\boldsymbol{b}\|}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg).$$

Equation (3.34) is of direct consequence.

To arrive at equation (3.35), define e_j as a $d\times 1$ vector with jth entry equal to 1 and with zeroes elsewhere. Then by equation (3.32), the (i,j) entry of the right-hand side of equation (3.35) is

$$\begin{aligned}
&\int_{\mathbb{R}^d}\big(\boldsymbol{e}_i^\top\boldsymbol{x}\big)\big(\boldsymbol{e}_j^\top\boldsymbol{x}\big)\Phi\big(a+\boldsymbol{b}^\top\boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x}\\
&\quad=\big(\boldsymbol{e}_i^\top\boldsymbol{e}_j\big)\int_{-\infty}^{\infty}\Phi\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx+\frac{\big(\boldsymbol{e}_i^\top\boldsymbol{b}\big)\big(\boldsymbol{e}_j^\top\boldsymbol{b}\big)}{\|\boldsymbol{b}\|^2}\int_{-\infty}^{\infty}\Phi\big(a+\|\boldsymbol{b}\|x\big)\phi''(x)dx\\
&\quad=\big(\boldsymbol{e}_i^\top\boldsymbol{e}_j\big)\phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg)\phi(x)dx-\frac{\big(\boldsymbol{e}_i^\top\boldsymbol{b}\big)\big(\boldsymbol{e}_j^\top\boldsymbol{b}\big)}{\|\boldsymbol{b}\|^2}\int_{-\infty}^{\infty}\Phi\big(a+\|\boldsymbol{b}\|x\big)\phi'(x)dx,\quad(3.37)
\end{aligned}$$

where equation (3.37) follows via integration by parts. The integrand in equation (3.37) is expressible as

$$(2\pi)^{-1}x\exp\bigg\{-\frac{1}{2}\Big(\big(a+\|\boldsymbol{b}\|x\big)^2+x^2\Big)\bigg\}=-x\phi\bigg(\frac{a}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg)\phi\bigg(\frac{x+a\|\boldsymbol{b}\|}{\sqrt{\boldsymbol{b}^\top\boldsymbol{b}+1}}\bigg).$$

Equation (3.35) follows. It is easy to prove equation (3.33) using the same method.

Consider the input function from equation (3.6). To obtain the projection of this function to the multivariate normal, we require closed form solutions to the equivalent of the zeroth, first and second moments. Using the $\otimes k$ notation given in equation (1.2) we can calculate each moment as

$$\mathcal{M}_k\quad\equiv\quad\int_{\mathbb{R}^d}\boldsymbol{x}^{\otimes k}\Phi\big(c_0+\boldsymbol{c}_1^\top\boldsymbol{x}\big)\exp\big((\boldsymbol{\eta}_1^{\text{input}})^\top\boldsymbol{x}+\boldsymbol{x}^\top\boldsymbol{H}_2^{\text{input}}\boldsymbol{x}\big)d\boldsymbol{x}\qquad(3.38)$$

and subsequently obtain the optimum natural parameters via algebraic manipulations. Note in the interest of brevity we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. We now work on the general case for all $\otimes k$. Using the matrix notation defined in equation (1.14)

$$\begin{aligned}
\mathcal{M}_k\quad=\quad&\int_{\mathbb{R}^d}\boldsymbol{x}^{\otimes k}\Phi\big(c_0+\boldsymbol{c}_1^\top\boldsymbol{x}\big)(2\pi)^{-d/2}\exp\left\{\begin{bmatrix}\boldsymbol{x}\\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top)\end{bmatrix}^\top\boldsymbol{\eta}-A(\boldsymbol{\eta})\right\}d\boldsymbol{x}\\
&\times\exp\big(A(\boldsymbol{\eta})\big)(2\pi)^{d/2}.
\end{aligned}$$

Using the inverse map of the natural parameters in equation (1.14)

$$\mathcal{M}_k\quad=\quad\int_{\mathbb{R}^d}\boldsymbol{x}^{\otimes k}\Phi\big(c_0+\boldsymbol{c}_1^\top\boldsymbol{x}\big)\phi_{\boldsymbol{\Sigma}}(\boldsymbol{x}-\boldsymbol{\mu})d\boldsymbol{x}\;(2\pi)^{d/2}\exp\big(A(\boldsymbol{\eta})\big).$$

We now implement a change of variable such that $\boldsymbol{z} = \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x} - \boldsymbol{\mu})$thus

$$\mathcal{M}_k = \int_{\mathbb{R}^d} \left(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}\right)^{\otimes k} \Phi\left(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + (\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1)^\top \boldsymbol{z}\right) \phi_{\boldsymbol{I}}(\boldsymbol{z}) d\boldsymbol{z}\ Z_1, \quad (3.39)$$

where $Z_1 = \exp\left(A(\boldsymbol{\eta}) + d/2\log(2\pi)\right)$. It is then easy to show that each of the kth moments are

$$\begin{aligned} Z_1^{-1}\mathcal{M}_0 &= \Phi(r_2), \\ Z_1^{-1}\mathcal{M}_1 &= \boldsymbol{\mu}\Phi(r_2) + 2\boldsymbol{\Sigma}\boldsymbol{c}_1 r_1^{-1}\phi(r_2), \\ Z_1^{-1}\mathcal{M}_2 &= (\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma})\Phi(r_2) + 2r_1^{-1}(\boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{\mu}^\top + \boldsymbol{\mu}\boldsymbol{c}_1^\top\boldsymbol{\Sigma} - 2r_2 r_1^{-1}\boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top\boldsymbol{\Sigma})\phi(r_2). \end{aligned}$$

The expressions for the optimal mean and variance parameters follow respectively, as

$$\boldsymbol{\mu}^* = E(\boldsymbol{x}) = \frac{\mathcal{M}_1}{\mathcal{M}_0} = \boldsymbol{\mu} + 2\boldsymbol{\Sigma}\boldsymbol{c}_1 r_1^{-1}\zeta'(r_2),$$

and

$$\boldsymbol{\Sigma}^* = E(\boldsymbol{x}\boldsymbol{x}^\top) - E(\boldsymbol{x})E(\boldsymbol{x})^\top = \frac{\mathcal{M}_2}{\mathcal{M}_0} - \frac{\mathcal{M}_1}{\mathcal{M}_0}\left(\frac{\mathcal{M}_1}{\mathcal{M}_0}\right)^\top = \boldsymbol{\Sigma} - (4\boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top\boldsymbol{\Sigma}r_1^{-2})\zeta''(r_2).$$

Converting back to non-vectorised natural parameter form using conversion defined in equation (1.7) leads to the required result.

3.5.2 Proof of Result 12

Consider an unnormalised multivariate normal density function $f(\boldsymbol{x}; \boldsymbol{\eta})$

$$f_{\text{UN}}(\boldsymbol{x}; \boldsymbol{\eta}) = \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta} \right\}.$$

Then the Kullback-Leibler divergence of f_{UN} from f_{input} is

$$\begin{aligned} \text{KL}(f_{\text{input}} \parallel f_{\text{UN}}) &= \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) \log\left(f_{\text{input}}(\boldsymbol{x})/f_{\text{UN}}(\boldsymbol{x}; \boldsymbol{\eta})\right) + f_{\text{UN}}(\boldsymbol{x}; \boldsymbol{\eta}) - f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ &= \mathcal{K}(\boldsymbol{\eta}) + \text{const}, \end{aligned}$$

where

$$\mathcal{K}(\boldsymbol{\eta}) \equiv (2\pi)^{d/2} \exp\left(\boldsymbol{\eta}_0 + A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2})\right) - \begin{bmatrix} \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \boldsymbol{x} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \end{bmatrix}^\top \boldsymbol{\eta}.$$

Note that the derivative vector of $\mathcal{K}(\boldsymbol{\eta})$ is

$$\mathsf{D}\mathcal{K}(\boldsymbol{\eta}) = (2\pi)^{d/2} \exp\left(\boldsymbol{\eta}_0 + A_{\text{N}}(\boldsymbol{\eta}_{1:2})\right) \begin{bmatrix} 1 \\ \mathsf{D}A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2})^\top \end{bmatrix}^\top - \begin{bmatrix} \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \boldsymbol{x} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \end{bmatrix}^\top.$$

Since the stationary condition to minimise the $\mathrm{KL}(f_{\text{input}} \parallel f_{\text{UN}})$ occurs at $\mathsf{D}\mathcal{K}(\boldsymbol{\eta})^\top = \mathbf{0}$,

$$(2\pi)^{d/2} \exp\left(\boldsymbol{\eta}_0 + A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2})\right) \begin{bmatrix} 1 \\ \nabla A_{\text{N}}(\boldsymbol{\eta}_{1:2}) \end{bmatrix} = \begin{bmatrix} \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \boldsymbol{x} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} \end{bmatrix}, \quad (3.40)$$

where $\nabla A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2}) \equiv \mathsf{D}A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2})^\top$ is the gradient vector of $A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2})$. It is then easy to show

$$\boldsymbol{\eta}_0^* = \log(C_{f_{\text{input}}}) - A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2}^*) - \frac{d}{2}\log(2\pi),$$

where

$$\boldsymbol{\eta}_{1:2}^* = (\nabla A_{\boldsymbol{N}})^{-1} \left(\begin{bmatrix} \int_{\mathbb{R}^d} \boldsymbol{x} \left(f_{\text{input}}(\boldsymbol{x})/C_{f_{\text{input}}}\right) d\boldsymbol{x} \\ \int_{\mathbb{R}^d} \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \left(f_{\text{input}}(\boldsymbol{x})/C_{f_{\text{input}}}\right) d\boldsymbol{x} \end{bmatrix} \right) \quad (3.41)$$

with existence and uniqueness of $(\nabla A_{\boldsymbol{N}})^{-1}$ being guaranteed by Proposition 3.2 of Wainwright & Jordan (2008),[66] and $C_{f_{\text{input}}} \equiv \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x}$. The Hessian matrix of $\mathcal{K}(\boldsymbol{\eta})$ is

$$\mathsf{H}\mathcal{K}(\boldsymbol{\eta}) = (2\pi)^{d/2} \exp\left(\boldsymbol{\eta}_0 + A_{\text{N}}(\boldsymbol{\eta}_{1:2})\right) \left\{ \begin{bmatrix} 1 \\ \nabla A_{\boldsymbol{N}}(\boldsymbol{\eta}_{1:2}) \end{bmatrix} \begin{bmatrix} 1 \\ \nabla A(\boldsymbol{\eta}_{1:2}) \end{bmatrix}^\top + \begin{bmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \mathsf{H}A_{\text{N}}(\boldsymbol{\eta}_{1:2}) \end{bmatrix} \right\}$$

By Proposition 3.1 of Wainwright & Jordan (2008),[66] A_{N} is strictly convex on its domain and thus $\mathsf{H}A(\boldsymbol{\eta}_{1:2})$ is positive definite. Thus, $\mathsf{H}\mathcal{K}(\boldsymbol{\eta})$ is positive definite for all $\boldsymbol{\eta}$ and so equation (2.40) is the unique maximiser of $\mathrm{KL}\left(f_{\text{input}} \parallel f_{\boldsymbol{UN}}\right)$. Therefore,

$$\text{proj}_{\text{UN}}[f_{\text{input}}](x) = \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta}^* \right\}$$

where $\boldsymbol{\eta}^*$ is as previously defined. However, $\boldsymbol{\eta}_{1:2}^*$ is the same natural parameter vector that arises via projection of $f_{\text{input}}/C_{f_{\text{input}}}$ onto the family of univariate normal density functions, thus

$$\text{proj}_{\boldsymbol{N}}[f_{\text{input}}/C_{f_{\text{input}}}](x) = \exp\left\{ \begin{bmatrix} \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix} \boldsymbol{\eta}_{1:2}^* - A_{\text{N}}(\boldsymbol{\eta}_{1:2}^*) \right\} (2\pi)^{-d/2}$$

which immediately leads to Result 3.

Chapter 4

Expectation propagation for one level logistic mixed models

Although a probit model can be used to handle data with a binary response variable, logistic models are used more frequently for the elegant interpretation they facilitate. However, since the integral arising in the projection required for fitting logistic GLMMs via EP does not have a closed form solution, the models explored in this section will be more computationally intensive than their probit counterparts. We now explore two methods of obtaining the appropriate update for the logistic link function. The benefit of implementing the message passing approach in the previous two chapters becomes obvious over the next two sections, as the logistic extension becomes easily available with minimal algebraic overheads.

This chapter is broken into two main sections. As in Chapter 2, we start with the simplest random intercepts only model in Section 4.1. The next Section 4.2 follows the work presented in Chapter 3 and extends our methodolgy to a general logistic model for any number of fixed and random effects. Since the previous two chapters explain the message passing frameworks in detail, we provide only the updated details and refer back to the previous Chapters when required.

The content of this chapter is published as: Hall, P., Johnstone, I. M., Ormerod, J. T., Wand, M. P., & Yu, J. C. F. (2020). Fast and Accurate Binary Response Mixed Model Analysis via Expectation Propagation, Journal of the American Statistical Association, 115:532, 1902–1916, DOI: 10.1080/01621459.2019.1665529

4.1 The simplest logistic mixed model

As in the probit case, we develop our methodology on the simplest GLMM with random intercepts only, where we consider only the parameter σ^2. For observed values of

$$y_{ij}; \quad 1 \leq i \leq m, \quad 1 \leq j \leq n,$$

where is $y_{ij} \in \{0, 1\}$, the simplified logistic binary mixed model can be shown to be

$$y_{ij}|u_i \overset{\text{iid}}{\sim} \text{Bernoulli}\big(\text{expit}(u_i)\big), \qquad u_i \overset{\text{iid}}{\sim} N(0, \sigma^2),$$

where $\text{expit}(x)$ is given by equation (1.9) and u_i is a scalar unobserved latent variable. We wish to find the maximiser of $\ell(\sigma^2)$ denoted by $\widehat{\sigma^2}$. The log-likelihood can be expressed as a sum

$$\ell(\sigma^2) = \sum_{i=1}^{m} \ell_i(\sigma^2),$$

where

$$\ell_i(\sigma^2) \equiv \log \int_{-\infty}^{\infty} \prod_{j=1}^{n} \text{expit}\big((2y_{ij} - 1)u_i\big)(2\pi\sigma^2)^{-1/2} \exp(-u_i^2/2\sigma^2) du_i, \tag{4.1}$$

and the maximum likelihood estimate of σ^2 is given by

$$\widehat{\sigma^2} = \underset{\sigma^2}{\text{argmax}}\ \ell(\sigma^2).$$

The best predictor is given by

$$\text{BP}(u_i) = \frac{\int_{-\infty}^{\infty} u_i \prod_{j=1}^{n} \text{expit}\big((2y_{ij} - 1)u_i\big) \exp(-u_i^2/2\sigma^2) du_i}{\int_{-\infty}^{\infty} \prod_{j=1}^{n} \text{expit}\big((2y_{ij} - 1)u_i\big) \exp(-u_i^2/2\sigma^2) du_i}.$$

Although we are working with the simplest model, the calculation of the maximum likelihood estimator and best predictors are complicated by the intractable integrals arising in equation (4.1). To calculate the log-likelihood surface of $\ell(\sigma^2)$ we explore an EP approximation scheme and compare it to a traditional quadrature approach.

The following subsections copy the structure of Chapter 2. In Subsection 4.1.1 we provide details of the quadrature approach to estimating the likelihood surface, then explain our novel method using EP in Subsection 4.1.2. We compare the likelihood surface of both methods in Subsection 4.1.3 and discuss best predictor computation in Subsection 4.1.4. Point estimate and confidence interval calculation is conducted

analogously to Section 2.4.1 and thus we do not repeat it.

4.1.1 Traditional quadrature likelihood approximation

Implementation of the adaptive quadrature via the Gauss-Kronrod method follows the same approach as in the probit case, where we utilise the R function `integrate()` included with base R. Although it is possible to directly implement the integrate function for calculation of intractable integrals arising in equation (4.1), doing so leads to results of small absolute value and as such are difficult to store. With this in mind, denoting $y_{ij}^{\dagger} \equiv 2y_{ij} - 1$, we suggest that for numerical stability each $\ell_i(\sigma^2)$ is calculated as

$$\frac{1}{2}\log(2\pi\sigma^2) + \ell_i(\sigma^2) = h(u_{0i}) + \log \int_{-\infty}^{\infty} \exp\big(h_i(u) - h_i(u_{0i})\big)du,$$

where

$$\begin{aligned} h_i(u) &\equiv \sum_{j=1}^{n} \left\{ y_{ij}^{\dagger} u - \log\Big(1 + \exp\big(y_{ij}^{\dagger} u\big)\Big) \right\} - \frac{u^2}{2\sigma^2}, \\ h_i'(u) &= \sum_{j=1}^{n} \left\{ y_{ij}^{\dagger} \Big(1 + \exp\big(y_{ij}^{\dagger} u\big)\Big)^{-1} \right\} - \frac{u}{\sigma^2} \end{aligned}$$

and u_{0i} is the root of h_i' which we recommend finding using a bisection search, where the starting values are selected -1 and 1 to be for the lower and upper bounds respectively.

4.1.2 Expectation propagation likelihood approximation

We now consider an EP approach to the approximate likelihood (denoted by $\underset{\sim}{\ell}(\sigma^2)$) which follows that of Section 2.2. The EP approximation is motivated by the minimisation of a KL divergence criterion, which in the logistic case is used to select an unnormalised normal density function to replace each

$$\operatorname{expit}\big((2y_{ij} - 1)u_i\big), \quad 1 \leq j \leq n$$

in equation (4.1). Subsequently, the integrand is proportional to a product of univariate normal density functions.

The goal of the EP problem is to find the KL projection of the input function onto the family of normal density functions. For logistic binary GLMMs, EP requires

repeated projection of the form

$$f_{\text{input}}(x) = \text{expit}(c_0 + c_1 x) \exp\left(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\right) \tag{4.2}$$

onto an unnormalised normal distribution (written in exponential form in equation (2.3)), where $\eta_1^{\text{input}} \in \mathbb{R}$ and $\eta_2^{\text{input}} < 0$, and c_0, c_1 and x follow from the probit case. As such, we seek $\boldsymbol{\eta}^*$ such that

$$\int_{-\infty}^{\infty} x^k \text{expit}(c_0 + c_1 x) \exp\left(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\right) dx = \int_{-\infty}^{\infty} x^k \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} dx, \tag{4.3}$$

where $\boldsymbol{\eta}^*$ is defined in equation (2.4). As before, obtaining the natural parameters $\boldsymbol{\eta}^*$ for projection onto the unnormalised normal family follows from obtaining the projection onto the normal family. As per Result 3, the optimal natural parameters η_1^* and η_2^*, are given according to the projection of the normalised function $f_{\text{input}}/C_{f_{\text{input}}}$ onto the normal family. We can subsequently use these optimal natural parameters to find the normalising natural parameter η_0^* via Result 4 and thus obtain the projection onto unnormalised normal family.

Thus to obtain the required projection, we first obtain the optimal natural parameters η_1^* and η_2^* to project onto the normal family. However, unlike the probit case no closed form solution exists for the integral arising in the required projection (left hand side of equation (4.3)). As in Nolan & Wand (2017)[47] we turn to the piecewise approximation of the expit function by Monahan & Stefanski (1989).[46] The key result can be summarised as,

$$\begin{aligned} \text{expit}(x) &\approx \text{expit}_{\text{MS}}(x), \\ \text{where} \quad \text{expit}_{\text{MS}}(x) &= \sum_{i=1}^{8} p_i \Phi(s_i x) \end{aligned} \tag{4.4}$$

and p_i and s_i are fixed constants given in Table 18.4.1 of Monahan & Stefanski (1989).[46] Thus we can write

$$\sum_{i=1}^{8} p_i \int_{-\infty}^{\infty} x^k \Phi(s_i c_0 + s_i c_1 x) \exp\left(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\right) dx \approx \int_{-\infty}^{\infty} x^k \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} dx$$

and implement Lemma 1 to solve the required integrals. Using simple algebraic manipulations analogous to the simplest probit case shown in equation (2.2) we arrive at Result 16.

Result 16. *Given f_{input} follows the form of equation (4.2), the projection onto the univariate normal family is given by*

$$proj_N[f_{input}] = \exp\Big(\boldsymbol{T}(x)^\top \boldsymbol{\eta}_{-1}^* - A(\boldsymbol{\eta}_{-1}^*)\Big)h(x)$$

where

$$\boldsymbol{\eta}_{-1}^* = k_{logistic}\big(\boldsymbol{\eta}_{-1}^{input}; c_0, c_1\big), \qquad \boldsymbol{\eta}_{-1}^{input} \equiv \begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}_{-1}^* \equiv \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix},$$

$k_{logistic}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1\right)$ *is as per Definition 15 and $\boldsymbol{T}(x)$ and $h(x)$ follow from Section 1.5.2.1.*

Definition 15. *For primary arguments $a_1 \in \mathbb{R}$ and $a_2 < 0$ and auxiliary arguments $c_0, c_1 \in \mathbb{R}$ the function $k_{logistic} : H \to H$ is given by*

$$k_{logistic}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1\right) = \begin{bmatrix} r_5(a_1 + c_1 r_3) \\ r_5 a_1 \end{bmatrix},$$

where

$$r_{1i} = \sqrt{2\big(2 - s_i^2 c_1^2 a_2^{-1}\big)}, \quad r_{2i} = s_i\big(2c_0 - c_1 a_2^{-1} a_1\big) r_{1i}^{-1}, \quad r_3 = \frac{2\sum_{i=1}^8 p_i s_i \phi(r_{2i}) r_{1i}^{-1}}{\sum_{i=1}^8 p_i \Phi(r_{2i})},$$

$$\tilde{r_3} = 4\sum_{i=1}^8 \left(\frac{p_i s_i^2 r_{2i}}{\sum_{i=1}^8 p_i \Phi(r_{2i})}\right), \quad r_4 = \frac{1}{2}\big(r_3^2 + \tilde{r_3}\big), \quad r_5 = \big(a_2 + r_4 c_1^2\big)^{-1} a_2.$$

Using Result 16 we now obtain the normalising natural parameter η_0^* to find the projection onto unnormalised normal family.

4.1.2.1 Projection onto the unnormalised normal family

Recall the moment matching problem described by equation (4.3) and Result 4. Then the normalising factor, can be shown to be

$$C_{f_{\text{input}}} = \int_{\mathbb{R}} f_{\text{input}}(x)dx = (2\pi)^{-1/2} \exp\left(A(\boldsymbol{\eta}^{\text{input}})\right) \sum_{i=1}^{8} p_i \Phi(r_{2i}),$$

where r_{2i} is given in Definition 10 and $A(\boldsymbol{\eta})$ is defined in Section 1.5. By Result 4

$$\eta_0^* = \log \sum_{i=1}^{8} p_i \Phi(r_{2i}) + \frac{1}{4}(\eta_1^*)^2/\eta_2^* - \frac{1}{4}(\eta_1^{\text{input}})^2/\eta_2^{\text{input}} + \frac{1}{2}\log(\eta_2^*/\eta_2^{\text{input}}).$$

Thus to obtain η_0^* we introduce Definition 16.

Definition 16. *Consider first primary scalar arguments a_1, a_2, b_1 and b_2, and auxiliary scalar arguments c_0 and c_1. The function $c_{logistic} : H \to \mathbb{R}$ is given by*

$$c_{logistic}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}; c_0, c_1\right) \equiv \log\left(\sum_{i=1}^{8} p_i \Phi(r_{2i})\right) + \frac{1}{4}b_1^2/b_2 - \frac{1}{4}a_1^2/a_2 + \frac{1}{2}\log(b_2/a_2),$$

where r_{2i} follow from Definition 15.

In summary, we can obtain the projection of the input function onto the univariate normal family using k_{logistic}, then find projection onto the unormalised univariate normal family using c_{logistic}. This is formalised in Result 17.

Result 17. *For an unnormalised input function of the form of equation (4.2),*

$$proj_{UN}[f_{input}] = \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \begin{bmatrix} \eta_0^* \\ \eta_1^* \\ \eta_2^* \end{bmatrix} \right\},$$

where

$$\begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix} = k_{logistic}\left(\begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}; c_0, c_1 \right)$$

and

$$\eta_0^* = c_{logistic}\left(\begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix}; c_0, c_1 \right).$$

We now explain how to implement the results shown in this section in a message passing framework.

4.1.2.2 Message passing formulation

As mentioned throughout this section, only minor changes to the message passing algorithm discussed in Section 2.2.2 are required to account for the logistic case. Note the components of the sum of the log-likelihood function $\ell_i(\sigma^2)$ are changed to

$$\ell_i(\sigma^2) = \log \int_{-\infty}^{\infty} \left(\prod_{j=1}^{n} p(y_{ij}|u_i) \right) p(u_i; \sigma^2) du_i, \tag{4.5}$$

where

$$p(y_{ij}|u_i) \equiv \text{expit}\big((2y_{ij} - 1)u_i\big) \quad \text{and} \quad p(u_i; \sigma^2) \equiv (2\pi\sigma^2)^{-1/2} \exp\big(-u_i^2/(2\sigma^2)\big)$$

are the conditional density functions of each response given its random effect and the density function of that random effect respectively.

Note the dependence structure of the product in equation (4.5) matches the probit

model shown in Figure 2.1. As such, we only need to update the messages depending on the conditional density functions of each response given its random effect. Following Section 2.2.2, this involves updating messages $m_{p(y_{ij}|u_i)\to u_i}(u_i)$ (as in equation (2.10)) to

$$m_{p(y_{ij}|u_i)\to u_i}(u_i) \longleftarrow \frac{\text{proj}_{\text{UN}}\left[\text{expit}(c_0 + c_{1_{ij}}u_i)\exp(\eta_1^{\otimes}u_i + \eta_2^{\otimes}u_i^2)\right]}{\exp(\eta_1^{\otimes}u_i + \eta_2^{\otimes}u_i^2)},$$

where $c_0 = 0$ and $c_{1_{ij}} \equiv 2y_{ij} - 1$. Utilising Result 6 leads to equation (2.16), where the linear and quadratic coefficient updates in equation (2.17) are changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\right)_{1:2} \longleftarrow k_{\text{logistic}}\left(\boldsymbol{\eta}_{1:2}^{\otimes}; c_0, c_{1_{ij}}\right) - \boldsymbol{\eta}_{1:2}^{\otimes} \tag{4.6}$$

and where the constant coefficient update in equation (2.18) is changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\right)_0 \longleftarrow c_{\text{logistic}}\left(\boldsymbol{\eta}_{1:2}^{\otimes}, \left(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\right)_{1:2} + \boldsymbol{\eta}_{1:2}^{\otimes}; c_0, c_{1_{ij}}\right). \tag{4.7}$$

In summary, Algorithm 3 applies to the logistic model, however equations (2.17) and (2.18) are replaced with equations (4.6) and (4.7). Additionally, the initialisation of $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$ now follows equation (4.9).

4.1.2.3 Starting values for the univariate logistic case

The EP message passing algorithm proposed relies on good starting values for convergence. We now derive starting values for $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$ using a Taylor series expansion. Note that

$$\log p(y_{ij}|u_i) = f(a_{ij}) - \log(2), \quad \text{where} \quad a_{ij} \equiv (2y_{ij} - 1)u_i,$$

and

$$\begin{aligned} f(x) &= \text{expit}(x), \\ f'(x) &= \frac{\exp(-x)}{\left(\exp(-x) + 1\right)^2}, \\ f''(x) &= \frac{\exp(-x)\left(1 - \exp(-x)\right)}{\left(\exp(-x) + 1\right)^3}. \end{aligned} \tag{4.8}$$

Let $\widehat{u}_i$ be a Laplace approximation to u_i. Now consider the following Taylor series expansion of the data dependent component of $\ell(\sigma^2)$

$$f(a_{ij}) = f(\widehat{a}_{ij}) + (u_i - \widehat{u}_i)(2y_{ij}-1)f'(\widehat{a}_{ij}) + \frac{1}{2}\big((u_i - \widehat{u}_i)(2y_{ij}-1)\big)^2 f''(\widehat{a}_{ij}) + ...$$
$$= \begin{bmatrix} 1 \\ u_i - \widehat{u}_i \\ (u_i - \widehat{u}_i)^2 \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + ...,$$

where $\widehat{a}_{ij} \equiv (2y_{ij}-1)\widehat{u}_i$ and

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} f(\widehat{a}_{ij}) \\ (2y_{ij}-1)f'(\widehat{a}_{ij}) \\ \frac{1}{2}f''(\widehat{a}_{ij}) \end{bmatrix}.$$

As in the probit case the quadratic approximation to $\log p(y_{ij}|u_i)$ based on Taylor expansion about $\widehat{u}_i$ is $\log \breve{p}(y_{ij}|u_i)$, where $\breve{p}(y_{ij}|u_i)$ is analogous to equation (2.22). By following the same logic it is easy to show

$$\eta^{\text{start}}_{p(y_{ij}|u_i)\to u_i} = \begin{bmatrix} \eta_0^{\text{start}} \\ (2y_{ij}-1)f'(\widehat{a}_{ij}) - f''(\widehat{a}_{ij})\widehat{u}_i \\ \frac{1}{2}f''(\widehat{a}_{ij}) \end{bmatrix}, \tag{4.9}$$

where

$$\eta_0^{\text{start}} = f(\widehat{a}_{ij}) - (2y_{ij}-1)f'(\widehat{a}_{ij})\widehat{u}_i + \frac{1}{2}(2y_{ij}-1)f''(\widehat{a}_{ij})\widehat{u}_i^2.$$

The comments at the end of Section 2.2.3 are also applicable for these starting values.

4.1.3 Evaluation of the estimates

Implementing the quadrature and EP approaches in the R computing environment, we now visually compare the accuracy of our likelihood approximation $\underset{\sim}{\ell}'(\sigma^2)$ to the exact likelihood surface $\ell(\sigma^2)$. Figure 4.1 plots the estimates of the likelihood surface for both methods. The data generated had $m = 30$ groups with $n = 5$ responses per group, and the true variance of $\sigma^2 = 0.19$. Using the quadrature method as exact,

the plot shows that although there are some discepencies on the tails of the likelihood surface, the approximate method follows the exact likelihood surface near the maximum. Additionally, the true value of σ^2 matches well with the maximum of the likelihood. With this in mind, we now turn to methodology for determination of its maximum with 95% confidence intervals.

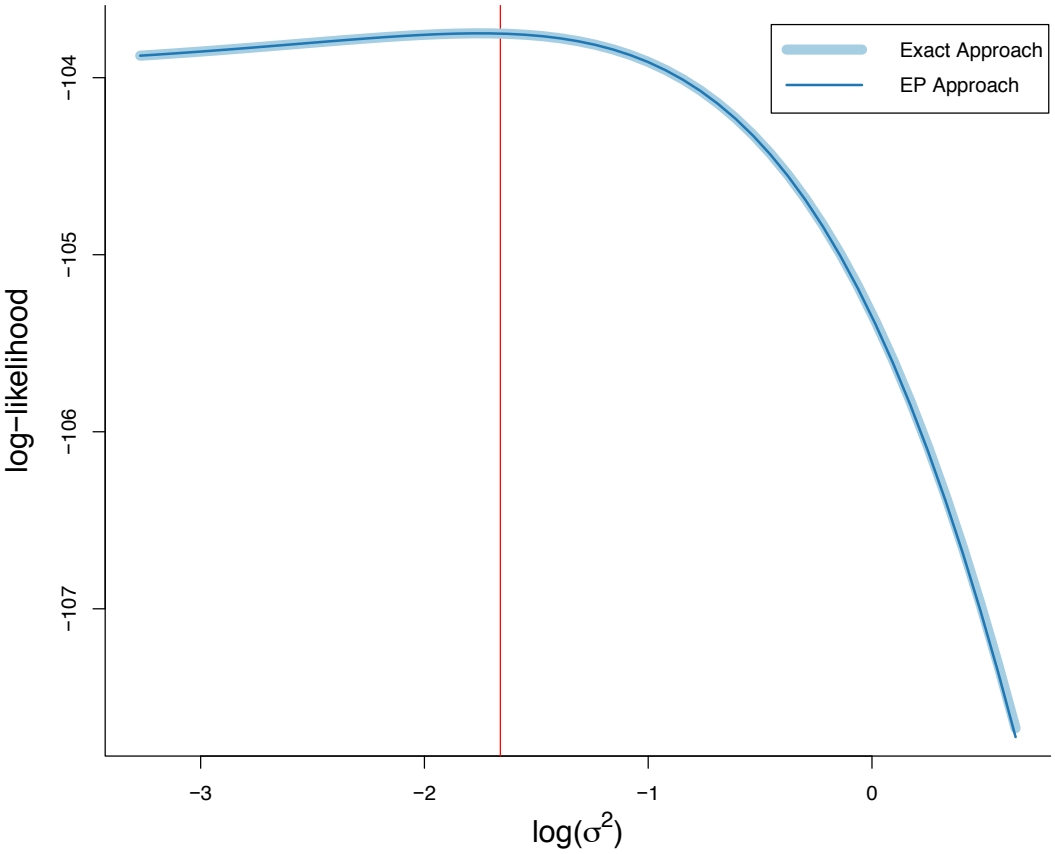


Figure 4.1: *A comparison plot of the log-likelihood surface over the parameter σ^2 for logistic models calculated using univariate quadrature and EP via the Monahan & Stefanski (1989)[46] approximation to the expit function. The true $\sigma^2 = 0.19$ and is represented on the log scale by the red line. The EP approximation is the dark blue line and the exact is the light blue line.*

4.1.4 Best predictor

Best prediction of u_i for logistic models via quadrature and EP is the same for the probit model presented in Section 2.5, although for the later we redefine

$$J_{si}(\sigma^2) \equiv \int_{-\infty}^{\infty} u_i^s \prod_{j=1}^{n} \text{expit}\big((2y_{ij}-1)u_i\big)(2\pi\sigma^2)^{-1/2}\exp(-u_i^2/2\sigma^2)du_i, \quad s=\{0,1,2\}.$$

The calculations are otherwise analogous.

4.2 General logistic mixed models

With knowledge from Section 4.1 on the random intercepts only model, we now extend our model to the more general case of GLMMs discussed in Section 1.7 that allow for any number of fixed and random effects. As such we respecify our model as

$$\begin{aligned} y_{ij}|\boldsymbol{u}_i \overset{\text{ind.}}{\sim} \text{Bernoulli}\Big(\text{expit}\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big), \quad \boldsymbol{u}_i \overset{\text{ind.}}{\sim} \boldsymbol{N}\big(\mathbf{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}\big), \\ 1 \leq i \leq m, \quad 1 \leq j \leq n_i, \end{aligned} \tag{4.10}$$

where the notation follows the general one level model presented in Section 1.8. The resulting log-likelihood can be written as

$$\ell\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}\right) = \sum_{i=1}^{m} \ell_i\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}\right),$$

where

$$\begin{aligned} \ell_i\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}\right) = &\log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \text{expit}\Big((2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big) \right\} \\ &\times |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(\frac{-1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{u}_i\right) d\boldsymbol{u}_i, \end{aligned}$$

and the best predictor of $\boldsymbol{u}_i$ is

$$\text{BP}(\boldsymbol{u}_i) \equiv \frac{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}_i \left\{ \prod_{j=1}^{n_i} \text{expit}\Big((2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big) \right\} \exp\left(-\tfrac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{u}_i\right) d\boldsymbol{u}_i}{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \text{expit}\Big((2y_{ij}-1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)\Big) \right\} \exp\left(-\tfrac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{u}_i\right) d\boldsymbol{u}_i}. \tag{4.11}$$

Quadrature for integrals greater than two dimensions becomes too computationally intensive for practical implementation, and as such we no longer consider it. The computations required for the multivariate logistic model via EP are mostly the same as the multivariate extension of the probit model, with the only changes being the starting values and projections onto the normalised and unnormalised inverse-logistic density functions.

The structure of this section follows that of the previous, first explaining the schematic of likelihood approximation using EP in Section 4.2.1 and the changes required from the probit model in Section 3.1. It concludes with the results of our simulation studies in Section 4.2.2. Details regarding computation of point estimates and confidence intervals are analogous to those presented in Section 3.2, whilst the same calculations in Section 3.3 can be used for best prediction of $\boldsymbol{u}_i$. We do not repeat either section again and instead refer readers to the previous work.

4.2.1 Expectation propagation likelihood approximation

The details of the EP approximation of the likelihood are largely analogous to those provided in Section 3.1 for the probit case. As before, the goal of the EP problem is to find the optimal natural parameters η_0, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ which minimise $\mathrm{KL}(f_{\text{input}} \parallel f_{\boldsymbol{UN}})$, where $f_{\boldsymbol{UN}}$ is defined by equation (3.4) and

$$f_{\text{input}}(\boldsymbol{x}) = \operatorname{expit}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\big) \exp\left(\big(\boldsymbol{\eta}_1^{\text{input}}\big)^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2^{\text{input}} \boldsymbol{x}\right), \tag{4.12}$$

where $\boldsymbol{\eta}_1^{\text{input}}$ is a $d \times 1$ vector and $\boldsymbol{H}_2^{\text{input}}$ is a $d \times d$ matrix, and c_0, $\boldsymbol{c}_1$ and $\boldsymbol{x}$ follow from the general probit case. As such, we seek an $\boldsymbol{\eta}^*$ such that

$$\begin{aligned}\int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \operatorname{expit}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\big) \exp\left\{ \begin{bmatrix} \boldsymbol{x} \\ \operatorname{vech}\big(\boldsymbol{x}\boldsymbol{x}^\top\big) \end{bmatrix}^\top \boldsymbol{\eta}^{\text{input}} \right\} d\boldsymbol{x} \\ = \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \operatorname{vech}\big(\boldsymbol{x}\boldsymbol{x}^\top\big) \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} d\boldsymbol{x}\end{aligned} \tag{4.13}$$

where $k \in \{0, 1, 2\}$. To obtain the required projection, we first obtain the optimal natural parameters $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ to project onto the multivariate normal family. For the generalised case, there are two possible solutions worth exploring to solve the otherwise

intractable integrals required for the projections, as per Result 18.

Result 18. *Given f_{input} follows the form of equation (4.12), the projection onto the multivariate normal family is given by*

$$proj_{\boldsymbol{N}}[f_{input}] = \exp\Big(\boldsymbol{T}(\boldsymbol{x})^\top \boldsymbol{\eta}_{-1}^* - A(\boldsymbol{\eta}_{-1}^*)\Big)h(\boldsymbol{x}),$$

where

$$\boldsymbol{\eta}_{-1}^* \equiv K_{logistic_{Quad}}(\boldsymbol{\eta}_{-1}^{input}; c_0, \boldsymbol{c}_1) \approx K_{logistic_{Approx}}(\boldsymbol{\eta}_{-1}^{input}; c_0, \boldsymbol{c}_1),$$

$$\boldsymbol{\eta}_{-1}^{input} \equiv \begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}_{-1}^{*} \equiv \begin{bmatrix} \boldsymbol{\eta}_1^{*} \\ \boldsymbol{\eta}_2^{*} \end{bmatrix},$$

$K_{logistic_{Quad}}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1\right)$ *is defined in Definition 17,*

$K_{logistic_{Approx}}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1\right)$ *is defined in Definition 18, and $\boldsymbol{T}(\boldsymbol{x})$ and $h(\boldsymbol{x})$ follow from Section 1.5.2.2.*

Definition 17. *For primary arguments $\boldsymbol{a}_1$ $(d \times 1)$ and $\boldsymbol{a}_2$ $\left(\frac{1}{2}d(d+1) \times 1\right)$ such that $vec^{-1}\Big(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\Big)$ is symmetric and positive definite, and auxiliary arguments $c_0 \in \mathbb{R}$ and $\boldsymbol{c}_1$ $(d \times 1)$, the function $K_{logistic_{Quad}} : H \to H$ is given by*

$$K_{logistic_{Quad}}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1\right) \equiv \begin{bmatrix} \mathbf{R}_5^\top(\boldsymbol{a}_1 + r_3\boldsymbol{c}_1) \\ \boldsymbol{D}_d^\top vec\left(\mathbf{R}_5^\top \boldsymbol{A}_2\right) \end{bmatrix}, \tag{4.14}$$

where

$$r_1 = \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\ \boldsymbol{c}_1, \quad r_2 = \big(2c_0 - \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1\big)r_1^{-1}, \quad r_3 = r_2 + 2\ \mathcal{C}_{b,1:0}(r_6, r_7)r_7,$$

$$r_4 = 2\big(\mathcal{C}_{b,1:0}(r_6, r_7)^2 - \mathcal{C}_{b,2:0}(r_6, r_7) - r_1\big)r_1^{-2}, \quad \mathbf{R}_5 = \big(\boldsymbol{A}_2 + r_4\boldsymbol{c}_1\boldsymbol{c}_1^\top\big)^{-1}\boldsymbol{A}_2,$$

$$r_6 = 1 - r_2, \quad r_7 = -r_1^{-1}, \quad \mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)},$$

$$\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)}, \quad b(x) = \log\left(1 + e^{-x}\right),$$

and $\mathcal{C}_b(k, r, q)$ is as per equation (4.24).

Definition 18. *For primary arguments* $\boldsymbol{a}_1$ $(d \times 1)$ *and* $\boldsymbol{a}_2$ $\left(\frac{1}{2}d(d+1) \times 1\right)$ *such that* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$ *is symmetric and positive definite, and auxiliary arguments* $c_0 \in \mathbb{R}$ *and* $\boldsymbol{c}_1$ $(d \times 1)$*, the function* $K_{logistic_{Approx}} : H \to H$ *is given by*

$$K_{logistic_{Approx}}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix}; c_0, \boldsymbol{c}_1\right) \equiv \begin{bmatrix}\mathbf{R}_5^\top(\boldsymbol{a}_1 + r_3\boldsymbol{c}_1)\\ \boldsymbol{D}_d^\top vec(\mathbf{R}_5^\top \boldsymbol{A}_2)\end{bmatrix}, \tag{4.15}$$

where

$$r_{1i} = \sqrt{2\left(2 - s_i^2 \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{c}_1\right)}, \quad r_{2i} = s_i\left(2c_0 - \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1\right)r_{1i}^{-1},$$

$$r_3 = \frac{2\sum_{i=1}^8 p_i s_i r_{1i}^{-1}\phi(r_{2i})}{\sum_{i=1}^8 p_i \Phi(r_{2i})}, \quad \tilde{r_3} = 4\sum_{i=1}^8\left(\frac{p_i s_i^2 r_{2i}}{\sum_{i=1}^8 p_i\Phi(r_{2i})}\right),$$

$$r_4 = \frac{1}{2}\left(r_3^2 + \tilde{r_3}\right), \quad \mathbf{R}_5 = \left(\boldsymbol{A}_2 + r_4\boldsymbol{c}_1^2\right)^{-1}\boldsymbol{A}_2.$$

The first solution using function $K_{\text{logistic}_{\text{Quad}}}$ as per Definition 17 requires solving each of the projections using univariate quadrature. Although this causes computational difficulties, it is still significantly faster than using multivariate quadrature. This approach involves utilising Lemma 2 to express the multivariate integrals required as univariate integrals. To account for the numerically unstable integrals, we express them using the form of $\mathcal{C}_b(p, q, r)$ as in Section 2.1 of Kim & Wand (2018).[32]

While optimising the likelihood surface via the EP algorithm, it is not unreasonable for the required integral to be calculated millions of times. As such, speed of integration for each projection becomes a vital issue. For univariate models, adaptive trapezoidal quadrature is a feasible option and provides high accuracy as well as an error criterion. However due to speed limitations its implementation is not practical for multivariate models. Although Gauss-Hermite quadrature with stored weights solves the computational speed issues, it is difficult to implement an error criterion to ensure a reasonable approximation of each integral is obtained. We leave this as an open problem and implement 100 point Gauss-Hermite quadrature for the simulations in this thesis. Details of the algebra required to arrive at Definition 17 are provided in Appendix 4.3.2.

