Parameter Estimation for Mixtures of Generalized Linear Mixed-effects Models

by

Lihua Wang

(Under the direction of Daniel Hall)

Abstract

Finite mixtures of simpler component models such as mixtures of normals and mixtures of generalized linear models (GLM) have proven useful for modelling data arising from a heterogeneous population, typically under an independence assumption. Mixed-effects models are often used to handle correlation as arises in longitudinal or other clustered data. In Chapter 3 of this dissertation, we present a more general class of models consisting of finite mixtures of generalized linear mixed effect models to handle correlation and heterogeneity simultaneously. For this class of models, we consider maximum likelihood (ML) as our main approach to estimation. Due to the complexity of the marginal loglikelihood of this model, the EM algorithm is employed to facilitate computation. To evaluate the integral in the E-step, when assuming normally distributed random effects, we consider numerical integration methods such as ordinary Gaussian quadrature (OGQ) and adaptive Gaussian quadrature (AGQ). We discuss nonparametric ML estimation (Aitkin, 1999) when we relax the normal assumption on the random effects. We also present the methods for computing the information matrix. In Chapter 4, restricted maximum likelihood method (REML) for Zero-Inflated (ZI) mixed effect models are developed. Zero-Inflated mixed effect models are submodels of two-component mixtures of GLMMs with one component degenerate to zero. For this type of

models, we adapt an estimator of variance components proposed by Liao and Lipsitz (2002) and think this method is more in the spirit of REML estimation in linear mixed effect models. This estimator is obtained based upon correcting the bias in the profile score function of the variance components. The idea is from McCullagh and Tibshirani (1990). The estimating procedure involves Monte Carlo EM algorithm which uses important sampling to generate random variates to construct Monte Carlo approximations at E-step. Simulation results show that the estimates of variance component parameters obtained from the REML method have significantly less bias than corresponding estimates from ML estimation method. In Chapter 5, we discuss some issues we encountered in the research and point out the potential topics for future research.

INDEX WORDS: Mixture models, Generalized linear mixed effect models, Maximum likelihood estimation, Zero-inflated models, Restricted maximum likelihood estimation

Parameter Estimation for Mixtures of Generalized Linear Mixed-effects Models

by

Lihua Wang

BEC, Renmin University of China, P. R. China, 1993

MEC, Renmin University of China, P. R. China, 1996

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2004

Parameter Estimation for Mixtures of Generalized Linear Mixed-effects Models

by

Lihua Wang

Approved:

Major Professor: Daniel Hall

Committee: Ishwar Basawa
Gauri Datta
Somnath Datta
Mary Meyer

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2004

Dedication

To my beloved mother ... ...

Acknowledgments

First, I would like to express my sincerely appreciation to my advisor Dr. Daniel Hall for his insight, wise direction, constant encouragement, and for his support in so many aspects. I would never finish my dissertation without them. He is such a wonderful advisor: not only knowledgeable, enthusiasm and confident, but also truly care for his students. He is a truly mentor and it has been such a pleasure to work under him.

I am also very grateful to my committee members: Dr. Ishwar Basawa, Dr. Gauri Datta, Dr. Mary Meyer and Dr. Somnath Datta for their assistance and valuable suggestions for my dissertation and also for all the help they have offered to me for the past 4 years. I also want to take this opportunity to thanks all the professors, stuff, my classmates and all my friends in UGA. I would not have so much fun without you. I wish you all the best.

Special thanks to my dearest friend Xiaohong Wang. She is a wonderful person and I am lucky to have her as best friend. Her care and encouragement drive me through the hard time. Thanks very much, Xiaohong, I treasure our friendship all life long.

I am so indebted to have a supportive family: my parents Huailiang Wang, Shuting Zhang and my brothers Jinhua Wang, Junhua Wang and Xihua Wang. Thanks you all for the long-term support and encouragement. I feel blissful with your love. Last, but not the least, I am deeply grateful to my husband, Ruijie Niu, for his understanding, love and always being there as a loyal listener. I still have a long way to go and I want to go with you.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Finite mixture models, such as mixtures of normals (Everitt and Hand, 1981; McLachlan and Basford, 1988) and mixtures of generalized linear models (Jansen, 1993; Dietz and Böhning, 1997) have proven useful for modeling data arising from a heterogeneous population, typically under an independence assumption. Mixed-effects models (Verbeke and Molenberghs, 2000; Breslow and Clayton, 1993) are often used to handle correlation as arises in longitudinal or other clustered data. There are situations where data not only exhibit heterogeneity but also are correlated by the experimental design. To better explain data with these characteristics, I develop a new class of regression models consisting of finite mixtures of generalized linear mixed effect models (mixtures of GLMMs) to handle correlation and heterogeneity simultaneously. This class can be viewed as an extension of finite mixtures of generalized linear models (Jansen, 1993) obtained by adding random effects to each component. Generalized linear models (GLMs), finite mixtures of GLMs and many other models are special cases of this broad class.

Parameter estimation is always one of the most important aspects of statistical inference for any model. Many previous efforts have been made at parameter estimation for GLMMs and mixtures of GLMs without random effects. In the mixed model context, Hall (2000) applied ML estimation and Yau and Lee (2001) applied hierarchical likelihood method to zero-inflated (ZI) mixed models. We present ML estimation with the EM algorithm for normal random effect mixture of GLMMs and nonparametric maximum likelihood (NPML) methods when assuming random effect distribution is unknown. Due to the difficulties of

evaluating the integral in the E step of the EM algorithm, numerical integration methods are employed.

One of the special cases of the two-component mixture occurs when one component is a degenerate distribution with point mass of one at zero. Such models are known as zero-inflated regression models and include zero-inflated Poisson (ZIP; Lambert, 1992), zero-inflated negative binomial, zero-inflated binomial (ZIB; Hall, 2000) and others (see Ridout, et al., 1998 for a review). When random effects are incorporated in these models, they become zero-inflated mixed models, which fall in the general class of mixtures of GLMMs that we consider here. That is, one component is zero, the other component is a GLMM type model. In this context, Hall (2000) considered maximum likelihood (ML) estimation for ZI-mixed Poisson and ZI-mixed binomial models. Yau and Lee (2001) considered an estimation method based on hierarchical or h-likelihood (Lee and Nelder, 1996). For estimation of the variance components associated with random effects in these models, the ML estimators are well known to be biased downward, which motivates a bias corrected variance component estimator, such as that provided by restricted maximum likelihood (REML, Patterson and Thompson, 1971) in a linear mixed model context. Yau and Lee propose a REML-like method of estimation which proceeds by iteratively fitting a linear mixed model via REML. From the perspective of the model fitting algorithm, this procedure is natural and appealing. However, it is not clear that this approach eliminates the regression parameter from the objective function for variance component estimation. That is, its connection to REML as a nuisance parameter elimination method is unclear. The accuracy of results from this method is also questionable (simulation study will be provided later). Alternatively, we adapt an estimator of variance components proposed by Liao and Lipsitz (2002) for ZI-mixed effect models. This estimator is obtained by correcting the bias in the profile score function of the variance components. The idea is from McCullagh and Tibshirani (1990). Based on our simulation results, this estimator has much less bias compared to the other two methods mentioned above.

The objectives of this dissertation are to present two-component GLMMs, to estimate the parameters of this model by ML and nonparametric ML estimation via EM algorithm, and to construct information matrix for the standard errors of parameter estimates. We also present REML estimation method for ZI-mixed effect models and carry out simulation study to compare different estimation methods in that context.

Chapter 2 describes some basic concepts, and reviews GLMMs, finite mixture models, the theory of the EM algorithm, and the theory of REML estimation. In Chapter 3, we formulate the two-component mixture of GLMMs, outline the EM algorithm and consider various methods of handling the required integration with respect to the random effects. At the end of Chapter 3, two real data examples are discussed and used to illustrate the methodology. In Chapter 4, REML estimation method is developed for ZI-mixed effect models. We also describe the algorithm to perform this method. ML estimation, REML estimation and Yau and Lee's method are compared via simulation study. A real data set is used to illustrate our method. Chapter 5 presents a discussion, including some potential future work topics.

Chapter 2

Literature Review and Preliminaries

This chapter provides a review of relevant literature as well as some statistical methods and techniques that will be needed in subsequent chapters.

2.1 Basic Concepts

2.1.1 Exponential dispersion family

Suppose a random variable Y_i (with mean μ_i) has a probability density function or probability mass function of the form of

$$f(y_i;\theta_i,\phi) = h(y_i,\phi)\exp\left\{\frac{y_i\theta_i - \kappa(\theta_i)}{a(\phi)}\right\}, \tag{2.1.1}$$

where ϕ is a (constant) dispersion parameter, θ_i is the natural or canonical parameter and can be expressed as some function of mean μ_i, and $\kappa(\theta_i)$ is the cumulant generating function. Then, the distribution of Y_i belongs to the exponential dispersion family. Many of the most commonly used distributions such as the normal, gamma, poisson and binomial are in this family.

2.1.2 GLM

To unify the analysis of normal and some important types of non-normal data, Nelder and Wedderburn (1972) proposed the generalized linear model (GLM). GLMs have three parts. They are:

- Systematic part (linear predictor): $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$.

- Random part: Y_i's are independent random variables each with $E(Y_i) = \mu_i$ and each with density (2.1.1).

- Link function: $g(\mu_i) = \eta_i$, where g(.) is a one to one and differentiable function. For canonical link, $\theta_i = \eta_i$.

For GLMs, we have $E(Y_i) = \mu_i = \kappa'(\theta_i)$ and $Var(Y_i) = a_i(\phi)\kappa''(\theta_i) = a_i(\phi)v(\mu_i)$, where $v(\mu_i)$ is called the variance function. In addition, $\theta_i = (\kappa')^{-1}[g^{-1}(\eta_i)]$.