The second approach denoted by $K_{\text{logistic}_{\text{Approx}}}$ as per Definition 18 involves using the piecewise approximation of the expit function by Monahan & Stefanski (1989)[46] in a similar manner to the previous random intercepts only section. By implementing this approximation, we are able to obtain closed form solutions to the integrals required. Details of the algebra required to arrive at Definition 18 are provided in Appendix 4.3.3.

Practical implementation of this approach in the higher dimensional setting proved to be troublesome with convergence issues arising in the EP algorithm. Figure 18.4.1 of Monahan & Stefanski (1989)[46] demonstrates the behaviour of the approximation compared to the expit function. An analysis showed that for some values arising in the required projections it is possible for the cumulative normal density function in the expit approximation to become very small, leading to underflow and causing convergence issues in the EP algorithm.

Using Result 18 we now show two methods to obtain the normalising natural parameter and subsequently, the projection onto unnormalised normal family.

4.2.1.1 Projection onto the unnormalised multivariate normal family

Obtaining the projection onto the unnormalised multivariate normal family involves similar algebra to the multivariate probit case. Recall the moment matching problem from equation (4.13) and Results 12 and 13. We require

$$C_f = \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} = \mathcal{M}_0(c_0, \boldsymbol{c}_1; \boldsymbol{\eta}),$$

where

$$\mathcal{M}_0(c_0, \boldsymbol{c}_1; \boldsymbol{\eta}) \equiv \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \text{expit}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\big) \exp\Big(\boldsymbol{\eta}_1^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2 \boldsymbol{x}\Big) d\boldsymbol{x},$$

and the right hand side is available by either quadrature or a piecewise approximation.

Using the quadrature method, we express

$$\mathcal{M}_0(c_0, \boldsymbol{c}_1; \boldsymbol{\eta}) = \mathcal{C}_b(0, r_6, r_7) Z_0 Z_1,$$

where $Z_0 = \exp\big((r_2/2)^2 r_1 + 0.5 \log(r_7/\pi)\big)$, $Z_1 = \exp\big(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi)\big)$ and r_6 and r_7 are given in Definition 17. Analogous to the argument given in the univariate case we then get

$$\begin{aligned} \eta_0^* &= \log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}(\boldsymbol{\eta}_1^*)^\top (\boldsymbol{H}_2^*)^{-1} \boldsymbol{\eta}_1^* \\ &\quad - \frac{1}{4}\big(\boldsymbol{\eta}_1^{\text{input}}\big)^\top \big(\boldsymbol{H}_2^{\text{input}}\big)^{-1} \boldsymbol{\eta}_1^{\text{input}} + \frac{1}{2}\log\big(|\boldsymbol{H}_2^*|/|\boldsymbol{H}_2^{\text{input}}|\big). \end{aligned}$$

To calculate η_0^* using the quadrature method we introduce Definition 19.

Definition 19. *Consider first, primary arguments* $\boldsymbol{a}_1$ *and* $\boldsymbol{b}_1$ *and auxiliary argument* $\boldsymbol{c}_1$ *where all three are* $d \times 1$*. Next consider arguments* $\boldsymbol{a}_2$ *and* $\boldsymbol{b}_2$ *which are* $\left(\frac{1}{2}d(d+1) \times 1\right)$ *such that both* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$ *and* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{b}_2\right)$ *are symmetric and positive definite. Finally note auxiliary argument* $c_0 \in \mathbb{R}$*. Then the function* $C_{logistic_{Quad}} : H \times H \to \mathbb{R}$ *is given by*

$$C_{logistic_{Quad}}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}; c_0, \boldsymbol{c}_1\right) \equiv \log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}\boldsymbol{b}_1^\top \boldsymbol{B}_2^{-1}\boldsymbol{b}_1 - \frac{1}{4}\boldsymbol{a}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1 + \frac{1}{2}\log\left(|\boldsymbol{B}_2|/|\boldsymbol{A}_2|\right),$$

where $\boldsymbol{A}_2 \equiv vec^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$*,* $\boldsymbol{B}_2 \equiv vec^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{b}_2\right)$*,* r_1*,* r_2*,* r_6 *and* r_7 *follow from Definition 17.*

Using the alternative using piecewise approximation,

$$\mathcal{M}_0 = \sum_{i=1}^{8} p_i \Phi(r_{2i})\ Z_1$$

where Z_1 is defined as in the quadrature approach and r_{2i} is given in Definition 18. It is then easy to show

$$\boldsymbol{\eta}_0^* = \log\left(\sum_{i=1}^{8} p_i \Phi(r_{2i})\right) + \frac{1}{4}(\boldsymbol{\eta}_1^*)^\top (\boldsymbol{H}_2^*)^{-1}\boldsymbol{\eta}_1^* - \frac{1}{4}\left(\boldsymbol{\eta}_1^{\text{input}}\right)^\top \left(\boldsymbol{H}_2^{\text{input}}\right)^{-1}\boldsymbol{\eta}_1^{\text{input}} + \frac{1}{2}\log\left(|\boldsymbol{H}_2^*|/|\boldsymbol{H}_2^{\text{input}}|\right)$$

and arrive at Definition 20.

Definition 20. *Consider first, primary arguments* $\boldsymbol{a}_1$ *and* $\boldsymbol{b}_1$ *and auxiliary argument* $\boldsymbol{c}_1$ *where all three are* $d \times 1$*. Next consider arguments* $\boldsymbol{a}_2$ *and* $\boldsymbol{b}_2$ *which are all* $\left(\frac{1}{2}d(d+1) \times 1\right)$ *such that both* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$ *and* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{b}_2\right)$ *are symmetric and positive definite. Finally note auxiliary argument* $c_0 \in \mathbb{R}$*. Then the function* $C_{logistic_{Approx}} : H \times H \to \mathbb{R}$ *is given by*

$$C_{logistic_{Approx}}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}; c_0, \boldsymbol{c}_1\right) \equiv \log\left(\sum_{i=1}^{8} p_i \Phi(r_{2i})\right) + \frac{1}{4}\boldsymbol{b}_1^\top \boldsymbol{B}_2^{-1}\boldsymbol{b}_1 - \frac{1}{4}\boldsymbol{a}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1 + \frac{1}{2}\log\left(|\boldsymbol{B}_2|/|\boldsymbol{A}_2|\right),$$

where $\boldsymbol{A}_2 \equiv vec^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$*,* $\boldsymbol{B}_2 \equiv vec^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{b}_2\right)$*,* r_1*,* r_2*,* r_6 *and* r_7 *follow from Definition 18.*

For either method, the projection onto the unormalised multivariate normal family can be obtained with Result 19.

Result 19. *For an unnormalised input function* f_{input} *of the form of equation (4.12),*

$$proj_{\boldsymbol{UN}}[f_{input}](\boldsymbol{x}) = \exp\left\{\begin{bmatrix}1\\ \boldsymbol{x}\\ vec\left(\boldsymbol{x}\boldsymbol{x}^\top\right)\end{bmatrix}^\top \begin{bmatrix}\eta_0^*\\ \boldsymbol{\eta}_1^*\\ \boldsymbol{\eta}_2^*\end{bmatrix}\right\},$$

where

$$\begin{bmatrix}\boldsymbol{\eta}_1^*\\ \boldsymbol{\eta}_2^*\end{bmatrix} = K_{logistic_{Quad}}\left(\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\end{bmatrix}; c_0, \boldsymbol{c}_1\right) \approx K_{logistic_{Approx}}\left(\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\end{bmatrix}; c_0, \boldsymbol{c}_1\right)$$

and

$$\eta_0^* = C_{logistic_{Quad}}\left(\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\eta}_1^*\\ \boldsymbol{\eta}_2^*\end{bmatrix}; c_0, \boldsymbol{c}_1\right) \approx C_{logistic_{Approx}}\left(\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\eta}_1^*\\ \boldsymbol{\eta}_2^*\end{bmatrix}; c_0, \boldsymbol{c}_1\right).$$

4.2.1.2 Message passing formulation

Only minor changes to Algorithm 6 are required to account for the logistic case. For the logistic model, the components of the sum of the log-likelihood function $\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ follow

$$\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left(\prod_{j=1}^{n_i} p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta}) \right) p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) d\boldsymbol{u}_i, \tag{4.16}$$

where

$$p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta}) \equiv \text{expit}\Big((2y_{ij} - 1)\big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big) \Big)$$

and

$$p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) \equiv |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \right)$$

are the conditional density functions of each response given its random effect and the density function of that random effect.

As the dependence structure of the product in equation (4.16) is the same as the probit model shown in Figure 3.1, we only need to update the messages depending on the conditional density functions of each response given its random effect. This involves updating messages $m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i)$ (equation (3.12))

$$m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \longleftarrow \frac{\text{proj}_{\text{UN}}\left[\text{expit}(c_{0_{ij}} + \boldsymbol{c}_{1_{ij}}^\top \boldsymbol{u}_i) \exp\left\{ \boldsymbol{u}_i^\top \boldsymbol{\eta}_1^{\otimes} + \big(\text{vech}(\boldsymbol{u}_i \boldsymbol{u}_i^\top)\big)^\top \boldsymbol{\eta}_2^{\otimes} \right\} \right]}{\exp\left\{ \boldsymbol{u}_i^\top \boldsymbol{\eta}_1^{\otimes} + \big(\text{vech}(\boldsymbol{u}_i \boldsymbol{u}_i^\top)\big)^\top \boldsymbol{\eta}_2^{\otimes} \right\}},$$

where $c_{0_{ij}} = (2y_{ij} - 1)\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{c}_{1_{ij}} = (2y_{ij} - 1)\boldsymbol{x}_{ij}^{\mathbf{R}}$. Following Section 3.1.2, we see that this task reduces down to adjusting the calculation of the optimal natural parameters $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ in equation (3.18), where the linear and quadratic coefficient updates in equation (3.19) given by K_{probit} are changed to either

$$\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)_{1:2} \leftarrow K_{\text{logistic}_{\text{Quad}}}\big(\boldsymbol{\eta}_{1:2}^{\otimes}; c_0, \boldsymbol{c}_{1_{ij}}\big) - \boldsymbol{\eta}_{1:2}^{\otimes}$$

or

$$\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)_{1:2} \leftarrow K_{\text{logistic}_{\text{Approx}}}\big(\boldsymbol{\eta}_{1:2}^{\otimes}; c_0, \boldsymbol{c}_{1_{ij}}\big) - \boldsymbol{\eta}_{1:2}^{\otimes}$$

(as per Definitions 17 and 18 respectively), and where the constant coefficient update in equation (3.20) given by C_{probit} is changed to either

$$\big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)_0 \leftarrow C_{\text{logistic}_{\text{Quad}}}\Big(\boldsymbol{\eta}_{1:2}^{\otimes}, \big(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\big)_{1:2} + \boldsymbol{\eta}_{1:2}^{\otimes}; c_0, \boldsymbol{c}_{1_{ij}}\Big)$$

or

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_0 \leftarrow C_{\text{logistic}_{\text{Approx}}}\left(\boldsymbol{\eta}_{1:2}^{\otimes}, \left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2} + \boldsymbol{\eta}_{1:2}^{\otimes}; c_0, \boldsymbol{c}_{1_{ij}}\right)$$

(as per Definitions 19 and 20 respectively).

Barring the aforementioned changes and the initialisation of $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ which now follows equation (4.17), the full algorithm for the approximation of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ follows Algorithm 6.

4.2.1.3 Starting values for the multivariate logistic case

Regardless of the projection methods, the EP algorithm for the logistic model uses the same starting values. We derive starting values using the same principles as in the probit case. Let

$$p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) = f(a_{ij}), \quad \text{where} \quad a_{ij} = (2y_{ij}-1)\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\boldsymbol{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right)$$

and $f(x)$ and its derivatives are as defined in equation (4.8). For a Laplace approximation of $\boldsymbol{u}_i$ denoted by $\widehat{\boldsymbol{u}}_i$ let

$$\widehat{a}_{ij} = (2y_{ij}-1)\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\boldsymbol{F}} + \widehat{\boldsymbol{u}}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right).$$

A Taylor series expansion of $f(a_{ij})$ evaluated at $\widehat{a}_{ij}$ leads to

$$\begin{aligned} f(a_{ij}) &= f(\widehat{a}_{ij}) + (2y_{ij}-1)f'(\widehat{a}_{ij})\left(\boldsymbol{u}_i^\top - \widehat{\boldsymbol{u}}_i^\top\right)\boldsymbol{x}_{ij}^{\mathbf{R}} + \frac{1}{2}f''(\widehat{a}_{ij})\left(\left(\boldsymbol{u}_i^\top - \widehat{\boldsymbol{u}}_i^\top\right)\boldsymbol{x}_{ij}^{\mathbf{R}}\right)^2 + \ldots \\ &= \begin{bmatrix} 1 \\ \boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i \\ \text{vech}\left(\left(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i\right)\left(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i\right)^\top\right) \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + \ldots, \end{aligned}$$

where

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} f(\widehat{a}_{ij}) \\ (2y_{ij}-1)f'(\widehat{a}_{ij})\boldsymbol{x}_{ij}^{\mathbf{R}} \\ \frac{1}{2}f''(\widehat{a}_{ij})\boldsymbol{D}_{d^{\mathbf{R}}}^\top \text{vec}\left(\boldsymbol{x}_{ij}^{\mathbf{R}}(\boldsymbol{x}_{ij}^{\mathbf{R}})^\top\right) \end{bmatrix}.$$

Following the same logic of the probit case, quadratic approximation to $\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ based on Taylor expansion about $\widehat{\boldsymbol{u}}_i$ is $\log \breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ as per equation (3.22) It is easy

to show

$$\boldsymbol{\eta}^{\text{start}}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})} = \begin{bmatrix} \eta_0^{\text{start}} \\ (2y_{ij}-1)f'(\widehat{a}_{ij})\boldsymbol{x}^{\mathbf{R}}_{ij} - f''(\widehat{a}_{ij})\boldsymbol{x}^{\mathbf{R}}_{ij}\big(\boldsymbol{x}^{\mathbf{R}}_{ij}\big)^\top \widehat{\boldsymbol{u}}_i \\ \frac{1}{2}f''(\widehat{a}_{ij})\boldsymbol{D}^\top_{d^{\mathbf{R}}}\text{vec}\Big(\boldsymbol{x}^{\mathbf{R}}_{ij}\big(\boldsymbol{x}^{\mathbf{R}}_{ij}\big)^\top\Big) \end{bmatrix}, \tag{4.17}$$

where

$$\eta_0^{\text{start}} = f(\widehat{a}_{ij}) - (2y_{ij}-1)f'(\widehat{a}_{ij})\boldsymbol{x}^{\mathbf{R}}_{ij}\widehat{\boldsymbol{u}}_i^\top + \frac{1}{2}f''(\widehat{a}_{ij})\widehat{\boldsymbol{u}}_{ij}^\top\boldsymbol{x}^{\mathbf{R}}_{ij}\big(\boldsymbol{x}^{\mathbf{R}}_{ij}\big)^\top\widehat{\boldsymbol{u}}_{ij}.$$

The final comments of Section 3.1.3 regarding obtaining Laplace approximations to $\boldsymbol{u}_i$ are also applicable for these starting values.

4.2.2 Simulation study

We now provide a simulation study analogous to that provided for the probit model, where we first compare a quadrature approach to EP and Laplace approximation approach for a random intercept model (i.e. $d^{\mathbf{R}} = 1$) and then test the empirical coverage of the EP approach on a random intercept model and slope model (i.e. $d^{\mathbf{R}} = 2$). For the simulation studies in this chapter we used the quadrature approach to obtaining the projections required for EP and did not assess the speed component.

4.2.2.1 Comparison of maximum likelihood estimates for univariate random effects

The first simulation study was repeated 1000 times, where datasets were simulated with true parameter values:

$$\beta_{\text{true}} = [0.37, 0.93]^\top \quad \text{and} \quad \sigma^2_{\text{true}} = -0.53. \tag{4.18}$$

There were 100 groups generated in the data with each group containing 10 measurements (i.e. $m = 100$, $n = 10$). The $\boldsymbol{x}^{\mathbf{F}}_{ij}$ and $\boldsymbol{x}^{\mathbf{R}}_{ij}$ vectors were of the form

$$\boldsymbol{x}^F_{ij} = [1, x_{1,ij}]^\top \quad \text{and} \quad \boldsymbol{x}^{\mathbf{R}}_{ij} = 1$$

where $x_{k,ij}$ was generated independently from a uniform distribution on the unit interval. The tolerance of error values were set to 10^{-5} for the EP scheme and the maximum number of iterations for optimisation was set to 1000. We compared our EP approach

to Laplace approximation and 100 point adaptive Gauss-Hermite quadrature. Both alternative approaches were implemented via the R function `glmer()` from the R package "lme4" (Bates, et al., 2018[5]).

The resulting estimates and 95% confidence intervals for each interpretable model parameter of the study are presented in Figure 4.3, where the numbers in the upper-right hand corner of each panel are the empirical coverage values based on all 1000 replicates. Only 20 randomly chosen replicates from each method are shown in the panels for ease of viewing, where Laplace approximation and EP are shown in black, super imposed adaptive Gauss-Hermite quadrature shown in grey. For the fixed slopes all three methods had the same empirical coverage of 96.3%. The methods also produced similar results for the fixed intercepts, although adaptive Gauss-Hermite quadrature performed the best, with 95.8%, ahead of the 95.7% of EP and 95.4% of Laplace approximation. However, for the variance parameter EP had 97.1% empirical coverage, notably higher Laplace approximation and adaptive Gauss-Hermite quadrature which produced empirical coverages of 93.8% and 94.5% respectively. It appears the results of the three methods are similar for settings of the data, however our method manages to marginally out perform the alternatives based on empirical coverage.

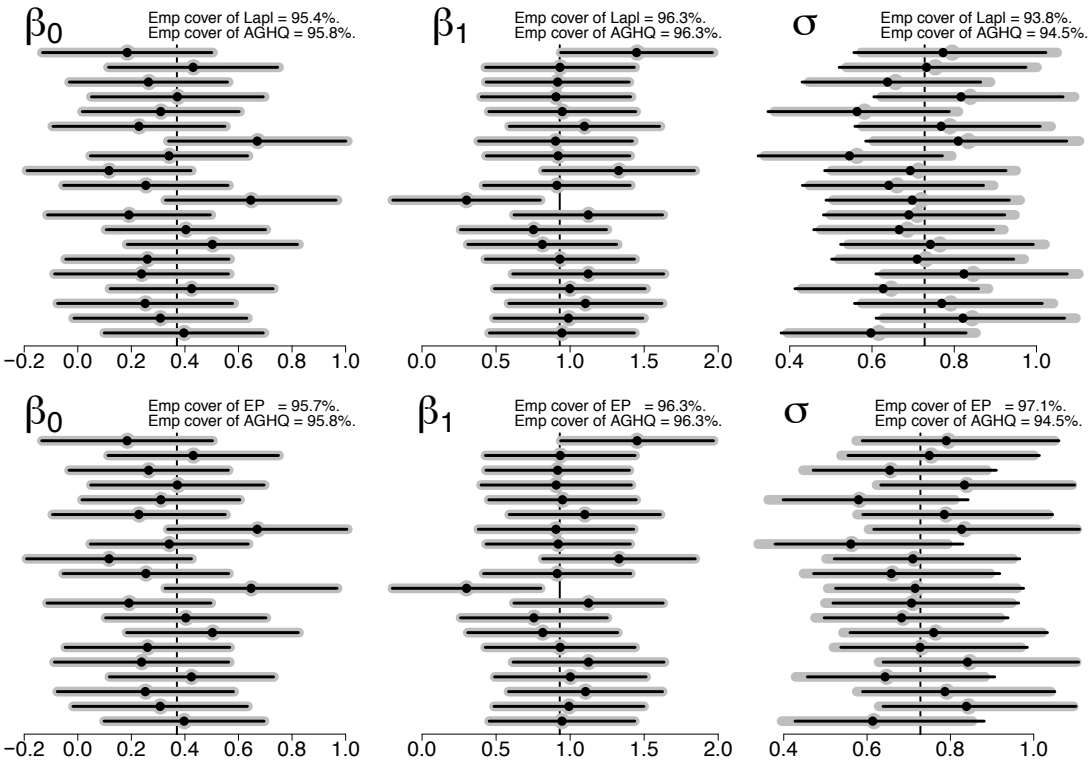


Figure 4.2: *Summary comparison between confidence interval coverage for the univariate model with true parameter values from equation (4.18) for models fit with adaptive Gauss-Hermite quadrature and Laplace approximation. The horizontal lines are the confidence intervals for 50 randomly chosen replications of the simulation study, the solid circular points indicate the corresponding point estimates, the vertical lines indicate true parameter values, the grey lines correspond to adaptive Gauss-Hermite quadrature, the black lines on the top row correspond to Laplace approximations and the black lines on the bottom row correspond to EP. The percentage given in the top right-hand corner of each panel is the empirical coverage over all* 1000 *replications.*

4.2.2.2 Maximum likelihood estimates for bivariate random effects

The simulation study was repeated 1000 times, where datasets were simulated according to equation (4.10) with arbitrarily chosen true parameter values

$$\boldsymbol{\beta}_{\text{true}} = [0.37, 0.93, -0.46, 0.08, -1.34, 1.09]^\top \quad \text{and} \quad \boldsymbol{\Sigma}_{\text{true}} = \begin{bmatrix} 0.73 & -0.52 \\ -0.52 & 0.95 \end{bmatrix}. \tag{4.19}$$

We note positive and negative correlation parameters are equally likely, but that neither has a large affect on the model from a mathematical perspective. Additionally, we note that correlations close to 1 pose numerical issues, and in reality we recommend

careful variable selection to avoid them. The number of groups in the data were fixed at $m = 1000$ with six measurements per group. The $\boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{x}_{ij}^{\mathbf{R}}$ vectors were of the form

$$\boldsymbol{x}_{ij}^{\mathbf{F}} = [1, x_{1_{ij}}, x_{2_{ij}}, x_{3_{ij}}, x_{4_{ij}}, x_{5_{ij}}]^\top \quad \text{and} \quad \boldsymbol{x}_{ij}^{\mathbf{R}} = [1, x_{1_{ij}}, x_{2_{ij}}]^\top$$

where $x_{k_{ij}}$ was generated independently from a uniform distribution on the unit interval. The tolerance of error values were set to 10^{-5} for the EP scheme and the maximum number of iterations for optimisation was set to 1000. The resulting estimates and 95% confidence intervals for each interpretable model parameter of the study are presented in Figure 4.3, where the numbers in the upper-right hand corner of each panel are the empirical coverage values based on all 1000 replicates. Only 50 randomly chosen replicates are shown in the each of the panels for ease of viewing. Across all 12 parameters estimated the empirical coverage showed excellent accuracy. While the empirical coverage for 94.3% and 94.6% was below 95% the estimates for the other fixed effects and variance parameters were as promised. The correlation parameter had 96.3% coverage which is slightly conservative.

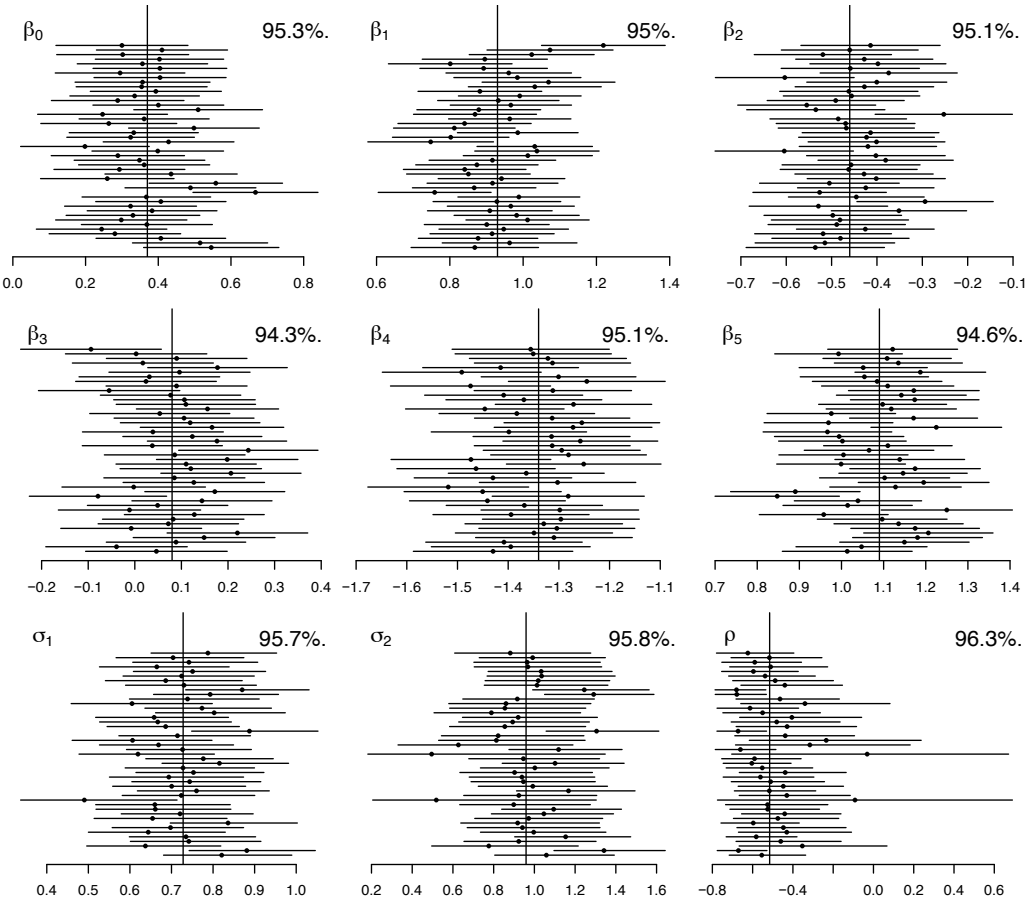


Figure 4.3: *Summary of confidence interval coverage for the simulation study with true parameter values from equation (4.19). The horizontal lines are the EP-based confidence intervals for 50 randomly chosen replications of the simulation study, the solid circular points indicate the corresponding point estimates and the vertical lines indicate true parameter values. The percentage given in the top right-hand corner of each panel is the empirical coverage over all* 1000 *replications.*

4.3 Appendix

4.3.1 Proof of Result 16

Using matrix notation and simple algebraic manipulations we can rewrite the target density of equation (4.2) as

$$\mathcal{M}_k = \sum_{i=1}^{8} p_i \int_{-\infty}^{\infty} x^k \Phi(s_i c_0 + s_i c_1 x) \exp\left\{ \begin{bmatrix} x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta} - A(\boldsymbol{\eta}) \right\} dx \exp\big(A(\boldsymbol{\eta})\big).$$

Taking the inverse of the natural parameters as per equation (1.5) and implementing a variable change where $u = (x-\mu)\sigma^{-1}$, such that $x = \mu + \sigma u$ and $dx = \sigma du$,

$$\mathcal{M}_k = \sum_{i=1}^{8} p_i \int_{-\infty}^{\infty} (\mu + \sigma u)^k \Phi\big(s_i(c_0 + c_1\mu) + s_i c_1 \sigma u\big)\phi(u) du Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta})\big)(2\pi)^{1/2}\sigma$. Thus, using Lemma 1,

$$Z_1^{-1}\mathcal{M}_0 = \sum_{i=1}^{8} p_i \Phi(r_{2i}),$$

$$Z_1^{-1}\mathcal{M}_1 = \sum_{i=1}^{8} p_i \big(\mu\Phi(r_{2i}) + 2s_i c_1 \sigma^2 r_{1i}^{-1}\phi(r_{2i})\big),$$

$$Z_1^{-1}\mathcal{M}_2 = \sum_{i=1}^{8} p_i \big(\mu^2\Phi(r_{2i}) + 4s_i c_1 \mu\sigma^2 r_{1i}^{-1}\phi(r_{2i}) + \sigma^2\Phi(r_{2i}) - 4s_i^2 c_1^2 \sigma^4 r_{2i} r_{1i}^{-2}\phi(r_{2i})\big),$$

where $r_{2i} = 2s_i(c_0 + c_1\mu)r_{1i}^{-1}$ and $r_{1i} = 2\big((s_i c_1\sigma)^2 + 1\big)^{-1/2}$. We can then show that the optimal mean and variance parameters are respectively

$$\mu^* = E(x) = \mu + c_1\sigma^2 r_3$$

and

$$(\sigma^*)^2 = E(x^2) - \big(E(x)\big)^2 = \sigma^2 - 2c_1^2\sigma^4 r_4,$$

where

$$r_3 = \frac{2\sum_{i=1}^{8} s_i p_i \phi(r_{2i}) r_{1i}^{-1}}{\sum_{i=1}^{8} p_i \Phi(r_{2i})},$$

$$r_4 = \frac{1}{2}(\tilde{r_3} + r_3^2) \quad \text{and} \quad \tilde{r_3} = \frac{4\sum_{i=1}^{8} s_i^2 p_i r_{2i} \phi(r_{2i}) r_{1i}^{-2}}{\sum_{i=1}^{8} p_i \Phi(r_{2i})}.$$

Using matrix notation, converting back to natural parameter form and implementing additional simplification, we arrive at the required result.

4.3.2 Proof of Definition 17

Using simple algebraic manipulations based on Lemma 2, we arrive at Lemma 4:

Lemma 4. *For integrals of the forms listed below, the corresponding solutions exist:*

$$\int_{\mathbb{R}^d} expit\big(a+\boldsymbol{b}^\top\boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} = \int_{-\infty}^{\infty} expit\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx, \tag{4.20}$$

$$\int_{\mathbb{R}^d} \boldsymbol{x}\, expit\big(a+\boldsymbol{b}^\top\boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}\int_{-\infty}^{\infty} x\, expit\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx, \tag{4.21}$$

$$\begin{aligned}\int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^\top expit\big(a+\boldsymbol{b}^\top\boldsymbol{x}\big)\phi_{\boldsymbol{I}}(\boldsymbol{x})d\boldsymbol{x} &= I_d\int_{-\infty}^{\infty} expit\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx \\ &+ \frac{\boldsymbol{b}\boldsymbol{b}^\top}{\boldsymbol{b}^\top\boldsymbol{b}}\Big(\int_{-\infty}^{\infty} x^2\, expit\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx - \int_{-\infty}^{\infty} expit\big(a+\|\boldsymbol{b}\|x\big)\phi(x)dx\Big),\end{aligned} \tag{4.22}$$

where $a\in\mathbb{R}$ and $\boldsymbol{b}$ is a $d\times 1$ vector.

We wish to obtain the projection of an input function following the form of equation (4.12) onto the multivariate normal family. Note in the interest of brevity we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. For the general case for all $\otimes k$

$$\mathcal{M}_k \equiv \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k}\text{expit}\big(c_0+\boldsymbol{c}_1^\top\boldsymbol{x}\big)\exp\Big(\boldsymbol{\eta}_1^\top\boldsymbol{x}+\boldsymbol{x}^\top\boldsymbol{H}_2\boldsymbol{x}\Big)d\boldsymbol{x}. \tag{4.23}$$

Using the matrix notation defined in equation (1.14) we can write

$$\begin{aligned}\mathcal{M}_k = \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k}\text{expit}\big(c_0+\boldsymbol{c}_1^\top\boldsymbol{x}\big)(2\pi)^{-d/2}\exp\left\{\begin{bmatrix}\boldsymbol{x}\\ \text{vech}\left(\boldsymbol{x}\boldsymbol{x}^\top\right)\end{bmatrix}^\top\boldsymbol{\eta}-A(\boldsymbol{\eta})\right\}d\boldsymbol{x} \\ \times(2\pi)^{d/2}\exp\big(A(\boldsymbol{\eta})\big).\end{aligned}$$

Taking the inverse of the natural parameters as per equation (1.14) and implementing the change of variables $\boldsymbol{x}=\boldsymbol{\mu}+\boldsymbol{\Sigma}^{1/2}\boldsymbol{u}$ leads to

$$\mathcal{M}_k = \int_{\mathbb{R}^d}\big(\boldsymbol{\mu}+\boldsymbol{\Sigma}^{1/2}\boldsymbol{u}\big)^{\otimes k}\text{expit}\Big(c_0+\boldsymbol{c}_1^\top\boldsymbol{\mu}+\big(\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\big)^\top\boldsymbol{u}\Big)\phi_I(\boldsymbol{u})d\boldsymbol{u}Z_1,$$

where $Z_1=\exp\big(A(\boldsymbol{\eta})+(d/2)\log(2\pi)\big)$. Using equations (4.20) - (4.22), we can now obtain each of the moments using only univariate quadrature. Note that the integrals required are renowned for being numerically unstable. For stability, we rewrite the integrals in the forms of those presented by Kim & Wand (2017),[32]

$$\mathcal{C}_b(k,r,q) = \int_{-\infty}^{\infty} x^k\exp\big(rx-qx^2-b(x)\big)dx, \tag{4.24}$$

where $b(x)=\log\big(1+\exp(x)\big)$. Using simple algebraic manipulations it is easy to show

$$\begin{aligned}\int_{-\infty}^{\infty} u^k\text{expit}\big(c_0+\boldsymbol{c}_1^\top\boldsymbol{\mu}+\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\|u\big)\phi(u)du \\ = \int_{-\infty}^{\infty}\big((r_2+2r_7x)(2r_7)^{-1/2}\big)^k\exp\big(r_6x-r_7x^2-b(x)\big)dxZ_0,\end{aligned}$$

where

$$r_1 = -2\boldsymbol{c}_1^\top \boldsymbol{\Sigma} \boldsymbol{c}_1, \quad r_2 = 2\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu}\big) r_1^{-1}, \quad r_6 = 1 - r_2, \quad r_7 = -r_1^{-1}$$

and

$$Z_0 = \exp\big((r_2/2)^2 r_1 + (1/2)\log(r_7/\pi)\big).$$

It follows,

$$\begin{aligned} Z_0^{-1} \int_{-\infty}^{\infty} \text{expit}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\|\ u\big)\phi(u)\ du &= \mathcal{C}_b(0, r_6, r_7), \\ Z_0^{-1} \int_{-\infty}^{\infty} u\,\text{expit}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\| u\big)\phi(u)\ du & \\ = \big(r_2\ \mathcal{C}_b(0, r_6, r_7) + 2r_7 \mathcal{C}_b(1, r_6, r_7)\big)(2r_7)^{-1/2}, & \end{aligned}$$

and

$$\begin{aligned} Z_0^{-1} \int_{-\infty}^{\infty} u^2 \text{expit}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\|\ u\big)\phi(u)\ du & \\ = 2\big(r_2\ \mathcal{C}_b(1, r_6, r_7) + r_7 \mathcal{C}_b(2, r_6, r_7)\big) - (r_2^2 r_1/2)\mathcal{C}_b(0, r_6, r_7). & \end{aligned}$$

It is then easy to find each of the kth moments

$$\begin{aligned} Z_0^{-1} Z_1^{-1} \mathcal{M}_0 &= \mathcal{C}_b(0, r_6, r_7), \\ Z_0^{-1} Z_1^{-1} \mathcal{M}_1 &= \boldsymbol{\mu}\ \mathcal{C}_b(0, r_6, r_7) + \boldsymbol{\Sigma}\boldsymbol{c}_1\big(2r_7 \mathcal{C}_b(1, r_6, r_7) + r_2 \mathcal{C}_b(0, r_6, r_7)\big), \\ Z_0^{-1} Z_1^{-1} \mathcal{M}_2 &= (\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma})\ \mathcal{C}_b(0, r_6, r_7) \\ &\quad + (\boldsymbol{\mu}\boldsymbol{c}_1^\top \boldsymbol{\Sigma} + \boldsymbol{\Sigma}\boldsymbol{c}_1 \boldsymbol{\mu}^\top)\ \big(r_2\ \mathcal{C}_b(0, r_6, r_7) + 2r_7 \mathcal{C}_b(1, r_6, r_7)\big) \\ &\quad + \boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top \boldsymbol{\Sigma}\ \big(4r_7^2 \mathcal{C}_b(2, r_6, r_7) + 4r_7 r_2\ \mathcal{C}_b(1, r_6, r_7) + (r_2^2 - 2r_7)\ \mathcal{C}_b(0, r_6, r_7)\big). \end{aligned}$$

Thus the optimal mean parameter for the projection is

$$\boldsymbol{\mu}^* = E(\boldsymbol{x}) = \frac{\mathcal{M}_1}{\mathcal{M}_0} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{c}_1\big(r_2 + 2r_7 \mathcal{C}_{b,1:0}(r_6, r_7)\big),$$

and the optimal variance parameter is

$$\begin{aligned} \boldsymbol{\Sigma}^* =& E\big(\boldsymbol{x}\boldsymbol{x}^\top\big) - E(\boldsymbol{x})E(\boldsymbol{x})^\top \\ =& \frac{\mathcal{M}_2}{\mathcal{M}_0} - \frac{\mathcal{M}_1}{\mathcal{M}_0}\left(\frac{\mathcal{M}_1}{\mathcal{M}_0}\right)^\top \\ =& \boldsymbol{\Sigma} + (2r_7)^2 \boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top \boldsymbol{\Sigma}\ \big(\mathcal{C}_{b,2:0}(r_6, r_7) - \mathcal{C}_{b,1:0}(r_6, r_7)^2 + r_1/2\big), \end{aligned}$$

where $\mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)}$ and $\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)}$. By converting back to natural optimal and input parameters we arrive at Definition 17.

4.3.3 Proof of Definition 18

We wish to obtain the projection of an input function following the form of equation (4.12) onto the multivariate normal family. Note in the interest of brevity we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. For the general case for all $\otimes k$. As before, we can obtain closed form solutions to the equivalent of the zeroth, first and second moments. Using the $\boldsymbol{x}^{\otimes k}$ notation as described in equation (1.2), we can write

$$\begin{aligned}\mathcal{M}_k &\equiv \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \text{expit}_{\text{MS}}\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\big) \exp\Big(\boldsymbol{\eta}_1^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2 \boldsymbol{x}\Big) d\boldsymbol{x} \\ &\equiv \sum_{i=1}^{8} p_i \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \Phi\big(s_i c_0 + s_i \boldsymbol{c}_1^\top \boldsymbol{x}\big) \exp\Big(\boldsymbol{\eta}_1^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2 \boldsymbol{x}\Big) d\boldsymbol{x},\end{aligned}$$

where $\text{expit}_{\text{MS}}(x)$ is given in equation (4.4). Using the matrix notation defined in equation (1.14) we can write

$$\begin{aligned}\mathcal{M}_k = &\sum_{i=1}^{8} p_i \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \Phi\big(s_i c_0 + s_i \boldsymbol{c}_1^\top \boldsymbol{x}\big) (2\pi)^{-d/2} \exp\left\{ \begin{bmatrix} \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta} - A(\boldsymbol{\eta}) \right\} d\boldsymbol{x} \\ &\times (2\pi)^{d/2} \exp\big(A(\boldsymbol{\eta})\big).\end{aligned}$$

Using the inverse of the natural parameters in equation (1.2) and implementing the change of variable $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{u}$

$$\mathcal{M}_k = \sum_{i=1}^{8} p_i \int_{\mathbb{R}^d} (\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{u})^{\otimes k} \Phi\Big(s_i\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu}\big) + \big(s_i \boldsymbol{\Sigma}^{1/2} \boldsymbol{c}_1\big)^\top \boldsymbol{u}\Big) \phi_I(\boldsymbol{u}) d\boldsymbol{u} Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi)\big)$. Then using Lemma 3 we can now obtain exact forms for each of the moments and thus calculate the optimal parameters for the projection. It is then easy to show each kth moment,

$$\begin{aligned}Z_1^{-1}\mathcal{M}_0 =& \sum_{i=1}^{8} p_i \Phi(r_{2i}), \\ Z_1^{-1}\mathcal{M}_1 =& \boldsymbol{\mu} \sum_{i=1}^{8} p_i \Phi(r_{2i}) + \boldsymbol{\Sigma}\boldsymbol{c}_1 \sum_{i=1}^{8} \big(2p_i s_i r_{1i}^{-1}\big)\phi(r_{2i}), \\ Z_1^{-1}\mathcal{M}_2 =& \big(\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma}\big) \sum_{i=1}^{8} p_i \Phi\left(r_{2i}\right) + \big(\boldsymbol{\mu}\boldsymbol{c}_1^\top \boldsymbol{\Sigma} + \boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{\mu}^\top\big) \sum_{i=1}^{8} \big(2p_i s_i r_{2i}^{-1}\big)\phi\left(r_{2i}\right) \\ &- \boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top\boldsymbol{\Sigma} \sum_{i=1}^{8} \big(4p_i s_i^2 r_{2i} r_{1i}^{-2}\big)\phi\left(r_{2i}\right),\end{aligned}$$

where $r_{1i} = 2\big(s_i^2 \boldsymbol{c}_1^\top \boldsymbol{\Sigma}\boldsymbol{c}_1 + 1\big)^{1/2}$ and $r_{2i} = 2s_i\big(c_0 + \boldsymbol{c}_1^\top\boldsymbol{\mu}\big)r_{1i}^{-1}$. The optimal mean and variance parameters for the projection follow and are respectively

$$\boldsymbol{\mu}^* = E(\boldsymbol{x}) = \frac{\mathcal{M}_1}{\mathcal{M}_0} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{c}_1 r_3$$

and

$$\boldsymbol{\Sigma}^* = E\big(\boldsymbol{x}\boldsymbol{x}^\top\big) - E(\boldsymbol{x})E(\boldsymbol{x})^\top = \frac{\mathcal{M}_2}{\mathcal{M}_0} - \frac{\mathcal{M}_1}{\mathcal{M}_0}\left(\frac{\mathcal{M}_1}{\mathcal{M}_0}\right)^\top = \boldsymbol{\Sigma} - 2r_4\boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top\boldsymbol{\Sigma},$$

where

$$r_4 = \frac{1}{2}\big(\tilde{r_3} + r_3^2\big), \quad r_3 = \frac{\sum_{i=1}^8 \big(2p_i s_i r_{1i}^{-1}\big)\phi(r_{2i})}{\sum_{i=1}^8 p_i\Phi(r_{2i})} \quad \text{and} \quad \tilde{r_3} = \frac{\sum_{i=1}^8 \big(4p_i s_i^2 r_{2i} r_{1i}^{-2}\big)\phi(r_{2i})}{\sum_{i=1}^8 p_i\Phi(r_{2i})}.$$

By converting both the input and optimal parameters back to natural parameters we arrive at the desired result.

Chapter 5

Expectation propagation for one level count response mixed models

Thus far, the novel methodology presented in this thesis has concerned datasets with binary response variables. We now consider Poisson and negative binomial models for datasets with count response variables. As the projections required for EP cannot be solved via closed form solutions, we implement the quadrature approach used for the logistic models. The message passing framework implemented in the binary case remains mostly unchanged, with only minor alterations required. Additionally, care must be taken during optimisation of the shape parameter for the negative binomial model. We negate any repeated details of work presented in previous chapters and refer readers back where appropriate.