2.1.3 Score function

If we denote the unknown parameters to be $\boldsymbol{\delta} = (\boldsymbol{\beta}^T, \phi)^T$ and the joint loglikelihood of independent $Y_1, \ldots, Y_n$ to be $\ell(\boldsymbol{\delta}; \boldsymbol{y}) = \sum_{i=1}^n \ell_i(\boldsymbol{\delta}; y_i) = \sum_{i=1}^n \log f(y_i; \boldsymbol{\delta})$, then the first derivative of the loglikelihood is the score function and can be expressed as:

$$S = S(\boldsymbol{\delta}) := \frac{\partial \ell(\boldsymbol{\delta}; \boldsymbol{y})}{\partial \boldsymbol{\delta}} = \sum_{i=1}^n \frac{\partial \ell_i(\boldsymbol{\delta}; y_i)}{\partial \boldsymbol{\delta}} = \sum_{i=1}^n S_i, \tag{2.1.2}$$

with $E(S) = E(S_i) = 0$. Further, we call the second derivative of the loglikelihood the Hessian matrix and denote it as H. The relationship between the score function and the Hessian matrix can be expressed as:

$$Var(S) = E(SS^T) = -E(H). \tag{2.1.3}$$

Each of the three terms in above equation is the Fisher information matrix which we denote $I(\boldsymbol{\delta})$. Usually, the Hessian matrix or Fisher information matrix are useful for deriving the standard errors of parameter estimates.

2.1.4 Asymptotic properties of ML estimates

Under certain regularity conditions (see Fahrmeir and Tutz, Ch.2), we have the following properties of ML estimates $\hat{\boldsymbol{\delta}}$ of $\boldsymbol{\delta}$:

- Consistency: As $n \to \infty$, $\hat{\boldsymbol{\delta}}_n \xrightarrow{p} \boldsymbol{\delta}$ (weak consistency); $\hat{\boldsymbol{\delta}}_n \to \boldsymbol{\delta}$ with probability 1 (strong consistency). Here $\hat{\boldsymbol{\delta}}_n$ denotes the sequence of ML estimates based on samples of size n.

- Asymptotic Normality:

$$\sqrt{n}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{I}^{-1}(\boldsymbol{\delta}))$$

  where $\boldsymbol{I}(\boldsymbol{\delta})$ is the Fisher information matrix defined in (2.1.3).

2.1.5 Over-dispersion

The existence of greater variation than predicted by the sampling model is called over-dispersion (Agresti, 1990). Over-dispersion is not uncommon in practice, especially for bounded count (seemingly binomial) data and unbounded count (seemingly Poisson) data. This phenomenon might arise in a number of ways, but there are two common causes. One is because of clustering in the population. This induces the observations on different subjects to be positively correlated rather than independent. Families, litters, etc, are common instances of natural clusters in populations. Another common way overdispersion happens is because the true sampling distribution is a mixture of different distributions such as Poisson etc.

2.1.6 Heterogeneity

A population is termed heterogeneous (Dietz and Böhning, 1997) if it contains subpopulations with different means, variances, relationships between response and covariates, or other distributional features. The heterogeneity is called unobserved if it is not known to which subpopulations the individuals of a sample belong.

2.2 GLMMs

GLMMs extend GLMs naturally by adding random terms in the linear predictor to account for overdispersion, correlation, and/or heterogeneity in the data. Since correlation is a natural feature of longitudinal and other clustered data, GLMMs have been used extensively for such data (Aitkin, 1996; Stiratelli, et al. , 1984; Zeger, et al. , 1988; etc). Generalized linear mixed models for clustered data are defined as follows.

Suppose that the observations on the ith cluster consist of responses y_{ij}, covariates $\boldsymbol{x}_{ij}$ and $\boldsymbol{z}_{ij}$ associated with the fixed and random effects respectively, for $i = 1, \ldots, K$ and $j = 1, \ldots, t_i$. Given a q dimensional vector of unobservable random effects $\boldsymbol{b}_i$, the y_{ij} are independent with means $E(y_{ij}|\boldsymbol{b}_i) = \mu_{ij}(\boldsymbol{b}_i)$ and variances $\text{var}(y_{ij}|\boldsymbol{b}_i) = a_i(\phi)v(\mu_{ij}(\boldsymbol{b}_i))$. We should note that the conditional means μ_{ij} depend on random effect $\boldsymbol{b}_i$. Similar to GLMs, the GLMM components are:

- Linear predictor: $\eta_{ij}(\boldsymbol{b}_i) = \boldsymbol{x}_{ij}^T\boldsymbol{\beta} + \boldsymbol{z}_{ij}\boldsymbol{b}_i$.
- Random part: Conditional on random effects $\boldsymbol{b}_i$, Y_{ij}'s are independent random variables with conditional densities belong to exponential dispersion family (2.1.1) and have conditional means and variances as above.
- Link function: $g(\mu_{ij}(\boldsymbol{b}_i)) = \eta_{ij}(\boldsymbol{b}_i)$
- $\boldsymbol{b}_i$ has mean $\mathbf{0}$ and follows distribution $\boldsymbol{F}$. Commonly $\boldsymbol{F}$ is assumed to be the multivariate normal with variance-covariance matrix $\boldsymbol{D} = \boldsymbol{D}(\boldsymbol{\theta})$.

Parameter estimation and statistical properties of such models have drawn a great deal of attention. As in GLMs, the generally preferred method of estimation is maximum likelihood. However, because of the nonlinearity of the model and the presence of random effects, obtaining the (marginal) likelihood of the model requires a difficult, often intractable, integration with respect to the random effects' distribution. Many approaches have been proposed to handle this integration so that ML can be accomplished. For lower dimensional random effects (e.g., ≤ 2), numerical integration methods such as ordinary Gaussian quadrature and adaptive Gaussian quadrature can be employed to evaluate the integral. For higher dimensional random effects, simulation-based approximation methods have been proposed to obtain the ML estimation. This type of method includes Monte Carlo EM (McCulloch, 1997), Monte Carlo Newton-Raphson (McCulloch, 1997), and simulated maximum likelihood where simulation is used to estimate the value of the likelihood directly (Geyer and Thompson, 1992; Gelfand and Carlin,1993; Durbin and Koopman, 1997). In addition, a

variety of approximate ML methods have been proposed. These approximate ML approaches are best categorized as estimating equation methods, and include penalized quasi-likelihood, marginal quasi-likelihood (Breslow and Clayton, 1993) and various similar methods that go by a variety of names (see Wolfinger and Lin, 1997 for a partial review and comparison). Marginal quasi-likelihood (MQL) is a computationally less demanding method than ML estimation and applies mainly to longitudinal data. The penalized quasi-likelihood (PQL) approach works reasonably well when the data are approximately normal but can be badly biased for highly non-normal data (Lin and Breslow, 1996).

Although GLMMs have successfully fitted a number of data sets, they fail many data sets that have multiple sources of variation. For instance, Olsen and Schafer (2001) motivated a two-part random effect model for semicontinuous longitudinal data because GLMMs fail to account for an excess of zeros in an otherwise continuous responses in the Adolescent Alcohol Prevention Trial. van Duijn and Bockenholt (1995) showed that a mixed model with gamma distribution did not fit well for a study on spelling errors made by Dutch school-children, which is repeated count data with heterogeneity coming from different classes. At the same time they showed mixture models can fit better. In order to explain this adequately, finite mixture models will be described next.

2.3 Finite Mixture Models

2.3.1 Basic Definition and Interpretation

Suppose $Y_1, \ldots, Y_n$ is a random sample of size n, where Y_i can be 1 dimensional random variable or p dimensional random vector, with probability density function $f(y_i)$ (or mass function in the discrete case) in a sample space R or R^p. We let $\boldsymbol{Y} = (Y_1^T, \ldots, Y_n^T)$ represent the entire sample, where T denotes the vector transpose. We also denote the realization of a random vector by the corresponding lower case letter; that is, we let $\boldsymbol{y} = (y_1, \ldots, y_n)$ represent the observed random sample where y_i is the observed value of the random vector Y_i. If the distribution of Y_i can be represented by a probability density function (p.d.f. hereafter)

of the form

$$f(y_i) = p_1 f_1(y_i) + \ldots + p_g f_g(y_i) \tag{2.3.1}$$

where

$$p_j > 0, \quad j = 1, \ldots, g > 1; \quad p_1 + \ldots + p_g = 1$$

and

$$f_j(.) \geq 0, \quad \int_{\boldsymbol{R}^p} f_j(.)dx = 1, \quad j = 1, \ldots, g$$

Then we say that Y_i has a $g - component\ finite\ mixture\ distribution$ and $f(y_i)$ is a $finite\ mixture\ density\ function$. The quantities $p_1, \ldots, p_g$ are called the mixing probabilities or the mixing proportions and $f_1(y_1), \ldots, f_g(y_g)$ are called the *component densities* of the mixture. It's easy to verify that $f(y_i)$ does define a p.d.f.

Including specific parametric forms, (2.3.1) can be written as

$$f(y_i|\boldsymbol{\delta}) = p_1 f_1(y_i|\boldsymbol{\theta}_1) + \ldots + p_g f_g(y_i|\boldsymbol{\theta}_g) \tag{2.3.2}$$

where $\boldsymbol{\delta} = (p_1, \ldots, p_g, \boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_g^T)^T$.

In our study of mixture models, the number of components are fixed. Of course, in many applications, the value of g should be estimated from the data together with parameter vector $\boldsymbol{\delta}$. In addition, there is no requirement that the component densities in (2.3.1) or (2.3.2) should all come from the same parametric family, but in most applications, this will be the case.

Mixture models provide a powerful tool to model unknown distributional shapes. For example, the method of kernel density estimation, say, with the Gaussian kernel, is essentially mixture modelling of a density, corresponding to a mixture of a large number of normals. By choosing appropriate components, mixture models are able to model quite complex distributions with each components representing a local area of the true distribution. Thus, they can handle situations where a single parametric distribution is unable to provide a satisfactory model for local variation in the observed data (McLachlan and Peel, 2000).

To illustrate, Lindsay (1995) supposes we have a population of animals consisting of two component types, say one is male, the other is female. We measure the characteristics such as length. Suppose length is normally distributed with different means in both components when considered alone. If we sample from the two components without label, the resulting distribution for length is a mixture of two normals.

2.3.2 Mixture of Normals

The earliest studies on finite mixture models were mainly about mixture of normals. That is, $f_j(y_i)$ in (2.3.1) takes the form of a normal density for all j. For example, a frequently used two-component mixture of normals has the form

$$f(y_i;\boldsymbol{\delta}) = p\phi(y_i;\mu_1,\sigma_1) + (1-p)\phi(y_i;\mu_2,\sigma_2),$$

where $\phi(.;\mu,\sigma)$ denotes the $N(\mu,\sigma^2)$ probability density function.