This chapter is broken into four main sections. Section 5.1 explains the simplest random intercepts only Poisson model, while Section 5.2 discusses it's extension to a general model for any number of fixed and random effects. Section 5.3 explains a simple model for inference on the shape parameter of the negative binomial model, while Section 5.5 explains its extension to a general model for any number of fixed and random effects.

5.1 The simplest Poisson mixed model

We consider a random intercepts only model first for simplicity with a balanced dataset, where all m groups have the same n number of observations in them. Note for the random intercept only model the EP approach affords no benefit over directly computing the integrals via quadrature, acting as a test bed for the general case which follows. For observed values of

$$y_{ij}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n,$$

where $y_{ij} \in \mathbb{Z}_{\geq 0}$, the form of the model is

$$y_{ij}|u_i \overset{\text{iid}}{\sim} \text{Poisson}\big(\exp(u_i); y_{ij}\big), \qquad u_i \overset{\text{iid}}{\sim} N(0, \sigma^2),$$

where $\text{Poisson}(\lambda; x)$ is given by equation (1.11) and u_i is a scalar unobserved latent variable. We wish to find the maximiser of $\ell(\sigma^2)$ denoted by $\widehat{\sigma^2}$ and the best predictor of the random effects $\text{BP}(u_i)$.

The log-likelihood of the simplified model can be written as

$$\ell(\sigma^2) = \sum_{i=1}^{m} \ell_i(\sigma^2),$$

where

$$\ell_i(\sigma^2) \equiv \log \int_{-\infty}^{\infty} \prod_{j=1}^{n} \text{Poisson}\big(\exp(u_i); y_{ij}\big)(2\pi\sigma^2)^{-1/2} \exp(-u_i^2/2\sigma^2) du_i, \tag{5.1}$$

and the maximum likelihood estimate of σ^2 is given by

$$\widehat{\sigma^2} = \underset{\sigma^2}{\text{argmax}}\ \ell(\sigma^2).$$

The best predictor of the random effect is given by

$$\text{BP}(u_i) = \frac{\int_{-\infty}^{\infty} u_i \prod_{j=1}^{n} \text{Poisson}\big(\exp(u_i); y_{ij}\big) \exp(-u_i^2/2\sigma^2) du_i}{\int_{-\infty}^{\infty} \prod_{j=1}^{n} \text{Poisson}\big(\exp(u_i); y_{ij}\big) \exp(-u_i^2/2\sigma^2)\ du_i}.$$

Using EP, we develop an approximation and sum each $\ell_i(\sigma^2)$ to obtain the full log-likelihood and best-predictors. We compare it to a traditional quadrature approach and show both methods provide reasonable and similar estimates.

The following subsections copy the structure of Chapter 4. In Subsection 5.1.1 we

provide details of the quadrature approach to estimating the likelihood surface, then explain our novel method using EP in Subsection 5.1.2. We compare the likelihood surface of both methods in Subsection 5.1.3, and outline computation of best predictors in Subsection 5.1.4. Point estimate and confidence interval calculation is conducted analogously to Section 2.4.1 and as such we do not repeat it.

5.1.1 Traditional quadrature likelihood approximation

Implementation of adaptive quadrature via Gauss-Kronrod follows the probit and expit cases explained in Sections 2.1 and 4.1.1 respectively, where we utilise the R function `integrate()` in the "stats" package (R Core Team, 2019[56]). Denoting $y_{i\cdot} \equiv \sum_{j=1}^{n} y_{ij}$, we suggest that for numerical stability each $\ell_i(\sigma^2)$ is calculated as

$$\frac{1}{2}\log(2\pi\sigma^2) + \ell_i(\sigma^2) = h(u_{0i}) + \log \int_{-\infty}^{\infty} \exp\big(h_i(u) - h_i(u_{0i})\big)du,$$

where

$$h_i(u) \equiv y_{i\cdot}u - n\exp(u) - \frac{u^2}{2\sigma^2},$$
$$h_i'(u) = y_{i\cdot} - n\exp(u) - \frac{u}{\sigma^2}$$

and u_{0i} is the root of h_i', which we recommend finding by a bisection search, where the starting values are selected -1 and 1 to be for the lower and upper bounds respectively.

5.1.2 Expectation propagation likelihood approximation

We now consider an EP approach to the approximate likelihood (denoted by $\underset{\sim}{\ell}(\sigma^2)$) as in Section 2.2. The EP approximation requires an unnormalised normal density function selected by minimising the KL-divergence criterion, to replace each

$$\text{Poisson}\big(\exp(u_i); y_{ij}\big), \quad 1 \le j \le n$$

in equation (5.1). For the case of Poisson GLMMs, the required projections onto an unnormalised univariate normal distribution can be written as

$$f_{\text{input}}(x) = \text{Poisson}\big(\exp(c_0 + c_1 x); c_2\big)\exp\big(\eta_1^{\text{input}}x + \eta_2^{\text{input}}x^2\big), \tag{5.2}$$

where $\eta_1^{\text{input}} \in \mathbb{R}$, $\eta_2^{\text{input}} < 0$, $c_0 = 0$, $c_1 = 1$, $c_2 = y_{ij}$ and $x = u_i$. Subsequently, the integrand is proportional to a product of univariate normal density functions. As such, we seek $\boldsymbol{\eta}^*$ such that

$$\int_{-\infty}^{\infty} x^k \text{Poisson}\big(\exp(c_0 + c_1 x); c_2\big) \exp\big(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\big) dx$$
$$= \int_{-\infty}^{\infty} x^k \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} dx. \qquad (5.3)$$

To obtain the required projection, we first obtain the optimal natural parameters η_1^* and η_2^* to project onto the normal family. Since the integral arising in the required projection does not have a closed form solution, we use univariate quadrature to obtain it. As before, caution must be exercised when calculating the integrals since they are prone to be numerically unstable. As such, each integral is calculated using the form of $\mathcal{C}_b(p, q, r)$ as in Section 2.1 of Kim & Wand (2018)[32] for numerical stability. By doing so we arrive at Result 20.

Result 20. *Given f_{input} follows the form of equation (5.2), the projection onto the univariate normal family is given by*

$$proj_N[f_{input}] = \exp\Big(\boldsymbol{T}(x)^\top \boldsymbol{\eta}_{-1}^* - A(\boldsymbol{\eta}_{-1}^*)\Big) h(x),$$

where

$$\boldsymbol{\eta}_{-1}^* = k_{Poisson}\big(\boldsymbol{\eta}_{-1}^{input}; c_0, c_1, c_2\big), \qquad \boldsymbol{\eta}_{-1}^{input} \equiv \begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}_{-1}^* \equiv \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix},$$

$k_{poisson}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1, c_2 \right)$ *is defined in Definition 21 and $\boldsymbol{T}(x)$ and $h(x)$ follow from equation (1.5.2.1).*

Definition 21. *For primary scalar arguments $a_1 \in \mathbb{R}$ and $a_2 < 0$, and auxiliary scalar arguments $c_0, c_1 \in \mathbb{R}$ and $c_2 \in \mathbb{Z}_{\geq 0}$, the function $k_{Poisson} : H \to H$ is given by*

$$k_{Poisson}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1, c_2\right) \equiv \begin{bmatrix} r_5(a_1 + r_3 c_1) \\ r_5 a_2 \end{bmatrix}, \tag{5.4}$$

where

$$r_1 = c_1^2 a_2^{-1}, \quad r_2 = \big(c_1 a_2^{-1} a_1 - 2c_0\big) r_7, \quad r_3 = r_2 + 2\mathcal{C}_{b,1:0}(r_6, r_7) r_7,$$

$$r_4 = 2\big(\mathcal{C}_{b,1:0}(r_6, r_7)^2 - \mathcal{C}_{b,2:0}(r_6, r_7) - r_1\big) r_7^2, \quad r_5 = \big(a_2 + r_4 c_1^2\big)^{-1} a_2,$$

$$r_6 = 1 - r_2, \quad r_7 = -(r_1)^{-1}, \quad \mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)},$$

$$\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)} \quad \textit{and} \quad b(x) = \exp(x).$$

A proof of Definition 21 is provided in Appendix 5.6.1. Using Result 20 we now obtain the normalising natural parameter η_0^* to find the projection onto unnormalised normal family.

5.1.2.1 Projection onto the unnormalised normal family

Recall the moment matching problem described by equation (2.6) and Result 4. Then the normalising factor can be shown to be

$$C_f = \int_{\mathbb{R}^d} f_{\text{input}}(x) dx = \mathcal{C}_b(0, r_6, r_7) Z_0 Z_1,$$

where $Z_0 = \exp\left((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\right)$, $Z_1 = \exp\left(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi) - \log\Gamma(c_2 + 1)\right)$ and r_6 and r_7 are given in Definition 21. By Result 4

$$\begin{aligned}\boldsymbol{\eta}_0^* = &\log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}(\boldsymbol{\eta}_1^*)^2/\boldsymbol{\eta}_2^* \\ &- \frac{1}{4}\big(\boldsymbol{\eta}_1^{\text{input}}\big)^2/\boldsymbol{\eta}_2^{\text{input}} + \frac{1}{2}\log\left(\boldsymbol{\eta}_2^*/\boldsymbol{\eta}_2^{\text{input}}\right) - \log\Gamma(c_2 + 1).\end{aligned}$$

Thus to obtain η_0^* we introduce Definition 22.

Definition 22. *For primary scalar arguments $a_1, b_1 \in \mathbb{R}$ and $a_2, b_2 < 0$ and auxiliary scalar arguments $c_0, c_1 \in \mathbb{R}$ and $c_2 \in \mathbb{Z}_{\geq 0}$ the function $c_{Poisson} : H \to \mathbb{R}$ is defined as:*

$$c_{Poisson}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}; c_0, c_1, c_2\right) \equiv \log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}a_1^2/a_2 - \frac{1}{4}b_1^2/b_2 + \frac{1}{2}\log(a_2/b_2) - \log\Gamma(c_2+1),$$

where r_1, r_2, r_6 and r_7 follow as defined in Definition 21.

In summary, the projection of the input function onto the unnormalised normal family is given in Result 21.

Result 21. *For an unnormalised input function of the form of equation (5.2),*

$$proj_{UN}[f_{input}] = \exp\left\{\begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \begin{bmatrix} \eta_0^* \\ \eta_1^* \\ \eta_2^* \end{bmatrix}\right\},$$

where

$$\begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix} = k_{Poisson}\left(\begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}; c_0, c_1, c_2\right)$$

and

$$\eta_0^* = c_{Poisson}\left(\begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix}; c_0, c_1, c_2\right).$$

We now show how to implement the results shown in this section in a message passing framework.

5.1.2.2 Message passing formulation

As mentioned, only minor changes to Algorithm 3 are required to account for the Poisson case. Note the components of the sum of the log-likelihood function $\ell_i(\sigma^2)$ are

$$\ell_i(\sigma^2) = \log \int_{-\infty}^{\infty} \Big(\prod_{j=1}^{n} p(y_{ij}|u_i)\Big) p(u_i; \sigma^2) du_i, \tag{5.5}$$

where

$$p(y_{ij}|u_i) \equiv \text{Poisson}\big(\exp(u_i), y_{ij}\big) \quad \text{and} \quad p(u_i;\sigma^2) \equiv (2\pi\sigma^2)^{-1/2}\exp\big(-u_i^2/(2\sigma^2)\big)$$

are respectively the conditional density functions of each response given its random effect and the density function of the random effect.

Since the dependence structure of the product in equation (5.5) matches that of the probit model shown in Figure 2.1, we only need to update the messages depending on the conditional density functions of each response given its random effect. Following Section 2.2.2, this involves updating messages $m_{p(y_{ij}|u_i)\to u_i}(u_i)$ as in equation (2.10) to

$$m_{p(y_{ij}|u_i)\to u_i}(u_i) \longleftarrow \frac{\text{proj}_{\text{UN}}\left[\text{Poisson}\big(\exp(c_0+c_1u_i);c_{2_{ij}}\big)\exp\left(\eta_1^{\otimes}u_i+\eta_2^{\otimes}u_i^2\right)\right]}{\exp(\eta_1^{\otimes}u_i+\eta_2^{\otimes}u_i^2)},$$

where $c_0=0$, $c_1=1$ and $c_{2_{ij}}=y_{ij}$. Obtaining the required projection follows from Result 21 analogous to equation (2.16), where the linear and quadratic coefficient updates in equation (2.17) are changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\right)_{1:2} \longleftarrow k_{\text{Poisson}}\left(\boldsymbol{\eta}_{1:2}^{\otimes};c_0,c_1,c_{2_{ij}}\right)-\boldsymbol{\eta}_{1:2}^{\otimes} \tag{5.6}$$

and where the constant coefficient update in equation (2.18) is changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\right)_0 \longleftarrow c_{\text{Poisson}}\left(\boldsymbol{\eta}_{1:2}^{\otimes},\left(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\right)_{1:2}+\boldsymbol{\eta}_{1:2}^{\otimes};c_0,c_1,c_{2_{ij}}\right). \tag{5.7}$$

In summary, Algorithm 3 applies to the Poisson model, however equations (2.17) and (2.18) are replaced with equations (5.6) and (5.7), with constant term inputs set as $c_0 \longleftarrow 0$, $c_1 \longleftarrow 1$ and $c_{2_{ij}} \longleftarrow y_{ij}$ for $1\le i\le m$, $1\le j\le n$. Additionally, the initialisation of $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$ follows equation (5.9).

5.1.2.3 Starting values for the Poisson case

We now derive starting values for the Poisson case using the same principles shown in the univariate probit and logistic case. Let

$$\log p(y_{ij}|u_i) = f(u_i,y_{ij}) - \log\Gamma(y_{ij}+1),$$

where

$$\begin{aligned} f(x, y_{ij}) = f(x) =& y_{ij}x - \exp(x), \\ f'(x) =& y_{ij} - \exp(x), \end{aligned}$$

and

$$f''(x) = -\exp(x). \tag{5.8}$$

Let $\widehat{u}_i$ be an approximation of u_i. Then a Taylor series expansion of $f(u_i)$ evaluated at $\widehat{u}_i$ leads to

$$\begin{aligned} f(u_i) =& f(\widehat{u}_i) + f'(\widehat{u}_i)(u_i - \widehat{u}_i) + \frac{1}{2}f''(\widehat{u}_i)(u_i - \widehat{u}_i)^2 + ... \\ =& \begin{bmatrix} 1 \\ u_i - \widehat{u}_i \\ (u_i - \widehat{u}_i)^2 \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + ... \,, \end{aligned}$$

where

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} f(\widehat{u}_i) \\ f'(\widehat{u}_i) \\ \frac{1}{2}f''(\widehat{u}_i) \end{bmatrix}.$$

The quadratic approximation to $\log p(y_{ij}|u_i)$ based on Taylor expansion about $\widehat{u}_i$ is $\log \breve{p}(y_{ij}|u_i)$ as per equation (2.22). By following the same logic of the previous cases it is easy to show

$$\boldsymbol{\eta}^{\text{start}}_{p(y_{ij}|u_i)\to u_i} = \begin{bmatrix} \eta_0^{\text{start}} \\ f'(\widehat{u}_i) - f''(\widehat{u}_i)\widehat{u}_i \\ \frac{1}{2}f''(\widehat{u}_i) \end{bmatrix}, \tag{5.9}$$

where

$$\eta_0^{\text{start}} = f(\widehat{u}_i) - f'(\widehat{u}_i)\widehat{u}_i + \frac{1}{2}f''(\widehat{u}_i)(\widehat{u}_i)^2.$$

The comments at the end of Section 2.2.3 apply for the these starting values.

5.1.3 Evaluation of the estimates

Implementing the quadrature and EP approaches in the R computing environment, we now visually compare the accuracy of our likelihood approximation $\underset{\sim}{\ell}'(\sigma^2)$ to the exact likelihood surface $\ell(\sigma^2)$. Figure 5.1 plots the estimates of the likelihood surface for both methods. The data generated had 50 groups with 5 responses per group (i.e. $m = 50$ and $n = 5$) and the true value of $\sigma^2 = 0.17$. Using the quadrature method as exact, the plot shows although there are some discepencies on the tails of the likelihood surface, the approximate method follows the exact likelihood surface around the maximum well. Additionally, the true value of σ^2 matches well with the maximum of the exact likelihood.

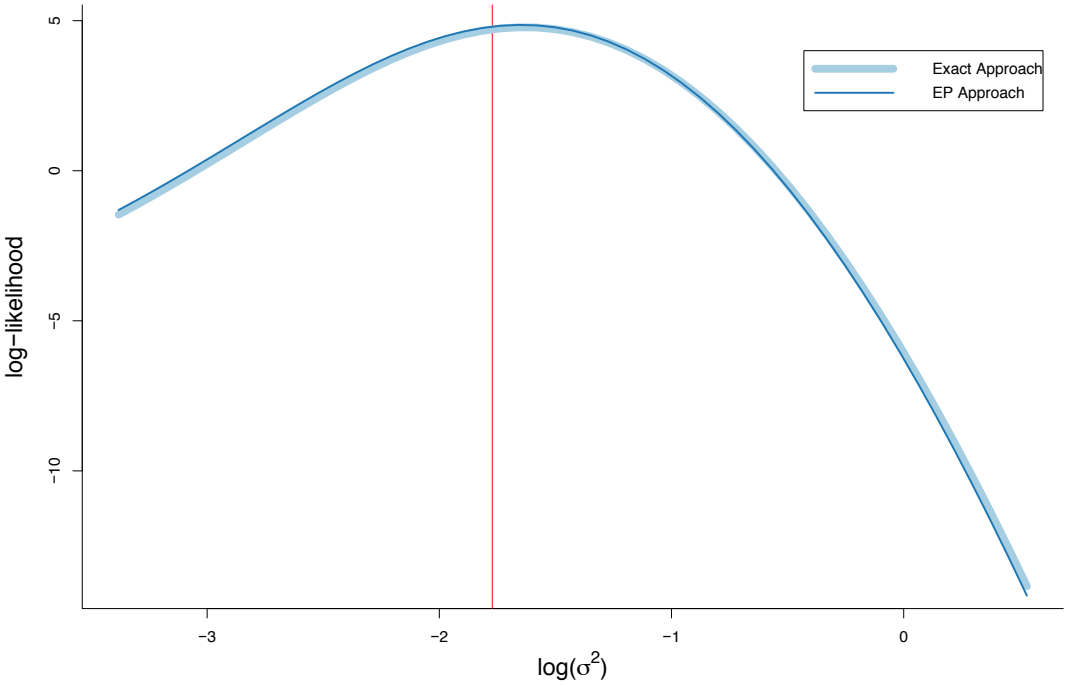


Figure 5.1: *A comparison plot of the log-likelihood surface over the parameter σ^2 for Poisson models calculated exactly using univariate quadrature and approximated via EP. The true $\sigma^2 = 0.17$ is represented on the log scale by the red line. The EP approximation is shown by the dark blue line and the exact by the light blue line.*

5.1.4 Best predictor

Best prediction for σ^2 via quadrature and EP is the same as presented in Section 2.5, although for the former we now redefine

$$J_{si}(\sigma^2) \equiv \int_{-\infty}^{\infty} u_i^s \prod_{j=1}^{n} \text{Poisson}\big(\exp(u_i); y_{ij}\big)(2\pi\sigma^2)^{-1/2} \exp\left(-u_i^2/(2\sigma^2)\right) du_i, \quad s \in \{0, 1, 2\}.$$

The calculations are otherwise analogous.

5.2 General Poisson mixed models

We now consider EP for the general Poisson model where any number of fixed and random effects can be specified. Consider

$$y_{ij}|\boldsymbol{u}_i \overset{\text{ind}}{\sim} \text{Poisson}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right); y_{ij}\Big), \quad \boldsymbol{u}_i \overset{\text{ind}}{\sim} \boldsymbol{N}\big(\mathbf{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}\big),$$

$$1 \leq i \leq m \quad \text{and} \quad 1 \leq j \leq n_i,$$

where the notation follows the general one level model presented in Section 1.8. The log-likelihood can be expressed as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{i=1}^{m} \ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}),$$

where

$$\ell_i\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}\right) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \text{Poisson}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right); y_{ij}\Big) \right\} \times |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \right) d\boldsymbol{u}_i,$$

and the best predictor of $\boldsymbol{u}_i$ is

$$\text{BP}(\boldsymbol{u}_i) \equiv \frac{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}_i \left\{ \prod_{j=1}^{n_i} \text{Poisson}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right); y_{ij}\Big) \right\} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \right) d\boldsymbol{u}_i}{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \text{Poisson}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right); y_{ij}\Big) \right\} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \right) d\boldsymbol{u}_i}. \tag{5.10}$$

Implementation of EP follows as per the previous models, with minor alterations to the algorithm to account for the count response variable.

We first explain likelihood approximation using EP in Subsection 5.2.1, before results of a simulation study are presented in Subsection 5.2.2. Details regarding computation of point estimates and confidence intervals are analogous to those presented in Section 3.2, whilst the same calculations in Section 3.3 can be used for best prediction of $\boldsymbol{u}_i$. We do not repeat either section again and instead refer readers to the previous work.

5.2.1 Expectation propagation likelihood approximation

EP centres around finding the optimal natural parameters $\eta_0, \boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ which minimise $\text{KL}(f_{\text{input}} \parallel f_{\boldsymbol{UN}})$, where $f_{\boldsymbol{UN}}$ is defined by equation (3.4) and

$$f_{\text{input}}(\boldsymbol{x}) = \text{Poisson}\Big(\exp\left(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\right); c_2\Big) \exp\left(\left(\boldsymbol{\eta}_1^{\text{input}}\right)^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2^{\text{input}} \boldsymbol{x}\right), \tag{5.11}$$

where $\boldsymbol{\eta}_1^{\text{input}}$ is a $d \times 1$ vector, $\boldsymbol{H}_2^{\text{input}}$ is a $d \times d$ matrix, $c_0 = \boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}}$, $c_2 = y_{ij}$, $\boldsymbol{c}_1 = \boldsymbol{x}_{ij}^{\mathbf{R}}$ and $x = \boldsymbol{u}_i$. As such, we seek an $\boldsymbol{\eta}^*$ to solve

$$\begin{aligned}
&\int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \text{Poisson}\Big(\exp\left(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\right); c_2\Big) \exp\left\{ \begin{bmatrix} \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta}^{\text{input}} \right\} d\boldsymbol{x} \\
&\qquad = \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} d\boldsymbol{x},
\end{aligned} \tag{5.12}$$

where $\boldsymbol{x}^{\otimes k}$ is as defined in equation (1.2). Thus to obtain the required projection, we first obtain the optimal natural parameters $\boldsymbol{\eta}_1^*$ and $\boldsymbol{\eta}_2^*$ to project onto the multivariate normal family as is presented in Result 22. As the integrals required to do so are not available in closed form solutions, we use the same properties as the expit for Poisson models to express the multivariate integrals as univariate integrals. These integrals are expressed as in Section 2.1 of Kim & Wand (2018)[32] as denoted by $\mathcal{C}_b(p, q, r)$ and solved using quadrature in a more efficient manner.

Result 22. *Given f_{input} follows the form of equation (5.11), the projection onto the multivariate normal family is given by*

$$proj_{\boldsymbol{N}}[f_{input}] = \exp\Big(\boldsymbol{T}(\boldsymbol{x})^\top \boldsymbol{\eta}_{-1}^* - A(\boldsymbol{\eta}_{-1}^*)\Big) h(\boldsymbol{x}),$$

where

$$\boldsymbol{\eta}_{-1}^* \equiv K_{Poisson}\big(\boldsymbol{\eta}_{-1}^{input}; c_0, \boldsymbol{c}_1, c_2\big),$$

$$\boldsymbol{\eta}_{-1}^{input} \equiv \begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}_{-1}^{*} \equiv \begin{bmatrix} \boldsymbol{\eta}_1^{*} \\ \boldsymbol{\eta}_2^{*} \end{bmatrix},$$

$K_{Poisson}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2 \right)$ *is defined in Definition 23 and $\boldsymbol{T}(\boldsymbol{x})$ and $h(\boldsymbol{x})$ follow from Section 1.5.2.2.*

Definition 23. *For primary arguments $\boldsymbol{a}_1$ $(d \times 1)$ and $\boldsymbol{a}_2$ $\big(\frac{1}{2}d(d+1) \times 1\big)$ such that $vec^{-1}\big(-(\boldsymbol{D}_d^+)^\top \boldsymbol{a}_2\big)$ is symmetric and positive definite, and auxiliary arguments $c_0 \in \mathbb{R}$, $c_2 \in \mathbb{Z}^+$ and $\boldsymbol{c}_1$ $(d \times 1)$, the function $K_{Poisson} : H \to H$ is given by*

$$K_{Poisson}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2 \right) \equiv \begin{bmatrix} \boldsymbol{R}_5^\top (\boldsymbol{a}_1 + r_3 \boldsymbol{c}_1) \\ \boldsymbol{D}_d^\top vec(\boldsymbol{R}_5^\top \boldsymbol{A}_2) \end{bmatrix}, \tag{5.13}$$

where

$$\boldsymbol{A}_2 \equiv vec^{-1}\big((\boldsymbol{D}_d^+)^\top \boldsymbol{a}_2\big), \quad r_1 = \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\, \boldsymbol{c}_1, \quad r_2 = \big(\boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1} \boldsymbol{a}_1 - 2c_0\big) r_7,$$

$$r_3 = r_2 + 2\, \mathcal{C}_{b,1:0}(r_6, r_7) r_7, \quad r_4 = 2\big(\mathcal{C}_{b,1:0}(r_6, r_7)^2 - \mathcal{C}_{b,2:0}(r_6, r_7) - r_1\big) r_7^2,$$

$$\boldsymbol{R}_5 = \big(\boldsymbol{A}_2 + r_4 \boldsymbol{c}_1 \boldsymbol{c}_1^\top\big)^{-1} \boldsymbol{A}_2, \quad r_6 = 1 - r_2, \quad r_7 = -(r_1)^{-1},$$

$$\mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)}, \quad \mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)}, \quad b(x) = \exp(x)$$

and $\mathcal{C}_b(k, r, q)$ is as per equation (5.42).

A proof of Definition 23 is given in Appendix 5.6.2. Using Result 22 we now obtain the normalising natural parameter η_0^* to find the projection onto unnormalised normal family.

5.2.1.1 Projection onto the unnormalised multivariate normal family

Recall the moment matching problem from equation (5.12) and Results 12 and 13. Then, we require

$$C_{f_{\text{input}}} = \int_{\mathbb{R}^d} f_{\text{input}}(\boldsymbol{x}) d\boldsymbol{x} = \mathcal{C}_b(0, r_6, r_7) Z_0 Z_1,$$

where $Z_0 = \exp\left((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\right)$, $Z_1 = \exp\left(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi) - \log\Gamma(c_2+1)\right)$ and r_6 and r_7 are given in Definition 23. Analogous to previous arguments, we then get

$$\begin{aligned}\boldsymbol{\eta}_0^* = &\log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}\left(\boldsymbol{\eta}_1^*\right)^\top \left(\boldsymbol{H}_2^*\right)^{-1} \boldsymbol{\eta}_1^* \\ &- \frac{1}{4}\left(\boldsymbol{\eta}_1^{\text{input}}\right)^\top \left(\boldsymbol{H}_2^{\text{input}}\right)^{-1} \boldsymbol{\eta}_1^{\text{input}} + \frac{1}{2}\log\left(|\boldsymbol{H}_2^*|/|\boldsymbol{H}_2^{\text{input}}|\right) - \log\Gamma(c_2+1).\end{aligned}$$

To find the normalising constant of the input function, we introduce Definition 24.

Definition 24. *Consider first, primary arguments $\boldsymbol{a}_1$ and $\boldsymbol{b}_1$ and auxiliary argument $\boldsymbol{c}_1$ where all three are $d \times 1$. Next consider arguments $\boldsymbol{a}_2$ and $\boldsymbol{b}_2$ which are all $\left(\frac{1}{2}d(d+1) \times 1\right)$ such that both $vec^{-1}\left(-(\boldsymbol{D}_d^+)^\top \boldsymbol{a}_2\right)$ and $vec^{-1}\left(-(\boldsymbol{D}_d^+)^\top \boldsymbol{b}_2\right)$ are symmetric and positive definite. Finally note auxiliary scalar argument $c_0, \in \mathbb{R}$, $c_2, \in \mathbb{Z}^+$. Then the function $C_{Poisson} : H \times H \to \mathbb{R}$ is given by*

$$\begin{aligned}C_{Poisson}\left(\begin{bmatrix}\boldsymbol{a}_1 \\ \boldsymbol{a}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix}; c_0, \boldsymbol{c}_1, c_2\right) \equiv &\log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) \\ &+ \frac{1}{4}\boldsymbol{b}_1^\top \boldsymbol{B}_2^{-1}\boldsymbol{b}_1 - \frac{1}{4}\boldsymbol{a}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1 + \frac{1}{2}\log\left(|\boldsymbol{B}_2|/|\boldsymbol{A}_2|\right) - \log\Gamma(c_2+1),\end{aligned}$$

where $\boldsymbol{A}_2 \equiv vec^{-1}\left((\boldsymbol{D}_d^+)^\top \boldsymbol{a}_2\right)$, $\boldsymbol{B}_2 \equiv vec^{-1}\left((\boldsymbol{D}_d^+)^\top \boldsymbol{b}_2\right)$, r_1, r_2, r_6 and r_7 follow from Definition 23.

In summary, the projection onto the unnormalised multivariate normal family is given by Result 23.

Result 23. *For an unnormalised input function of the form of equation (5.11),*

$$proj_{\boldsymbol{UN}}\left[f_{input}\right](\boldsymbol{x}) = \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{x} \\ vec\left(\boldsymbol{x}\boldsymbol{x}^\top\right) \end{bmatrix}^\top \begin{bmatrix} \eta_0^* \\ \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}\right\},$$

where

$$\begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix} = K_{Poisson}\left(\begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2\right)$$

and

$$\eta_0^* = C_{Poisson}\left(\begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2\right).$$

5.2.1.2 Message passing formulation

As mentioned throughout this section, only minor changes to Algorithm 6 are required to account for the Poisson case. Each component of the sum of the log-likelihood function $\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ can be written as

$$\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left(\prod_{j=1}^{n_i} p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta})\right) p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) d\boldsymbol{u}_i, \tag{5.14}$$

where

$$p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta}) \equiv \text{Poisson}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), y_{ij}\Big)$$

and

$$p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) \equiv |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i\right)$$

are respectively the conditional density functions of each response given its random effect and the density function of that random effect.

As the dependence structure of the product in equation (5.14) is the same as the probit model shown in Figure 3.1, we only need to update the messages depending on the conditional density functions of each response given its random effect. This involves

updating messages $m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i)$ as in equation (3.12)

$$m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \leftarrow \frac{\text{proj}_{\boldsymbol{UN}}\left[\text{Poisson}\Big(\exp\left(c_{0_{ij}}+\boldsymbol{c}_{1_{ij}}^\top\boldsymbol{u}_i, c_{2_{ij}}\right)\Big)\exp\left\{\boldsymbol{u}_i^\top\boldsymbol{\eta}_1^\otimes+\left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top\boldsymbol{\eta}_2^\otimes\right\}\right]}{\exp\left\{\boldsymbol{u}_i^\top\boldsymbol{\eta}_1^\otimes+\left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top\boldsymbol{\eta}_2^\otimes\right\}},$$

where we set the constant terms:

$$c_{0_{ij}} \leftarrow \boldsymbol{\beta}^\top\boldsymbol{x}_{ij}^{\mathbf{F}}; \quad \boldsymbol{c}_{1_{ij}} \leftarrow \boldsymbol{x}_{ij}^{\mathbf{R}}; \quad c_{2_{ij}} \leftarrow y_{ij}; \quad 1\le i\le m, \quad 1\le j\le n_i.$$

Following Section 3.1.2, this task reduces down to adjusting the calculation of the optimal natural parameters $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ in equation (3.18) for the Poisson model, where the linear and quadratic coefficient updates in equation (3.19) given by K_{probit} are changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2} \leftarrow K_{\text{Poisson}}\left(\boldsymbol{\eta}_{1:2}^\otimes; c_{0_{ij}}, \boldsymbol{c}_{1_{ij}}, c_{2_{ij}}\right) - \boldsymbol{\eta}_{1:2}^\otimes \tag{5.15}$$

as per Definition 23, and where the constant coefficient update in equation (3.20) given by C_{probit} is changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_0 \leftarrow C_{\text{Poisson}}\left(\boldsymbol{\eta}_{1:2}^\otimes, \left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2} + \boldsymbol{\eta}_{1:2}^\otimes; c_{0_{ij}}, \boldsymbol{c}_{1_{ij}}, c_{2_{ij}}\right)$$

as per Definition 24.

Barring the aforementioned changes and the initialisation of $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ with the values discussed in Section 5.2.1.3, the full algorithm for the approximation of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ follows Algorithm 6.

5.2.1.3 Starting values for the Poisson case

We now derive starting values for the Poisson case using the same principles shown in the probit and expit case. Let

$$\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) = f(a_{ij}; y_{ij}) - \log\Gamma(y_{ij}+1),$$

where $a_{ij} = \boldsymbol{\beta}^\top\boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top\boldsymbol{x}_{ij}^{\mathbf{R}}$ and $f(x)$ is defined as in equation (5.8). Then

$$(a_{ij} - \widehat{a}_{ij}) = (\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top\boldsymbol{x}_{ij}^{\mathbf{R}}$$

where $\widehat{a}_{ij} = \boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \widehat{\boldsymbol{u}}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}$ and $\widehat{\boldsymbol{u}}$ is an approximation of $\boldsymbol{u}_i$. A Taylor series expansion of $f(a_{ij})$ evaluated at $\widehat{a}_{ij}$ leads to

$$\begin{aligned} f(a_{ij}) =& f(\widehat{a}_{ij}) + f'(\widehat{a}_{ij})(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}} + \frac{1}{2} f''(\widehat{a}_{ij}) \big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)^2 + \dots \\ =& \begin{bmatrix} 1 \\ \boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i \\ \operatorname{vech}\big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top\big) \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + \dots , \end{aligned}$$

where

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} f(\widehat{a}_{ij}) \\ f'(\widehat{a}_{ij}) \boldsymbol{x}_{ij}^{\mathbf{R}} \\ \frac{1}{2} f''(\widehat{a}_{ij}) \boldsymbol{D}_{d^{\mathbf{R}}}^\top \operatorname{vec}\big(\boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top\big) \end{bmatrix}.$$

Quadratic approximation to $\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ based on Taylor expansion about $\widehat{\boldsymbol{u}}_i$ is $\log \breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ as per equation (3.22). By following the same logic of the previous cases it is easy to show

$$\boldsymbol{\eta}^{\text{start}}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})} = \begin{bmatrix} \eta_0^{\text{start}} \\ f'(\widehat{a}_{ij}) \boldsymbol{x}_{ij}^{\mathbf{R}} - f''(\widehat{a}_{ij}) \boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top \widehat{\boldsymbol{u}}_i \\ \frac{1}{2} f''(\widehat{a}_{ij}) \boldsymbol{D}_{d^{\mathbf{R}}}^\top \operatorname{vec}\big(\boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top\big) \end{bmatrix},$$

where

$$\eta_0^{\text{start}} = f(\widehat{a}_{ij}) - f'(\widehat{a}_{ij}) (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top \widehat{\boldsymbol{u}}_{ij} + \frac{1}{2} f''(\widehat{a}_{ij}) \widehat{\boldsymbol{u}}_{ij}^\top \boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top \widehat{\boldsymbol{u}}_{ij}.$$

5.2.2 Simulation study

We now provide results of a simulation study for a random intercept model with 1000 replicates. Datasets were simulated with true parameter values

$$\boldsymbol{\beta}_{\text{true}} = [0.38, 0.93]^\top \quad \text{and} \quad \sigma^2_{\text{true}} = -0.53, \tag{5.16}$$

generated with 50 groups and 5 measurements per group (i.e. $m = 50$ and $n = 5$). The $\boldsymbol{x}_{ij}^{\mathbf{F}}$ and $\boldsymbol{x}_{ij}^{\mathbf{R}}$ vectors were of the form

$$\boldsymbol{x}_{ij}^{\mathbf{F}} = [1, x_{1_{ij}}]^\top \quad \text{and} \quad \boldsymbol{x}_{ij}^{\mathbf{R}} = 1,$$

where $x_{1_{ij}}$ was generated independently from a uniform distribution on the unit interval. The tolerance of error values were set to 10^{-5} for the EP scheme and the maximum number of iterations for optimisation was set to 1000.

We compare our EP methodology to Laplace approximation and adaptive Gauss-Hermite quadrature using 100 quadrature points. Both alternative methods were implemented via the R function `glmer()` from the R package "lme4" (Bates, et al., 2018[5]). Since in the Poisson case quadrature is required for each projection of EP, EP is considerably slower than Laplace approximations and Gauss-Hermite quadrature. Additionally we note that to run the simulations within a reasonable time frame, they were seperated and run across mulitple high performance computers with of varying specification.

The resulting estimates and 95% confidence intervals for each interpretable model parameter of the study are presented in Figure 4.3, where the upper-right hand corner of each panel shows the empirical coverage values based on all 1000 replicates. Only 20 randomly chosen replicates from each method are shown in the panels for ease of viewing, where Laplace approximation and EP are shown by black lines, super imposed on grey lines showing the quadrature approach. Across the fixed effects parameters the empirical coverage of EP is within 0.2% of quadrature, where as Laplace approximations are marginally lower, particularly for the fixed slope which had coverage of 93.6%. The EP empirical coverage for the variance parameter was 1.6% higher than the expected 95%, where Laplace and quadrature are much closer coverage to 95%, with coverages of 94.9% and 95.1% respectively. Although we are aware that a comparison of time is obscured by factors such as language of implementation and computer performance, the average time for the EP routine over 100 replications was 1166.0054 seconds, whilst Laplace approximations and adaptive Gauss-Hermite took an average of 0.3108 and 0.3133 seconds respectively.

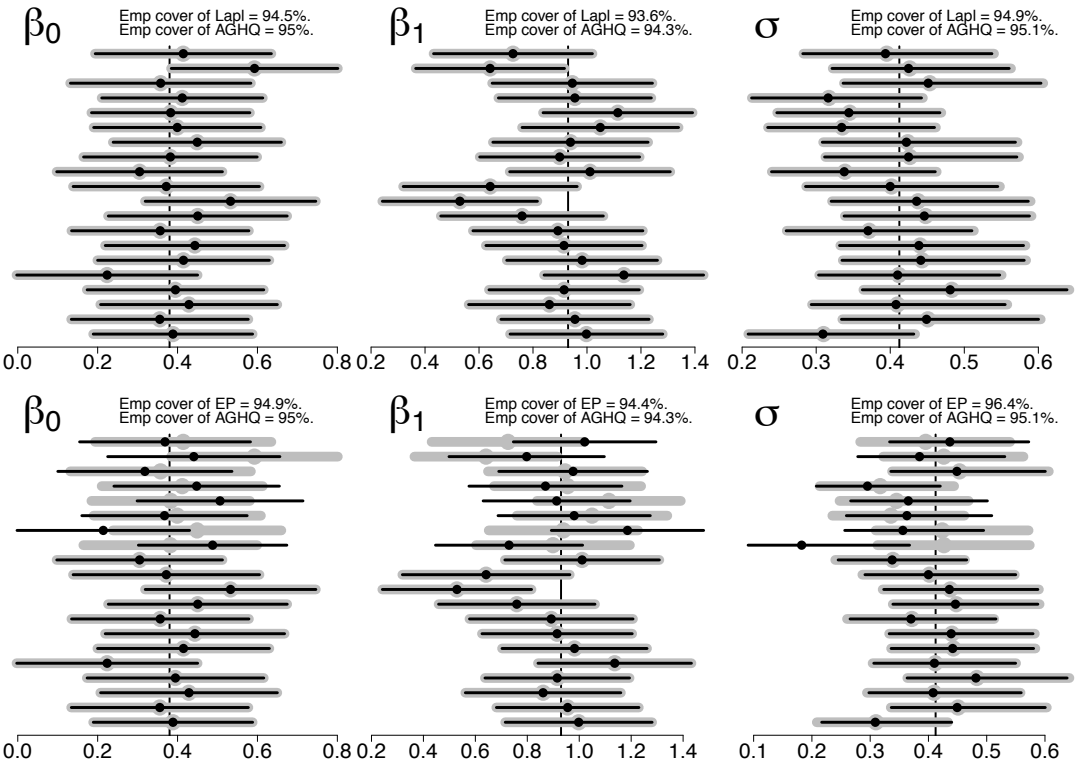


Figure 5.2: *Summary comparison between confidence interval coverage for the univariate model with true parameter values from equation (5.16) for models fit with adaptive Gauss-Hermite quadrature and Laplace approximation. The horizontal lines are the confidence intervals for 20 randomly chosen replications of the simulation study, the solid circular points indicate the corresponding point estimates, the vertical lines indicate true parameter values, the grey lines correspond to adaptive Gauss-Hermite quadrature, the black lines on the top row correspond to Laplace approximation approach and the black lines on the bottom row correspond to EP approximation approach. The percentage given in the top right-hand corner of each panel is the empirical coverage over all* 1000 *replications.*

5.3 The simplest negative binomial models

The negative binomial model follows naturally from the Poisson model. It provides the added benefit of a shape parameter κ, which facilitates handling over-dispersed count data. Specifically, as κ approaches zero the variance of the data becomes infinite. As κ increases to infinity the negative binomial distribution begins to resemble the Poisson distribution. Large values of κ can inturn lead to numerical issues, making the negative binomial model a particularly tricky one to handle. Analysts should determine whether to use Poisson or negative binomial links based on the nature of the data.

Since the negative binomial model brings the additional challenge of estimating the shape parameter κ, we begin developing our model on the case where we consider only the parameter κ. For this model we fix the variance between groups σ^2_{Fixed} and aim to estimate κ. Additionally, we assume a balanced dataset with m-groups and n-observations per group. For observed values of

$$y_{ij}; \quad 1 \leq i \leq m, \quad 1 \leq j \leq n,$$

where $y_{ij} \in \mathbb{Z}_{\geq 0}$, the negative binomial model is

$$y_{ij}|u_i; \kappa \overset{\text{iid}}{\sim} \text{NB}\big(\exp(u_i), \kappa; y_{ij}\big), \quad u_i \overset{\text{iid}}{\sim} N\big(0, \sigma^2_{\text{Fixed}}\big),$$

where $\sigma^2_{\text{Fixed}} \in \mathbb{R}_{>0}$ is a fixed number, $\kappa \in \mathbb{R}_{>0}$, $\text{NB}(y, \mu, \kappa)$ is given by equation (1.9) and u_i is a scalar unobserved latent variable. We wish to find the maximiser of $\ell(\kappa)$ denoted by $\widehat{\kappa}$. Note that we do not find best predictors of u_i as they are assumed to be known in this setting. The likelihood of model can be expressed as

$$\ell(\kappa) = \sum_{i=1}^{m} \ell_i(\kappa),$$

where

$$\ell_i(\kappa) \equiv \log \int_{-\infty}^{\infty} \prod_{j=1}^{n} \text{NB}\big(\exp(u_i), \kappa; y_{ij}\big) \left(2\pi\sigma^2_{\text{Fixed}}\right)^{-1/2} \exp\big(-u_i^2/2\sigma^2_{\text{Fixed}}\big) du_i, \tag{5.17}$$

and the maximum likelihood estimate of κ is given by

$$\widehat{\kappa} = \underset{\kappa}{\text{argmax}}\ \ell(\kappa).$$

Given the integral arising in the calculation of the log-likelihood surface of $\ell(\kappa)$ does not have a closed form solution, we develop an EP scheme to approximate it and compare the result to a quadrature approach.