2.3.3 Mixture of GLMs

For a mixture of g component distributions of GLMs in proportions $p_1,\ldots,p_g$, we have the density of the ith response variable Y_i is given by:

$$f(y_i;\boldsymbol{\delta}) = \sum_{j=1}^{g} p_j f_j(y_i;\theta_{ij},\phi_j) \tag{2.3.3}$$

where $f_j(y_i;\theta_{ij},\phi_j)$ has the form (2.1.1) and the link function $\eta_{ij} = \boldsymbol{x}_i^T\boldsymbol{\beta}_j$. In applications, the mixing probability may also be modelled as functions of some vector of covariates $\boldsymbol{w}_i$ associated with the response. The generalized logit transform is commonly used to express the relationship between a multinomial probability vector and a covariate vector. This choice leads to

$$p_j(\boldsymbol{w}_i|\boldsymbol{\gamma}) = \exp(\boldsymbol{w}_i^T\boldsymbol{\gamma}_j)/\left\{1+\sum_{h=1}^{g-1}\exp(\boldsymbol{w}_i^T\boldsymbol{\gamma}_h)\right\} \quad (j=1,\ldots,g) \tag{2.3.4}$$

where $\boldsymbol{\gamma}_g = \mathbf{0}$ and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1^T,\ldots,\boldsymbol{\gamma}_{g-1}^T)^T$ contains the logistic regression coefficients.

The extension of mixtures of normals to mixtures of GLMs greatly enlarges the model class and has successfully fitted many data sets that have overdispersion relative to a standard GLM. For example, Wang, Cockburn and Puterman (1998) dealt with overdispersion in patent data by applying a finite mixture of Poisson regression models; Wang and Puterman (1998) modelled extra-binomial variation by mixed logistic regression models instead of quasi-likelihood or beta-binomial regression and emphasize that this mixture model provides an interpretable alternative to other approaches. ZIP (Lambert, 1992) and ZIB (Hall, 2000) models also fall into this category. They are designed for Poisson or binomial data with extra zeros (zero-inflated data). The advantage to mixtures of GLMs in some applications is not only an improvement in fit to the data but also better understanding of the data-generating mechanism.

Most parameter estimation methods for mixture models can be classified into two categories. One is the likelihood-based approach and the other is the Bayesian approach. ML estimation is greatly facilitated by the EM algorithm, while the Bayesian approach has benefitted from the development of the Gibbs sampler (Gelfand and Smith, 1990). Important papers on the Bayesian analysis of mixture following MCMC methods include Diebolt and Robert (1994) and Escobar and West (1995).

Although finite mixtures of normals and finite mixtures of GLMs can model heterogeneity in the data, they are inappropriate when both heterogeneity and intra-cluster correlation exist. We present a new class of models based on finite mixtures of GLMMs which can handle such a situation and which is an extension of mixtures of GLMs.

2.3.4 Mixture of GLMMs

This new class of models combines the properties of GLMMs and mixtures of GLMs. They are formed by adding random effects to each component in a mixture of GLMs. Similar to mixture of GLMs, suppose we have g components, where the ℓth component appears in the

population with proportion p_ℓ. Then the g-component mixture of GLMMs are of the form

$$f(y_{ij}|\boldsymbol{b}_i) = \sum_{\ell=1}^{g} p_\ell f_\ell(y_{ij}|\boldsymbol{b}_i; \theta_{\ell ij}, \phi_\ell), \tag{2.3.5}$$

where $i, i = 1, \ldots, K$ index the clusters; $j, j = 1, \ldots, t_i$ index the observations in cluster i; $\boldsymbol{b}_i$ is the q dimensional vector of random effects for the ith cluster which is assumed to be independent from one cluster to the next with a multivariate normal distribution $MVN_q(\mathbf{0}, \boldsymbol{D})$; $f_\ell(y_{ij}|\boldsymbol{b}_i; \theta_{\ell ij}, \phi_\ell)$ are the GLMMs described in Section 2.2; and p_ℓ have some regression form such as (2.3.4).

By comparing the equation (2.3.5) with (2.3.3) and Section 2.2, it is clear that if there is no random effect, (2.3.5) becomes (2.3.3), if there is only one component, then (2.3.5.) becomes a GLMM as described in Section 2.2. Due to these connections, some parameter estimation methods and derivations of standard errors for GLMMs and mixture of GLMs can be adapted to analyze mixtures of GLMMs.

In particular, when we assume normal random effects, the marginal likelihood is hard to evaluate, which is the same situation we have for GLMMs. Hence numerical integration methods such as ordinary Gaussian quadrature and adaptive Gaussian quadrature and simulation-based methods such as importance sampling used there can be adapted here. Second, because of the mixture structure, the EM algorithm will facilitate the fitting procedure. So that we may relax the assumption of normality on the random effects, we will consider a nonparametric ML approach. We will restrict our attention to 2-component GLMMs in Chpater 3 and to ZI-mixed effect models in Chapter 4.

2.4 EM Algorithm

2.4.1 Formulation of The EM Algorithm

The Expectation-Maximization (EM) algorithm is a widely used algorithm for maximum likelihood estimation in "incomplete-data" situations. Some incomplete-data situations are obvious such as missing data, censored observations, truncated distribution, etc; some are

not so clear such as a mixture distribution. Hence we need an appropriate formulation of the incompleteness to facilitate the application of the EM algorithm so that we obtain a computational benefit.