Subsection 5.3.1 provides details of the quadrature approach to estimating the likelihood surface. We then explain our novel method using EP in Subsection 5.3.2, before comparing the likelihood surface of both methods in Subsection 5.3.3 and explaining point estimate and confidence interval calculation in Section 5.3.4.

5.3.1 Traditional quadrature likelihood approximation

Implementation of the adaptive quadrature via the Gauss-Kronrod method follows the same approach as in the Poisson case, where we utilise the R function `integrate()` in the "stats" package (R Core Team, 2019[56]). We suggest that for numerical stability each $\ell_i(\kappa)$ arising in equation (5.17) is calculated as

$$\ell_i(\kappa) = h(u_{0i}) + \log \int_{-\infty}^{\infty} \exp\big(h_i(u) - h_i(u_{0i})\big) du,$$

where

$$\begin{aligned} h_i(u) &\equiv y_{i.}u - \frac{u^2}{2\sigma^2_{\text{Fixed}}} - (n\kappa + y_{i.}) \log\big(\kappa + \exp(u)\big), \\ h_i'(u) &\equiv y_{i.} - \frac{u}{\sigma^2_{\text{Fixed}}} - \frac{(n\kappa + y_{i.}) \exp(u)}{\log\big(\kappa + \exp(u)\big)} \end{aligned}$$

and $\boldsymbol{u}_{0i}$ is the root of h_i'. We recommend finding $\boldsymbol{u}_{0i}$ using a bisection search, where the starting values are selected -1 and 1 to be for the lower and upper bounds respectively.

5.3.2 Expectation propagation likelihood approximation

We now consider an EP approach to the approximate likelihood denoted by $\underset{\sim}{\ell}(\kappa)$. In the negative binomial case, the EP approximation requires an unnormalised normal density function to replace each

$$\text{NB}\big(\exp(u_i), \kappa; y_{ij}\big), \quad 1 \leq j \leq n$$

in equation (5.17), such that the KL-divergence criterion is minimised. For the case of negative binomial GLMMs, the required projections onto an unnormalised univariate

normal distribution are

$$f_{\text{input}}(x) = \text{NB}\big(\exp(c_0 + c_1 x), \kappa; c_2\big) \exp\left(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\right), \tag{5.18}$$

where $\eta_1^{\text{input}} \in \mathbb{R}$, $\eta_2^{\text{input}} < 0$, $c_0 = 0$, $c_1 = 1$, $c_2 = y_{ij}$, $\kappa \in \mathbb{R}_{\geq 0}$ and $x = u_i$. Subsequently, the integrand is proportional to a product of univariate normal density functions. As such, we seek $\boldsymbol{\eta}^*$ such that

$$\begin{aligned}\int_{-\infty}^{\infty} x^k \text{NB}\big(\exp(c_0 + c_1 x), \kappa; c_2\big) \exp\big(\eta_1^{\text{input}} x + \eta_2^{\text{input}} x^2\big) dx \\ = \int_{-\infty}^{\infty} x^k \exp\left\{ \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} dx,\end{aligned} \tag{5.19}$$

Thus to obtain the required projection, we first obtain the optimal natural parameters η_1^* and η_2^* to project onto the normal family. Since the integral arising in the required projection does not have a closed form solution, we use univariate quadrature to obtain it. As before, it is important to use caution when calculating the integrals, since they are prone to be numerically unstable. For this reason we express each integral as in Section 2.1 of Kim & Wand (2018)[32] as denoted by $\mathcal{C}_b(p, q, r)$. Using algebra similar to that required for the Poisson model, we arrive at Result 24.

Result 24. *Given f_{input} follows the form of equation (5.18), the projection onto the univariate normal family is given by*

$$proj_N[f_{input}] = \exp\Big(\boldsymbol{T}(x)^\top \boldsymbol{\eta}_{-1}^* - A(\boldsymbol{\eta}_{-1}^*)\Big) h(x),$$

where

$$\boldsymbol{\eta}_{-1}^* = k_{NB}\big(\boldsymbol{\eta}_{-1}^{input}; c_0, c_1, c_2, \kappa\big), \qquad \boldsymbol{\eta}_{-1}^{input} \equiv \begin{bmatrix} \eta_1^{input} \\ \eta_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}_{-1}^* \equiv \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix},$$

$k_{NB}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1, c_2, \kappa \right)$ *is defined in Definition 25 and $\boldsymbol{T}(x)$ and $h(x)$ follow from Section 1.5.2.1.*

Definition 25. *For primary scalar arguments $a_1 \in \mathbb{R}$ and $a_2 \in \mathbb{R}_{\geq 0}$ and auxiliary scalar arguments $c_0, c_1 \in \mathbb{R}$, $c_2 \in \mathbb{Z}_{\geq 0}$ and $\kappa \in \mathbb{R}_{\geq 0}$ the function $k_{NB} : H \to H$ is given by*

$$k_{NB}\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; c_0, c_1, c_2, \kappa\right) \equiv \begin{bmatrix} r_5(a_1 + r_3 c_1) \\ r_5 a_2 \end{bmatrix},$$

where

$$r_1 = c_1^2 a_2^{-1}, \quad r_2 = \big(c_1 a_2^{-1} a_1 - 2c_0\big) r_7, \quad r_3 = r_2 + 2\, \mathcal{C}_{b,1:0}(r_6, r_7) r_7,$$

$$r_4 = 2\big(\mathcal{C}_{b,1:0}(r_6, r_7)^2 - \mathcal{C}_{b,2:0}(r_6, r_7) - r_1\big) r_7^2, \quad r_5 = \big(a_2 + r_4 c_1^2\big)^{-1} a_2,$$

$$r_6 = 1 - r_2, \quad r_7 = -r_1^{-1}, \quad \mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)},$$

$$\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2, r_6, r_7)}{\mathcal{C}_b(0, r_6, r_7)}, \quad b(x) = (c_2 + \kappa) \log\big(\exp(x) + \kappa\big).$$

and $\mathcal{C}_b(k, r, q)$ follows from equation (5.42).

A proof of Definition 25 is given in Appendix 5.6.3. Using Result 24 we now obtain the normalising natural parameter η_0^* to find the projection onto unnormalised normal family.

5.3.2.1 Projection onto the unnormalised normal family

Recall the moment matching problem described by equation (5.19) and Result 4. Then the normalising factor can be shown to be

$$C_f = \int_{\mathbb{R}^d} f_{\text{input}}(x) dx = \mathcal{C}_b(0, r_6, r_7) Z_0 Z_1,$$

where

$$Z_1 = \exp\left(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi) + \log(c_2 + \kappa) + \kappa \log \kappa - \log \Gamma(c_2 + 1) - \log \Gamma(\kappa)\right),$$

$Z_0 = \exp\left((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\right)$, and r_6 and r_7 are given in Definition 25. By Result 4

$$\begin{aligned}\boldsymbol{\eta}_0^* = &\log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}(\boldsymbol{\eta}_1^*)^2/\boldsymbol{\eta}_2^* - \frac{1}{4}(\boldsymbol{\eta}_1^{\text{input}})^2/\boldsymbol{\eta}_2^{\text{input}} \\ &+ \frac{1}{2}\log(\boldsymbol{\eta}_2^*/\boldsymbol{\eta}_2^{\text{input}}) + \log\Gamma(c_2+\kappa) + \kappa\log\kappa - \log\Gamma(c_2+1) - \log\Gamma(\kappa).\end{aligned}$$

Definition 26 follows.

Definition 26. *For primary scalar arguments $a_1 \in \mathbb{R}$ and $a_2 \in \mathbb{R}_{\geq 0}$ and auxiliary scalar arguments $c_0, c_1 \in \mathbb{R}$, $c_2 \in \mathbb{Z}_{\geq 0}$ and $\kappa \in \mathbb{R}_{\geq 0}$ the function $c_{NB} : H \to \mathbb{R}$ is defined as:*

$$\begin{aligned}c_{NB}\left(\begin{bmatrix}a_1\\a_2\end{bmatrix}, \begin{bmatrix}b_1\\b_2\end{bmatrix}; c_0, c_1, c_2, \kappa\right) \equiv &\log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}a_1^2/a_2 \\ &-\frac{1}{4}b_1^2/b_2 + \frac{1}{2}\log(a_2/b_2) + \log\Gamma(c_2+\kappa) + \kappa\log\kappa - \log\Gamma(c_2+1) - \log\Gamma(\kappa),\end{aligned}$$

where r_1, r_2, r_6 and r_7 follows from Definition 25.

Following the work of this section, the projection of the input function onto the unnormalised normal family is given in Result 25.

Result 25. *For an unnormalised input function of the form of equation (2.5),*

$$proj_{UN}[f_{input}] = \exp\left\{\begin{bmatrix}1\\x\\x^2\end{bmatrix}^\top \begin{bmatrix}\eta_0^*\\\eta_1^*\\\eta_2^*\end{bmatrix}\right\},$$

where

$$\begin{bmatrix}\eta_1^*\\\eta_2^*\end{bmatrix} = k_{NB}\left(\begin{bmatrix}\eta_1^{input}\\\eta_2^{input}\end{bmatrix}; c_0, c_1, c_2, \kappa\right)$$

and

$$\eta_0^* = c_{NB}\left(\begin{bmatrix}\eta_1^{input}\\\eta_2^{input}\end{bmatrix}, \begin{bmatrix}\eta_1^*\\\eta_2^*\end{bmatrix}; c_0, c_1, c_2, \kappa\right).$$

We now show how to implement the results shown in this section in a message passing framework.

5.3.2.2 Message passing formulation

Only minor changes to Algorithm 3 are required to account for the negative binomial case. The components of the sum of the log-likelihood function $\ell_i(\kappa)$ are

$$\ell_i(\kappa) = \log \int_{-\infty}^{\infty} \Big(\prod_{j=1}^{n} p(y_{ij}|u_i, \kappa) \Big) p\big(u_i; \sigma^2_{\text{Fixed}}\big) du_i, \tag{5.20}$$

where

$$p(y_{ij}|u_i, \kappa) \equiv \text{NB}\big(\exp(u_i), \kappa; y_{ij}\big)$$

and

$$p\big(u_i; \sigma^2_{\text{Fixed}}\big) \equiv \big(2\pi\sigma^2_{\text{Fixed}}\big)^{-1/2} \exp\big(-u_i^2/(2\sigma^2_{\text{Fixed}})\big)$$

are respectively the conditional density functions of each response given its random effect and the density function of that random effect.

Since the dependence structure of the product in equation (5.5) matches that of the probit model shown in Figure 2.1, we only need to update the messages depending on the conditional density functions of each response given its random effect. Following Section 2.2.2, this involves updating messages $m_{p(y_{ij}|u_i)\to u_i}(u_i)$ as in equation (2.10)

$$m_{p(y_{ij}|u_i)\to u_i}(u_i) \leftarrow \frac{\text{proj}_{\text{UN}}\left[\text{NB}\big(\exp(c_0 + c_1 u_i), \kappa; c_{2_{ij}}\big) \exp\left(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2\right)\right]}{\exp\left(\eta_1^{\otimes} u_i + \eta_2^{\otimes} u_i^2\right)},$$

where $c_0 = 0$, $c_1 = 1$, $c_{2_{ij}} = y_{ij}$ and $\kappa \in \mathbb{R}_{\geq 0}$. Utilising Result 21 leads to equation (2.16), where the linear and quadratic coefficient updates in equation (2.17) are changed to

$$\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_{1:2} \leftarrow k_{\text{NB}}\big(\boldsymbol{\eta}^{\otimes}_{1:2}; c_0, c_1, c_{2_{ij}}, \kappa\big) - \boldsymbol{\eta}^{\otimes}_{1:2} \tag{5.21}$$

and where the constant coefficient update in equation (2.18) is changed to

$$\big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_0 \leftarrow c_{\text{NB}}\Big(\boldsymbol{\eta}^{\otimes}_{1:2}, \big(\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}\big)_{1:2} + \boldsymbol{\eta}^{\otimes}_{1:2}; c_0, c_1, c_{2_{ij}}, \kappa\Big). \tag{5.22}$$

In summary, Algorithm 3 applies to the Poisson model, however equations (2.16) and (2.18) are replaced with equations (5.21) and (5.22), with constant term inputs set as $c_0 \leftarrow 0$, $c_1 \leftarrow 1$ and $c_{2_{ij}} \leftarrow y_{ij}$ for $1 \leq i \leq m$, $1 \leq j \leq n$. Additionally, the initialisation of $\boldsymbol{\eta}_{p(y_{ij}|u_i)\to u_i}$ follows equation (5.24).

5.3.2.3 Starting values for the negative binomial case

We now derive starting values for the negative binomial case using the same principles shown in the previous cases. Let

$$\log p(y_{ij}|u_i,\kappa) = f(u_i, y_{ij}) + \log\Gamma(y_{ij}+\kappa) - \log\Gamma(y_{ij}+1) - \log\Gamma(\kappa) + \kappa\log\kappa,$$

where

$$\begin{aligned} f(y_{ij}, x, \kappa) =& f(x) = y_{ij}x - (y+\kappa)\log\big(\exp(x)+\kappa\big), \\ & f'(x) = y_{ij} - \frac{(y+\kappa)\exp(x)}{\exp(x)+\kappa}, \end{aligned}$$

and

$$f''(x) = -\frac{\kappa(y+\kappa)\exp(x)}{\big(\exp(x)+\kappa\big)^2}. \tag{5.23}$$

Let $\widehat{u}_i$ be an approximation of u_i. Then a Taylor series expansion of $f(u_i)$ evaluated at $\widehat{u}_i$ leads to

$$\begin{aligned} f(u_i) &= f(\widehat{u}_i) + f'(\widehat{u}_i)(u_i - \widehat{u}_i) + \frac{1}{2}f''(\widehat{u}_i)(u_i - \widehat{u}_i)^2 + ... \\ &= \begin{bmatrix} 1 \\ u_i - \widehat{u}_i \\ (u_i - \widehat{u}_i)^2 \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + ... \,, \end{aligned}$$

where

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} f(\widehat{u}_i) \\ f'(\widehat{u}_i) \\ \frac{1}{2}f''(\widehat{u}_i) \end{bmatrix}.$$

By following the same logic of the probit case it is easy to show

$$\boldsymbol{\eta}^{\text{start}}_{p(y_{ij}|u_i)\to u_i} = \begin{bmatrix} \eta_0^{\text{start}} \\ f'(\widehat{u}_i) - f''(\widehat{u}_i)\widehat{u}_i \\ \frac{1}{2}f''(\widehat{u}_i) \end{bmatrix}, \tag{5.24}$$

where

$$\eta_0^{\text{start}} = f(\widehat{u}_i) - f'(\widehat{u}_i)\widehat{u}_i + \frac{1}{2}f''(\widehat{u}_i)(\widehat{u}_i)^2.$$

The comments at the end of Section 2.2.3 are also applicable for the these starting values.

5.3.3 Evaluation of the estimates

Implementing the quadrature and EP approaches in the R computing environment, we now visually compare the accuracy of our likelihood approximation $\underset{\sim}{\ell}'(\kappa)$ to the exact likelihood surface $\ell(\kappa)$. Figure 5.3 plots the estimates of the likelihood surface for both methods. The data generated had 100 groups with 5 responses per group and the true value of $\kappa = 15$ and $\sigma^2 = 0.25$. Using the quadrature method as exact, the plot shows that although there are some discepencies on the tails of the likelihood surface, the approximate method follows the exact likelihood surface around the true maximum well. Additionally, the true value of κ matches well with the maximum of the exact likelihood.

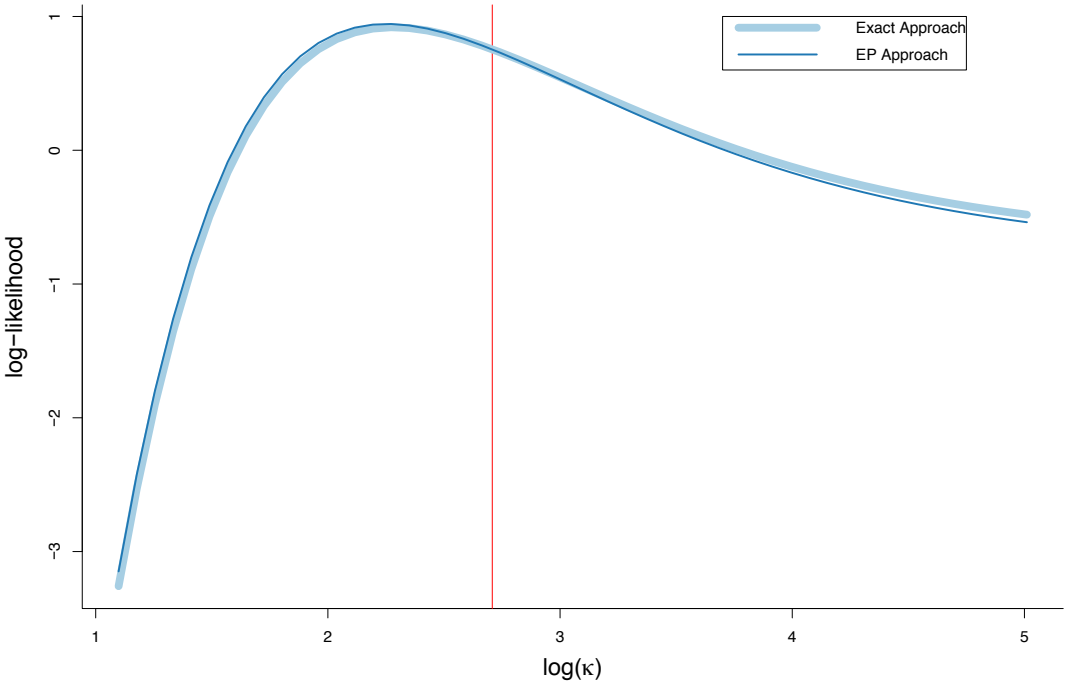


Figure 5.3: *A comparison plot of the log-likelihood surface over the parameter κ for negative binomial models calculated exactly using quadrature and approximated via EP. The true $\kappa = 15$ is represented on the log scale by the red line. The EP approximation is shown by the dark blue line and the exact by the light blue line.*

5.3.4 Computation of point estimates and confidence intervals

The maximum likelihood estimator for κ via quadrature and EP are respectively given by $\widehat{\kappa} = \underset{\kappa}{\text{argmax}}\ \ell(\kappa^2)$ and $\widehat{\underset{\sim}{\kappa}} = \underset{\kappa}{\text{argmax}}\ \underset{\sim}{\ell}(\kappa)$. As before, to find their stationary points we require the first derivative of the likelihood functions denoted by $\ell'(\kappa)$ and $\underset{\sim}{\ell}'(\kappa)$. Calculation of the second derivative, denoted by $\ell''(\kappa)$ and $\underset{\sim}{\ell}''(\kappa)$, facilitates calculation of confidence intervals.

The constraints imposed on the κ parameter to positive numbers mean that it is more appropriate to work with the parameter in a transformed space,

$$\xi \equiv \log(\kappa) = g(\kappa).$$

Since the maximum likelihood estimator is asymptotically normally distributed and

$g'(\kappa) = \kappa^{-1}$,

$$\widehat{\xi} \sim N\left(\xi_{\text{true}}, \frac{1}{(\widehat{\kappa})^2\big(-\ell''(\widehat{\kappa})\big)}\right).$$

Thus for a 95% confidence interval we expect that

$$0.95 \approx P\left(\widehat{\xi} - \frac{1.96}{\sqrt{(\widehat{\kappa})^2\big(-\ell''(\widehat{\kappa})\big)}} < \xi_{\text{true}} < \widehat{\xi} + \frac{1.96}{\sqrt{(\widehat{\kappa})^2\big(-\ell''(\widehat{\kappa})\big)}}\right).$$

Setting

$$\xi_{\text{low}} = \log(\widehat{\kappa}) - \frac{1.96}{\sqrt{(\widehat{\kappa})^2\big(-\ell''(\widehat{\kappa})\big)}} \quad \text{and} \quad \xi_{\text{upp}} = \log(\widehat{\kappa}) + \frac{1.96}{\sqrt{(\widehat{\kappa})^2\big(-\ell''(\widehat{\kappa})\big)}},$$

then the lower and upper 95% confidence intervals for parameter κ are given by

$$\kappa_{\text{low}} = \exp(\xi_{\text{low}}) \quad \text{and} \quad \kappa_{\text{upp}} = \exp(\xi_{\text{upp}}).$$

5.3.4.1 Derivative approximation

We consider only the quasi-Newtown solutions to the derivatives required for both point estimates and confidence intervals. To find the maximum of the likelihood surface and to obtain the second derivative, we use both the Nelder-Mead and BFGS algorithms via the R function `optim()` in the "stats" package (R Core Team, 2019[56]). As this is discussed clearly in Section 2.4.2 we do not repeat it.

5.4 General negative binomial model

We now conisder the extension to more general models where any number of fixed and random effects can be specified. This general model also allows for unbalanced datasets of m-groups with n_i observations per group. The form of this model is

$$y_i|\boldsymbol{u}_i \overset{\text{ind}}{\sim} \text{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), \kappa; y_{ij}\Big), \quad \boldsymbol{u}_i \overset{\text{ind}}{\sim} \boldsymbol{N}\big(\mathbf{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}\big),$$

$$1 \le i \le m, \quad 1 \le j \le n_i \quad \text{and} \quad \kappa > 0,$$

where the notation follows the general one level model presented in Section 1.8. The log-likelihood can be expressed as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa) = \sum_{i=1}^{m} \ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa),$$

where

$$\begin{aligned}\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \mathrm{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), \kappa; y_{ij} \Big) \right\} \\ \times |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\Big(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \Big) d\boldsymbol{u}_i,\end{aligned}$$

and the best predictor of $\boldsymbol{u}_i$ is

$$\mathrm{BP}(\boldsymbol{u}_i) \equiv \frac{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \boldsymbol{u}_i \left\{ \prod_{j=1}^{n_i} \mathrm{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), \kappa; y_{ij} \Big) \right\} \exp\big(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \big) d\boldsymbol{u}_i}{\int_{\mathbb{R}^{d^{\mathbf{R}}}} \left\{ \prod_{j=1}^{n_i} \mathrm{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), \kappa; y_{ij} \Big) \right\} \exp\big(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \big) d\boldsymbol{u}_i}. \tag{5.25}$$

Implementation of EP follows as per the previous models, with minor alterations to the algorithm to account for the count response variable.

We first explain likelihood approximation using EP in Subsection 5.4.1, before explaining computation of point estimates and confidence intervals in Section 5.4.2. Results of a simulation study are presented in Subsection 5.4.3. The same calculations in Section 3.3 can be used for best prediction of $\boldsymbol{u}_i$. We do not repeat them again and instead refer readers to the previous work.

5.4.1 Expectation propagation likelihood approximation

EP can be used to approximate the likelihood by updating and summing the natural parameter updates. We wish to find the optimal natural parameters η_0, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ which minimise $\mathrm{KL}(f_{\text{input}} \parallel f_{\text{UN}})$, where f_{UN} is defined by equation (3.4) and

$$f_{\text{input}}(\boldsymbol{x}) = \mathrm{NB}\Big(\exp\left(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\right), \kappa; c_2 \Big) \exp\Big(\big(\boldsymbol{\eta}_1^{\text{input}}\big)^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2^{\text{input}} \boldsymbol{x} \Big), \tag{5.26}$$

where $\boldsymbol{\eta}_1^{\text{input}}$ is a $d \times 1$ vector, $\boldsymbol{H}_2^{\text{input}}$ is a $d \times d$ matrix, $c_0 = \boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}}$, $c_2 = y_{ij}$, $\boldsymbol{c}_1 = \boldsymbol{x}_{ij}^{\mathbf{R}}$ and $\boldsymbol{x} = \boldsymbol{u}_i$. As such, we seek an $\boldsymbol{\eta}^*$ to solve

$$\int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \text{NB}\Big(\exp\left(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}\right), \kappa; c_2\Big) \exp\left\{ \begin{bmatrix} \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta}^{\text{input}} \right\} d\boldsymbol{x} = \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k} \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{x} \\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \boldsymbol{\eta}^* \right\} d\boldsymbol{x} \tag{5.27}$$

where $\boldsymbol{x}^{\otimes k}$ is as defined in equation (1.2). Thus to obtain the required projection, we first obtain the optimal natural parameters $\boldsymbol{\eta}_1^*$ and $\boldsymbol{\eta}_2^*$ to project onto the multivariate normal family as is presented in Result 26. As the integrals required to do so are not available in closed form solutions, we use the same properties as the expit and Poisson models to express the multivariate integrals as univariate integrals. These integrals are expressed in Section 2.1 of Kim & Wand (2018)[32] as denoted by $\mathcal{C}_b(p, q, r)$ and solved using quadrature in a more efficient manner.

Result 26. *Given f_{input} follows the form of equation (5.26), the projection onto the multivariate normal family is given by*

$$proj_N[f_{input}] = \exp\Big(\boldsymbol{T}(\boldsymbol{x})^\top \boldsymbol{\eta}_{-1}^* - A(\boldsymbol{\eta}_{-1}^*)\Big) h(\boldsymbol{x}),$$

where

$$\boldsymbol{\eta}_{-1}^* \equiv K_{NB}\big(\boldsymbol{\eta}_{-1}^{input}; c_0, \boldsymbol{c}_1, c_2; \kappa\big),$$

$$\boldsymbol{\eta}_{-1}^{input} \equiv \begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}, \qquad \boldsymbol{\eta}_{-1}^* \equiv \begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix},$$

$K_{NB}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2; \kappa \right)$ *is defined in Definition 27 and $\boldsymbol{T}(\boldsymbol{x})$ and $h(\boldsymbol{x})$ follow from Section 1.5.2.2.*

Definition 27. *For primary arguments* $\boldsymbol{a}_1$ $(d \times 1)$ *and* $\boldsymbol{a}_2$ $\left(\frac{1}{2}d(d+1) \times 1\right)$ *such that* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$ *is symmetric and positive definite, and auxiliary arguments* $c_0, \in \mathbb{R}$, $c_2, \in \mathbb{Z}_{\geq 0}$, $\kappa \in \mathbb{R}_{\geq 0}$ *and* $\boldsymbol{c}_1$ $(d \times 1)$, *the function* $K_{NB} : H \to H$ *is given by*

$$K_{NB}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix}; c_0, \boldsymbol{c}_1, c_2, \kappa\right) \equiv \begin{bmatrix}\mathbf{R}_5^\top(\boldsymbol{a}_1 + r_3\boldsymbol{c}_1)\\ \boldsymbol{D}_d^\top vec(\mathbf{R}_5^\top \boldsymbol{A}_2)\end{bmatrix}, \tag{5.28}$$

where

$$r_1 = \boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\ \boldsymbol{c}_1, \quad r_2 = (\boldsymbol{c}_1^\top \boldsymbol{A}_2^{-1}\boldsymbol{a}_1 - 2c_0)r_7, \quad r_3 = r_2 + 2\ \mathcal{C}_{b,1:0}(r_6, r_7, \kappa)r_7,$$

$$r_4 = 2\big(\mathcal{C}_{b,1:0}(r_6, r_7, \kappa)^2 - \mathcal{C}_{b,2:0}(r_6, r_7, \kappa) - r_1\big)r_7^2, \quad \boldsymbol{R}_5 = (\boldsymbol{A}_2 + r_4\boldsymbol{c}_1\boldsymbol{c}_1^\top)^{-1}\boldsymbol{A}_2,$$

$$r_6 = 1 - r_2, \quad r_7 = -r_1^{-1}, \quad \mathcal{C}_{b,1:0}(r_6, r_7, \kappa) = \frac{\mathcal{C}_b(1, r_6, r_7, \kappa)}{\mathcal{C}_b(0, r_6, r_7, \kappa)},$$

$$\mathcal{C}_{b,2:0}(r_6, r_7, \kappa) = \frac{\mathcal{C}_b(2, r_6, r_7, \kappa)}{\mathcal{C}_b(0, r_6, r_7, \kappa)}, \quad b(x; \kappa) = (c_2 + \kappa)\log\big(\exp(x) + \kappa\big)$$

and $\mathcal{C}_b(k, r, q)$ *follows from equation (5.42).*

A proof of Definition 27 is given in Appendix 5.6.4. Using Result 26 we now obtain the normalising natural parameter η_0^* to find the projection onto unnormalised normal family.

5.4.1.1 Projection onto the unnormalised multivariate normal family

We now obtain the projection onto the unnormalised multivariate normal family. Recall the moment matching problem from equation (5.27) and Results 12 and 13. Then, we require

$$C_f = \int_{\mathbb{R}^d} f_{\text{input}}(x)dx = \mathcal{C}_b(0, r_6, r_7)Z_0Z_1,$$

where

$$Z_1 = \exp\left(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi) + \log\Gamma(c_2 + \kappa) + \kappa\log\kappa - \log\Gamma(c_2 + 1) - \log\Gamma(\kappa)\right),$$

$Z_0 = \exp\left((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\right)$ and r_6 and r_7 are given in Definition 27. Analogous to previous arguments

$$\begin{aligned}\boldsymbol{\eta}_0^* = &\log \mathcal{C}_b(0, r_6, r_7) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) + \frac{1}{4}(\boldsymbol{\eta}_1^*)^\top (\boldsymbol{H}_2^*)^{-1}\boldsymbol{\eta}_1^* \\ &- \frac{1}{4}(\boldsymbol{\eta}_1^{\text{input}})^\top (\boldsymbol{H}_2^{\text{input}})^{-1}\boldsymbol{\eta}_1^{\text{input}} + \frac{1}{2}\log\left(|\boldsymbol{H}_2^*|/|\boldsymbol{H}_2^{\text{input}}|\right) + \log\Gamma(c_2+\kappa) \\ &+ \kappa\log\kappa - \log\Gamma(c_2+1) - \log\Gamma(\kappa).\end{aligned}$$

To find the normalising constant of the input function, we introduce a function given in Definition 28.

Definition 28. *Consider first, primary arguments* $\boldsymbol{a}_1$ *and* $\boldsymbol{b}_1$ *and auxiliary argument* $\boldsymbol{c}_1$ *where all three are* $d\times 1$*. Next consider arguments* $\boldsymbol{a}_2$ *and* $\boldsymbol{b}_2$ *which are all* $\left(\frac{1}{2}d(d+1)\times 1\right)$ *such that both* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$ *and* $vec^{-1}\left(-\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{b}_2\right)$ *are symmetric and positive definite. Finally note auxiliary scalar argument* $c_0 \in \mathbb{R}$*,* $c_2, \in \mathbb{Z}_{\geq 0}$ *and* $\kappa \in \mathbb{R}_{\geq 0}$*. Then the function* $C_{NB} : H \times H \to \mathbb{R}$ *is given by*

$$\begin{aligned}C_{NB}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}; c_0, \boldsymbol{c}_1, c_2, \kappa\right) \equiv &\log \mathcal{C}_b(0, r_6, r_7, \kappa) + (r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi) \\ &+ \frac{1}{4}\boldsymbol{b}_1^\top \boldsymbol{B}_2^{-1}\boldsymbol{b}_1 - \frac{1}{4}\boldsymbol{a}_1^\top \boldsymbol{A}^{-1}\boldsymbol{a}_1 + \frac{1}{2}\log\left(|\boldsymbol{B}_2|/|\boldsymbol{A}_2|\right) \\ &+ \log\Gamma(c_2+\kappa) + \kappa\log\kappa - \log\Gamma(c_2+1) - \log\Gamma(\kappa),\end{aligned}$$

where $\boldsymbol{A}_2 \equiv vec^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{a}_2\right)$*,* $\boldsymbol{B}_2 \equiv vec^{-1}\left(\left(\boldsymbol{D}_d^+\right)^\top \boldsymbol{b}_2\right)$*,* r_1*,* r_2*,* r_6 *and* r_7 *follow from Definition 27.*

In summary, the projection onto the unnormalised multivariate normal family is given by Result 27.

Result 27. *For an unnormalised input function following the form of equation (5.26),*

$$proj_{\boldsymbol{UN}}\left[f_{input}\right](x) = \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{x} \\ vec(\boldsymbol{x}\boldsymbol{x}^\top) \end{bmatrix}^\top \begin{bmatrix} \eta_0^* \\ \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}\right\},$$

where

$$\begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix} = K_{NB}\left(\begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2, \kappa\right)$$

and

$$\eta_0^* = C_{NB}\left(\begin{bmatrix} \boldsymbol{\eta}_1^{input} \\ \boldsymbol{\eta}_2^{input} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\eta}_1^* \\ \boldsymbol{\eta}_2^* \end{bmatrix}; c_0, \boldsymbol{c}_1, c_2, \kappa\right).$$

5.4.1.2 Message passing formulation

Only minor changes to Algorithm 6 are required to account for the negative binomial case. Note the components of the sum of the log-likelihood function $\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa)$ are

$$\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \left(\prod_{j=1}^{n_i} p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta})\right) p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) d\boldsymbol{u}_i, \tag{5.29}$$

where

$$p(y_{ij}|\boldsymbol{u}_i; \boldsymbol{\beta}) \equiv \mathrm{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), \kappa; y_{ij}\Big)$$

and

$$p(\boldsymbol{u}_i; \boldsymbol{\Sigma}) \equiv |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i\right)$$

are the conditional density functions of each response given its random effect and the density function of that random effect respectively.

As the dependence structure of the product in equation (5.14) is the same as the probit model shown in Figure 3.1, we only need to update the messages depending on the conditional density functions of each response given its random effect. This involves

updating messages $m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i)$ as in equation (3.12)

$$m_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}(\boldsymbol{u}_i) \leftarrow \frac{\text{proj}_{\text{UN}}\left[\text{NB}\Big(\exp\left(c_0+\boldsymbol{c}_{1_{ij}}^\top\boldsymbol{u}_i\right),\kappa;c_{2_{ij}}\Big)\exp\left\{\boldsymbol{u}_i^\top\boldsymbol{\eta}_1^{\otimes}+\left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top\boldsymbol{\eta}_2^{\otimes}\right\}\right]}{\exp\left\{\boldsymbol{u}_i^\top\boldsymbol{\eta}_1^{\otimes}+\left(\text{vech}(\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right)^\top\boldsymbol{\eta}_2^{\otimes}\right\}},$$

where we set the constant terms:

$$c_{0_{ij}} \leftarrow \boldsymbol{\beta}^\top\boldsymbol{x}_{ij}^{\mathbf{F}}; \quad \boldsymbol{c}_{1_{ij}} \leftarrow \boldsymbol{x}_{ij}^{\mathbf{R}}; \quad c_{2_{ij}} \leftarrow y_{ij}; \quad 1\leq i\leq m, \quad 1\leq j\leq n_i.$$

Following Section 3.1.2, we see that this task reduces down to adjusting the calculation of the optimal natural parameters $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ in equation (3.18) for the Poisson model, where the linear and quadratic coefficient updates in equation (3.19) given by K_{probit} are changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2} \leftarrow K_{\text{NB}}\left(\boldsymbol{\eta}_{1:2}^{\otimes};c_{0_{ij}},\boldsymbol{c}_{1_{ij}},c_{2_{ij}},\kappa\right)-\boldsymbol{\eta}_{1:2}^{\otimes} \tag{5.30}$$

as per Definition 27, and where the constant coefficient update in equation (3.20) given by C_{probit} is changed to

$$\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_0 \leftarrow C_{\text{NB}}\left(\boldsymbol{\eta}_{1:2}^{\otimes},\left(\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}\right)_{1:2}+\boldsymbol{\eta}_{1:2}^{\otimes};c_{0_{ij}},\boldsymbol{c}_{1_{ij}},\boldsymbol{c}_{2_{ij}},\kappa\right)$$

as per Definition 28.

Barring the aforementioned changes and the initialisation of $\boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})\to\boldsymbol{u}_i}$ with the ones discussed in Section 5.4.1.3, the full algorithm for the approximation of $\ell(\boldsymbol{\beta},\boldsymbol{\Sigma})$ follows Algorithm 6 exactly.

5.4.1.3 Starting values for the negative binomial case

We now derive starting values for the negative binomial case using the same principles shown in the other cases. Let

$$\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta}) = f(a_{ij})+\log\Gamma(y_{ij}+\kappa)-\log\Gamma(y_{ij}+1)-\log\Gamma(\kappa)+\kappa\log\kappa,$$

where $a_{ij}=\boldsymbol{\beta}^\top\boldsymbol{x}_{ij}^{\mathbf{F}}+\boldsymbol{u}_i^\top\boldsymbol{x}_{ij}^{\mathbf{R}}$ and $f(x)$ is defined as in equation (5.23). Then

$$(a_{ij}-\widehat{a}_{ij}) = (\boldsymbol{u}_i-\widehat{\boldsymbol{u}}_i)^\top\boldsymbol{x}_{ij}^{\mathbf{R}}$$

where $\widehat{a}_{ij} = \boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \widehat{\boldsymbol{u}}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}$ and $\widehat{\boldsymbol{u}}$ is an approximation of $\boldsymbol{u}_i$. A Taylor series expansion of $f(a_{ij})$ evaluated at $\widehat{a}_{ij}$ leads to

$$\begin{aligned} f(a_{ij}) &= f(\widehat{a}_{ij}) + f'(\widehat{a}_{ij})(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}} + \frac{1}{2} f''(\widehat{a}_{ij})\big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\big)^2 + \dots \\ &= \begin{bmatrix} 1 \\ \boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i \\ \operatorname{vech}\big((\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)(\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i)^\top\big) \end{bmatrix}^\top \breve{\boldsymbol{\eta}}_{ij} + \dots , \end{aligned}$$

where

$$\breve{\boldsymbol{\eta}}_{ij} = \begin{bmatrix} f(\widehat{a}_{ij}) \\ f'(\widehat{a}_{ij}) \boldsymbol{x}_{ij}^{\mathbf{R}} \\ \frac{1}{2} f''(\widehat{a}_{ij}) \boldsymbol{D}_{d^{\mathbf{R}}}^\top \operatorname{vec}\big(\boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top\big) \end{bmatrix}.$$

Quadratic approximation to $\log p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ based on Taylor expansion about $\widehat{\boldsymbol{u}}_i$ is $\log\ \breve{p}(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})$ as per equation (3.22). By following the same logic of previous cases it is easy to show

$$\boldsymbol{\eta}^{\text{start}}_{p(y_{ij}|\boldsymbol{u}_i;\boldsymbol{\beta})} = \begin{bmatrix} \eta_0^{\text{start}} \\ f'(\widehat{a}_{ij}) \boldsymbol{x}_{ij}^{\mathbf{R}} - f''(\widehat{a}_{ij}) \boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top \widehat{\boldsymbol{u}}_i \\ \frac{1}{2} f''(\widehat{a}_{ij}) \boldsymbol{D}_{d^{\mathbf{R}}}^\top \operatorname{vec}\big(\boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top\big) \end{bmatrix},$$

where

$$\eta_0^{\text{start}} = f(\widehat{a}_{ij}) - f'(\widehat{a}_{ij})(\boldsymbol{x}_{ij}^{\mathbf{R}})^\top \widehat{\boldsymbol{u}}_{ij} + \frac{1}{2} f''(\widehat{a}_{ij}) \widehat{\boldsymbol{u}}_{ij}^\top \boldsymbol{x}_{ij}^{\mathbf{R}} (\boldsymbol{x}_{ij}^{\mathbf{R}})^\top \widehat{\boldsymbol{u}}_{ij}.$$

5.4.2 Computation of point estimates and confidence intervals

We denote the maximum likelihood approximation using EP as $\widehat{\boldsymbol{\beta}}, \widehat{\underset{\sim}{\boldsymbol{\Sigma}}}, \widehat{\kappa} = \underset{\boldsymbol{\beta},\underset{\sim}{\boldsymbol{\Sigma}},\kappa}{\operatorname{argmax}}\, \underset{\sim}{\ell}(\boldsymbol{\beta}, \underset{\sim}{\boldsymbol{\Sigma}}, \kappa)$. As before, to find their stationary points, we use the Nelder-Mead and BFGS algorithms for optimisation via the R function `optim()` in the package "stats" (R Core Team, 2019[56]). As in the κ-only model, additional care must be taken to ensure optimisation of the parameter κ is on an unconstrained space. We do this using a log transform as in the univariate case i.e. $\xi = \log(\kappa)$. For clarity we now show the changes required from the steps given in Section 3.2:

1. Steps (a) - (b) for converting $\boldsymbol{\Sigma}$ to the unconstrained space $\boldsymbol{\theta}$ follows from before.

We must also now convert κ to the unconstrained space ξ.

(c) Convert κ to the unconstrained space ξ

$$\xi = \log(\kappa).$$

2. A quasi-Newton optimisation method can now be used to obtain the maximum likelihood estimate of $\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \xi\right)$,

$$\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\xi}\right) = \underset{\theta \in \mathbb{R}^{d^{\boldsymbol{\Sigma}}(d^{\boldsymbol{\Sigma}}+1)/2}}{\operatorname{argmax}} \ell(\boldsymbol{\beta}, \boldsymbol{\theta}, \xi).$$

We suggest conducting an initial search via the Nelder-Mead method, with refinements by BFGS algorithm. Both can be implemented via the `optim()` R function in the "stats" package (R Core Team, 2019[56]).

3. The step converting $\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}\right)$ to $\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}\right)$ follows from before.

4. Now obtain the Hessian matrix $H\ell\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\xi}\right)$ at the maximum $\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\xi}\right)$ using the quasi-Newton method BFGS, which as before can be implemented via `optim()`. Although we need values of $(\boldsymbol{\beta}, \boldsymbol{\omega}, \xi)$ to be returned in the Hessian, the constraints on these parameters mean the Hessian should still be calculated on the $(\boldsymbol{\beta}, \boldsymbol{\theta}, \xi)$ space, which is computed as before.

5. Form $100(1-\alpha)\%$ confidence intervals for the entries of $(\boldsymbol{\beta}, \boldsymbol{\omega})$ using

$$\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\omega}} \\ \widehat{\xi} \end{bmatrix} \pm \Phi^{-1}\left(1 - \frac{1}{2}\alpha\right) \sqrt{-\operatorname{diag}\left\{\left(\boldsymbol{H}\ell\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\xi}\right)\right)^{-1}\right\}}.$$

6. Back transform the confidence interval limits for $\boldsymbol{\omega}$ corresponding to the standard deviation and correlation parameters as before and back transform ξ to κ by exponentiating it.

5.4.3 Simulation study

We now provide a simulation study repeated 1000 times comparing EP to a quadrature approach and Laplace approximation. The simulation study in this chapter does not assess the speed component.

Datasets were generated with true parameter values:

$$\beta_{\text{true}} = [0.38, 0.93]^\top, \quad \sigma^2_{\text{true}} = 0.25 \quad \text{and} \quad \kappa_{\text{true}} = 15. \tag{5.31}$$

There were 100 groups generated in the data with each group containing 5 measurements (i.e. $m = 100$, $n = 5$). The $\boldsymbol{x}^{\mathbf{F}}_{ij}$ and $\boldsymbol{x}^{\mathbf{R}}_{ij}$ vectors were of the form

$$\boldsymbol{x}^{\mathbf{F}}_{ij} = [1, x_{1_{ij}}]^\top \quad \text{and} \quad \boldsymbol{x}^{\mathbf{R}}_{ij} = 1,$$

where $x_{k,ij}$ was generated independently from a uniform distribution on the unit interval. The tolerance of error for the EP scheme was set to 10^{-5} and 1000 iterations were allowed for optimisation. We compare the EP approach to the Laplace approximation and 100 point adaptive Gauss-Hermite quadrature. Both alternative approaches were implemented via the R function `glmer()` from the package "lme4" (Bates, et al., 2018[5]).

Point estimates and 95% confidence intervals for each interpretable model parameter of the study are presented in Figure 5.4, where the number in the upper-right hand corner of each panel are the empirical coverage values based on all 1000 replicates. Only 20 randomly chosen replicates from each method are shown in the panels for ease of viewing. Laplace approximation and EP are shown in black, super imposed on adaptive Gauss-Hermite quadrature approach shown in grey.

None of the three methods provided over 95% empirical coverage for the fixed intercept. Noteably they were within 0.3% of the highest coverage, which was EP at 94.4%. Although for the fixed slope all methods had coverage above 95%, the quadrature aproach delivered the highest coverage of 95.9%, 0.4% above the coverage of EP and 0.7% above Laplace approximations. EP provided the best estimates of 94.3% for the random intercept parameter by some margin. Surprisingly, quadrature provided coverage than Laplace approximation for this parameter, with 81.3% opposed to 92.8%.

Neither Laplace approximation nor Gauss-Hermite quadrature support confidence interval calculation for the shape parameter via the R function `glmer()`. Differences between the point estimates of Laplace approximation and adaptive Gauss-Hermite quadrature exist for small κ values, although for larger values they are minimal. Point estimates of EP differ from adaptive Gauss-Hermite quadrature across the range of values, however match with Laplace approximation values for small values of κ. Although confidence intervals via EP provides reasonable coverage of 94.4%, the large width of intervals may not lend itself well to meaningful inference. It is left as an open topic for future research.

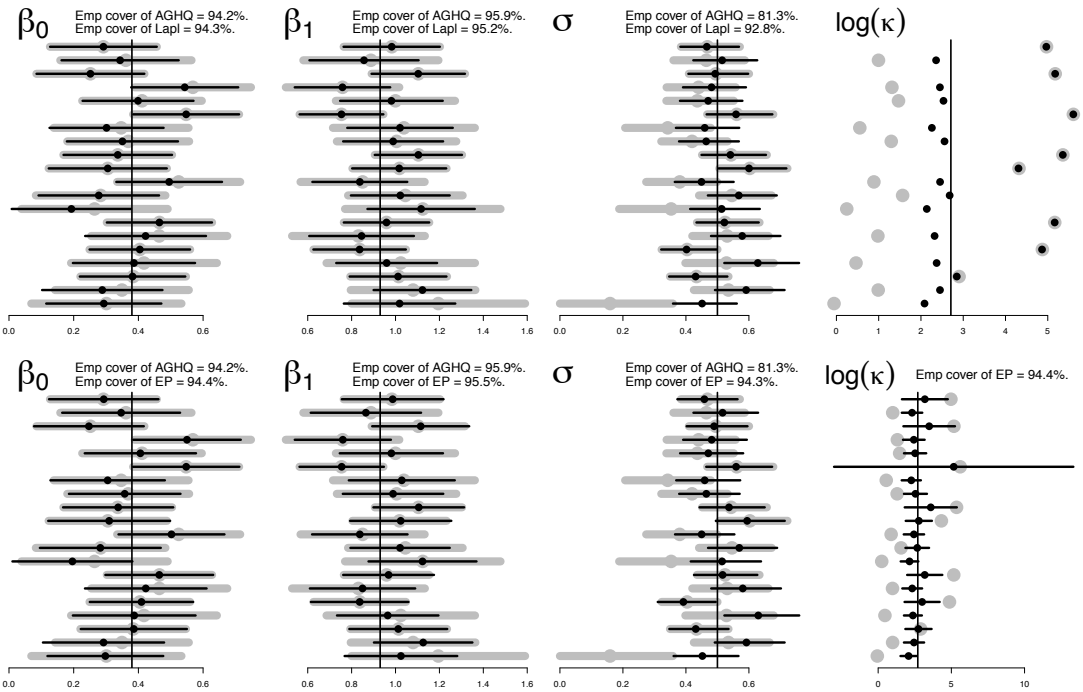


Figure 5.4: *Plots of confidence interval coverage for the univariate model with true parameter values from equation (5.31) for models fit with adaptive Gauss-Hermite quadrature, Laplace approximation and EP. The horizontal lines are the confidence intervals for 20 randomly chosen replications of the simulation study, the solid circular points indicate the point estimates and the vertical lines indicate true parameter values. The Laplace and EP are shown in black, superimposed over the grey adaptive Gauss-Hermite quadrature estimates. The percentage given in the top right-hand corner of each panel is the empirical coverage over all* 1000 *replications.*

5.5 Varying dispersion negative binomial model

We give an indication of how the proposed general negative binomial model can be extended to the case of varying dispersion, where each group has its own shape parameter. Rather than estimating a unique κ for each group, we instead estimate an error term ν_i for each group and multiply κ by the exponent of this term. The form of this model is

$$y_{ij}|\boldsymbol{u}_i \overset{\text{ind}}{\sim} \text{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}^{\mathbf{F}}_{ij} + \boldsymbol{u}_{ij}^\top \boldsymbol{x}^{\mathbf{R}}_{ij}\right), \kappa \exp(\nu_i)\Big), \quad \boldsymbol{u}_i \overset{\text{ind}}{\sim} \boldsymbol{N}\left(\mathbf{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}\right), \quad \nu_i \overset{\text{ind}}{\sim} N\left(0, \sigma^2_\nu\right),$$
$$1 \leq i \leq m, \quad 1 \leq j \leq n_i \quad \kappa > 0, \quad \text{and} \quad \nu_i \in \mathbb{R},$$

where the notation follows the general one level model presented in Section 1.8. The log-likelihood can be expressed as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa, \sigma_{\nu}^2) = \sum_{i=1}^{m} \ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa, \sigma_{\nu}^2),$$

where

$$\begin{aligned}\ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \kappa, \sigma_{\nu}^2) = \log \int_{\mathbb{R}^{d^{\mathbf{R}}}} \int_{-\infty}^{\infty} \Big\{ \prod_{j=1}^{n_i} \mathrm{NB}\Big(\exp\left(\boldsymbol{\beta}^\top \boldsymbol{x}_{ij}^{\mathbf{F}} + \boldsymbol{u}_i^\top \boldsymbol{x}_{ij}^{\mathbf{R}}\right), \kappa; y_{ij} \Big) \Big\} \\ \times |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\Big(-\frac{1}{2} \boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_i \Big) (2\pi\sigma_{\nu}^2)^{-1/2} \exp\Big(-\frac{\nu_i^2}{2\sigma_{\nu}^2} \Big) d\nu_i d\boldsymbol{u}_i.\end{aligned} \tag{5.32}$$

Although it seems that the extention of our methodology to this model is complicated, the likelihood as in equation (5.32) can be obtained by message passing on the updated factor graph in Figure 5.5 in a similar way to the general negative binomial model. We leave further development of this model for future research.

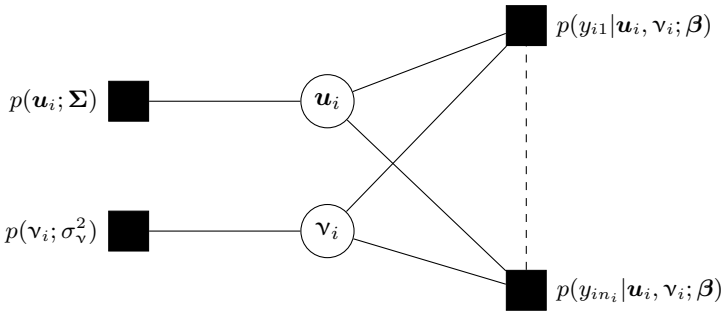


Figure 5.5: *Factor graph representation of the product structure of the integrand in equation (5.32).*

5.6 Appendix

5.6.1 Proof of Definition 21

From this point on we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. The kth moment of the input function in equation (5.2) is given by

$$\mathcal{M}_k = \int_{-\infty}^{\infty} x^k \Psi(c_0 + c_1 x; c_2)(2\pi)^{-1/2} \exp\left\{ \begin{bmatrix} x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta} - A(\boldsymbol{\eta}) \right\} dx$$
$$\times (2\pi)^{1/2} \exp\big(A(\boldsymbol{\eta})\big)\Gamma(c_2+1)^{-1},$$

where $\Psi(x;c) = \exp\big(cx - b(x)\big)$ and $b(x) = \exp(x)$. Using the inverse map of the natural parameters in equation (1.14) and implementing the change of variable $x = \mu + \sigma u$,

$$\mathcal{M}_k = \int_{-\infty}^{\infty} (\mu + \sigma u)^k \Psi\big(c_0 + c_1\mu + \sigma c_1 u; c_2\big)\phi(u)du Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta}) + \frac{1}{2}\log(2\pi) - \log\Gamma(c_2+1)\big)$. Each of the required moments can be obtained via univariate quadrature. For numerical stability we rearrange the integral that arises in the form presented by Kim & Wand (2017),[32]

$$\mathcal{C}_b(k, r, q) = \int_{-\infty}^{\infty} x^k \exp\big(rx - qx^2 - b(x)\big)dx, \tag{5.33}$$

where $b(x) = \exp(x)$. Using simple algebraic manipulations, it is easy to show the integrals required for the calculation of each moment can be expressed as

$$\int_{-\infty}^{\infty} u^k \Psi(c_0 + c_1\mu + \sigma c_1 u; c_2)\phi(u)du$$
$$= \int_{-\infty}^{\infty} \big((r_2 + 2r_7 x)(2r_7)^{-1/2}\big)^k \exp\big(r_6 x - r_7 x^2 - b(x)\big)dx Z_0,$$

where

$$r_1 = -2\sigma^2 c_1^2, \quad r_2 = 2(c_0 + c_1\mu)r_1^{-1}, \quad r_6 = c_2 - r_2, \quad r_7 = -r_1^{-1},$$
$$b(x) = \exp(x) \quad \text{and} \quad Z_0 = \exp\big((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\big).$$

The explicit forms of the integrals required follow easily,

$$Z_0^{-1}\int_{-\infty}^{\infty} \Psi(c_0 + c_1\mu + \sigma c_1 u; c_2)\phi(u)du = \mathcal{C}_b(0, r_6, r_7),$$
$$Z_0^{-1}\int_{-\infty}^{\infty} u\ \Psi(c_0 + c_1\mu + \sigma c_1 u; c_2)\phi(u)du$$
$$= \big(r_2\ \mathcal{C}_b(0, r_6, r_7) + 2r_7\mathcal{C}_b(1, r_6, r_7)\big)(2r_7)^{-1/2},$$
$$Z_0^{-1}\int_{-\infty}^{\infty} u^2\ \Psi(c_0 + c_1\mu + \sigma c_1 u; c_2)\phi(u)du$$
$$= 2\big(r_2\mathcal{C}_b(1, r_6, r_7) + r_7\mathcal{C}_b(2, r_6, r_7)\big) - \frac{r_2^2 r_1}{2}\mathcal{C}_b(0, r_6, r_7).$$

Implementing these forms, each of the moments are

$$
\begin{aligned}
Z_0^{-1}Z_1^{-1}\mathcal{M}_0 &= \mathcal{C}_b(0, r_6, r_7),\\
Z_0^{-1}Z_1^{-1}\mathcal{M}_1 &= \mu\mathcal{C}_b(0, r_6, r_7) + \sigma\big(r_2\mathcal{C}_b(0, r_6, r_7) + 2r_7\mathcal{C}_b(1, r_6, r_7)\big)(2r_7)^{-1/2},\\
Z_0^{-1}Z_1^{-1}\mathcal{M}_2 &= (\mu^2+\sigma^2)\mathcal{C}_b(0, r_6, r_7) + 2\mu c_1\sigma^2\big(r_2\mathcal{C}_b(0, r_6, r_7) + 2r_7\mathcal{C}_b(1, r_6, r_7)\big)\\
&\quad + \sigma^4 c_1^2\big(4r_7^2\mathcal{C}_b(2, r_6, r_7) + 4r_7r_2\mathcal{C}_b(1, r_6, r_7) + (r_2^2 - 2r_7)\mathcal{C}_b(0, r_6, r_7)\big).
\end{aligned}
$$

Letting $\mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}$ and $\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}$, the optimal mean parameter for the projection is

$$
\mu^* = E(x) = \frac{\mathcal{M}_1}{\mathcal{M}_0} = \mu + \sigma^2 c_1\big(r_2 + 2r_7\mathcal{C}_{b,1:0}(r_6, r_7)\big),
$$

and the optimal variance parameter is

$$
\begin{aligned}
(\sigma^2)^* =& E(x^2) - E(x)^2 = \frac{\mathcal{M}_2}{\mathcal{M}_0} - \left(\frac{\mathcal{M}_1}{\mathcal{M}_0}\right)^2\\
=& \sigma^2 + (2r_7)^2\sigma^4 c_1^2\big(\mathcal{C}_{b,2:0}(r_6, r_7) - \mathcal{C}_{b,1:0}(r_6, r_7)^2 + r_1/2\big).
\end{aligned}
$$

By converting back to natural optimal and input parameters we arrive at Definition 17.

5.6.2 Proof of Definition 23

Using simple algebraic manipulations based on Lemma 2, we arrive at Lemma 5:

Lemma 5. *For integrals of the forms listed below, the corresponding solutions exist:*

$$
\int_{\mathbb{R}^d} \Psi\big(a + \boldsymbol{b}^\top\boldsymbol{x}; c\big)\phi(\boldsymbol{x})d\boldsymbol{x} = \int_{-\infty}^{\infty} \Psi\big(a + \|\boldsymbol{b}\|x; c\big)\phi(x)dx, \tag{5.34}
$$

$$
\int_{\mathbb{R}^d} \boldsymbol{x}\ \Psi\big(a + \boldsymbol{b}^\top\boldsymbol{x}; c\big)\phi(\boldsymbol{x})d\boldsymbol{x} = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}\int_{-\infty}^{\infty} x\ \Psi\big(a + \|\boldsymbol{b}\|x; c\big)\phi(x)dx, \tag{5.35}
$$

$$
\begin{aligned}
\int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^\top\ \Psi\big(a + \boldsymbol{b}^\top\boldsymbol{x}; c\big)\phi(\boldsymbol{x})d\boldsymbol{x} &= I_d\int_{-\infty}^{\infty} \Psi\big(a + \|\boldsymbol{b}\|x; c\big)\phi(x)dx \\
&\quad + \frac{\boldsymbol{b}\boldsymbol{b}^\top}{\boldsymbol{b}^\top\boldsymbol{b}}\left(\int_{-\infty}^{\infty} x^2\Psi\big(a + \|\boldsymbol{b}\|x; c\big)\phi(x)dx - \int_{-\infty}^{\infty} \Psi\big(a + \|\boldsymbol{b}\|x; c\big)\phi(x)dx\right),
\end{aligned} \tag{5.36}
$$

where $a \in \mathbb{R}$, $\boldsymbol{b}$ is a $d \times 1$ vector, $c \in \mathbb{Z}^+$, $\Psi(x; c) = \exp\big(cx - b(x)\big)$ and $b(x) = \exp(x)$.

We wish to obtain the projection of an input function following the form of equation (5.11) onto the multivariate normal family. Note in the interest of brevity we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. Using the $\boldsymbol{x}^{\otimes k}$ notation as described in equation (1.2), the kth moment of the input function is given by

$$
\begin{aligned}
\mathcal{M}_k \quad = \quad & \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k}\Psi\big(c_0 + \boldsymbol{c}_1^\top\boldsymbol{x}; c_2\big)(2\pi)^{-d/2}\exp\left\{\begin{bmatrix}\boldsymbol{x}\\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top)\end{bmatrix}^\top\boldsymbol{\eta} - A(\boldsymbol{\eta})\right\}d\boldsymbol{x}\\
& \times(2\pi)^{d/2}\exp\big(A(\boldsymbol{\eta})\big)\Gamma(c_2+1)^{-1}.
\end{aligned}
$$

Using the inverse map of the natural parameters in equation (1.14) and implementing a change of variable, where $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{u}$,

$$\mathcal{M}_k \quad = \quad \int_{\mathbb{R}^d} \big(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{u}\big)^{\otimes k} \Psi\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + (\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1)^\top \boldsymbol{u}; c_2\big)\phi_{\boldsymbol{I}}(\boldsymbol{u}) d\boldsymbol{u} Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi) - \log\Gamma(c_2+1)\big)$. We can then use Lemma 5 to obtain each of the required moments via univariate quadrature. For numerical stability we rearrange the integral that arises in the form presented by Kim & Wand (2017)[32]

$$\mathcal{C}_b(k, r, q) = \int_{-\infty}^{\infty} x^k \exp\big(rx - qx^2 - b(x)\big) dx, \tag{5.37}$$

where $b(x) = \exp(x)$. Using simple algebraic manipulations, it is easy to show

$$\begin{aligned}
&\int_{-\infty}^{\infty} u^k \Psi\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\| u; c_2\big)\phi(u) du \\
&\qquad = \int_{-\infty}^{\infty} \big((r_2 + 2r_7 x)(2r_7)^{-1/2}\big)^k \exp\big(r_6 x - r_7 x - b(x)\big) dx Z_0,
\end{aligned}$$

where

$$r_1 = -2\boldsymbol{c}_1^\top \boldsymbol{\Sigma}\boldsymbol{c}_1, \quad r_2 = 2\big(c_0 + \boldsymbol{c}_1^\top\boldsymbol{\mu}\big)r_1^{-1}, \quad r_6 = c_2 - r_2, \quad r_7 = -r_1^{-1},$$

$$b(x) = \exp(x) \quad \text{and} \quad Z_0 = \exp\big((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\big).$$

It follows,

$$\begin{aligned}
Z_0^{-1} \int_{-\infty}^{\infty} \Psi\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\| u; c_2\big)\phi(u)\ du &= \mathcal{C}_b(0, r_6, r_7), \\
Z_0^{-1} \int_{-\infty}^{\infty} u\ \Psi\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\| u; c_2\big)\phi(u)\ du & \\
= \big(r_2\mathcal{C}_b(0, r_6, r_7) + 2r_7\mathcal{C}_b(1, r_6, r_7)\big)(2r_7)^{-1/2}, & \\
Z_0^{-1} \int_{-\infty}^{\infty} u^2\ \Psi\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\|\ u; c_2\big)\phi(u)\ du & \\
= 2\big(r_2\mathcal{C}_b(1, r_6, r_7) + r_7\mathcal{C}_b(2, r_6, r_7)\big) - \frac{r_2^2 r_1}{2}\mathcal{C}_b(0, r_6, r_7). &
\end{aligned}$$

Implementing these forms, each of the moments can be derived in a similar way to the probit cases. For the zeroth moment where $k = 0$,

$$\begin{aligned}
Z_0^{-1} Z_1^{-1} \mathcal{M}_0 &= \mathcal{C}_b(0, r_6, r_7), \\
Z_0^{-1} Z_1^{-1} \mathcal{M}_1 &= \boldsymbol{\mu}\mathcal{C}_b(0, r_6, r_7) + \boldsymbol{\Sigma}^{1/2}\big(r_2\mathcal{C}_b(0, r_6, r_7) + 2r_7\mathcal{C}_b(1, r_6, r_7)\big)(2r_7)^{-1/2}, \\
Z_0^{-1} Z_1^{-1} \mathcal{M}_2 &= \big(\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma}\big)\mathcal{C}_b(0, r_6, r_7) \\
&\quad + \big(\boldsymbol{\mu}\boldsymbol{c}_1^\top\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{\mu}^\top\big)\big(r_2\mathcal{C}_b(0, r_6, r_7) + 2r_7\mathcal{C}_b(1, r_6, r_7)\big) \\
&\quad + \boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top\boldsymbol{\Sigma}\big(4r_7^2\mathcal{C}_b(2, r_6, r_7) + 4r_7 r_2\mathcal{C}_b(1, r_6, r_7) + (r_2^2 - 2r_7)\mathcal{C}_b(0, r_6, r_7)\big).
\end{aligned}$$

First, let $\mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}$ and $\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}$. Then the optimal mean parameter for the projection is

$$\boldsymbol{\mu}^* = E(\boldsymbol{x}) = \frac{\mathcal{M}_1}{\mathcal{M}_0} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{c}_1\big(r_2 + 2r_7\mathcal{C}_{b,1:0}(r_6, r_7)\big),$$

and the optimal variance parameter is

$$\begin{aligned}\boldsymbol{\Sigma}^* &= E(\boldsymbol{x}\boldsymbol{x}^\top) - E(\boldsymbol{x})E(\boldsymbol{x})^\top = \frac{\mathcal{M}_2}{\mathcal{M}_0} - \frac{\mathcal{M}_1}{\mathcal{M}_0}\left(\frac{\mathcal{M}_1}{\mathcal{M}_0}\right)^\top \\ &= \boldsymbol{\Sigma} + (2r_7)^2\boldsymbol{\Sigma}\boldsymbol{c}_1\boldsymbol{c}_1^\top\boldsymbol{\Sigma}\big(\mathcal{C}_{b,2:0}(r_6, r_7) - \mathcal{C}_{b,1:0}(r_6, r_7)^2 + r_1/2\big),\end{aligned}$$

where $\mathcal{C}_{b,1:0}(r_6, r_7) = \frac{\mathcal{C}_b(1,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}$ and $\mathcal{C}_{b,2:0}(r_6, r_7) = \frac{\mathcal{C}_b(2,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}$. By converting back to natural optimal and input parameters we arrive at Definition 17.

5.6.3 Proof of Definition 25

We wish to obtain the projection of an input function following the form of equation (5.18) onto the univariate normal family. Note in the interest of brevity we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. Using the x^k notation as described in equation (1.2), the kth moment of the input function is given by

$$\begin{aligned}\mathcal{M}_k = \int_{\mathbb{R}} x^k \Upsilon\big(c_2; c_0 + c_1^\top x, \kappa\big)(2\pi)^{-1/2} \exp\left\{\begin{bmatrix} x \\ x^2 \end{bmatrix}^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\right\} dx \\ \times \frac{\Gamma(c_2+\kappa)\kappa^\kappa}{\Gamma(c_2+1)\Gamma(\kappa)}(2\pi)^{1/2}\exp\big(A(\boldsymbol{\eta})\big),\end{aligned}$$

where $\Upsilon(c; x; \kappa) = \exp\big(cx - b(x; c, \kappa)\big)$ and $b(c; x; \kappa) = (c+\kappa)\log\big(\exp(x) + \kappa\big)$. Using the inverse map of the natural parameters in equation (1.14) and implementing a change of variable $x = \mu + \sigma u$

$$\mathcal{M}_k = \int_{\mathbb{R}} (\mu + \sigma u)^k \Upsilon\big(c_2; c_0 + c_1\mu + (\sigma c_1)^\top u; \kappa\big)\phi(u)du Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta}) + \frac{1}{2}\log(2\pi) + \log(c_2+\kappa) + \kappa\log\kappa - \log\Gamma(c_2+1) - \log\Gamma(\kappa)\big)$. We can then use Lemma 6 to obtain each of the required moments via univariate quadrature. For numerical stability we rearrange the integral that arise in the form presented by Kim & Wand (2017)[32]

$$\mathcal{C}_b(k, r, q) = \int_{-\infty}^{\infty} x^k \exp\big(rx - qx^2 - b(x; c_2, \kappa)\big)dx. \tag{5.38}$$

Using simple algebraic manipulations, it is easy to show

$$\begin{aligned}&\int_{-\infty}^{\infty} u^k \Upsilon(c_0 + c_1\mu + \sigma c_1 u; c_2)\phi(u)du \\ &\qquad = \int_{-\infty}^{\infty} \big((r_2 + 2r_7x)(2r_7)^{-1/2}\big)^k \exp\big(r_6x - r_7x - b(x)\big)dx\ Z_0,\end{aligned}$$

where,

$$r_1 = -2c_1^\top \sigma c_1, \quad r_2 = 2(c_0 + c_1^\top \mu) r_1^{-1}, \quad r_6 = c_2 - r_2, \quad r_7 = -r_1^{-1},$$
$$b(x) = \exp(x) \quad \text{and} \quad Z_0 = \exp\left((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\right).$$

Using simple algebraic manipulations analogous to the Poisson case in Section 5.6.1 it is easy to arrive at Definition 25.

5.6.4 Proof of Definition 27

Using simple algebraic manipulations based on Lemma 2, we arrive at Lemma 6:

Lemma 6. *For integrals of the forms listed below, the corresponding solutions exist:*

$$\int_{\mathbb{R}^d} \Upsilon\big(c, a + \boldsymbol{b}^\top \boldsymbol{x}, \kappa\big)\phi(\boldsymbol{x})d\boldsymbol{x} = \int_{-\infty}^{\infty} \Upsilon\big(c, a + \|\boldsymbol{b}\|x, \kappa\big)\phi(x)dx, \tag{5.39}$$

$$\int_{\mathbb{R}^d} \boldsymbol{x}\Upsilon\big(c, a + \boldsymbol{b}^\top \boldsymbol{x}, \kappa\big)\phi(\boldsymbol{x})d\boldsymbol{x} = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}\int_{-\infty}^{\infty} x\ \Upsilon\big(c, a + \|\boldsymbol{b}\|x, \kappa\big)\phi(x)dx, \tag{5.40}$$

$$\int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^\top \Upsilon\big(c, a + \boldsymbol{b}^\top \boldsymbol{x}, \kappa\big)\phi(\boldsymbol{x})d\boldsymbol{x} = I_d \int_{-\infty}^{\infty} \Upsilon\big(c, a + \|\boldsymbol{b}\|x, \kappa\big)\phi(x)dx$$
$$+\frac{\boldsymbol{b}\boldsymbol{b}^\top}{\boldsymbol{b}^\top\boldsymbol{b}}\Big(\int_{-\infty}^{\infty} x^2\Upsilon\big(c, a + \|\boldsymbol{b}\|x, \kappa\big)\phi(x)dx - \int_{-\infty}^{\infty} \Upsilon\big(c, a + \|\boldsymbol{b}\|x, \kappa\big)\phi(x)dx\Big), \tag{5.41}$$

where $a \in \mathbb{R}$, $\boldsymbol{b}$ *is a* $d \times 1$ *vector,* $c \in \mathbb{Z}^+$, $\Upsilon(c, x, \kappa) = \exp\big(cx - b(c, x, \kappa)\big)$ *and* $b(c, x, \kappa) = (c + \kappa)\log\big(\exp(x) + \kappa\big)$.

We wish to obtain the projection of an input function following the form of equation (5.11) onto the multivariate normal family. Note in the interest of brevity we represent the input parameter $\boldsymbol{\eta}^{\text{input}}$ as $\boldsymbol{\eta}$. Using the $\boldsymbol{x}^{\otimes k}$ notation as described in equation (1.2), the kth moment of the input function is given by

$$\mathcal{M}_k = \int_{\mathbb{R}^d} \boldsymbol{x}^{\otimes k}\Upsilon\big(c_2, c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}, \kappa\big)(2\pi)^{-d/2}\exp\left\{\begin{bmatrix}\boldsymbol{x}\\ \text{vech}(\boldsymbol{x}\boldsymbol{x}^\top)\end{bmatrix}^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\right\}d\boldsymbol{x}$$
$$\times \frac{\Gamma(c_2 + \kappa)\kappa^\kappa}{\Gamma(c_2 + 1)\Gamma(\kappa)}(2\pi)^{d/2}\exp\big(A(\boldsymbol{\eta})\big).$$

Using the inverse map of the natural parameters in equation (1.14) and implementing a change of variable where $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{u}$,

$$\mathcal{M}_k = \int_{\mathbb{R}^d} (\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{u})^{\otimes k}\Upsilon\Big(c_2, c_0 + \boldsymbol{c}_1^\top\boldsymbol{\mu} + (\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1)^\top\boldsymbol{u}, \kappa\Big)\phi_{\boldsymbol{I}}(\boldsymbol{u})d\boldsymbol{u}Z_1,$$

where $Z_1 = \exp\big(A(\boldsymbol{\eta}) + \frac{d}{2}\log(2\pi) + \log(c_2 + \kappa) + \kappa\log\kappa - \log\Gamma(c_2 + 1) - \log\Gamma(\kappa)\big)$. We can then Lemma 6 to obtain each of the required moment via univariate quadrature.

For numerical stability we rearrange the integral that arises in the form presented by Kim & Wand (2017),[32]

$$\mathcal{C}_b(k, r, q) = \int_{-\infty}^{\infty} x^k \exp\big(rx - qx^2 - b(x)\big)dx, \tag{5.42}$$

where $b(x, \kappa) = (c_2 + \kappa)\log\big(\exp(x) + \kappa\big)$. Using simple algebraic manipulations, it is easy to show

$$\begin{aligned}\int_{-\infty}^{\infty} u^k \Upsilon\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu} + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{c}_1\|u, c_2\big)\phi(u)du \\ = \int_{-\infty}^{\infty} \big((r_2 + 2r_7 x)(2r_7)^{-1/2}\big)^k \exp\big(r_6 x - r_7 x - b(x)\big)dx \; Z_0,\end{aligned}$$

where

$$r_1 = -2\boldsymbol{c}_1^\top \boldsymbol{\Sigma}\boldsymbol{c}_1, \quad r_2 = 2\big(c_0 + \boldsymbol{c}_1^\top \boldsymbol{\mu}\big)r_1^{-1}, \quad r_6 = c_2 - r_2, \quad r_7 = -r_1^{-1},$$

$$b(x) = \exp(x) \quad \text{and} \quad Z_0 = \exp\left((r_2/2)^2 r_1 + \frac{1}{2}\log(r_7/\pi)\right).$$

By following the Poisson case in Section 5.6.2 it is easy to arrive at Definition 27.

Chapter 6

Expectation propagation for two level and crossed random effects probit models

Having explored models for both count and binary response data with one level of nesting, we now attempt to handle models for crossed random effects and two level structures. We work with probit models since they have closed form solutions as in Chapter 3, and show how the same key results from Chapter 3 are applicable to the higher level models in this chapter. We start by applying our work to the crossed random effects model in Section 6.1, before explaining how it can also be implemented for two level models in Section 6.2.

6.1 The general probit crossed mixed model

We first extend our methodology from the one level probit case to crossed random effects GLMMs. We aim to find approximations of the maximum likelihood estimates for parameters $\boldsymbol{\beta}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ with 95% confidence intervals.

The crossed random effects model specification is given in equation (6.1),

$$\begin{gathered} y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'} \overset{\text{ind.}}{\sim} \text{Bernoulli}\left\{\Phi\left(\boldsymbol{\beta}^\top \boldsymbol{x}^{\mathbf{F}}_{ii'j} + \boldsymbol{u}_i^\top \boldsymbol{x}^{\mathbf{R}}_{ii'j} + (\boldsymbol{u}'_{i'})^\top \boldsymbol{x}^{\mathbf{R}'}_{ii'j}\right)\right\}, \\ \boldsymbol{u}_i \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}}}, \boldsymbol{\Sigma}) \quad \text{independently of} \quad \boldsymbol{u}'_{i'} \overset{\text{ind.}}{\sim} \boldsymbol{N}(\boldsymbol{0}_{d^{\mathbf{R}'}}, \boldsymbol{\Sigma}'), \\ 1 \le i \le m, \quad 1 \le i' \le m', \quad 1 \le j \le n_{ii'}, \end{gathered} \tag{6.1}$$

The content of this chapter is published as: Hall, P., Johnstone, I. M., Ormerod, J. T., Wand, M. P., & Yu, J. C. F. (2020). Fast and Accurate Binary Response Mixed Model Analysis via Expectation Propagation, Journal of the American Statistical Association, 115:532, 1902–1916, DOI: 10.1080/01621459.2019.1665529

where the notation follows that of the crossed random effects model in Section 1.8.

The form of the log-likelihood is

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}') = \log \int_{\mathbb{R}^{md^{\mathbf{R}}+m'd^{\mathbf{R}'}}} \left(\prod_{(i,i'):n_{ii'}>0} \prod_{j=1}^{n_{ii'}} p(y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'}; \boldsymbol{\beta}) \right) p(\boldsymbol{u}, \boldsymbol{u}'; \boldsymbol{\Sigma}, \boldsymbol{\Sigma}') d \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{u}' \end{bmatrix}, \tag{6.2}$$

where

$$p(y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'}; \boldsymbol{\beta}) = \Phi\Big\{ (2y_{ii'j} - 1)\Big(\boldsymbol{\beta}^\top \boldsymbol{x}^{\mathbf{F}}_{ii'j} + \boldsymbol{u}_i^\top \boldsymbol{x}^{\mathbf{R}}_{ii'j} + (\boldsymbol{u}'_{i'})^\top \boldsymbol{x}^{\mathbf{R}'}_{ii'j}\Big) \Big\},$$

$$p(\boldsymbol{u}, \boldsymbol{u}'; \boldsymbol{\Sigma}, \boldsymbol{\Sigma}') \quad \text{is} \quad \boldsymbol{N}\left(\begin{bmatrix} \mathbf{0}_{md^{\mathbf{R}}} \\ \mathbf{0}_{m'd^{\mathbf{R}'}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{I}_m \otimes \boldsymbol{\Sigma} & \mathbf{0}_{md^{\mathbf{R}} \times m'd^{\mathbf{R}'}} \\ \mathbf{0}_{m'd^{\mathbf{R}'} \times md^{\mathbf{R}}} & \boldsymbol{I}_{m'} \otimes \boldsymbol{\Sigma}' \end{bmatrix} \right),$$

and

$$\boldsymbol{u} \equiv \left[\boldsymbol{u}_1^\top \dots \boldsymbol{u}_m^\top\right]^\top, \quad \boldsymbol{u}' \equiv \left[(\boldsymbol{u}'_1)^\top \dots (\boldsymbol{u}'_m)^\top\right]^\top.$$

The joining of the random effects vectors $\boldsymbol{u}$ and $\boldsymbol{u}'$ into one random vector from a single density function allows a form that is more amenable to EP. For likelihood inference with crossed random effects the integrals required grow with the number of random effects as well as the number of groups, i.e. when $d^{\mathbf{R}} = d^{\mathbf{R}'} = 1$ they are over the $m + m'$-dimensional space. Although it is possible to reduce the integral to m $d^{\mathbf{R}}$-dimensional inner intergrals and outer integrals over $\mathbb{R}^{m'd^{\mathbf{R}'}}$ (alternatively m' $d^{\mathbf{R}'}$-dimensional inner intergrals and outer integrals over $\mathbb{R}^{md^{\mathbf{R}}}$ is also possible), the dimension of the integral we wish to solve is still dependent on group size. Since the current form is more amenable to EP we do not conduct such a simplification here.

In this section we first explain the schematic of likelihood approximation using EP in Subsection 6.1.1, before exploring computation of point estimates and confidence intervals in Subsection 6.1.2. The results of our simulation studies are presented in Subsection 6.1.3.

6.1.1 Expectation propagation likelihood approximation

Although it seems that the extention of our methodology to crossed random effects is complicated, by expressing the likelihood as in equation (6.2) the updates for the projection onto the multivariate normal family from the general one level model apply to the crossed case. Specifically, EP for the crossed case centres around finding the

optimal natural parameters η_0, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ which minimise $\mathrm{KL}(f_{\text{input}} \parallel f_{\boldsymbol{UN}})$, where the approximating density $f_{\boldsymbol{UN}}$ is defined to be the unnormalised multivariate normal density in equation (3.4), and the target density $f_{\text{input}}(\boldsymbol{x})$ can be reparameterised to have the same form as the one level model

$$f_{\text{input}}(\boldsymbol{x}) = \Phi(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}) \exp\left((\boldsymbol{\eta}_1^{\text{input}})^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2^{\text{input}} \boldsymbol{x}\right), \tag{6.3}$$

where $\boldsymbol{\eta}_1^{\text{input}}$ is a $d \times 1$ vector, $\boldsymbol{H}_2^{\text{input}}$ is a $d \times d$ matrix, $c_0 = (2y_{ii'j} - 1)\boldsymbol{\beta}^\top \boldsymbol{x}_{ii'j}^{\mathbf{F}}$, $\boldsymbol{c}_1 = (2y_{ii'j} - 1)\left[(\boldsymbol{x}_{ii'j}^{\mathbf{R}})^\top \ (\boldsymbol{x}_{ii'j}^{\mathbf{R}'})^\top\right]^\top$ and $\boldsymbol{x} = \left[\boldsymbol{u}_i^\top \ (\boldsymbol{u}'_{i'})^\top\right]^\top$. It is easy to see that the moment matching problem required to find the optimal natural parameters to project onto the normalised multivariate normal family can be solved analogously to the one level probit model using Result 14, and the full projection onto the unnormalised multivariate normal family is given in Result 15.

6.1.1.1 Message passing formulation

The message passing scheme required for crossed random effects follows that of the one level model in Section 3.1.2, with minor ammendments. We expicitly show some of the updated messages here with the aim to clarify how the previous projection results are reused. Note the alternate expression of the prior

$$p(\boldsymbol{u}, \boldsymbol{u}'; \boldsymbol{\Sigma}, \boldsymbol{\Sigma}') \equiv \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \\ \operatorname{vech}\left(\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \end{bmatrix}^\top \right) \end{bmatrix}^\top \boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')} \right\},$$

where

$$\boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')} = \begin{bmatrix} -\frac{md^{\mathbf{R}}}{2}\log|2\pi\boldsymbol{\Sigma}| - \frac{m'd^{\mathbf{R}'}}{2}\log|2\pi\boldsymbol{\Sigma}'| \\ \mathbf{0}_{md^{\mathbf{R}}+m'd^{\mathbf{R}'}} \\ -\frac{1}{2}\boldsymbol{D}^\top_{m'd^{\mathbf{R}'}+md^{\mathbf{R}}} \operatorname{vec}\left(\begin{bmatrix} \boldsymbol{I}_m \otimes \boldsymbol{\Sigma} & \mathbf{0}_{md^{\mathbf{R}} \times m'd^{\mathbf{R}'}} \\ \mathbf{0}_{m'd^{\mathbf{R}'} \times md^{\mathbf{R}}} & \boldsymbol{I}_{m'} \otimes \boldsymbol{\Sigma}' \end{bmatrix}^{-1} \right) \end{bmatrix}. \tag{6.4}$$

The structure of equation (6.2) in factor graph form is shown in Figure 6.1, where the

circular stochastic node corresponds to the random vector $\left[\boldsymbol{u}^\top(\boldsymbol{u}')^\top\right]^\top$, the solid squares indicate the factor nodes and the dependencies of the factor nodes on the stochastic node are demonstrated through the edges.

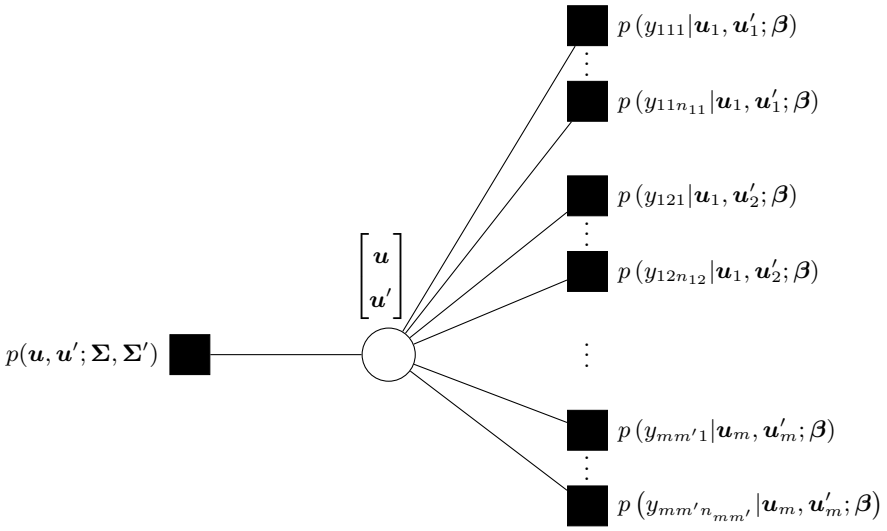


Figure 6.1: *Factor graph representation of the product structure of the integrand in equation (6.2) for crossed random effects models.*

Suppose that

$$p(y_{ii'j}|\boldsymbol{u}_i, \boldsymbol{u}'_{i'}; \boldsymbol{\beta}) = \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \\ \text{vech}\left(\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \end{bmatrix}^\top\right) \end{bmatrix}^\top \boldsymbol{\eta}_{ii'j}\right\}, \quad 1 \leq j \leq n_{ii'},$$

are initialised to be unnormalised multivariate normal density functions in $\left[\boldsymbol{u}_i^\top(\boldsymbol{u}'_{i'})^\top\right]^\top$. Then, for each $j = 1, ..., n_{ii'}$, the $\boldsymbol{\eta}_{ii'j}$ update of the EP algorithm involves minimisation

of

$$\mathrm{KL}\Bigg(p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta}) \prod_{j'\neq j}^{n_{ii'}} \underset{\sim}{p}(y_{ii'j'}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta}) \times p(\boldsymbol{u},\boldsymbol{u}';\boldsymbol{\Sigma},\boldsymbol{\Sigma}') \Big\| \prod_{(i,i'):n_{ii'}>0} \prod_{j'=1}^{n_{ii'}} \underset{\sim}{p}(y_{ii'j'}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta}) p(\boldsymbol{u},\boldsymbol{u}';\boldsymbol{\Sigma},\boldsymbol{\Sigma}') \Bigg) \tag{6.5}$$

as a function of $(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})$. As mentioned Result 15 can be used to update $\boldsymbol{\eta}_{ii'j}$ in an iterative procedure until convergence.