Let $\boldsymbol{Y}$ be the random vector corresponding to the observed data $\boldsymbol{y}$. The distribution of $\boldsymbol{Y}$ is $f(\boldsymbol{y};\boldsymbol{\delta})$, where $\boldsymbol{\delta}$ is a vector-valued parameter taking values in Ω. In addition, let $\boldsymbol{y}^c$ be the complete data vector with distribution function $f^c(\boldsymbol{y}^c|\boldsymbol{\delta})$and $\boldsymbol{u}$ be the missing data vector. We have $\boldsymbol{y}^c = (\boldsymbol{y}^T, \boldsymbol{u}^T)^T$. In the EM algorithm context, the observed data vector $\boldsymbol{y}$ is viewed as a function of the complete data vector $\boldsymbol{y}^c$, where the relationship is as follows (McLachlan and Krishnan, 1997):

$$f(\boldsymbol{y};\boldsymbol{\delta}) = \int_{\chi(\boldsymbol{y})} f^c(\boldsymbol{y}^c;\boldsymbol{\delta}) d\boldsymbol{y}^c.$$

Here we suppose two sample spaces χ and y, and we observe the incomplete data vector $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$ in y instead of observing the complete data vector $\boldsymbol{y}^c$ in χ. There is a many to one mapping from χ to y.

Let $L(\boldsymbol{\delta};\boldsymbol{y})$, $\ell(\boldsymbol{\delta};\boldsymbol{y})$ be the observed data likelihood and loglikelihood, respectively, and let $L^c(\boldsymbol{\delta};\boldsymbol{y}^c)$, $\ell^c(\boldsymbol{\delta};\boldsymbol{y}^c)$ be the complete data likelihood and loglikelihood. In the EM algorithm, we do not maximize $\ell(\boldsymbol{\delta};\boldsymbol{y})$ directly to get ML estimates, but iteratively maximize $\ell^c(\boldsymbol{\delta};\boldsymbol{y}^c)$ averaged over all possible values of the missing data $\boldsymbol{u}$. That is, the objective function is defined to be $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) = E[\ell^c(\boldsymbol{\delta};\boldsymbol{y}^c)|\boldsymbol{y},\boldsymbol{\delta}^{(h)}]$, and we iteratively maximize $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$.

In more detail, the $(h+1)$th iteration for obtaining ML estimates via the EM algorithm is as follows:

Step 0: Specify a starting value $\boldsymbol{\delta}^0$, and a convergence criterion.

Step 1(E-Step): Calculate $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ as defined above. This requires evaluation of the conditional expectation of the unobservables given the observables.

Step 2(M-Step): Find the value of $\boldsymbol{\delta}^{(h+1)}$ in Ω which maximizes $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$. That is, find $\boldsymbol{\delta}^{(h+1)}$ such that

$$Q(\boldsymbol{\delta}^{(h+1)}|\boldsymbol{\delta}^{(h)}) \geq Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$$

for all $\boldsymbol{\delta} \in \Omega$.

Steps 1 and 2 are alternated repeatedly until the convergence criterion set in step 0 is obtained.

2.4.2 Theory of The EM Algorithm

(1) Monotonicity: The EM algorithm increases the observed likelihood $L(\boldsymbol{\delta}|\boldsymbol{y})$ at each iteration, that is, $L(\boldsymbol{\delta}^{(h+1)}|\boldsymbol{y}) \geq L(\boldsymbol{\delta}^{(h)}|\boldsymbol{y})$ for $h = 0, 1, \ldots$. (Dempster, Laird and Rubin, 1977).

(2) Let $f^c(\boldsymbol{y}^c|\boldsymbol{y};\boldsymbol{\delta})$ be sufficiently smooth, and suppose a sequence of EM iterates $\boldsymbol{\delta}^{(h)}$ satisfies

$$\frac{\partial Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial \boldsymbol{\delta}}|_{\boldsymbol{\delta}=\boldsymbol{\delta}^{(h+1)}} = 0$$

and $\boldsymbol{\delta}^{(h)}$ converge to some value $\boldsymbol{\delta}^*$. Then it follows that

$$\frac{\partial \ell(\boldsymbol{\delta};\boldsymbol{y})}{\partial \boldsymbol{\delta}}|_{\boldsymbol{\delta}=\boldsymbol{\delta}^*} = 0$$

That is, if the iterates $\boldsymbol{\delta}^{(h)}$ converge, they converge to a stationary point of $L(\boldsymbol{\delta};\boldsymbol{y})$. This implies that when there are multiple stationary points (local or global maximizers, saddle points), the algorithm may not converge to the global maximum.

(3) When there are multiple stationary points (local or global maximizers, saddle points), convergence of the EM sequence $\boldsymbol{\delta}^{(h)}$ to either type depends on the choice of starting value. When $L(\boldsymbol{\delta};\boldsymbol{y})$ is unimodal in Ω with $\boldsymbol{\delta}^*$ being the only stationary point of $L(\boldsymbol{\delta};\boldsymbol{y})$, and $\partial Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}/\partial\boldsymbol{\delta}$ is continuous in $\boldsymbol{\delta}$ and $\boldsymbol{\delta}^{(h)}$, then $\boldsymbol{\delta}^{(h)}$ converges to the unique maximizer $\boldsymbol{\delta}^*$ of $L(\boldsymbol{\delta};\boldsymbol{y})$ (the unique ML estimates), irrespective of its starting point.

(4) Dempster, Laird and Rubin (1977) showed the convergence of the EM algorithm is linear and the rate of convergence depends on the amount of missing information about $\boldsymbol{\delta}$. Hence it's possible that the EM algorithm can be very slow if a large portion of data are missing.