The process of minimising the KL divergence in equation (6.5) can be expressed as a message from the factor $p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})$ to the stochastic node $\left[\boldsymbol{u}_i^\top (\boldsymbol{u}'_{i'})^\top\right]^\top$ as

$$\begin{aligned} &m_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \\ &\quad \longleftarrow \frac{\mathrm{proj}_{\boldsymbol{UN}}\left[m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\; p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\right](\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}{m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}, \\ &\qquad 1\le i\le m,\quad 1\le i'\le m',\quad 1\le j\le n_{ii'}, \end{aligned} \tag{6.6}$$

and the update of the message passed from $p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')$ to $(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})$ is

$$\begin{aligned} &m_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \\ &\quad \longleftarrow \frac{\mathrm{proj}_{\boldsymbol{UN}}\left[m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\right](\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}{m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}. \end{aligned} \tag{6.7}$$

Similarly, the updates of stochastic node to factor messages are

$$\begin{aligned} &m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \\ &\quad \longleftarrow m_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \prod_{j'\neq j}^{n_{ii'}} m_{p(y_{ii'j'}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}), \\ &\qquad 1\le i\le m,\quad 1\le i'\le m',\quad 1\le j\le n_{ii'}, \end{aligned} \tag{6.8}$$

and

$$m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \longleftarrow \prod_{j=1}^{n_{ii'}} m_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}). \tag{6.9}$$

We now express the key messages in their simplest natural parameter form. Recall

the natural parameter expression of the prior in equation (6.4) and that the unnormalised normal density constraint is enforced on equations (6.6) and (6.7). Then

$$m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})$$
$$\longleftarrow \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \\ \operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}^\top\right)\end{bmatrix}^\top \boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}\right\}. \tag{6.10}$$

By substituting the above forms into equation (6.7) it is easy to show the message $m_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \longleftarrow p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')$ is constant throughout the message passing updates. As such, we now set

$$\boldsymbol{\eta}_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})} \longleftarrow \boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}. \tag{6.11}$$

For convenience, we denote the natural parameter vector

$$\boldsymbol{\eta}^{\otimes} \equiv \boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}.$$

Following the derivation of equation (6.10) it is easy to show

$$m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})$$
$$= \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \\ \operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}^\top\right)\end{bmatrix}^\top \boldsymbol{\eta}^{\otimes}\right\}$$
$$= \exp(\eta_0^{\otimes}) \exp\left\{\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}^\top \boldsymbol{\eta}_1^{\otimes} + \left\{\operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}^\top\right)\right\}^\top \boldsymbol{\eta}_2^{\otimes}\right\}.$$

Substituting this into equation (6.6) and following simplifications analogous to the

general one level case leads to

$$
\begin{aligned}
&m_{p(y_{ij}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \\
&\qquad \longleftarrow \text{proj}_{\boldsymbol{UN}}\left[\Phi\left(c_{0_{ii'j}}+\boldsymbol{c}_{1_{ii'j}}^{\top}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}\right)\right. \\
&\qquad\qquad \left.\times \exp\left\{\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}^{\top}\boldsymbol{\eta}_1^{\otimes}+\left\{\text{vech}\left(\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}^{\top}\right)\right\}^{\top}\boldsymbol{\eta}_2^{\otimes}\right\}\right] \\
&\qquad\qquad \times\left(\exp\left\{\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}^{\top}\boldsymbol{\eta}_1^{\otimes}+\left\{\text{vech}\left(\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}^{\top}\right)\right\}^{\top}\boldsymbol{\eta}_2^{\otimes}\right\}\right)^{-1},
\end{aligned}
$$

where $c_{0_{ii'j}} = (2y_{ii'j}-1)\boldsymbol{\beta}^{\top}\boldsymbol{x}^{\mathbf{F}}_{ii'j}$ and $\boldsymbol{c}_{1_{ii'j}} = (2y_{ii'j}-1)\left[(\boldsymbol{x}^{\mathbf{R}}_{ii'j})^{\top}\ (\boldsymbol{x}^{\mathbf{R}'}_{ii'j})^{\top}\right]^{\top}$. Utilising Result 15,

$$
\begin{aligned}
&m_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'}) \\
&\qquad \longleftarrow \exp\left\{\begin{bmatrix}1\\ \boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\\ \text{vech}\left(\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}^{\top}\right)\end{bmatrix}^{\top}\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\right\},
\end{aligned}
$$

where the linear and quadratic coefficient updates are

$$
\left(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\right)_{1:2} \longleftarrow K_{\text{probit}}\left(\boldsymbol{\eta}^{\otimes}_{1:2}; c_{0_{ii'j}}, \boldsymbol{c}_{1_{ii'j}}\right) - \boldsymbol{\eta}^{\otimes}_{1:2}
$$

and the constant coefficient update is

$$
\begin{aligned}
&\left(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\right)_0 \\
&\qquad \longleftarrow C_{\text{probit}}\left(\boldsymbol{\eta}^{\otimes}_{1:2}, \left(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\right)_{1:2} + \boldsymbol{\eta}^{\otimes}_{1:2}; c_{0_{ii'j}}, \boldsymbol{c}_{1_{ii'j}}\right).
\end{aligned}
$$

Using the simplification of equation (6.6) and (6.7), equation (6.8) can be shown to be

$$m_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})$$
$$\longleftarrow \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i \\ \boldsymbol{u}'_{i'} \\ \operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i \\ \boldsymbol{u}'_{i'}\end{bmatrix}^\top\right)\end{bmatrix}^\top \boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}\right\},$$

where

$$\boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})} + \sum_{j'\neq j}\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}.$$

Once convergence is reached, the EP approximation $\underset{\sim}{\ell}(\boldsymbol{\beta},\boldsymbol{\Sigma},\boldsymbol{\Sigma}')$ is given by

$$\begin{aligned}\underset{\sim}{\ell}(\boldsymbol{\beta},\boldsymbol{\Sigma},\boldsymbol{\Sigma}') = \log\int_{\mathbb{R}^{md^{\mathbf{R}}+m'd^{\mathbf{R}'}}}&\left(\prod_{(i,i')n_{ii'}>0}\prod_{j=1}^{n_{ii'}} m_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\right)\\ &\times m_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\; d\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}.\end{aligned} \tag{6.12}$$

Using natural parameters the integral arising in equation (6.12) can be approximated by

$$\begin{aligned}&\int_{\mathbb{R}^{md^{\mathbf{R}}+m'd^{\mathbf{R}'}}}\left(\prod_{(i,i')n_{ii'}>0}\prod_{j=1}^{n_{ii'}} m_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\right)\\ &\qquad\times m_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\; d\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}'_{i'}\end{bmatrix}\\ &=(2\pi)^{(md^{\mathbf{R}}+m'd^{\mathbf{R}'})/2}\exp\left\{\left(\boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}+\sum_{(i,i')n_{ii'}>0}\sum_{j=1}^{n_{ii'}}\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\right)_0\right.\\ &\qquad\left.+A\left(\left(\boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}+\sum_{(i,i')n_{ii'}>0}\sum_{j=1}^{n_{ii'}}\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\right)_{1:2}\right)\right\}.\end{aligned}$$

The full algorithm for the approximation of $\ell(\boldsymbol{\beta},\boldsymbol{\Sigma},\boldsymbol{\Sigma}')$ using EP is provided in Algorithm 7.

Algorithm 7 *Explicit form of algorithm used for the message passing approach to EP*

Inputs: $y_{ii'j}$, $\boldsymbol{x}^{\mathbf{F}}_{ii'j}$, $\boldsymbol{x}^{\mathbf{R}}_{ii'j}$, $\boldsymbol{x}^{\mathbf{R}'}_{ii'j}$, $1 \leq i \leq m$, $1 \leq i' \leq m'$, $1 \leq j \leq n_{ii'}$,

$$\boldsymbol{\beta}(d^{\mathbf{F}} \times 1),\ \boldsymbol{\Sigma}(d^{\mathbf{R}} \times d^{\mathbf{R}}),\ \boldsymbol{\Sigma}'(d^{\mathbf{R}'} \times d^{\mathbf{R}'}),\ \text{are symmetric and positive definite.}$$

Set constants: $c_{0_{ii'j}} \longleftarrow (2y_{ii'j}-1)\boldsymbol{\beta}^\top \boldsymbol{x}^{\mathbf{F}}_{ii'j}$; $\quad \boldsymbol{c}_{1_{ii'j}} \longleftarrow (2y_{ii'j}-1)\big[(\boldsymbol{x}^{\mathbf{R}}_{ii'j})^\top\ (\boldsymbol{x}^{\mathbf{R}'}_{ii'j})^\top\big]^\top$,

$$1 \leq i \leq m, \quad 1 \leq i' \leq m', \quad 1 \leq j \leq n_{ii'};$$

$$\boldsymbol{\eta}_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}$$

$$\longleftarrow \boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')} \equiv \begin{bmatrix} -\frac{md^{\mathbf{R}}}{2}\log|2\pi\boldsymbol{\Sigma}| - \frac{m'd^{\mathbf{R}'}}{2}\log|2\pi\boldsymbol{\Sigma}'| \\ \mathbf{0}_{md^{\mathbf{R}}+m'd^{\mathbf{R}'}} \\ -\frac{1}{2}\boldsymbol{D}^\top_{m'd^{\mathbf{R}'}+md^{\mathbf{R}}}\operatorname{vec}\left(\begin{bmatrix} \boldsymbol{I}_m \otimes \boldsymbol{\Sigma} & \mathbf{0}_{md^{\mathbf{R}}\times m'd^{\mathbf{R}'}} \\ \mathbf{0}_{m'd^{\mathbf{R}'}\times md^{\mathbf{R}}} & \boldsymbol{I}_{m'} \otimes \boldsymbol{\Sigma}' \end{bmatrix}^{-1}\right) \end{bmatrix},$$

$$1 \leq i \leq m, \quad 1 \leq i' \leq m'.$$

Initialise: $\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}$, $1 \leq i \leq m$, $1 \leq i' \leq m'$, $1 \leq j \leq n_{ii'}$, using Laplace approximations.

Cycle:

$$\operatorname{SUM}\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big) \longleftarrow \sum_{(i,i')n_{ii'}>0}\sum_{j=1}^{n_{ii'}} \boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}$$

For $(i,i') \in \{1,...,m\} \times \{1,...,m'\}$:

For $j = 1,...,n_{ii'}$:

$$\boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{u}_i,\boldsymbol{u}'_{i'},\boldsymbol{\Sigma},\boldsymbol{\Sigma}')\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}$$
$$+ \operatorname{SUM}\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big) - \boldsymbol{\eta}_{p(y_{ij}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}$$

$$\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big)_{1:2}$$
$$\longleftarrow K_{\text{probit}}\left(\Big(\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}\Big)_{1:2}; c_{0_{ii'j}}, c_{1_{ii'j}}\right)$$
$$- \Big(\boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}\Big)_{1:2}$$

until convergence of all natural parameters vectors.

Algorithm 8 *Continuation of Algorithm 7 used for the message passing approach to EP*

For $(i, i') \in \{1, ..., m\} \times \{1, ..., m'\}$:

For $j = 1, ..., n_{ii'}$:

$$\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big)_0 \longleftarrow C_{\text{probit}}\bigg(\Big(\boldsymbol{\eta}_{\boldsymbol{u}_i\to p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}\Big)_{1:2}, \Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big)_{1:2} + \Big(\boldsymbol{\eta}_{(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})\to p(y_{ij}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})}\Big)_{1:2}; c_{0_{ii'j}}, c_{1_{ii'j}}\bigg).$$

$$\text{SUM}\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big) \longleftarrow \sum_{(i,i')n_{ii'}>0}\sum_{j=1}^{n_{ii'}}\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}$$

Output: The full approximate log-likelihood is given by

$$\underset{\sim}{\ell}(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}') = \frac{1}{2}(md^{\mathbf{R}} + m'd^{\mathbf{R}'})\log(2\pi) + \bigg(\boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')} + \text{SUM}\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big)\bigg)_0 + A\bigg\{\bigg(\boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')} + \text{SUM}\Big(\boldsymbol{\eta}_{p(y_{ii'j}|\boldsymbol{u}_i,\boldsymbol{u}'_{i'};\boldsymbol{\beta})\to(\boldsymbol{u}_i,\boldsymbol{u}'_{i'})}\Big)\bigg)_{1:2}\bigg\}$$

where, $A(\boldsymbol{\eta})$ is defined as in equation (1.7) and $\boldsymbol{\eta}_{(\boldsymbol{\Sigma},\boldsymbol{\Sigma}')}$ follows from equation (2.21).

6.1.2 Computation of point estimates and confidence intervals

The numerical maximisation of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}')$ is a simple extension of the one level case. As before, we must ensure the search for the maxima of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ occur over the cone of symmetric positive definite matrices, which we accomplished with a re-parameterisation (Bateman & Pinheiro, 2000[55]). For the general crossed random effects case the following procedure is recommended:

1. Convert $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ to the unconstrained space $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ as follows:

   (a) Obtain the spectral decomposition of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$,

   $$\boldsymbol{\Sigma} = \boldsymbol{u}_{\boldsymbol{\Sigma}}\,\text{diag}(\boldsymbol{\lambda}_{\boldsymbol{\Sigma}})\boldsymbol{u}_{\boldsymbol{\Sigma}}^{\top} \quad \text{and} \quad \boldsymbol{\Sigma}' = \boldsymbol{u}'_{\boldsymbol{\Sigma}'}\,\text{diag}(\boldsymbol{\lambda}'_{\boldsymbol{\Sigma}'})(\boldsymbol{u}'_{\boldsymbol{\Sigma}'})^{\top}.$$

   (b) Used the spectral decomposition to obtain the matrix logarithm of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$,

   $$\log(\boldsymbol{\Sigma}) = \boldsymbol{u}_{\boldsymbol{\Sigma}}\,\text{diag}\big(\log(\boldsymbol{\lambda}_{\boldsymbol{\Sigma}})\big)\boldsymbol{u}_{\boldsymbol{\Sigma}}^{\top} \quad \text{and} \quad \log(\boldsymbol{\Sigma}') = \boldsymbol{u}'_{\boldsymbol{\Sigma}'}\,\text{diag}\big(\log(\boldsymbol{\lambda}'_{\boldsymbol{\Sigma}'})\big)(\boldsymbol{u}'_{\boldsymbol{\Sigma}'})^{\top}.$$

(c) Used $\log(\boldsymbol{\Sigma})$ and $\log(\boldsymbol{\Sigma}')$ to obtain $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ respectively,

$$\boldsymbol{\theta} \equiv \text{vech}\Big(\frac{1}{2}\log(\boldsymbol{\Sigma})\Big) \quad \text{and} \quad \boldsymbol{\theta}' \equiv \text{vech}\Big(\frac{1}{2}\log(\boldsymbol{\Sigma}')\Big).$$

2. Obtain the maximum likelihood estimate of $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}'})$ using a quasi-Newton optimisation method,

$$(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}'}) = \text{argmax}\ \ell(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\theta}').$$

We conducted an initial search via Nelder-Mead with refinements by BFGS, which can be implemented via the "optim" R function.

3. Convert $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}'})$ to $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\omega}'})$ as follows:

(a) Obtain the spectral decomposition of $\text{vech}^{-1}(\widehat{\boldsymbol{\theta}})$ and $\text{vech}^{-1}(\widehat{\boldsymbol{\theta}'})$:

$$\text{vech}^{-1}(\widehat{\boldsymbol{\theta}}) = \boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}\ \text{diag}(\boldsymbol{\lambda}_{\widehat{\boldsymbol{\theta}}})\ (\boldsymbol{u}_{\widehat{\boldsymbol{\theta}}})^\top \quad \text{and} \quad \text{vech}^{-1}(\widehat{\boldsymbol{\theta}'}) = \boldsymbol{u}'_{\widehat{\boldsymbol{\theta}'}}\ \text{diag}(\boldsymbol{\lambda}'_{\widehat{\boldsymbol{\theta}'}})\ (\boldsymbol{u}'_{\widehat{\boldsymbol{\theta}'}})^\top.$$

(b) Obtain $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}\ \text{diag}\ \big(\exp(2\boldsymbol{\lambda}_{\widehat{\boldsymbol{\theta}}})\big)\boldsymbol{u}_{\widehat{\boldsymbol{\theta}}}^\top$ and $\widehat{\boldsymbol{\Sigma}'} = \boldsymbol{u}'_{\widehat{\boldsymbol{\theta}'}}\ \text{diag}\ \big(\exp(2\boldsymbol{\lambda}'_{\widehat{\boldsymbol{\theta}'}})\big)(\boldsymbol{u}'_{\widehat{\boldsymbol{\theta}'}})^\top$.

(c) i. If $d^{\mathbf{R}} = 1$, then $\widehat{\boldsymbol{\omega}} = \frac{1}{2}\log(\widehat{\boldsymbol{\Sigma}})$. If $d^{\mathbf{R}'} = 1$, then $\widehat{\boldsymbol{\omega}'} = \frac{1}{2}\log(\widehat{\boldsymbol{\Sigma}'})$.

ii. If $d^{\mathbf{R}} > 1$, then

$$\widehat{\boldsymbol{\omega}} = \begin{bmatrix} \frac{1}{2}\log\big(\text{diag}(\widehat{\boldsymbol{\Sigma}})\big) \\ \tanh^{-1}\Big\{\text{vecbd}(\widehat{\boldsymbol{\Sigma}})\Big/\sqrt{\text{vecbd}\big(\text{diag}(\widehat{\boldsymbol{\Sigma}})\text{diag}(\widehat{\boldsymbol{\Sigma}})^\top\big)}\ \Big\} \end{bmatrix}.$$

If $d^{\mathbf{R}'} > 1$, then

$$\widehat{\boldsymbol{\omega}'} = \begin{bmatrix} \frac{1}{2}\log\big(\text{diag}(\widehat{\boldsymbol{\Sigma}'})\big) \\ \tanh^{-1}\Big\{\text{vecbd}(\widehat{\boldsymbol{\Sigma}'})\Big/\sqrt{\text{vecbd}\big(\text{diag}(\widehat{\boldsymbol{\Sigma}'})\text{diag}(\widehat{\boldsymbol{\Sigma}'})^\top\big)}\ \Big\} \end{bmatrix}.$$

4. Obtain the Hessian matrix $H\ell(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\omega}'})$ at the maximum $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\omega}'})$ using the quasi-Newton method BFGS, which as before can be implemented via `optim()`. The constraints on the parameters $(\boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\omega}')$ means the Hessian should still be calculated on the $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\theta}')$ space. The conversion can be obtained as follows:

(a) Form the symmetric matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ as follows:

i. If $d^{\mathbf{R}} = 1$, then $\boldsymbol{\Sigma} = \exp(2\boldsymbol{\omega})$. If $d^{\mathbf{R}'} = 1$ then $\boldsymbol{\Sigma}' = \exp(2\boldsymbol{\omega}')$.

ii. If $d^{\mathbf{R}} > 1$, then let $\boldsymbol{\omega}_1$ denote the first $d^{\mathbf{R}}$ entries of $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_2$ denote the remaining $\frac{1}{2}d^{\mathbf{R}}(d^{\mathbf{R}} - 1)$ entries of $\boldsymbol{\omega}$. Set $\text{diag}(\boldsymbol{\Sigma}) = \exp(2\boldsymbol{\omega}_1)$. Obtain

the below-diagonal entries of $\boldsymbol{\Sigma}$ so that

$$\text{vecbd}(\boldsymbol{\Sigma}) = \tanh(\boldsymbol{\omega}_2) \odot \text{vecbd}\big(\exp(\boldsymbol{\omega}_1)\exp(\boldsymbol{\omega}_1)^\top\big)$$

holds. Obtain the above-diagonal entries of $\boldsymbol{\Sigma}$ such that symmetry of $\boldsymbol{\Sigma}$ is enforced. If $d^{\mathbf{R}'} > 1$, then let $\boldsymbol{\omega}'_1$ denote the first $d^{\mathbf{R}'}$ entries of $\boldsymbol{\omega}'$ and $\boldsymbol{\omega}'_2$ denote the remaining $\frac{1}{2}d^{\mathbf{R}'}(d^{\mathbf{R}'}-1)$ entries of $\boldsymbol{\omega}'$. Set $\text{diag}(\boldsymbol{\Sigma}') = \exp(2\boldsymbol{\omega}'_1)$. Obtain the below-diagonal entries of $\boldsymbol{\Sigma}'$ so that

$$\text{vecbd}(\boldsymbol{\Sigma}') = \tanh(\boldsymbol{\omega}'_2) \odot \text{vecbd}\big(\exp(\boldsymbol{\omega}'_1)\exp(\boldsymbol{\omega}'_1)^\top\big)$$

holds. Obtain the above-diagonal entries of $\boldsymbol{\Sigma}'$ such that symmetry of $\boldsymbol{\Sigma}'$ is enforced.

iii. Obtain the spectral decomposition:

$$\boldsymbol{\Sigma} = \boldsymbol{u}_{\boldsymbol{\Sigma}}\text{diag}(\boldsymbol{\lambda}_{\boldsymbol{\Sigma}})\boldsymbol{u}_{\boldsymbol{\Sigma}}^\top \quad \text{and} \quad \boldsymbol{\Sigma}' = \boldsymbol{u}'_{\boldsymbol{\Sigma}'}\text{diag}(\boldsymbol{\lambda}'_{\boldsymbol{\Sigma}'})(\boldsymbol{u}'_{\boldsymbol{\Sigma}'})^\top.$$

iv. Obtain

$$\boldsymbol{\theta} = \text{vech}\Big\{\frac{1}{2}\boldsymbol{u}_{\boldsymbol{\Sigma}}\text{diag}\big(\log(\boldsymbol{\lambda}_{\boldsymbol{\Sigma}})\big)\boldsymbol{u}_{\boldsymbol{\Sigma}}^\top\Big\}$$

and

$$\boldsymbol{\theta}' = \text{vech}\Big\{\frac{1}{2}\boldsymbol{u}'_{\boldsymbol{\Sigma}'}\text{diag}\big(\log(\boldsymbol{\lambda}'_{\boldsymbol{\Sigma}'})\big)(\boldsymbol{u}'_{\boldsymbol{\Sigma}'})^\top\Big\}.$$

5. Form $100(1-\alpha)\%$ confidence intervals for the entries of $(\boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\omega}')$ using

$$\begin{bmatrix}\widehat{\boldsymbol{\beta}}\\ \widehat{\boldsymbol{\omega}}\\ \widehat{\boldsymbol{\omega}'}\end{bmatrix} \pm \Phi^{-1}\Big(1-\frac{1}{2}\alpha\Big)\sqrt{-\text{diag}\Big\{\big(\boldsymbol{H}\ell(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\omega}'})\big)^{-1}\Big\}}.$$

6. Back transform the confidence interval limits for the $\boldsymbol{\omega}$ component, to correspond to the standard deviation and correlation parameters as follows:

$$\begin{bmatrix}\sqrt{\text{diag}(\boldsymbol{\Sigma})}\\ \text{vecbd}(\boldsymbol{\Sigma})\big/\sqrt{\text{vecbd}\big(\text{diag}(\boldsymbol{\Sigma})\text{diag}(\boldsymbol{\Sigma})^\top\big)}\end{bmatrix}$$

and

$$\begin{bmatrix}\sqrt{\text{diag}(\boldsymbol{\Sigma}')}\\ \text{vecbd}(\boldsymbol{\Sigma}')\big/\sqrt{\text{vecbd}\big(\text{diag}(\boldsymbol{\Sigma}')\text{diag}(\boldsymbol{\Sigma}')^\top\big)}\end{bmatrix}.$$

6.1.3 Simulation study

Three simulation studies were conducted, each for 1000 replicates on University of Technology Sydney Interactive High Performance Computing facility Jupiter node with eight 3.6 gigahertz processors and 32 gigabytes random access memory.

6.1.3.1 Comparison with MCMC and Laplace approximation maximum likelihood for crossed random effects

The observations were simulated 1000 times according to equation (6.1) with true parameter values

$$\boldsymbol{\beta}_{\text{true}} = [-0.58, 1.07]^\top \quad , \boldsymbol{\Sigma}_{\text{true}} = \sigma^2_{\text{true}} = 0.32, \quad \text{and} \quad \boldsymbol{\Sigma}'_{\text{true}} = (\sigma^2_{\text{true}})' = 0.47. \quad (6.13)$$

The number of groups in the data were $m = 10$, $m' = 6$ with the number of measurements in each group fixed at $n_{ii'} = 3$ for all (i, i'). The $\boldsymbol{x}^{\mathbf{F}}_{ii'j}$, $\boldsymbol{x}^{\mathbf{R}}_{ii'j}$ and $\boldsymbol{x}^{\mathbf{R}'}_{ii'j}$vectors were of the form

$$\boldsymbol{x}^{\mathbf{F}}_{ii'j} = [1, x_{ii'j}]^\top, \quad \boldsymbol{x}^{\mathbf{R}}_{ii'j} = 1 \quad \text{and} \quad \boldsymbol{x}^{\mathbf{R}'}_{ii'j} = 1$$

where $x_{ii'j}$ was generated independently from a uniform distribution on the unit interval. The tolerance of error values for the EP algorithm was set to 10^{-5} and a maximum of 100 Nelder-Mead search iterations was used. The model described was fit using each of the following methods:

1. Laplace approximation implemented via the function `glmer()` in the R package "lme4" (Bates, et al., 2018[5]).
2. EP as described in this Section, initalised using Laplace approximations from the function `glmer()` in the R package "lme4" (Bates, et al., 2018[5]).
3. Markov Chain Monte Carlo with flat priors.

We compare point estimates and confidence intervals produced by each of the three methods in Figure 6.2. Our studies show that the quality of results produced by each method varied across parameters. MCMC provided the highest coverage (98.6%) for the fixed intercept, while EP and Laplace approximations provided estimates which were over confident (92.7% and 93.2% respectively). For the fixed slope all methods provided approximately 95% coverage, with Laplace approximations providing the highest empirical coverage (95.1%). No method provided good results for variance

parameter estimates. Laplace approximations had the highest empirical coverage for the variance parameter of the first level (69.5% compared to 63% from EP), whilst EP had the highest empirical coverage for the variance parameter of the crossed level (89.2% compared to 85.9% from Laplace approximations). MCMC provided the worst coverage for the first and crossed level of (56.5% and 69.2% respectively). Overall it seems that Laplace approximations perform marginally better than EP for this model and data senario.

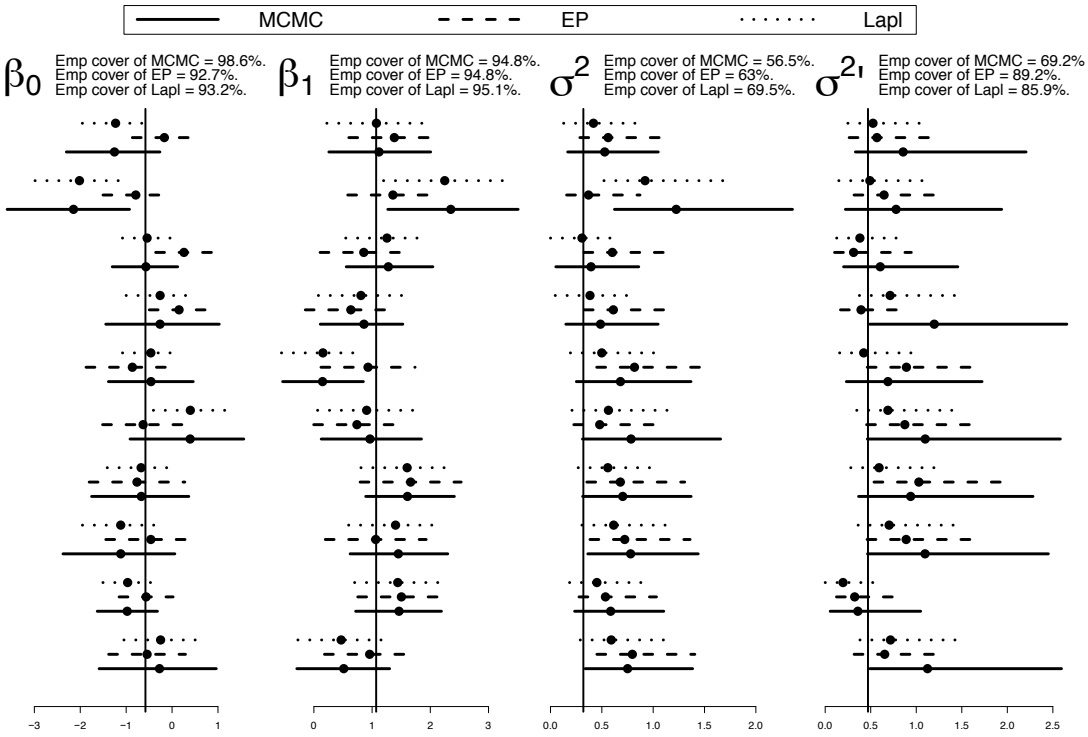


Figure 6.2: *Comparison of point estimates and 95% confidence intervals for the simulation study with true parameter values given by equation (6.13). We display 10 randomly chosen replications of the simulation study described. The vertical lines indicate true parameter values and the percentages displayed at the top of each panel are empirical coverages over all 1000 replications for each method involved in the comparison.*

6.2 The general probit two level mixed model

We now extend our methodology for the probit case to deal with two level GLMMs, utilising key results from Chapter 3 as in the crossed case. Our aim is to find an approximation to the maximum likelihood estimates of the parameters $\boldsymbol{\beta}$, $\boldsymbol{\Sigma}^{\mathbf{L}1}$ and $\boldsymbol{\Sigma}^{\mathbf{L}2}$

with 95% confidence intervals.

The probit model with two levels of nesting is specified in equation (6.14),

$$\begin{gathered} y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2} \overset{\text{ind.}}{\sim} \text{Bernoulli}\left\{\Phi\Big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ijk}^{\mathbf{F}} + (\boldsymbol{u}_i^{\mathbf{L}1})^\top \boldsymbol{x}_{ijk}^{\mathbf{R}1} + (\boldsymbol{u}_{ij}^{\mathbf{L}2})^\top \boldsymbol{x}_{ijk}^{\mathbf{R}2}\Big)\right\}, \\ \boldsymbol{u}_i^{\mathbf{L}1} \overset{\text{ind.}}{\sim} \boldsymbol{N}(\mathbf{0}_{d^{\mathbf{R}1}}, \boldsymbol{\Sigma}^{\mathbf{L}1}) \quad \text{independently of} \quad \boldsymbol{u}_{ij}^{\mathbf{L}2} \overset{\text{ind.}}{\sim} \boldsymbol{N}(\mathbf{0}_{d^{\mathbf{R}2}}, \boldsymbol{\Sigma}^{\mathbf{L}2}), \\ 1 \leq i \leq m, \quad 1 \leq j \leq n_i, \quad 1 \leq k \leq o_{ij}, \end{gathered} \tag{6.14}$$

where the response $y_{ijk} \in \{0, 1\}$ and predictor vectors $\boldsymbol{x}_{ijk}^{\mathbf{F}}$, $\boldsymbol{x}_{ijk}^{\mathbf{R}1}$ and $\boldsymbol{x}_{ijk}^{\mathbf{R}2}$ correspond to the kth set of measurements within the jth inner group within the ith outer group. The number of outer groups is m and the number of inner groups in the ith outer group is n_i. The sample size of the jth group in the ith outer group is o_{ij}. $\boldsymbol{x}_{ijk}^{\mathbf{R}1}$ and $\boldsymbol{x}_{ijk}^{\mathbf{R}2}$ have respective dimensions $d^{\mathbf{R}1} \times 1$ and $d^{\mathbf{R}2} \times 1$.

The form of the log-likelihood is

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) \equiv \sum_{i=1}^{m} \ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}),$$

where

$$\begin{aligned} \ell_i(\boldsymbol{\beta}, \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) = \log \int_{\mathbb{R}^{d^{\mathbf{R}1}+d^{\mathbf{R}2}}} \prod_{j=1}^{n_i} \prod_{k=1}^{o_{ij}} p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta}) \\ \times p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})\, d \begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix}, \end{aligned} \tag{6.15}$$

$$p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta}) = \Phi\Big\{(2y_{ijk} - 1)\Big(\boldsymbol{\beta}^\top \boldsymbol{x}_{ijk}^{\mathbf{F}} + (\boldsymbol{u}_i^{\mathbf{L}1})^\top \boldsymbol{x}_{ijk}^{\mathbf{R}1} + (\boldsymbol{u}_{ij}^{\mathbf{L}2})^\top \boldsymbol{x}_{ijk}^{\mathbf{R}2}\Big)\Big\}$$

and

$$\begin{aligned} p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) \equiv & |2\pi\boldsymbol{\Sigma}^{\mathbf{L}1}|^{-1/2} |2\pi\boldsymbol{\Sigma}^{\mathbf{L}2}|^{-1/2} \\ & \times \exp\left\{-\frac{1}{2} \begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\Sigma}^{\mathbf{L}1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{\mathbf{L}2} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix}\right\}. \end{aligned}$$

Unlike the crossed random effects case, the integrals required are not dependent on the number of groups. Subsection 6.2.1 explains the schematic of likelihood approximation for the two level probit model using EP. It is easy to see how computation of point

estimates and confidence intervals for $(\boldsymbol{\beta}, \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})$ using `optim()` in R follows the same process as the crossed model in Subsection 6.1.2, and as such we do not repeat it. Due to time constraints we did not conduct a simulation study and as such no results are presented.

6.2.1 Expectation propagation likelihood approximation

As in the crossed random effects case, the results facilitating the projection of the target density onto the unnormalised multivariate normal family for the general one level probit model can be reused for the two level probit model. For the two level model EP involves finding the optimal natural parameters $\eta_0, \boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ which minimise $\mathrm{KL}(f_{\text{input}} \parallel f_{\boldsymbol{UN}})$, where the unnormalised multivariate normal density $f_{\boldsymbol{UN}}$ is defined by equation (3.4) and the target density $f_{\text{input}}(\boldsymbol{x})$ can be written as

$$f_{\text{input}}(\boldsymbol{x}) = \Phi(c_0 + \boldsymbol{c}_1^\top \boldsymbol{x}) \exp\left((\boldsymbol{\eta}_1^{\text{input}})^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{H}_2^{\text{input}} \boldsymbol{x}\right), \tag{6.16}$$

where $\boldsymbol{\eta}_1^{\text{input}}$ is a $d \times 1$ vector, $\boldsymbol{H}_2^{\text{input}}$ is a $d \times d$ matrix, $c_0 = (2y_{ijk} - 1)\boldsymbol{\beta}^\top \boldsymbol{x}_{ijk}^{\mathbf{F}}$, $\boldsymbol{c}_1 = (2y_{ijk} - 1)\left[(\boldsymbol{x}_{ijk}^{\mathbf{R}1})^\top \ (\boldsymbol{x}_{ijk}^{\mathbf{R}2})^\top\right]^\top$ and $\boldsymbol{x} = \left[(\boldsymbol{u}_i^{\mathbf{L}1})^\top \ (\boldsymbol{u}_{ij}^{\mathbf{L}2})^\top\right]^\top$. It is easy to see we require $\boldsymbol{\eta}^*$ to solve equation (3.7), where the optimal natural parameters $\boldsymbol{\eta}_1^*$ and $\boldsymbol{\eta}_2^*$ to project onto the multivariate normal family are given by Result 14, and the projection on the unnormalised multivariate normal density is given by Result 15.

6.2.1.1 Message passing formulation

The message passing scheme required for the two level model follows that of Section 3.1.2, with minor ammendments. For clarity we expicitly show it here. Note the alternate matrix expression of the prior

$$p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) \equiv \exp\left\{ \begin{bmatrix} 1 \\ \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \\ \operatorname{vech}\left(\begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix}^\top \right) \end{bmatrix}^\top \boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})} \right\},$$

where

$$\boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})} \equiv \begin{bmatrix} -\frac{d^{\mathbf{R}1}}{2}\log|2\pi\boldsymbol{\Sigma}^{\mathbf{L}1}| - \frac{d^{\mathbf{R}2}}{2}\log|2\pi\boldsymbol{\Sigma}^{\mathbf{L}2}| \\ \mathbf{0}_{d^{\mathbf{R}1}+d^{\mathbf{R}2}} \\ -\frac{1}{2}\boldsymbol{D}^{\top}_{d^{\mathbf{R}1}+d^{\mathbf{R}2}}\mathrm{vec}\left(\begin{bmatrix} \boldsymbol{\Sigma}^{\mathbf{L}1} & \mathbf{0}_{d^{\mathbf{R}1}\times d^{\mathbf{R}2}} \\ \mathbf{0}_{d^{\mathbf{R}2}\times d^{\mathbf{R}1}} & \boldsymbol{\Sigma}^{\mathbf{L}2} \end{bmatrix}^{-1}\right) \end{bmatrix}. \tag{6.17}$$

The structure of equation (6.2) in factor graph form is shown in Figure 6.3, where the circular stochastic node corresponds to the random vector $(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$, the solid squares indicate the factor nodes and the dependencies of the factor nodes on the stochastic node are demonstrated through the edges. Suppose that

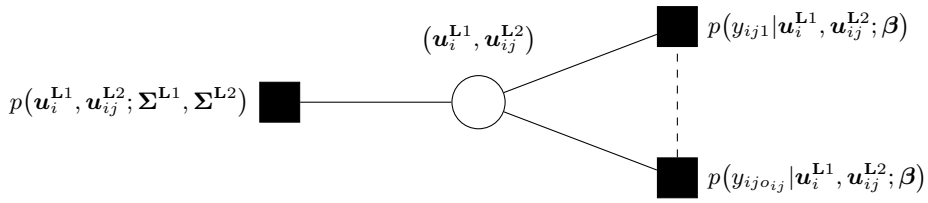


Figure 6.3: *Factor graph representation of the product structure of the integrand in equation (6.15) for random effects models with two levels of nesting.*

$$p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta}) = \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \\ \mathrm{vech}\left(\begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \end{bmatrix}^{\top}\right) \end{bmatrix}^{\top} \boldsymbol{\eta}_{ijk}\right\}, \quad 1 \le k \le o_{ij},$$

are initialised to be unnormalised multivariate normal density functions in $(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$. Then, for each $k = 1, ..., o_{ij}$, the $\boldsymbol{\eta}_{ijk}$ update involves minimisation of

$$\mathrm{KL}\Bigg(p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta}) \prod_{j=1}^{n_i}\prod_{k'\neq k}^{o_{ij}} \underset{\sim}{p}(y_{ijk'}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta}) \times p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) \Bigg\| \prod_{j=1}^{n_i}\prod_{k'=1}^{o_{ij}} \underset{\sim}{p}(y_{ijk'}|\boldsymbol{u}_i^{\mathbf{L}1}\boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta}) p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})\Bigg) \tag{6.18}$$

as a function of $(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$. We can achieve this by using Result 15 to update $\boldsymbol{\eta}_{ijk}$ in an iterative procedure until it converges.

The process of minimising the KL divergence in equation (6.18) can be expressed as a message from the factor $p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta})$ to the stochastic node $(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$ as

$$m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$$
$$\longleftarrow \frac{\text{proj}_{\boldsymbol{UN}}\left[m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})\; p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\beta})\right](\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})}{m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})},$$
$$1 \le i \le m, \quad 1 \le j \le n_i, \quad 1 \le k \le o_{ij}, \tag{6.19}$$

and the update of the message passed from $p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})$ to $(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$ is

$$m_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$$
$$\longleftarrow \frac{\text{proj}_{\boldsymbol{UN}}\left[m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}) p(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}; \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})\right](\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})}{m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})}. \tag{6.20}$$

The updates of stochastic node to factor messages are

$$m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2})$$
$$= m_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}) \prod_{j=1}^{n_i} \prod_{k' \ne k}^{o_{ij}} m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}),$$
$$1 \le i \le m, \quad 1 \le j \le n_i, \quad 1 \le k \le o_{ijk}, \tag{6.21}$$

and

$$m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}) = \prod_{j=1}^{n_i} \prod_{k'=1}^{o_{ij}} m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1}, \boldsymbol{u}_{ij}^{\mathbf{L}2}). \tag{6.22}$$

Recall the natural parameter expression of the prior in equation (6.17) and that the unnormalised normal density constraint is enforced on equations (6.19) and (6.20).

Then

$$m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})$$
$$= \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \\ \operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}^\top\right)\end{bmatrix}^\top \boldsymbol{\eta}_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}\right\}. \tag{6.23}$$

By substituting the above forms into equation (6.20) it is easy to show the message $m_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2}) \longleftarrow p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})$ is constant throughout the message passing updates. As such, we now set

$$\boldsymbol{\eta}_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})} \longleftarrow \boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}. \tag{6.24}$$

For convenience, we denote the natural parameter vector

$$\boldsymbol{\eta}^{\otimes} \equiv \boldsymbol{\eta}_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}.$$

Following the derivation of equation (6.23) it is easy to show

$$m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})$$
$$= \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \\ \operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}^\top\right)\end{bmatrix}^\top \boldsymbol{\eta}^{\otimes}\right\}$$
$$= \exp(\eta_0^{\otimes}) \exp\left\{\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}^\top \boldsymbol{\eta}_1^{\otimes} + \left\{\operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}^\top\right)\right\}^\top \boldsymbol{\eta}_2^{\otimes}\right\}.$$

Substituting this into equation (6.19) and following simplifications analogous to the

general one level case leads to

$$
\begin{aligned}
&m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2})}(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2}) \\
&\quad\longleftarrow \mathrm{proj}_{\boldsymbol{UN}}\left[\Phi\left(c_{0_{ijk}}+\boldsymbol{c}_{1_{ijk}}^{\top}\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}\right)\right. \\
&\qquad\left.\times\exp\left\{\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}^{\top}\boldsymbol{\eta}_1^{\otimes}+\left\{\mathrm{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}^{\top}\right)\right\}^{\top}\boldsymbol{\eta}_2^{\otimes}\right\}\right] \\
&\qquad\times\left(\exp\left\{\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}^{\top}\boldsymbol{\eta}_1^{\otimes}+\left\{\mathrm{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}^{\top}\right)\right\}^{\top}\boldsymbol{\eta}_2^{\otimes}\right\}\right)^{-1},
\end{aligned}
$$

where $c_{0_{ijk}} = (2y_{ijk}-1)\boldsymbol{\beta}^{\top}\boldsymbol{x}_{ijk}^{\mathbf{F}}$ and $\boldsymbol{c}_{1_{ijk}} = (2y_{ijk}-1)\left[(\boldsymbol{x}_{ijk}^{\mathbf{R}1})^{\top}(\boldsymbol{x}_{ijk}^{\mathbf{R}2})^{\top}\right]^{\top}$. Utilising Result 15,

$$
\begin{aligned}
&m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2})}(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2}) \\
&\qquad\longleftarrow\exp\left\{\begin{bmatrix}1\\ \boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\\ \mathrm{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathrm{L}1}\\ \boldsymbol{u}_{ij}^{\mathrm{L}2}\end{bmatrix}^{\top}\right)\end{bmatrix}^{\top}\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2})}\right\},
\end{aligned}
$$

where the linear and quadratic coefficient updates are

$$
\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2})}\right)_{1:2} \longleftarrow K_{\mathrm{probit}}\left(\boldsymbol{\eta}_{1:2}^{\otimes};c_{0_{ijk}},\boldsymbol{c}_{1_{ijk}}\right)-\boldsymbol{\eta}_{1:2}^{\otimes}
$$

and the constant coefficient update is

$$
\begin{aligned}
\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2})}\right)_0 \longleftarrow & C_{\mathrm{probit}}\left(\boldsymbol{\eta}_{1:2}^{\otimes},\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathrm{L}1},\boldsymbol{u}_{ij}^{\mathrm{L}2})}\right)_{1:2}\right. \\
&\left.+\boldsymbol{\eta}_{1:2}^{\otimes};c_{0_{ijk}},\boldsymbol{c}_{1_{ijk}}\right).
\end{aligned}
$$

Using the simplification of equation (6.19) and (6.20), equation (6.21) can be shown to

be

$$m_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})$$
$$\longleftarrow \exp\left\{\begin{bmatrix} 1 \\ \boldsymbol{u}_i^{\mathbf{L}1} \\ \boldsymbol{u}_{ij}^{\mathbf{L}2} \\ \operatorname{vech}\left(\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}^\top\right)\end{bmatrix}^\top \boldsymbol{\eta}_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}\right\},$$

where

$$\boldsymbol{\eta}_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})} + \sum_{j'\neq j}\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}.$$

Once convergence is reached, the EP approximation of each $\underset{\sim}{\ell}_i(\boldsymbol{\beta},\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})$ is given by

$$\underset{\sim}{\ell}_i(\boldsymbol{\beta},\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2}) = \log\int_{\mathbb{R}^{d^{\mathbf{R}1}+d^{\mathbf{R}2}}}\prod_{j=1}^{n_i}\prod_{k=1}^{o_{ij}}\left(m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\right)$$
$$\times\, m_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\; d\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}. \tag{6.25}$$

Using natural parameters the integral arising in equation (6.25) can be approximated by

$$\int_{\mathbb{R}^{d^{\mathbf{R}1}+d^{\mathbf{R}2}}}\prod_{j=1}^{n_i}\prod_{k=1}^{o_{ij}}\left(m_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\right)$$
$$\times\, m_{p(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\; d\begin{bmatrix}\boldsymbol{u}_i^{\mathbf{L}1}\\ \boldsymbol{u}_{ij}^{\mathbf{L}2}\end{bmatrix}$$
$$= (2\pi)^{(d^{\mathbf{R}1}+d^{\mathbf{R}2})/2}\exp\left\{\left(\boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})} + \sum_{j=1}^{n_i}\sum_{k=1}^{o_{ij}}\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right)_0\right.$$
$$\left. + A\left(\left(\boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})} + \sum_{j=1}^{n_i}\sum_{k=1}^{o_{ij}}\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right)_{1:2}\right)\right\}.$$

The full algorithm for the approximation of $\ell(\boldsymbol{\beta},\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})$ using EP is provided in

Algorithm 9.