2.4.3 Incomplete Data Structure of Mixture Problem

Obtaining the ML estimates of the parameters in mixture density (2.3.1) becomes easier with the EM algorithm if we regard the component from which each datum comes as the missing

data. Corresponding to the formulation of the mixture density in Section 2.3.1, we define the vector of indicator variables $\boldsymbol{U}_i = (U_{i1}, \ldots, U_{ig})^T$ with realization $\boldsymbol{u}_i = (u_{i1}, \ldots, u_{ig})^T$ by $u_{i\ell} = 1$, if y_i is from component ℓ, otherwise, $u_{i\ell} = 0$. If $(Y_1, \ldots, Y_n)$ are i.i.d, then $(\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n)$ are i.i.d according to a multinomial distribution consisting of one draw from g components with probabilities $p_{i1}, \ldots, p_{ig}$ respectively. We can write

$$\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n \overset{iid}{\sim} Mult_g(1, \boldsymbol{p})$$

Treating $\boldsymbol{y}$ as observed data and $\boldsymbol{u}$ as the missing data, then the complete data $\boldsymbol{y}^c = (\boldsymbol{y}, \boldsymbol{u})$ has loglikelihood:

$$\begin{aligned} \log f(\boldsymbol{y}, \boldsymbol{u}; \boldsymbol{\delta}) &= \log[f(\boldsymbol{y}|\boldsymbol{u}; \boldsymbol{\delta}) f(\boldsymbol{u}; \boldsymbol{\delta})] \\ &= \sum_{\ell=1}^{g} \sum_{i=1}^{n} u_{i\ell} \{\log p_{i\ell} + \log f_\ell(y_i; \boldsymbol{\delta})\} \end{aligned}$$

Following the formulation of EM algorithm in Section 2.4.1 , we define $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ in the mixture problem as:

$$\begin{aligned} Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) &= E[\log f(\boldsymbol{y}, \boldsymbol{u}; \boldsymbol{\delta})|\boldsymbol{y}; \boldsymbol{\delta}^{(h)}] \\ &= \sum_{\ell=1}^{g} \sum_{i=1}^{n} E\{u_{i\ell}\{\log p_{i\ell} + \log f_\ell(y_i; \boldsymbol{\delta})\}|y_i; \boldsymbol{\delta}^{(h)}\} \\ &= \sum_{\ell=1}^{g} \sum_{i=1}^{n} \hat{u}_{i\ell}^{(h)} \log p_{i\ell} + \sum_{\ell=1}^{g} \sum_{i=1}^{n} \hat{u}_{i\ell}^{(h)} \log f_\ell(y_i; \boldsymbol{\delta}), \end{aligned}$$

where

$$\hat{u}_{i\ell}^{(h)} = E[u_{i\ell}|y_i; \boldsymbol{\delta}^{(h)}] = \frac{p_{i\ell} f_\ell(y_i; \boldsymbol{\delta}^{(h)})}{f(y_i; \boldsymbol{\delta}^{(h)})}$$

for $i = 1, \ldots, n$ and $\ell = 1, \ldots, g$

Hence, by using the incomplete structure of mixture problem, the M step of the EM algorithm has been separated into two parts: one involves only the mixing probabilities, the other involves only the component distributions. That means fitting the mixture model can be done by iteratively fitting standard non-mixture models since we can solve $g+1$ estimating equations in M-step that have the form of weighted GLM score equations. Thus, estimation in the mixture problem is greatly simplified with the EM algorithm.

2.5 Monte Carlo EM Algorithm via Importance Sampling for GLMMs

2.5.1 Importance Sampling

Monte Carlo integration (e.g., Tanner, 1993) can be carried out using sets of random variates picked from any arbitrary probability distribution. The choice of distribution obviously makes a difference to the efficiency of the method. For example, Monte Carlo integration carried out using uniform probability distributions gives very poor estimates of high-dimensional integrals and is not a useful method of approximation. In 1953, however, Metropolis et. al. introduced an algorithm that enabled the incorporation of "importance sampling" into Monte Carlo integration. The idea is to choose a distribution that generates values that are in the region where the integrand is large because this region is where the most important contributions are made to the value of the integral.

To be more specific, assume the following integration problem:

$$h(y) = \int f(y|x)g(x)dx$$

If we can not directly sample from $g(x)$, importance sampling can be used. Let $I(x)$ be a density that is easy to sample from and that approximates $g(x)$ (see Tanner, 1993). We draw i.i.d. samples $x_1, \ldots, x_m$ from $I(x)$. Then the above integral is approximated by

$$\hat{h}_m(y) = \int f(y|x)g(x)dx \approx \frac{1}{m}\sum_{i=1}^{m} w_i f(y|x_i),$$

where $w_i = g(x_i)/I(x_i)$. The distribution $I(x)$ is called the importance sampler. The theoretical basis for this estimator is the strong law of large numbers, which says as $m \to \infty$, $\hat{h}_m(y) \to h(y)$, almost surely. For a general discussion of importance sampling, see Hesterberg (1990).