Algorithm 9 *Explicit form of algorithm used for the message passing approach to EP*

Inputs: y_{ijk}, $\boldsymbol{x}^{\mathbf{F}}_{ijk}$, $\boldsymbol{x}^{\mathbf{R}1}_{ijk}$, $\boldsymbol{x}^{\mathbf{R}1}_{ijk}$, $1 \leq i \leq m$, $1 \leq j \leq n_i$, $1 \leq k \leq o_{ij}$,

$$\boldsymbol{\beta}(d^{\mathbf{F}} \times 1), \boldsymbol{\Sigma}^{\mathbf{L}1}(d^{\mathbf{R}1} \times d^{\mathbf{R}1}), \boldsymbol{\Sigma}^{\mathbf{L}2}(d^{\mathbf{R}2} \times d^{\mathbf{R}2}), \text{ are symmetric and positive definite.}$$

Set constants: $c_{0_{ijk}} \longleftarrow (2y_{ijk} - 1)\boldsymbol{\beta}^{\top}\boldsymbol{x}^{\mathbf{F}}_{ijk}$; $\quad \boldsymbol{c}_{1_{ijk}} \longleftarrow (2y_{ijk} - 1)\left[(\boldsymbol{x}^{\mathbf{R}1}_{ijk})^{\top}(\boldsymbol{x}^{\mathbf{R}2}_{ijk})^{\top}\right]^{\top}$,

$$1 \leq i \leq m, \quad 1 \leq j \leq n_i, \quad 1 \leq k \leq o_{ij};$$

$$\boldsymbol{\eta}_{p(\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\Sigma}, \boldsymbol{\Sigma}') \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}$$

$$\longleftarrow \boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2})} \equiv \begin{bmatrix} -\frac{d^{\mathbf{R}1}}{2}\log|2\pi\boldsymbol{\Sigma}^{\mathbf{L}1}| - \frac{d^{\mathbf{R}2}}{2}\log|2\pi\boldsymbol{\Sigma}^{\mathbf{L}2}| \\ \mathbf{0}_{d^{\mathbf{R}1}+d^{\mathbf{R}2}} \\ -\frac{1}{2}\boldsymbol{D}^{\top}_{d^{\mathbf{R}1}+d^{\mathbf{R}2}} \operatorname{vec}\left(\begin{bmatrix} \boldsymbol{\Sigma}^{\mathbf{L}1} & \mathbf{0}_{d^{\mathbf{R}1} \times d^{\mathbf{R}2}} \\ \mathbf{0}_{d^{\mathbf{R}2} \times d^{\mathbf{R}1}} & \boldsymbol{\Sigma}^{\mathbf{L}2} \end{bmatrix}^{-1} \right) \end{bmatrix},$$

$$1 \leq i \leq m, \quad 1 \leq j \leq n_i, \quad 1 \leq j \leq o_{ij}.$$

For $i = 1, ..., m$:

Initialise: $\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}2}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta}) \to (\boldsymbol{u}^{\mathbf{L}2}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}$, $1 \leq i \leq m$, $1 \leq j \leq n_i$, $1 \leq j \leq o_{ij}$, using Laplace approximations.

Cycle:

$$\operatorname{SUM}\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta}) \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}\right) \longleftarrow \sum_{j=1}^{n_i}\sum_{k=1}^{o_{ij}} \boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta}) \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}$$

For $j = 1, ..., n_i$:

For $k = 1, ..., o_{ij}$:

$$\boldsymbol{\eta}_{(\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}) \to p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}, \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}$$

$$+ \operatorname{SUM}\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta}) \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}\right) - \boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta}) \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}$$

$$\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta}) \to (\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij})}\right)_{1:2}$$

$$\longleftarrow K_{\text{probit}}\left(\left(\boldsymbol{\eta}_{(\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}) \to p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta})}\right)_{1:2}; c_{0_{ijk}}, c_{1_{ijk}}\right)$$

$$- \left(\boldsymbol{\eta}_{(\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}) \to p(y_{ijk}|\boldsymbol{u}^{\mathbf{L}1}_i, \boldsymbol{u}^{\mathbf{L}2}_{ij}; \boldsymbol{\beta})}\right)_{1:2}$$

until convergence of all natural parameters vectors.

Algorithm 10 *Continuation of Algorithm 9 used for the message passing approach to EP*

For $j = 1, ..., n_i$:

For $k = 1, ..., o_{ij}$:

$$\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right)_0 \longleftarrow C_{\text{probit}}\Bigg(\left(\boldsymbol{\eta}_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}\right)_{1:2},$$
$$\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right)_{1:2} + \left(\boldsymbol{\eta}_{(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})\to p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})}\right)_{1:2};$$
$$c_{0_{ijk}}, c_{1_{ijk}}\Bigg).$$

$$\text{SUM}\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right) \longleftarrow \sum_{j=1}^{n_i}\sum_{k=1}^{o_{ij}} \boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}$$

Output: The full approximate log-likelihood is given by

$$\underset{\sim}{\ell}(\boldsymbol{\beta}, \boldsymbol{\Sigma}^{\mathbf{L}1}, \boldsymbol{\Sigma}^{\mathbf{L}2}) = \frac{m(d^{\mathbf{R}1} + d^{\mathbf{R}2})}{2}\log(2\pi)$$
$$+ \sum_{i=1}^{m}\Bigg\{\left(\boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})} + \text{SUM}\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right)\right)_0$$
$$+ A\left\{\left(\boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})} + \text{SUM}\left(\boldsymbol{\eta}_{p(y_{ijk}|\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2};\boldsymbol{\beta})\to(\boldsymbol{u}_i^{\mathbf{L}1},\boldsymbol{u}_{ij}^{\mathbf{L}2})}\right)\right)_{1:2}\right\}\Bigg\},$$

where $A(\boldsymbol{\eta})$ is defined as in equation (1.7) and $\boldsymbol{\eta}_{(\boldsymbol{\Sigma}^{\mathbf{L}1},\boldsymbol{\Sigma}^{\mathbf{L}2})}$ follows from equation (2.21).

Chapter 7

Applications of expectation propagation for one level probit mixed models

Regardless of the amount of care taken during data collection, real data is subject to the laws of the environment it is collected in; that is, it is often full of subtle yet difficult features and traits, providing unforeseen challenges to analysts when fitting models. The facets of real data are not easily replicated in simulations due to the stochastic environment in which they are created. As such, each real dataset provides a unique opportunity for rigorous testing of statistical methodology. Not only does this help to expose unknown weaknesses which can be either acknowledged or further developed, but can also help highlight strengths. In Section 7.1 we introduce the R package created for implementing the methodology in the probit case as explained in Chapter 3. Section 7.2 and 7.3 this cover applications of our methodology on two real datasets, each with unique challenges. We fit models for each dataset and compare the fits to current methodology, and in the case of the second dataset contribute some useful findings.

7.1 R package "glmmEP"

To facilitate implementation of the methodology presented in this thesis we developed the R package "glmmEP" (Wand & Yu, 2020[69]). Due to computational issues with the integrals required to solve non-probit models and time constraints, the package is restricted to one level probit models. Although the package uses an R interface, all computations relating to EP are conducted in Fortran77 to reduce computing

time. The R function `glmmEP()` is used to fit a probit GLMM using our methodology, where the user must supply a vector of binary responses, a matrix of fixed effects, a matrix of random effects, and a vector with identification numbers for each group. Calling `summary.glmmEP()` returns the maximum likelihood estimates and corresponding confidence intervals for the fixed effects parameters. Confidence levels are set using the `confLev` argument in the function `glmmEP.control()`. Unlike other packages we also provide confidence intervals for random effects parameters. A vignette with more details is available by calling `glmmEPvignette()`.

7.2 Modelling immunisation of Guatemalan children

In 1987, the National Survey of Maternal and Child Health was conducted in Guatemala on a sample of 5160 women age 15 to 44. The study utilised a questionaire about prenatal care and immunisation status of chidren born during the study period. Pebley, Goldman & Rodríguez (1996)[54] provide a thorough explaination of the 1987 study dataset and conduct an analysis.

We used the dataset `guImmun` from the R package "mlmRev" (Bates, Maechler & Bolker, 2019[4]) which is a smaller version of the full dataset of the 1987 National Survey of Maternal and Child Health, with only 2519 observations and 13 variables. Of the 13 we selected 8, which are summarised in Table (7.1). This dataset has a low number of observations per grouping variable and as such is a difficult model to estimate. Our aim is to compare the fits of current GLMM methodology to that provided by our methodology.

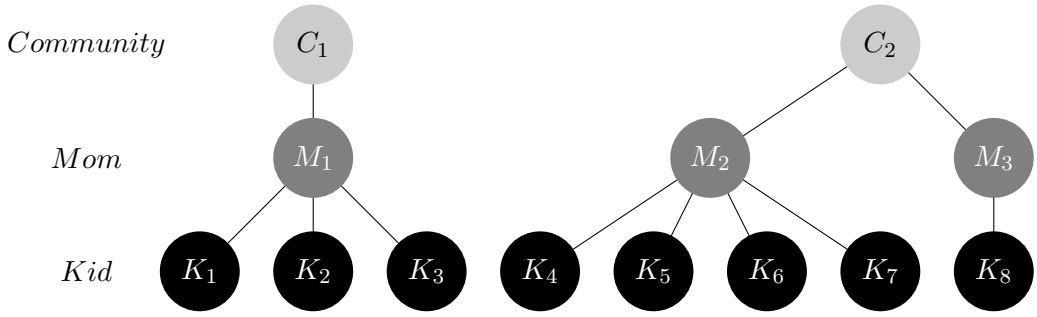


Figure 7.1: *A plot showing the two level multiple membership structure of the data, specifically kids nested under mothers nested under community. The first row shows each community, the second shows each mother in each community and the third row shows the kids of each mother in each community.*

Table 7.1: *A list of the variables in the* `guImmun` *dataset used for model fitting.*

Variable name	**Description**
`immun`	A two-level factor whether or not a child has their complete set of immunisations.
`pcInd81`	A continuous variable of indigenous population percentage in the community child lived in during 1981 census.
`kid2p`	A two-level factor specifying if the child was two years or older.
`momEd`	A three-level factor with the mother's level of school education. The levels are not finished primary school, finished primary school, and finished secondary school.
`husEd`	A four-level factor variable with the husband's level of school education. The levels are not finished primary school, finished primary school, finished secondary school, and unknown.
`momWork`	A two-level factor variable whether the child's mother had ever worked outside the home.
`rural`	A two-level factor variable indicating whether or not the child's location is rural or urban.
`mom`	A multilevel factor variable that codes the children's mothers.

The random intercepts and slopes probit mixed model we fit follows in equation (7.1),

$$\begin{aligned}\mathcal{I}(\texttt{immun}_{ij} = Y)|u_{0i}, u_{1i} \overset{\text{ind.}}{\sim} \text{Bernoulli}\Big\{&\Phi\big(\beta_0 + u_{0i} + (\beta_1 + u_{1i})\texttt{pcInd81}_{ij} \\ &+\beta_2\mathcal{I}(\texttt{kid2p}_{ij} = Y) + \beta_3\mathcal{I}(\texttt{momEd}_{ij} = S) + \beta_4\mathcal{I}(\texttt{husEd}_{ij} = S) \\ &+\beta_5\mathcal{I}(\texttt{momWork}_{ij} = Y) + \beta_6\mathcal{I}(\texttt{rural}_{ij} = Y)\big)\Big\},\end{aligned} \tag{7.1}$$

where $\mathcal{I}(\mathcal{P}) = 1$ when $\mathcal{P}$ is true and 0 otherwise, and $\texttt{immun}_{ij}$ is the immunity value for the jth child of the ith mother for $1 \leq i \leq 1596$, $1 \leq j \leq n_i$ and $n_i \in \{1, 2, 3\}$. Other variables in the model are defined analogously. We assume our bivariate random effects vectors satisfy

$$\begin{bmatrix} u_{0i} \\ u_{1i} \end{bmatrix} \overset{\text{ind.}}{\sim} \boldsymbol{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \right). \tag{7.2}$$

Note that although the data does have an additional level of nesting we do not consider it given the limitations in methodology as discussed in Chapter 6. Also we note that although Pebley, Goldman & Rodríguez (1996)[54] suggest use of a logistic model, the speed issues discussed in Chapter 4 limit us to a probit link.

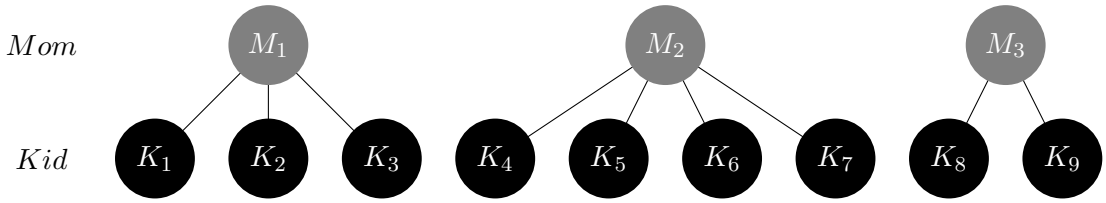


Figure 7.2: *A plot showing the two level structure of the data, specifically kids nested under mothers. The first row shows each mother, the second shows each kid of each mother.*

The fixed effects included in our model were selected using a generalised additive model selection scheme implemented via the R package "gamsel" (Chouldechova, Hastie & Spinu, 2018[15]) using an overlap grouped least absolute shrinkage selection operator (Jacob, et al., 2009[30]), where the most parisimonious model within one standard error from the minimum was chosen (Hastie, Tibshirani & Friedman, 2009[28]). The package allows including terms as non-linear splines in the model. Error estimates determined using 10 fold cross-validation suggest a good choice for the penalty parameter of a model fitted with least absolute shrinkage selection operator is $\lambda = 2.227$. Although the selected model suggests including primary parental education as a binary variable, given the ordinal nature of education we instead include whether or not the parents finished secondary school for interpretation sake. Linearity between indigenous population percentage in the child's community and their immunisation status was also confirmed visually using spline fits in the aforementioned package.

We fit the model specfied in equation (7.1) with the four following methods and compare the results:

1. **Markov Chain Monte Carlo** via the function `stan()` from the R package "rstan" (Stan Development Team, 2018[59]), using a Bayesian version of the model specified in equation (7.1) with independent $N(0, 1010)$ distributed diffuse priors for $\beta_0, \ldots, \beta_6$, and a member of the marginally noninformative family of covariance matrix priors described in Huang & Wand (2013)[29] for the 2×2 covariance matrix in equation (7.2). In the notation of the same paper, the hyperparameters were set to $\nu = 2$ and $A_1 = A_2 = 10^5$. A warm-up of 200000 was used followed by samples of 10000 retained for inference.

2. **Laplace approximation** via the function `glmer()` in the R package "lme4" (Bates, et al., 2018[5]).

3. **Expectation propagation** as described in Chapter 3 via our function `glmmEP()` in the R package "glmmEP" (Wand & Yu, 2020[69]).

4. **Data cloning** via the function `dclone()` in the R package "dclone" (Solymos, 2010[61]), with 10 clones.

We assume that the Markov Chain Monte Carlo approach gives close to exact results since a quadrature approach is not suitable for more than one random effect. Although other approaches exist, we include a comparison to the data cloning approach since it provides a reliable frequentist inference. The EP point estimates and approximate 95% confidence intervals are given in Table 7.2. With the exception of those involving parental education, each of the parameters is seen to be statistically significantly different from zero. As examples, the EP 95% confidence interval for β_1 of (-1.08, -0.454) indicates a lower prevalence of immunization in communities with higher percentages of indigenous people and the 95% confidence interval for σ_2 of (1.54, 4.35) shows that there is significant heterogeneity in the indigenous percentage effect across the 1595 families.

Figure 7.3 shows comparision of the fixed effects point estimates and approximate 95% confidence intervals of the four methods. Here we use Markov chain Monte Carlo as the gold standard method and apply the assumption that Markov chain Monte Carlo based 95% credible intervals are close to the 95% confidence intervals based on exact maximum likelihood. Although each method provided similar answers regarding significance of parameters, their point estimates and confidence interval coverage were notably different. Only Markov chain Monte Carlo produced a confidence interval for the intercept parameter which was significant. The other methods produced confidence intervals for the intercept parameter with a high proximity to 0, suggesting a lack of evidence to include it. Fixed effect estimates provided by data cloning show considerable bias and reduced standard errors, most notably for indigenous population percentage in the child's community and age of the child, where point estimates for each parameter were outside the credible interval of the Markov chain Monte Carlo approach. Using data cloning to fit this model took approximately 3hrs on a contempary laptop, considerably slower than the 12 seconds taken by EP. We also note data cloning has quite a few tuning parameters such as the number of clones and prior hyperparameter values. Given the slow speed of data cloning, it was difficult to assess the sensitivity of parameter choice.

Laplace approximation of the model performed slightly better than the data cloning approach, however was still poor. Given the multilevel model we fit has a low number of

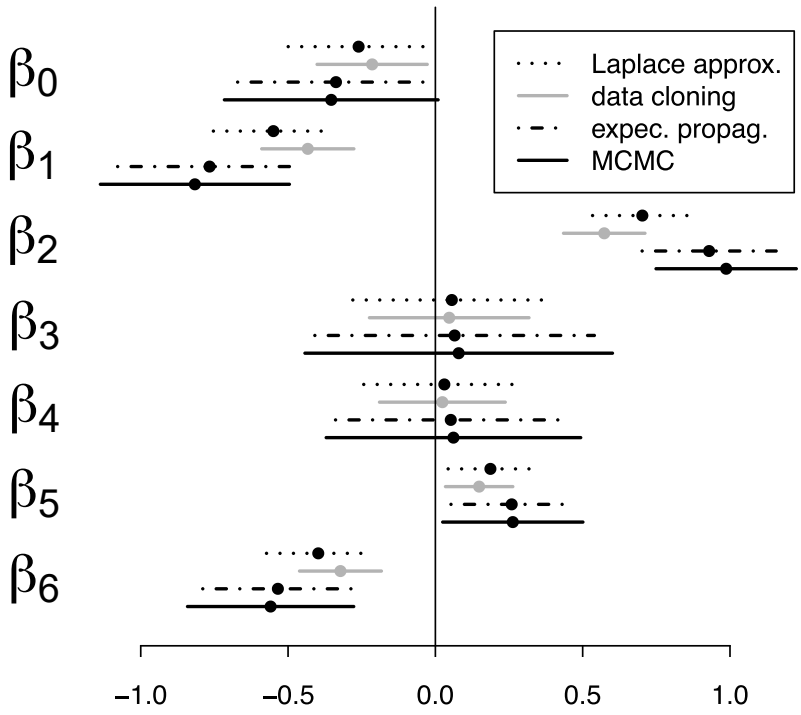


Figure 7.3: *Visual comparison of approximate 95% confidence/credible intervals for* $\beta_0, ..., \beta_6$ *for three approaches to fitting the probit mixed model equation (7.1) to the Guatemala immunization data. The approaches are Laplace approximation, data cloning, EP and Markov chain Monte Carlo (MCMC) with details given in the text.*

observations per group, the breakdown of Laplace approximation's asymptotic properties and a subsequent drop in performance is expected and consistent with literature, as documented in Vonesh (1996).[65] Additionally, we note that unlike our R function `glmmEP()`, the R function `glmer()` does not produce confidence intervals for random effect parameters. As such, although it took 6 seconds (less than half the time of EP) to fit the model with confidence intervals for the fixed effects only, to include confidence intervals for random effects as well took 124 seconds.

Expectation propagation provided consistent point estimates and confidence interval converage close to Markov chain Monte Carlo estimates whilst being considerably cheaper to compute. For this application, it was clearly the standout method.

Table 7.2: *A table of approximate maximum likelihood estimates and corresponding upper and lower 95% confidence intervals given by the EP methodology for the parameters in model equations (7.1) and (7.2).*

Parameter	**95%C.I. low**	**Estimate**	**95%C.I. upp**
β_0	-0.6711	-0.3373	-0.0035
β_1	-1.0783	-0.7663	-0.4543
β_2	0.7018	0.9291	1.1565
β_3	-0.4090	0.0653	0.5396
β_4	-0.3388	0.0523	0.4434
β_5	0.0531	0.2591	0.4650
β_6	-0.7895	-0.5345	-0.2795
σ_1	1.1622	1.5370	2.0328
σ_2	1.5407	2.5887	4.3494
ρ	-0.9486	-0.7821	-0.2766

7.3 Modelling donor attendance of the Australian Red Cross Blood Service

Blood donation is currently one of the only ways to collect blood for therapeutic use and as such is vital for healthcare in all nations. Given the often voluntary nature of blood donations, the rate of donor attendance can be quite variable. Such variations have implications for both financial and human resources of the health care system (Boksmati, et al., 2016).[9] As such, blood collection services seek to understand the factors involved in donor attendance.Although an abundance of work exists in understanding how to maintain steady donations, such as Charbonneau, et al. (2015),[13] Bagot, et al. (2013)[2] and Gemelli, et al. (2017),[20] most have small datasets and fail to provide an analysis using modern statistical tools.

Our dataset provided by the Australian Red Cross Blood Service has has over 3 million donation records collected in Queensland from 2015 to 2017. Table 7.3 provides a description of each of the 43 variables recorded. We aim to identify key factors that contribute to blood donors' appointment results (i.e. whether or not the donor will attend their scheduled donation), whilst also including more factors and a larger sample size than other studies. Given the size of the data, computationally efficient methods are required. As such, we compare GLMM fitting using our methodology to Laplace approximation. Furthermore, since the starting values of our EP algorithm depended on estimates from Laplace approximation, fitting this model allows an insight into the

feasibility of the new methodology.

Table 7.3: *A table that shows each of the variables in the analysis, explains what they mean, and lists the type of data they are.*

Variable name	**Description**
`ID`	A multilevel factor variable with each donors' ID number.
`attendance`	A two-level factor whether the donors either *attended* or *absent* their appointment. Didn't attend is the reference.
`age`	A continuous variable with the donor's age in years.
`sex`	A two-level factor variable with the donor's gender. Female is the reference.
`bloodType`	A nine-level factor variable with donor's the blood type. It can be $A+$, $A-$, $AB-$, $AB+$, $B-$, $B+$, $O-$, $O+$ or $Unknown$.
`pltDon`	A count variable with the number of platelet donations the donor has had.
`WBDon`	A count variable with the number of whole blood donations the donor has had.
`plsDon`	A count variable with the number of plasma donations the donor has had.
`apptTime`	A continuous positive variable with time of the appointment, measured from midnight.
`apptWkday`	A six-level factor varible with weekday of the appointment from 2 to 7, where 2,3,4,5,6,7 are weekdays Monday, Tuesday, Wednesday, Thursday, Friday, Saturday respectively.
`apptMth`	A 12-level factor variable with the month of the appointment.
`apptSeas`	A four-level factor variable with season of the appointment.
`apptYear`	A three-level factor variable with the year of the appointment.
`apptPubHolRad`	A five-level factor variable with the number of days either before or after the appointment to the nearest public holiday. If a public holiday is not within seven days of an appointment, this is 0. If an appointment is one day from a public holiday, this is 1. If an appointment is two days from a public holiday, this is 2. If an appointment is three day from a public holiday, this is 3, etc.

`apptCreDelta`	A continuous variable with the difference between date of creating the booking and actual donation date.
`apptBookType`	A seven-level factor with the type of booking made. *In centre* bookings are those made after the previous donation. *MCC inbound* is where the donor calls the centre. *MCC inbound* is where the centre calls the donor. *MCC unknown* is a booking made over the phone, but it is unsure who called who. *Community relations* are bookings made through community outreach. *Portal* are bookings made using the portal software and *Webbooking* are those made over the web.
`colTypeSched`	A four-level factor variable with the type of blood collection procedure scheduled to be conducted. It can be either *plasmapheresis*, *plateletpheresis*, *whole blood* or *no blood*.
`colTypeTaken`	A four-level factor variable with the type of blood collection procedure conducted. It can be either *plasmapheresis*, *plateletpheresis*, *whole blood* or *no blood* taken during appointment.
`centreID`	A multilevel factor variable with the centre where blood collection was scheduled. There are 25 centres represented by unique codes.
`centreType`	A two-level factor with the type of centre where the collection of blood was scheduled, i.e. whether it was a *static* collection room or a *mobile* blood van.
`centreLoc`	A two level factor whether the scheduled collection centre was *regional* or *metro*. Here *regional* is the reference category.
`lastDefType`	A 354 level factor with the donor's last deferral type code.
`lastDefLen`	A continuous variable with length of time in weeks since the donor last deferred an appointment. This is set to 0 for when the donor has not had a previous deferral.
`lastDefEndLen`	A continuous variable with the length of time in weeks since the donors last deferral ended. This is set to 0 for when the donor has not had a previous deferral.
`inbndSMS`	A count variable with the number of SMS to the Red Cross sent by the donor.

`inbndPhone`	A count variable with the number of phone calls to the Red Cross by the donor.
`inbndEmail`	A count variable with the number of emails sent to the Red Cross by the donor.
`inbndUnknown`	A count variable with the number of unknown communications sent to the Red Cross by the donor.
`inbndInternet`	A count variable with the number of internet communications to the Red Cross by the donor.
`outbndPhone`	A count variable with the number of phone calls to the donor by the Red Cross.
`outbndSMS`	A count variable with the number of SMS to the donor sent by the Red Cross.
`outbndEmail`	A count variable with the number of emails sent to the donor by the Red Cross.
`outbndUnknown`	A count variable with the number of unknown communications sent to the donor by the Red Cross.
`outbndInternet`	A count variable with the number of internet communications to the donor by the Red Cross.
`lastTTC`	A continuous variable with the time it took in the last appointment before the donor was on the couch ready for the phlebotomist to take blood.
`lastBleedTime`	A continuous variable with the time it took in the last appointment from needle in to needle out. This is linked to collection type.
`lastDonTime`	A continuous variable the total time it took from when the donor entered the Red Cross to when they finished giving blood, i.e. sum of `lastTTC` and `lastDonTime`.
`lastDonMultiArm`	A two-level factor variable whether or not the donor was punctured in one or both arms. $True$ if one arm and $False$ if both.
`lastDAEVVR`	A two-level factor variable with the last donor adverse event, specifically whether a vasovagal reaction took place. $True$ if one did happen and $False$ if one did not.

`lastDAEnonVVR` A two-level factor variable with the last donor adverse event, specifically whether a non-vasovagal reaction took place. *True* if one did happen and *False* if one did not.

7.3.1 Data cleaning

The dataset was pooled from many different sources and required cleaning. Donors were free to attend any room as many times as they wished in any location in Queensland, and as such, many donors attended different rooms for their appointments. The underlying structure of the data is quite complicated, involving donations nested in donors, crossed with rooms, where rooms are nested under area (see Figure 7.4). Additionally there is a time effect to the true structure of the data. However, given the limitations of the methodology presented in this thesis, we restrict the data to have one level of nesting; donations nested in donors (see Figure 7.5). We acheive this by filtering the donors so that only visits to their most popular room were retained.

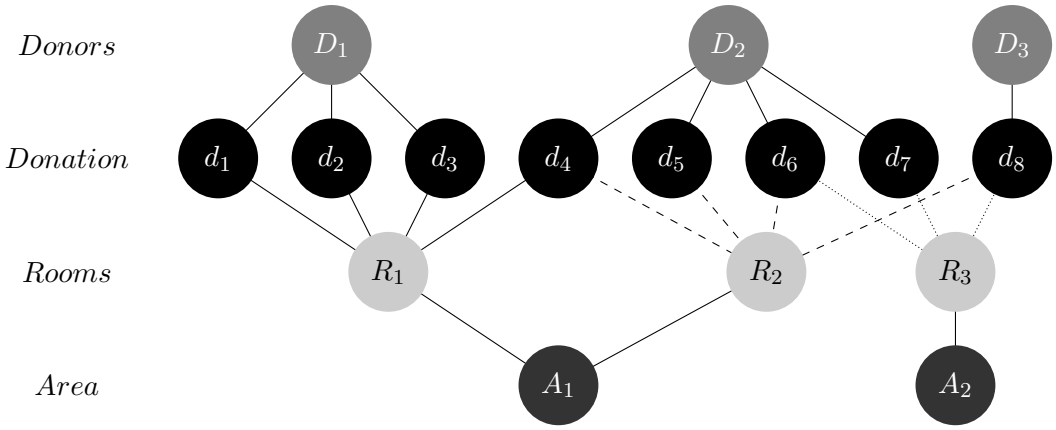


Figure 7.4: *A plot showing the two level multiple membership structure of the data, specifically donations nested under donors crossed with rooms, which are nested under area. The first row shows each donor, the second shows each donation of each donor, the third shows which room each donation was done in, and the fourth shows the area of each room. Note the time aspect of the data is not included in this graph.*

Some variables were reported as they were used in the collection rooms and as such were not in a format that lent themselves to statistical analysis. `apptTime`, which was in a date time format, was converted to seconds from midnight. The date component was also separated into years.

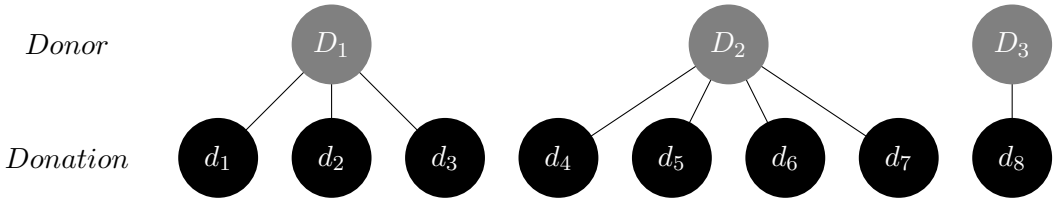


Figure 7.5: *A plot showing the one level structure of the data, specifically donations nested under donors. The first row shows each donor, the second shows each donation of each donor.*

Patient appointment result was recorded as a six-level factor with outcomes being; attended, blood was taken, cancelled, rescheduled, active or unknown. We converted it to a two-level factor called `attendance`, where donors either attended or were absent, with *absent* as the reference category.

Negative values for `apptCreDelta` were present in the dataset for patients that attended without booking in advance. These negative values were created as the donor's appointment time was recorded after their blood was taken by staff. Since it is not useful for prediction of donors attendance rate, we remove them from the dataset.

There are also a number of missing values present in the data. For the cases of `WBDon`, `pltDon`, `plsDon` and `lastDefLen` missing values may be indicative of first time donors. Given that the analysis provided is primarily a demonstration of methodology, we do not explore this and instead conduct a complete case analysis. A spurious observation where donor's age meant their last deferral length was not possible was also removed from the data.

The counts of contact for each communication method between the Red Cross and the donor were dichotomised from count variables that indicated how many times each form of communication had been conducted on each donor, to a binary variable recording whether there was no communication or at least 1 communication.

Finally there were a number of donors with an extremely high number of observations in the dataset (the highest donor had 2247 bookings). It is speculated that these were automated bookings based on various information flags, that were either not attended or cancelled. Since donors with extremely high numbers of bookings were not uncommon we did not remove them from our dataset.

Table 7.4: *A table reporting the shapes and recommended fits of each continuous variable from the R package "gamsel" (Chouldechova, Hastie & Spinu, 2018[15])*

Variable name	**Form selected**
`age`	Linear
`WBDon`	Bend point at 16.
$\log(\texttt{plsDon}+1)$	Bend points at 1.5, 2.5 and 3.5.
$\log(\texttt{pltDon}+1)$	Indicator of equalling zero and for positive, bend points at 1.5, 2 and 3.
`apptTime`	Bend points at 0.45 and 0.53.
`apptCreDelta`	Bend points at 25 and 50.
$\log(\texttt{lastDefLen}+1)$	Bend point at 1.2.
$\log(\texttt{lastDefEndLen}+1)$	Bend points at 1.6, 2.7 and 4.
$\log(\texttt{lastTTC}+1)$	Indicator of equalling zero and, for positive, bend points at 2, 2.5 and 3.
`lastBleedTime`	Bend points at 40 and 60.

7.3.2 Modelling continuous variables

We used the R package "gamsel" (Chouldechova, Hastie & Spinu, 2018[15]) to explore the relationship between each continuous covariate in the dataset and the response variable. Specifically, the R function `gamsel()` was used to automatically select whether variables were linear or non-linear using an overlap grouped least absolute shrinkage selection operator (Tibshirani, 1996[64]) via the R package "gamsel" (Chouldechova, Hastie & Spinu, 2018[15]) as in Section 7.2. This showed `plsDon`, `pltDon`, `lastDefLen`, `lastDefEndLen` and `lastTTC` had skewed non-linear relationships with `attendance`. As such we implemented a transform of $\log(x+1)$ and re-ran the `gamsel()` routine. To account for the non-linearity of some variables we decided to use a broken stick model where the break points were decided by visual inspection of the plots generated by `gamsel()` in conjunction with histograms. The results of this exploration are given in Table 7.4.

Categorical variables were broken into several binary variables each. The reference level of each variable is given in Table 7.3. All continuous variables were also scaled to have mean 0. After cleaning 2430370 observations remained.

7.3.3 Initial model

We first attempted to fit an exploratory model via Laplace approximation. This model included all variables as fixed effects, a random intercept, and random slopes for age. The probit mixed model we fit follows in equation (7.3), where $\mathcal{I}(\mathcal{P}) = 1$ when the logical statement $\mathcal{P}$ is true, $\mathcal{B}(\mathcal{P}; x) = 0$ when the logical statement is false, $\mathcal{B}(\mathcal{P}; x) = x$ when the logical statement is true, and $\texttt{attendance}_{ij}$ is the `attendance` value for the ith donation of the jth donor for $1 \leq i \leq 55657$, $1 \leq j \leq n_i$ and $n_i \in \{1, ..., 2247\}$. Other variables are defined analogously to $\texttt{attendance}_{ij}$. We assume our bivariate random effects vectors satisfy

$$\begin{bmatrix} u_{0i} \\ u_{1i} \end{bmatrix} \overset{\text{ind.}}{\sim} \boldsymbol{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \right). \tag{7.4}$$

Given both the data size and number of fixed effect parameter estimates in the model, the R function `glmer()` in the package "lme4" (Bates, et al., 2018[5]) was unable fit the model. As such, we turned to the function `glmmTMB()` (Magnusson, et al., 2019[39]) from the package "glmmTMB", which utilises the R package "Template Model Builder" (Kristensen, 2018[33]) as a backend for model fitting and as such provides a speed advantage for GLMMs with high numbers of fixed and random effects (Bolker, 2019[7]).

Fitting our model using `glmmTMB()` with all 2430370 observations that remained after cleaning on a high performance computing cluser took approximately 27hrs and used up to 50gb random access memory. We then attempted to fit the same model using our EP approach via a modified version of our R function `glmmEP()` (as in the R package "glmmEP"), with starting values given by the Laplace approximations from `glmmTMB()`. To find the maximum likelihood estimates for the parameters, we first tried using the R function `optim()` to compute a Nelder Mead algorithm with refinements via Broyden Fletcher Goldfarb Shanno algorithm as discussed in Section 3.2. After 2 days of continuous computing time on a high performance computing cluster the Nelder Mead phase of the routine had not completed, and it was decided to try a different method. In particular the unconstrained optimisation by quadratic approximation algorithm was selected for implementation via the function `uobyqa()` in the package "minqa" (Bates, et al., 2015[6]). The `uobyqa()` approach was significantly faster than the scheme using `optim()` approach, completing in approximately 1 day, and as such we use `uobyqa()` for further investigations.

We tried two methods to calculate the hessian matrix to calculate the confidence intervals for EP. The first method was using Broyden Fletcher Goldfarb Shanno algorithm via the R function `optim()` with `hessian = TRUE`, and the second was using the function `hessian()` from the R package "numDeriv" (Gilbert & Varadhan, 2019[22]). Both methods failed due to singularities. As such we were not able to obtain confidence intervals for EP. We did not make further attempts to obtain the hessian required since the initial fit was simply exploratory.

Except for the random effects components the difference between point estimates provided by Laplace approximation and EP were minimal. This is most likely due to the high number of donations per donor, which vary between 1 and 2247. The difference in random effects estimates are shown in Table 7.5. Finally we note that majority of the parameter estimates given by Laplace approximation were highly significant, which can be attributed to the large size of the dataset used in the study.

Table 7.5: *Table comparing the random effects point estimates obtained by Laplace approximations and by EP.*

Parameter	**Laplace approx.**	**EP approx.**
σ_1	0.4547	0.4544
σ_2	0.2263	0.2261
ρ	0.0538	0.0248

$$\mathcal{I}(\texttt{attendance}_{ij} = \textit{attended})|u_{0i}, u_{1i} \overset{\text{ind.}}{\sim} \text{Bernoulli}\Big\{\Phi\Big\{\beta_0 + u_{0i} + (\beta_1 + u_{1i})\texttt{age}_{ij}$$
$$+\beta_2\ \mathcal{I}(\texttt{sex}_{ij} = \textit{male}) + \beta_3\ \mathcal{I}(\texttt{bloodType}_{ij} = A+) + \beta_4\ \mathcal{I}(\texttt{bloodType}_{ij} = A-)$$
$$+\beta_5\ \mathcal{I}(\texttt{bloodType}_{ij} = AB-) + \beta_6\ \mathcal{I}(\texttt{bloodType}_{ij} = AB+) + \beta_7\ \mathcal{I}(\texttt{bloodType}_{ij} = B+)$$
$$+\beta_8\ \mathcal{I}(\texttt{bloodType}_{ij} = B-) + \beta_9\ \mathcal{I}(\texttt{bloodType}_{ij} = O+) + \beta_{10}\ \mathcal{I}(\texttt{bloodType}_{ij} = O-)$$
$$+\beta_{11}\ \texttt{WBDon}_{ij} + \beta_{12}\ \mathcal{B}(\texttt{WBDon}_{ij} > 16) + \beta_{13}\ \log(\texttt{plsDon} + 1)$$
$$+\beta_{14}\ \mathcal{B}(\log(\texttt{plsDon} + 1) > 1.5) + \beta_{15}\ \mathcal{B}(\log(\texttt{plsDon} + 1) > 2.5)$$
$$+\beta_{16}\ \mathcal{B}(\log(\texttt{plsDon} + 1) > 3.5) + \beta_{17}\ \log(\texttt{pltDon} + 1)$$
$$+\beta_{18}\ \mathcal{I}(\log(\texttt{pltDon} + 1) = 0) + \beta_{19}\ \mathcal{B}(\log(\texttt{pltDon} + 1) > 1.5)$$
$$+\beta_{20}\ \mathcal{B}(\log(\texttt{pltDon} + 1) > 2) + \beta_{21}\ \mathcal{B}(\log(\texttt{pltDon} + 1) > 3) + \beta_{22}\ \texttt{apptTime}$$
$$+\beta_{23}\ \mathcal{B}(\texttt{apptTime} > 0.45) + \beta_{24}\ \mathcal{B}(\texttt{apptTime} > 0.53) + \beta_{25}\ \mathcal{I}(\texttt{apptWkday} = 3)$$
$$+\beta_{26}\ \mathcal{I}(\texttt{apptWkday} = 4) + \beta_{27}\ \mathcal{I}(\texttt{apptWkday} = 5) + \beta_{28}\ \mathcal{I}(\texttt{apptWkday} = 6)$$
$$+\beta_{29}\ \mathcal{I}(\texttt{apptWkday} = 7) + \beta_{30}\ \mathcal{I}(\texttt{apptMth} = 2) + \beta_{31}\ \mathcal{I}(\texttt{apptMth} = 3)$$
$$+\beta_{32}\ \mathcal{I}(\texttt{apptMth} = 4) + \beta_{33}\ \mathcal{I}(\texttt{apptMth} = 5) + \beta_{34}\ \mathcal{I}(\texttt{apptMth} = 6)$$
$$+\beta_{35}\ \mathcal{I}(\texttt{apptMth} = 7) + \beta_{36}\ \mathcal{I}(\texttt{apptMth} = 8) + \beta_{37}\ \mathcal{I}(\texttt{apptMth} = 9)$$
$$+\beta_{38}\ \mathcal{I}(\texttt{apptMth} = 10) + \beta_{39}\ \mathcal{I}(\texttt{apptMth} = 11) + \beta_{40}\ \mathcal{I}(\texttt{apptMth} = 12)$$
$$+\beta_{41}\ \mathcal{I}(\texttt{apptYear} = 2016) + \beta_{42}\ \mathcal{I}(\texttt{apptYear} = 2017) + \beta_{43}\ \mathcal{I}(\texttt{apptPubHolRad} = 1)$$
$$+\beta_{44}\ \mathcal{I}(\texttt{apptPubHolRad} = 2) + \beta_{45}\ \mathcal{I}(\texttt{apptPubHolRad} = 3) + \beta_{46}\ \mathcal{I}(\texttt{apptPubHolRad} = 4)$$
$$+\beta_{47}\ \texttt{apptCreDelta} + \beta_{48}\ \mathcal{B}(\texttt{apptCreDelta} > 25) + \beta_{49}\ \mathcal{B}(\texttt{apptCreDelta} > 50)$$
$$+\beta_{50}\ \mathcal{I}(\texttt{apptBookType} = \textit{community relations}) + \beta_{51}\ \mathcal{I}(\texttt{apptBookType} = \textit{in centre})$$
$$+\beta_{52}\ \mathcal{I}(\texttt{apptBookType} = \textit{NCC inbound}) + \beta_{53}\ \mathcal{I}(\texttt{apptBookType} = \textit{NCC outbound})$$
$$+\beta_{54}\ \mathcal{I}(\texttt{apptBookType} = \textit{portal}) + \beta_{55}\ \mathcal{I}(\texttt{apptBookType} = \textit{web booking})$$
$$+\beta_{56}\ \mathcal{I}(\texttt{colTypeSched} = \textit{no blood}) + \beta_{57}\ \mathcal{I}(\texttt{colTypeSched} = \textit{plasmapheresis})$$
$$+\beta_{58}\ \mathcal{I}(\texttt{colTypeSched} = \textit{plateletpheresis}) + \beta_{59}\ \mathcal{I}(\texttt{colTypeTaken} = \textit{no blood})$$
$$+\beta_{60}\ \mathcal{I}(\texttt{colTypeTaken} = \textit{plasmapheresis}) + \beta_{61}\ \mathcal{I}(\texttt{colTypeTaken} = \textit{plateletpheresis})$$
$$+\beta_{62}\ \mathcal{I}(\texttt{centreType} = \textit{static}) + \beta_{63}\ \mathcal{I}(\texttt{centreType} = \textit{metro}) + \beta_{64}\ \log(\texttt{lastDefLen} + 1)$$
$$+\beta_{65}\ \mathcal{B}(\log(\texttt{lastDefLen} + 1) > 1.2) + \beta_{66}\ \log(\texttt{lastDefEndLen} + 1)$$
$$+\beta_{67}\ \mathcal{B}(\log(\texttt{lastDefEndLen} + 1) > 1.6) + \beta_{68}\ \mathcal{B}(\log(\texttt{lastDefEndLen} + 1) > 2.7)$$
$$+\beta_{69}\ \mathcal{B}(\log(\texttt{lastDefEndLen} + 1) > 4) + \beta_{70}\ \mathcal{I}(\texttt{inbndPhone} \leq 1) + \beta_{71}\ \mathcal{I}(\texttt{inbndSMS} \leq 1)$$
$$+\beta_{72}\ \mathcal{I}(\texttt{inbndEmail} \leq 1) + \beta_{73}\ \mathcal{I}(\texttt{inbndInternet} \leq 1) + \beta_{74}\ \mathcal{I}(\texttt{outbndPhone} \leq 1)$$
$$+\beta_{75}\ \mathcal{I}(\texttt{outbndSMS} \leq 1) + \beta_{76}\ \mathcal{I}(\texttt{outbndEmail} \leq 1) + \beta_{77}\ \mathcal{I}(\texttt{outbndInternet} \leq 1)$$
$$+\beta_{78}\ \mathcal{I}(\texttt{outbndLetter} \leq 1) + \beta_{79}\ \log(\texttt{lastTTC} + 1) + \beta_{80}\ \mathcal{I}(\log(\texttt{lastTTC} + 1) = 0)$$
$$+\beta_{81}\ \mathcal{B}(\log(\texttt{lastTTC} + 1) > 2) + \beta_{82}\ \mathcal{B}(\log(\texttt{lastTTC} + 1) > 2.5)$$
$$+\beta_{83}\ \mathcal{B}(\log(\texttt{lastTTC} + 1) > 3) + \beta_{84}\ \texttt{lastBleedTime} + \beta_{85}\ \mathcal{B}(\texttt{lastBleedTime} > 40)$$
$$+\beta_{86}\ \mathcal{B}(\texttt{lastBleedTime} > 60) + \beta_{87}\ \mathcal{I}(\texttt{lastDonMultiArm} = \textit{True})$$
$$+\beta_{88}\ \mathcal{I}(\texttt{lastDAEVVR}_{ij} = \textit{True}) + \beta_{89}\ \mathcal{I}(\texttt{lastDAEnonVVR}_{ij} = \textit{True})\Big\}\Big\}, \qquad (7.3)$$

7.3.4 Second model

With insights from the previous model, we aimed to improve model fit and obtain confidence intervals for the EP approach. We conducted variable reduction of the fixed effects using least absolute shrinkage selection operator (Tibshirani, 1996[64]) via the R package "glmnet" (Friedman, J., et al., 2020[17]). A plot of the error estimates given the penalty parameter λ determined via 10 fold cross-validation is shown in Figure 7.8. Although grounds for using the "one-standard-error" rule are established (Hastie, Tibshirani & Friedman, 2009[28]) we instead used $\lambda = \exp(-6)$ as suggested by Figure 7.8, since it provides a simpler model without an excessive increase in the standard error. Again we included a random intercept and slope for age. The probit mixed model we fit follows in equation (7.5), where $\texttt{attendance}_{ij}$, $\mathcal{B}(\mathcal{P}; x)$ and $\mathcal{I}(\mathcal{P})$ are as previously

$$
\begin{aligned}
\mathcal{I}(\texttt{attendance}_{ij} = \textit{attended}) | u_{0i}, u_{1i} \overset{\text{ind.}}{\sim} \text{Bernoulli}\Big\{ \Phi\Big\{ & \beta_0 + u_{0i} + (\beta_1 + u_{1i})\texttt{age}_{ij} \\
& + \beta_2\, \mathcal{I}(\texttt{sex}_{ij} = \textit{male}) + \beta_3\, \mathcal{I}(\texttt{bloodType}_{ij} = A+) + \beta_4\, \mathcal{I}(\texttt{bloodType}_{ij} = B+) \\
& + \beta_5\, \texttt{WBDon}_{ij} + \beta_6\, \mathcal{B}(\texttt{WBDon}_{ij} > 16) + \beta_7\, \log\big(\texttt{plsDon} + 1\big) \\
& + \beta_8\, \mathcal{B}\big(\log(\texttt{plsDon} + 1) > 1.5 \log\big) + \beta_9\, \mathcal{B}\big(\log(\texttt{plsDon} + 1) > 2.5\big) \\
& + \beta_{10}\, \mathcal{B}\big(\log(\texttt{plsDon} + 1) > 3.5\big) + \beta_{11}\, \mathcal{I}\big(\log(\texttt{pltDon}_{ij} + 1) = 0\big) \\
& + \beta_{12}\, \texttt{apptTime} + \beta_{13}\, \mathcal{B}(\texttt{apptTime} > 0.45) + \beta_{14}\, \mathcal{B}(\texttt{apptTime} > 0.53) \\
& + \beta_{15}\, \mathcal{I}(\texttt{apptWkday}_{ij} = 3) + \beta_{16}\, \mathcal{I}(\texttt{apptWkday}_{ij} = 7) \\
& + \beta_{17}\, \mathcal{I}(\texttt{apptMth}_{ij} = 11) + \beta_{18}\, \mathcal{I}(\texttt{apptMth}_{ij} = 12) \\
& + \beta_{19}\, \mathcal{I}(\texttt{apptYear}_{ij} = 2017) + \beta_{20}\, \texttt{apptCreDelta} \\
& + \beta_{21}\, \mathcal{B}(\texttt{apptCreDelta} > 25) + \beta_{22}\, \mathcal{B}(\texttt{apptCreDelta} > 50) \\
& + \beta_{23}\, \mathcal{I}(\texttt{apptBookType}_{ij} = \textit{MCC unknown}) \\
& + \beta_{24}\, \mathcal{I}(\texttt{colTypeSched}_{ij} = \textit{plasmapheresis collection}) \\
& + \beta_{25}\, \mathcal{I}(\texttt{colTypeTaken}_{ij} = \textit{plasmapheresis collection}) \\
& + \beta_{26}\, \log(\texttt{lastDefLen} + 1) + \beta_{27}\, \mathcal{B}\big(\log(\texttt{lastDefLen} + 1) > 1.2\big) \\
& + \beta_{28}\, \log(\texttt{lastDefEndLen} + 1) + \beta_{29}\, \mathcal{I}(\texttt{inbndPhone} > 1) \\
& + \beta_{30}\, \mathcal{I}(\texttt{outbndPhone} > 1) + \beta_{31}\, \mathcal{I}(\texttt{outbndSMS} > 1) \\
& + \beta_{32}\, \mathcal{I}(\texttt{outbndEmail} \geq 1) + \beta_{33}\, \mathcal{I}(\texttt{outbndLetter} \geq 1) \Big\} \Big\}, \qquad (7.5)
\end{aligned}
$$

defined. Other variables are defined analogously to $\texttt{attendance}_{ij}$. We assumed our bivariate random effects vectors satisfy equation (7.4).

As before we first obtained fits using Laplace approximation via `glmmTMB()` and used

them as starting values for the EP algorithm. Again both models were extremely heavy computationally, with each method taking approximately one day to fit. A difference is noticable between the estimates of β_{20}, where Laplace approximation provided much tighter confidence intervals than EP. This is a trend that is consistent across almost all parameters estimated, however much like the first fit, the fixed effect confidence intervals and point estimates for Laplace approximations are extremely close to those provided by EP. Figure 7.7 shows a comparison between Laplace approximation and EP. The random effects estimates were also extremely close between the two methods, as shown in Table 7.6. Differences between the models random effects predictions are somewhat visible in Figure 7.6, which shows best predictions of the random intercepts plotted against best predictions of slopes for 24 randomly chosen donors. While these differences are quite small, they are certainly notable for certain donors, such as donor 10, 18 and 29. As a whole however, there is not much separating the fit of each model. We present the point estimates with upper and lower confidence intervals for each parameter in Table 7.7.

7.3.4.1 Results of model fit

All of the parameters estimated in the model were statisically different from zero, except the parameter β_{10} (95% confidence intervals of $(-0.004, 0.0448)$) which was included to account for the nonlinear relationship of the number of plasma donations and attendance. Due to the large dataset, all the fixed effects parameters estimated have extremely tight confidence intervals, with the distance between upper and lower intervals ranging from 0.0035 to 0.0780. However, many variables have coefficients that are small in magnitude and as such although they are significant statistically their impact is minimal in reality.

Age (β_1 with 95% confidence interval $(0.0139, 0.0148)$) increased the probability of donation attendance, as did being male (β_2 with 95% confidence interval $(0.0626, 0.074)$) over being female. Blood types $A+$ and $B+$ (β_3 with 95% confidence interval $(0.0068, 0.0187)$ and β_4 with 95% confidence interval $(0.0159, 0.0293)$ respectively) increased the probability of donation attendance compared to all other blood types, although $B+$ had a larger effect on donation attendance than $A+$.

The number of whole blood donations per donor had a varying effect on donor attendance, in both magnitude and direction. For low values it decreased donation attendance (β_5 with 95% confidence interval $(-0.0046, -0.0023)$), however began increasing it after 16 donations (β_6 with 95% confidence interval $(0.0128, 0.0177)$). The relationship between the number of plasma donations per donor and attendance also varies in

magnitude, but is always negative (β_7 with 95% confidence interval $(-0.0849, -0.0676)$). It weakens between three (β_8 with 95% confidence interval $(0.0561, 0.0919)$) and 11 plasma donations (β_9 with 95% confidence interval $(-0.0575, -0.0158)$), before returning beyond original levels after 32 plasma donations (β_{10} with 95% confidence interval $(-0.004, 0.0448)$). Donors with no platelet donations (β_{11} with 95% confidence interval $(0.0017, 0.013)$) had increased attendance compared to those with platelet donations.

Donor last deferal length has a non-linear relationship with attendance, initially decreasing attendance up to approximately 2.5 days (β_{26} with 95% confidence interval $(-0.0243, -0.012)$), then increasing after this point (β_{27} with 95% confidence interval $(0.0684, 0.0869)$). Time since the last deferal ended was positively linked with donation attendance probability (β_{28} with 95% confidence interval $(0.0005, 0.0041)$).

Both scheduled and collected plasmapheresis donations (β_{24} with 95% confidence interval $(0.0276, 0.0331)$ and β_{25} with 95% confidence interval $(0.0052, 0.0103)$ respectively) were linked with increasing donation attendance.

Appointment time (β_{12} with 95% confidence interval $(-0.8924, -0.7444)$) was linked with decreasing attendance. Much like previous parameters, it has a non-linear relationship with attendance. After appointment time becomes greater than 0.45 (i.e. after approximately 10:45am) (β_{13} with 95% confidence interval $(0.3564, 0.6778)$) the negative effect on probability of attendance decreases, with a further decrease occuring for appointment times greater than 0.53 (i.e. after approximately 12:45am) (β_{14} with 95% confidence interval $(0.5093, 0.7761)$). Note there is an even spread of attendance times between .3 and 0.8 (i.e., 7:12am to 7:12pm), even though it can range from 0 to 1 for all patients, where 0 is 0hrs from midnight and 1 is 24hrs from midnight (i.e., each hour of the day adds 1/24).

Both the appointment month and day affect attendance probability. Appointments on Wednesdays and Saturdays (β_{15} with 95% confidence interval $(0.0139, 0.0184)$ and β_{16} with 95% confidence interval $(0.0059, 0.0116)$ respectively) were positively linked with donation attendance compared to other days in the week, although the effect size of appointments on Saturdays is half that on Wednesdays. The same trend is true for November and December appointments (β_{17} with 95% confidence interval $(0.008, 0.0115)$ and β_{18} with 95% confidence interval $(0.019, 0.0225)$ respectively), where both months were positively linked with donation attendance compared to other months. We also note donors in 2017 had an increased probability of attending appointments (β_{19} with 95% confidence interval $(0.1095, 0.1142)$).

The length of time between booking and attending the appointment (β_{20} with 95% confidence interval $(-0.0298, -0.0292)$) was negatively linked with donation attendance. As the length of time between booking and attending the appointment becomes greater than 25 days length (β_{21} with 95% confidence interval $(0.0266, 0.0276)$) negative effect becomes minimised, before increasing again as the length of time between booking and attending the appointment becomes greater than 50 days length (β_{22} with 95% confidence interval $(-0.0019, -0.001)$).

Appointments with booking type MCC unknown (β_{23} with 95% confidence interval $(0.2577, 0.2619)$) were the most positively linked variable with probability of donation attendance.

We found more than one text message from the Red Cross to the donor (β_{31} with 95% confidence interval $(0.0846, 0.0887)$) led to the largest increase in probability of donation attendance of any communication method. Letters and emails sent from the Red Cross to the donor lead to smaller increases in attendance probability (β_{33} with 95% confidence interval $(0.0123, 0.0162)$ and β_{32} with 95% confidence interval $(0.0065, 0.0106)$ respectively). Phone calls from the Red Cross to the donor and from the donor to the Red Cross were both linked to a decrease in the probability of appointment attendance (β_{30} with 95% confidence interval $(-0.0355, -0.0313)$ and β_{29} with 95% confidence interval $(-0.0137, -0.0096)$ respectively).

The random intercept parameter (σ_1 with 95% confidence interval $(0.4826, 0.4952)$) shows attendance probability of donors varies, while the random slope (σ_2 with 95% confidence interval $(0.2352, 0.2573)$) suggests the effect of age varies even less between patients. The correlations of the random intercept and slope (ρ with 95% confidence interval $(0.0017, 0.0591)$) are minimal and are likely to be negligible in reality.

In summary, from our model we are able to identify the following useful trends:

- The effect of age is dependent on the donor although in general older patients are more likely to attend their appointments, as are males compared to females. These results are supported by Gemelli, et al. (2017)[20] and Charbonneau, et al. (2015)[13] who shows older patients and males are more frequent donors. The variability of the effect of age may be caused by older patients being more experienced donors, in addition to other behavioural effects that occur with aging. Charbonneau, et al. (2015)[13] suggest females may stop attending due pregnancy, or may stop miss donations based on their menstrual cycle.
- Donors with blood type $B+$ are associated with attendance more than other blood

groups. This contradicts Gemelli, et al. (2017)[20] who found negative blood groups more common than positive groups.

- High numbers of previous plasma donations are linked to decreasing attendance, while the number of platelet and whole blood donations have negligible effect. Charbonneau, et al. (2015)[13] also supports the idea that the number of plasma donations has an affect on donation attendance. This may be because there is a large gap required between whole blood donations (nearly two months), where as plasma donations can be once a month. Although platelet donations can be once every seven days, donations can take up to 3 hours which may deter patients from returning.
- A long last deferral length is favourable for attendance, while time since the last deferral does not have any effect. Although the reasoning behind this is not entirely clear the result aligns with the findings in Spekman, et. al. (2019).[40]
- Donors with scheduled plasmapheresis donations are most likely to attend compared to other appointments. This might be because the minimum time between appointments is the shortest of all the appointment types.
- Early and late appointment times seem to have opposite effects on attendance, where early appointments decrease attendance and late appointments increase attendance. This may be due to patients work commitments, or having greater confidence to donate in later parts of the day after eating several meals.
- Wednesday and December are associated more with attendance than other weekdays and months respectively. Although it is difficult to intuit why Wednesday is associated in increasing donor attendance, it does make sense that people may have spare time over the holiday period during December, or may be influenced by the festive season.
- The booking type MCC unknown seems to give the largest improvement in donor attendance, however there is not much intuition behind this.
- More than one outbound text message led to the largest increase in attendance of the communication methods, followed by outbound letters. Gemelli et. al. (2018)[21] supports that SMS increased the odds of donors attendance. Outbound and inbound phone calls decreased attendance, possibly due to the large number of cancellation calls, deferrals or missing patient calls.

Table 7.6: *Table comparing the random effects point estimates obtained by Laplace approximations and by EP for the model specified in Equation (7.5).*

Parameter	**Laplace approx.**	**EP approx.**
σ_1	0.4876	0.4889
σ_2	0.2441	0.2460
ρ	0.0300	0.0304

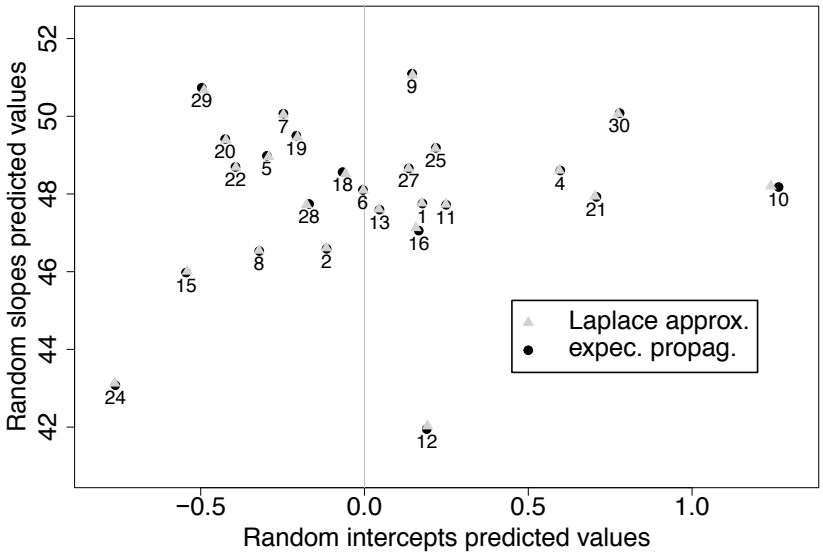


Figure 7.6: *Comparison plot for best predictions of the random intercepts and slopes for 24 donors between Laplace approximation and EP.*

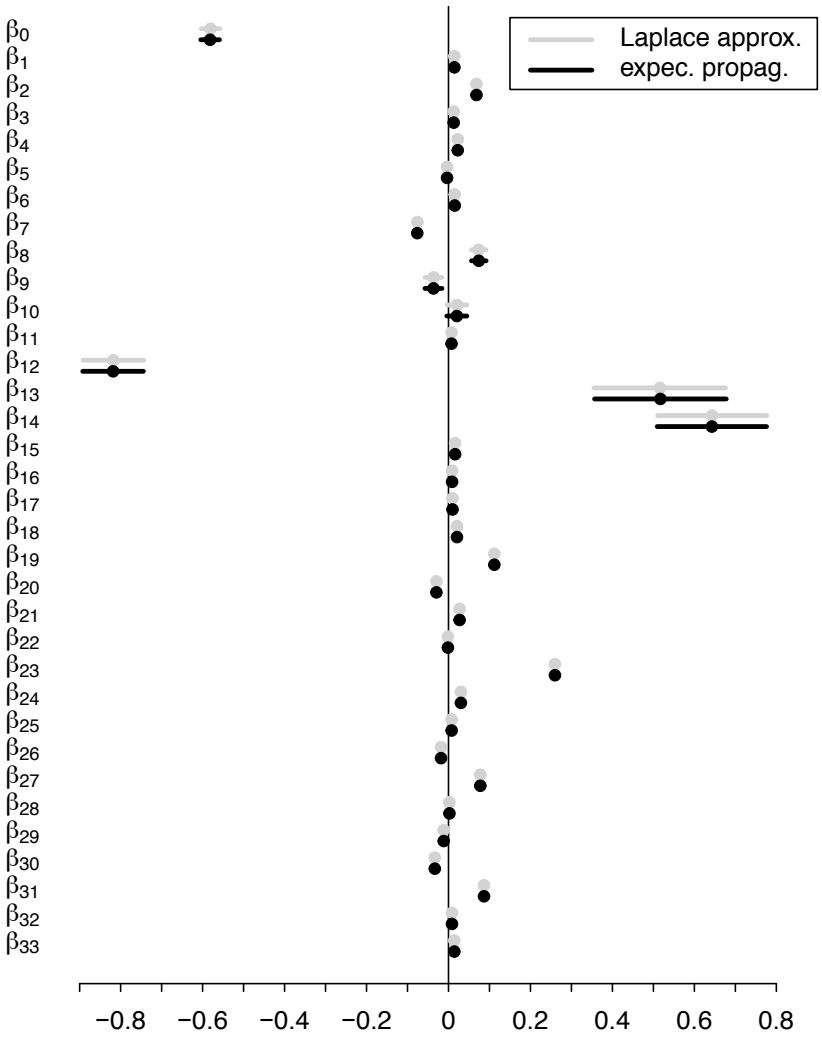


Figure 7.7: *Comparison plot of the confidence intervals and point estimates obtained with Laplace approximations and EP for the fixed effect parameters corresponding to the model specified in equation (7.5).*

7.4 Appendix

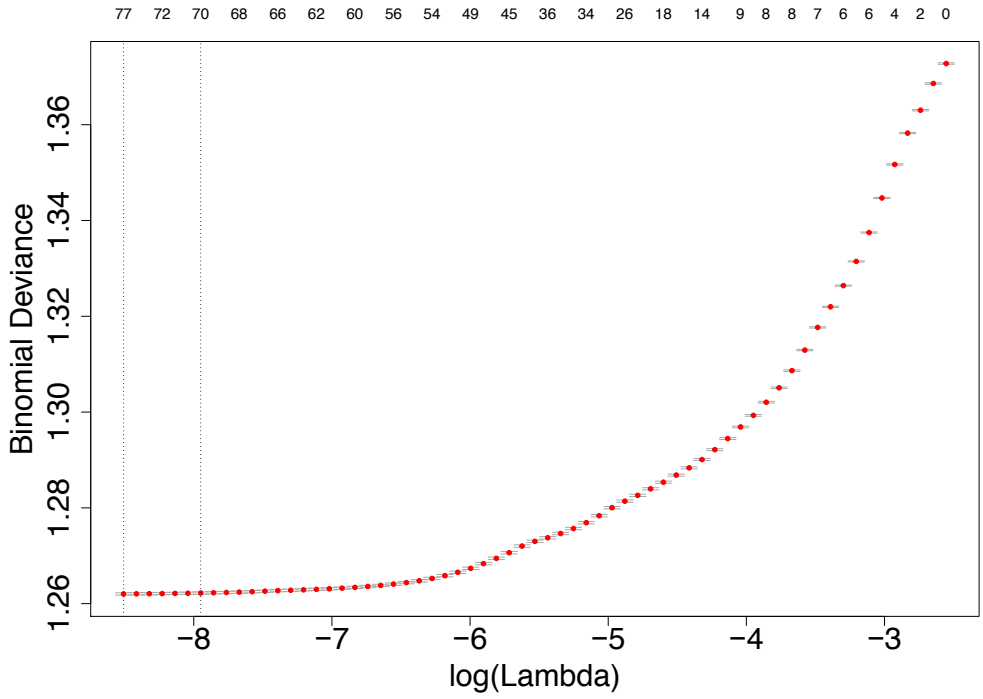


Figure 7.8: *Plot of error estimates given the penalty parameter λ determined via 10 fold cross-validation. The values shown horizontally on top of the plot are the number of variables included in the model.*

Table 7.7: *Table listing the fixed effects point estimates obtained by EP according to Equation (7.5). Note that they are the same as those by EP.*

Parameter	**95% C.I. low**	**Estimate**	**95% C.I. high**
β_0	-0.6045	-0.5816	-0.5588
β_1	0.0139	0.0144	0.0148
β_2	0.0626	0.0683	0.074
β_3	0.0068	0.0128	0.0187
β_4	0.0159	0.0226	0.0293
β_5	-0.0046	-0.0034	-0.0023
β_6	0.0128	0.0152	0.0177
β_7	-0.0849	-0.0762	-0.0676
β_8	0.0561	0.074	0.0919
β_9	-0.0575	-0.0366	-0.0158
β_{10}	-0.004	0.0204	0.0448
β_{11}	0.0017	0.0074	0.013

Parameter	**95% C.I. low**	**Estimate**	**95% C.I. high**
β_{12}	-0.8924	-0.8184	-0.7444
β_{13}	0.3564	0.5171	0.6778
β_{14}	0.5093	0.6427	0.7761
β_{15}	0.0139	0.0162	0.0184
β_{16}	0.0059	0.0087	0.0116
β_{17}	0.008	0.0098	0.0115
β_{18}	0.019	0.0207	0.0225
β_{19}	0.1095	0.1119	0.1142
β_{20}	-0.0298	-0.0295	-0.0292
β_{21}	0.0266	0.0271	0.0276
β_{22}	-0.0019	-0.0014	-0.001
β_{23}	0.2577	0.2598	0.2619
β_{24}	0.0276	0.0303	0.0331
β_{25}	0.0052	0.0077	0.0103
β_{26}	-0.0243	-0.0182	-0.012
β_{27}	0.0684	0.0777	0.0869
β_{28}	5e-04	0.0023	0.0041
β_{29}	-0.0137	-0.0116	-0.0096
β_{30}	-0.0355	-0.0334	-0.0313
β_{31}	0.0846	0.0867	0.0887
β_{32}	0.0065	0.0085	0.0106
β_{33}	0.0123	0.0143	0.0162
σ_1	0.4826	0.4889	0.4952
σ_2	0.2352	0.246	0.2573
ρ_{12}	0.0017	0.0304	0.0591

Chapter 8

Discussion and conclusion

This thesis aimed to add to the body of methodology used for frequentist inference of GLMMs by using the typically Bayesian idea of EP. We begun developing our methodology in Chapter 2 on a random intercept only probit model, as well as exploring possible solutions for computing confidence intervals. We explained how EP is used in our model, how we obtained starting values for the EP algorithm, and that the EP approximation of the likelihood surface is very similar around the maximum to the exact likelihood. Three possible solutions for obtaining confidence intervals were explored and we considered how they performed via a simulation study. For this study, approach II of EP suffered significantly in terms of computational performance. Keeping in mind the eventual extension to higher dimensional random effects, it was most practical to use a quasi-Newtown optimisation approach due to the algebraic cost of implementing the other approaches. However, assuming the random effects dimension remains constant, approach I is note worthy for its potential computational speed given it requires only a bisection search. Best prediction theory for the univariate model was also covered.

The generalised case of the probit model from Chapter 2 is explored in Chapter 3. Specifically, in this chapter we developed methodology to handle any number of fixed and random effects for one level models. We first explained how to obtain the EP likelihood approximation and the starting values required for the algorithm, before covering how to compute point estimates and confidence intervals using the quasi-Newton method selected in Chapter 2, as well as best predictors of the random effects. Our simulation study comparing the main approaches (Laplace approximation and Gauss-Hermite quadrature) and as well as a Markov chain Monte Carlo based method proved our EP approach is more consistent and accurate than other approaches in a variety of settings. In practice our methodology is hindered computationally given it is dependent

on Laplace approximations for starting values. Additional computational performance may be found by implementing more advanced optimisation algorithms over a limited search area. Although Greene (2012)[23] suggests against the use of BFGS optimisation, simulation studies throughout the thesis provide evidence that it is suitable, with good emperical coverage provided for maximum likelihood estimates.

We next explored methods for handling logistic models with one level of nesting in Chapter 4. The projections for the logistic model were difficult to solve since closed form solutions do not exist for them. In the univariate case we considered the use of a piecewise approximation, which compared well to the exact likelihood. However, the piecewise approximation did not extend well to the multivariate setting, and as such we explored solving the projections using quadrature. We proved a series of results which allowed the expression of the required multivariate integrals as univariate integrals. Additionally we showed how to express these integrals in a stable manner for computation. The quadrature method compared well to Laplace approximation with regards to accuracy, however had poor computational speed. This speed problem was caused by the use of trapisoidal quadrature. Although it was possible to solve these integrals using Gauss-Hermite quadrature, it was not possible to gauge the level of accuracy the quadrature provided. To our knowledge there is not a quadrature rule that allows approximation of the integral occuring in this chapter within an error bound. We left this as an open problem for future research.

Following the work of Chapter 4, in Chapter 5 we explored Poisson and negative-binomial models for count data. As with the logistic model, the projections required for both count models were not available with closed form solutions. As such we utilised similar results to express the integrals required in a stable and computationally efficient form. In the Poisson model we showed that for the settings of the simulation study, EP provided marginally superior coverage to Laplace approximations, however Gauss-Hermite quadrature appeared the closest to 95% empirical coverage. Similar findings were presented in the negative binomial model, where our EP method provided marginally better results than Laplace approximations and Gauss-Hermite quadrature. We provided confidence intervals for the shape parameter in the negative binomial model, which to our knowledge is not available elsewhere in literature. Given the time limitations of the thesis and complexity of the coding involved we did not present results of multivariate studies for count data. Additionally, we note that although were are able to give confidence intervals for the shape parameter for the negative binomial model, further research into the implications of doing so are required. Finally a careful analysis of the negative binomial model implementation provided in this thesis may be benefitial to providing better numerical stability as the shape parameter approaches infinity.

We investigated the extensions to higher level models in Chapter 6. We explained how crossed random effects and two level models can be framed and implemented using the message passing framework shown in the previous chapters, with only minor revisions to the algebra. Additionally we showed the simple alterations for calculation of confidence intervals. Simulation results showed the performance of the EP approach was similar to that of Laplace approximations, although given the challenging dataset no method performed particularly well. Due to time limitations, we were not able to conduct a more indepth exploration of EP in these models, and leave this as an open problem. Although we did not explicitly show it, the extensions shown in Chapters 3-4 should help give an idea of the steps required to accomodate for crossed random effects models with different link functions and responses types.

In Chapter 7 we applied methodology developed in Chapter 3 to two datasets. In the first dataset, we showed how EP is useful for data that is difficult to model with current methods. Specifically, we showed that for data with a low number of observations per grouping variable, EP is not only as accurate as MCMC, but also provides a good compromise in terms of speed. In the second dataset, where the number of observations per grouping variable was high, we showed that there is only a very minor difference between EP and Laplace approximation. Given the computational cost of implementing EP for this dataset, Laplace approximation was favourable. In this second analysis we also provided useful information on the practicalities of fitting large GLMMs particularly from the software standpoint, where not only do we suggest useful alternatives in the case where the ubiquitous "lme4" package breaks, but also contribute our own software package "glmmEP" (Wand & Yu, 2020[69]). Although we provide some interesting and novel findings from our analysis, a further exploration of the dataset considering things such as missing values and the spacio-temporal structure of the data could be useful for further inference. Additionally, it is possible to obtain starting values for EP by using a sample of the whole dataset, which may lead to computational savings. Further research into this may be of great use for applications of the EP methodology to large datasets. Finally, the large number of events for some patients is relevant to the underlying nature of the dataset and should not be ommited. In no major way would Figure 7.6 be effected by these observations other than both EP and Laplace approximation having similar estimates of fixed and random effects for those patients. The author notes that covergence issues experienced in the first model fit for the Australian Red Cross blood data was caused by highly correlated fixed effects.

In conclusion, this thesis has led to the development of novel methodology for one level binary and count GLMMs, as well as binary crossed random effects GLMMs. We created an efficient message passing framework for researchers to expand and develop

upon, allowing EP, an idea typically implemented in Bayesian settings, to be used for frequentist inference. Additionally we contributed the R package "glmmEP" (Wand & Yu, 2020[69]) and showed our method provides consistent and accurate parameter estimates for GLMMs, often with a clear improvements in empirical coverage variance parameters. Our work contributes to the underdeveloped frequentist setting for datasets with low numbers of observations per group.

References

[1] Azzalini, A. (2020). sn: The Skew-Normal and Related Distributions Such as the Skew-t (Version 1.5-5). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/sn/index.html

[2] Bagot, K. L., Bove, L. L., Masser, B. M., Bednall, T. C., & Buzza, M. (2013). Perceived deterrents to being a plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion, 53(5), 1108–1119.

[3] Barthelme, S. (2016). Part 1: The Expectation-Propagation algorithm, lecture recording, Week 5: "Bayesian statistics and algorithms", Centre International de Rencontres Mathématiques, delivered 2 March 2016.

[4] Bates, D., Maechler, M., & Bolker, B. (2019). mlmRev: Examples from Multilevel Modelling Software Review (Version 1.0-7). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/mlmRev/index.html

[5] Bates, D., Maechler, M., Bolker, B., Walker, S., & Christensen, R. H. B. (2018). lme4: Linear Mixed-Effects Models using 'Eigen' and S4 (Version 1.1-17). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/lme4/lme4.pdf

[6] Bates, D., Mullen, K. M., Nash, J. C., & Varadhan, R. (2014). minqa: Derivative-free optimization algorithms by quadratic approximation (Version 1.2.4). Retrieved from https://cran.r-project.org/web/packages/minqa/index.html

[7] Bolker, B. (2019). Getting started with the glmmTMB package. https://cran.r-project.org/web/packages/glmmTMB/vignettes/glmmTMB.pdf

[8] Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., & White, J. S. S. (2009). Generalized linear mixed models: a practical guide for ecology and evolution. Trends in Ecology & Evolution, 24(3), 127–135.

[9] Boksmati, N., Butler-Henderson, K., Anderson, K., & Sahama, T. (2016). The Effectiveness of SMS Reminders on Appointment Attendance: a Meta-Analysis. Journal of Medical Systems, 40(4), 90

[10] Borenstein, M., Hedges, L. V., Higgins, J. P. T., & Rothstein, H. R. (2009). Introduction to Meta-Analysis. West Sussex: John Wiley & Sons Ltd.

[11] Breslow, N. E., & Clayton, D. G. (1993). Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association, 88(421), 9–25.

[12] Carlin, B. P., & Gelfand, A. E. (1991). An for iterative Monte Carlo method nonconjugate Bayesian analysis. Statistics and Computing, 1, 119–128.

[13] Charbonneau, J., Cloutier, M. S. & Carrier, É. (2015). Why Do Blood Donors Lapse or Reduce Their Donation's Frequency? Transfusion Medicine Reviews, 30(1), 1–5.

[14] Chesher, A. (1997). Non-normal variation and regression to the mean. Stat Methods Med Res. 6(2), 147–66.

[15] Chouldechova, A., Hastie, T., & Spinu, V. (2018). gamsel: Fit Regularization Path for Generalized Additive Models. Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/gamsel/gamsel.pdf

[16] Collins, D. (2008). The performance of estimation methods for generalized linear mixed models. (Doctor of Phiosophy), University of Wollongong, Wollongong.

[17] Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N. & Qian, J., (2020). glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models. Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/glmnet/glmnet.pdf

[18] Gelman, A. (2006). Multilevel (hierarchical) modeling: What it can and cannot do. Technometrics, 48(3), 241–251.

[19] Gelman, A. & Hill, J. (2007). Data analysis using regression and multilevel/hierarchical models. Cambridge, Cambridge University Press.

[20] Gemelli, C. N., Hayman, J., & Waller, D. (2017). Frequent whole blood donors: understanding this population and predictors of lapse. Transfusion, 57(1), 108–114.

[21] Gemelli, C. N., Carver, A., Garn, A., Wright, S. T. & Davison, T. E. (2018). Evaluation of the impact of a personalized postdonation short messaging service on the retention of whole blood donors. Transfusion, 58(3), 701–709.

[22] Gilbert, P., & Varadhan, R. (2019). numDeriv: Accurate Numerical Derivatives (Version 2016.8-1.1). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/numDeriv/index.html

[23] Greene, W. H. (2012). Econometric Analysis. (Seventh ed.): Prentice Hall.

[24] Guo, J., Gabry, J., & Goodrich, B. (2017). rstan: R Interface to Stan (Version 2.19.2). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/rstan/index.html

[25] Hadfield, J. (2017). MCMCglmm: MCMC Generalised Linear Mixed Models (Version 2.25). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/MCMCglmm/index.html

[26] Hall, P., Johnstone, I. M., Ormerod, J. T., Wand, M. P., & Yu, J. C. F. (2020). Fast and Accurate Binary Response Mixed Model Analysis via Expectation Propagation, Journal of the American Statistical Association, 115:532, 1902–1916, DOI: 10.1080/01621459.2019.1665529.

[27] Handayani, D., Notodiputro, K. A., Sadik, K., & Kurnia, A. (2017). A comparative study of approximation methods for maximum likelihood estimation in generalized linear mixed models (GLMM). AIP Conference Proceedings, 1827, 020033.

[28] Hastie, T., Tibshirani, R., & Friedman, J. (2009). Shrinkage Methods. In The Elements of Statistical Learning: Data Mining, Inference, and Prediction. (Second ed., 61–79): Springer.

[29] Huang, A., & Wand, M. P. (2013). Simple Marginally Noninformative Prior Distributions for Covariance Matrices. Bayesian Analysis, 2(2), 439–452.

[30] Jacob, L., Obozinski, G. & Vert, J. P. (2009). Group lasso with overlap and graph lasso. 26th Annual International Conference on Machine Learning Conference Proceedings, 433—440.

[31] Kim, A. S. I. & Wand, M. P. (2016). The Explicit Form of Expectation Propagation for a Simple Statistical Model. Electronic Journal of Statistics, 10(1), 550–581.

[32] Kim, A. S. I. & Wand, M. P. (2017). On expectation propagation for generalised, linear and mixed models. Australian and New Zealand Journal of Statistics, 60(1), 75–102.

[33] Kristensen, K. (2018). TMB: Template Model Builder: A General Random Effect Tool Inspired by 'ADMB' (Version 1.7.15). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/TMB/index.html

[34] Lauritzen, S. L. (1992). Propagation of probabilities, means and variances in mixed graphical association models. Journal of American Statistical Association, 87(420), 1098–1108.

[35] Lele, S. R., Dennis, B. & Lutscher, F. (2007). Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods. Ecology Letters, 10(7), 551–563.

[36] Liu, Q. & Pierce, D. A. (1994). A Note on Gauss-Hermite Quadrature. Biometrika, 81(3), 624–629.

[37] Luts, J., & Wand, M. (2015). Variational Inference for Count Response Semiparametric Regression. Bayesian Analysis, 10(4), 991–1023.

[38] Magnus, J. R. & Neudecker, H. (1999). Kronecker products, the vec operator and the Moore-Penrose inverse. Matrix Differential Calculus with Applications in Statistics and Econometrics. 3 ed.: Wiley, 112.

[39] Magnusson, A., Skaug, H., Nielsen, A., Berg, C., Kristensen, K., Maechler, M., ... Bolker, B. (2019). glmmTMB: Generalized Linear Mixed Models using Template Model Builder (Version 0.2.3). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/glmmTMB/index.html

[40] Spekman, M. L. C., van Tilburg, T. G. & Merz, E. M. (2019). Do deferred donors continue their donations? A large-scale register study on whole blood donor return in the Netherlands. Transfusion, 59(12), 3657–3665.

[41] McCulloch, C. E., Searle, S. R., & Neuhaus, J. M. (2008). Generalized, Linear, and Mixed Models, New York, John Wiley & Sons.

[42] McCulloch, C. E., & Neuhaus, J. M. (2012). Prediction of random effects in linear and generalized linear models under model misspecification. Biometrics, 67(1), 270–279.

[43] Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. Seventeenth Conference on Uncertainty in Artificial Intelligence Conference Proceedings, 362–369.

[44] Minka, T. P. 2005. Divergence measures and message passing. Microsoft Research Technical Report Series, MSR-TR-2005-173, 1–17.

[45] Minka, T., Guiver, J., Winn, J. & Zaykov, Y. 2014. Infer.NET 2.6. Microsoft Research Cambridge. http://research.microsoft.com/infernet

[46] Monahan, J. F., & Stefanski, L. A. (1989). Normal scale mixture approximations to F*(z) and computation of the logistic-normal integral. In N. Balakrishnan (Ed.), Handbook of the Logistic Distribution. 123, 529–540.

[47] Nolan, T. H., & Wand, M. P. (2017). Accurate logistic variational message passing: algebraic and numerical details. Stat, 6(1), 102–112.

[48] Novomestky, F. (2020). matrixcalc: Collection of functions for matrix calculations (Version 1.0-3). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/matrixcalc/index.html

[49] Ogden, H. E. (2015). A sequential reduction method for inference in generalized linear mixed models. Electronic Journal of Statistics, 9(1), 135–152.

[50] Ogden, H. (2019). glmmsr: Fit a Generalized Linear Mixed Model (Version 0.2.3). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/glmmsr/index.html

[51] Opper, M. (1999). A Bayesian Approach to On-line Learning. In D. Saad (Ed.), On-Line Learning in Neural Networks (363–378): Cambridge University Press.

[52] Opper, M. & Winther, O. (2000). Gaussian processes for classification: Mean field algorithms. Neural Computation, 12(11), 2655–2684.

[53] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, California, Morgan Kaugmann Publishers, Inc.

[54] Pebley, A., Goldman, N.,& Rodríguez, G. (1996). Prenatal and delivery care and childhood immunization in Guatemala: Do family and community matter? Demography, 33(2), 231–247.

[55] Pinheiro, J. C., & Bates, D. M. (2000). Mixed-Effects Models in S and S-PLUS, New York, Springer.

[56] R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[57] Raudenbush, S. W. (1993). A Crossed Random Effects Model for Unbalanced Data with Applications in Cross-Sectional and Longitudinal Research. Journal of Educational Statistics. 18(4), 321–349.

[58] Rohde, D. & Wand, M. P. (2015). Semiparametric Mean Field Variational Bayes: General Principles and Numerical Issues. Journal of Machine Learning Research, 17(172), 1–47.

[59] Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. http://mc-stan.org/.

[60] Scott, M. A., Simonoff, J. S., & Marx, B. D. (2013). The SAGE Handbook of Multilevel Modeling.

[61] Solymos, P. (2010). dclone:Data Cloning in R. The R Journal, 2(2), 29–37.

[62] Steenbergen, M. R., & Jones, B. S. (2002). Modeling Multilevel Data Structures. American Journal of Political Science, 46(1), 218–237.

[63] Teunissen, P. J. G. (2007). Best prediction in linear models with mixed integer/real unknowns: theory and application. Journal of Geodesy 81, 759–780.

[64] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1), 267–288.

[65] Vonesh, E. F. (1996). A note on the use of Laplace's approximation for nonlinear mixed-effects models. Biometrika, 83(2), 447–452.

[66] Wainwright, M. J. & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference. Foundations and Trends in Machine Learning, 1.

[67] Wand, M. P & Jones, M. C. (1994). Kernel Smoothing. Chapman & Hall/CRC Monographs on Statistics & Applied Probability (60). Boca Raton, Chapman & Hall.

[68] Wand, M. P. (2017). Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing. Journal of the American Statistical Association, 112(517), 137–168.

[69] Wand, M. P. & Yu, J. C. F. (2020). glmmEP: Generalized Linear Mixed Model Analysis via Expectation Propagation (Version 1.0-3.1). Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/glmmEP/index.html

[70] Winn, J. & Bishop, C. (2005). Variational Message Passing. Journal of Machine Learning Research, 6, 661–694

[71] Wolfinger, R., & O'connell, M. (1993). Generalized linear mixed models a pseudo-likelihood approach. Journal of Statistical Computation and Simulation, 48, 233–243.