2.5.2 Monte Carlo EM Algorithm for GLMMs

McCulloch (1994) describes a Monte Carlo EM algorithm (MCEM) based on the Gibbs sampler that can handle complicated mixed model structure but is limited to a binary

response with a probit link. A Monte Carlo EM algorithm (MCEM) based on the Metropolis algorithm (Tanner, 1993) is developed by McCulloch (1997) to deal with more general type of GLMMs. Booth and Hobert (1999) proposed two new implementations of the EM algorithm for GLMMs. One of these methods uses importance sampling to generate random variates to construct Monte Carlo approximations at the E-step. This is different from the MCEM described by McCulloch (1994, 1997). In each iteration of McCulloch's algorithm a Markov chain with stationary distribution equal to the exact conditional distribution of $\boldsymbol{b}$ given $\boldsymbol{y}$ is used to approximate the E-step. Booth and Hobert (1999) state that "the use of random samples has significant advantages over dependent samples arising from Markov chains" (see Booth and Hobert, 1999, p.266-267 for detailed comparisons.) Because of difficulties of assessing convergence to stationarity and the error in estimates, Evans and Swartz (1995) comment that "Markov chain methods are recommended only when there is no adequate alternatives." This is reiterated by Jones and Hobert (2001) who state, "before resorting to MCMC, one should try the Monte Carlo methods based on independent samples, for example, rejection sampling or important sampling." (Jones and Hobert, 2001, p.331)

For simplification, let $\boldsymbol{\delta} = (\boldsymbol{\beta}, \phi, \boldsymbol{\theta})$ be the unknown parameter vector for GLMMs in section 2.2. We further assume the unknown random effects $\boldsymbol{b}$ play the role of missing data. Then the complete data vector can be written as $(\boldsymbol{u}, \boldsymbol{b})$ and the EM algorithm computes

$$\begin{aligned} Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) &= E\{\ell^c(\boldsymbol{\delta}; \boldsymbol{y}, \boldsymbol{b})|\boldsymbol{y}; \boldsymbol{\delta}^{(h)}\} \\ &= \sum_{i=1}^{K} \int \ell^c(\boldsymbol{\delta}; \boldsymbol{y}_i, \boldsymbol{b}_i) f(\boldsymbol{b}_i|\boldsymbol{y}_i; \boldsymbol{\delta}^{(h)}) d\boldsymbol{b}_i. \end{aligned} \tag{2.5.1}$$

The integrals in equation (2.5.1) are now with respect to the random effects $\boldsymbol{b}$ only. We consider an importance sampling approach to approximate this integral.

Suppose $I(\boldsymbol{b}_i), i = 1, \ldots, K$ are the importance samplers which have similar distributional shape as $f(\boldsymbol{y}_i|\boldsymbol{b}_i; \boldsymbol{\delta}^{(h)})\phi_q(\boldsymbol{b}_i)$. Here $\boldsymbol{b}_{i1}, \ldots, \boldsymbol{b}_{im}$ are independently drawn from $I(\boldsymbol{b}_i)$. Then equation (2.5.1) can be approximated by

$$Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) \approx \sum_{i=1}^{K} \frac{\sum_{\ell=1}^{m} w_{i\ell}^* \ell^c(\boldsymbol{\delta}; \boldsymbol{y}_i, \boldsymbol{b}_{i\ell})}{\sum_{\ell=1}^{m} w_{i\ell}^*}$$

$$
\begin{aligned}
&= \sum_{i=1}^{K}\sum_{\ell=1}^{m} w_{i\ell}\ell^c(\boldsymbol{\delta};\boldsymbol{y}_i,\boldsymbol{b}_{i\ell}) \\
&= \sum_{i=1}^{K}\sum_{j=1}^{t_i}\sum_{\ell=1}^{m} w_{i\ell}\log f(y_{ij}|\boldsymbol{b}_{i\ell};\boldsymbol{\beta},\phi) + \sum_{i=1}^{K}\sum_{\ell=1}^{m} w_{i\ell}\log\phi_q(\boldsymbol{b}_{i\ell};\boldsymbol{\theta}), \qquad (2.5.2)
\end{aligned}
$$

where $w^*_{i\ell} = f(\boldsymbol{y}_i|\boldsymbol{b}_{i\ell};\boldsymbol{\beta}^{(h)})\phi_q(\boldsymbol{b}_{i\ell};\boldsymbol{\theta}^{(h)})/I(\boldsymbol{b}_{i\ell})$ and $w_{i\ell} = w^*_{i\ell}/\sum_{\ell=1}^{m} w^*_{i\ell}$.

Booth and Hobert (1999) point out that a good choice for the importance sampler $I(\boldsymbol{b}_i)$ for the conditional distribution of $\boldsymbol{b}_i$ given $\boldsymbol{y}_i$ is a multivariate t-density with approximately the same mean and covariance as the true conditional distribution of $\boldsymbol{b}_i$ given $\boldsymbol{y}_i$. That is to say, suppose we can find the mean and covariance of $f(\boldsymbol{b}_i|\boldsymbol{y}_i)$, then the multivariate t-density we use as importance sampler has this mean and variance.

To find the mean and variance of $f(\boldsymbol{b}_i|\boldsymbol{y}_i)$, define

$$h(\boldsymbol{b}_i|\boldsymbol{y}_i;\boldsymbol{\delta}^{(h)}) = L_i^{-1}\exp\{\ell(\boldsymbol{b}_i)\}$$

where L_i is an unknown normalizing constant given by $\int f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\phi^{(h)})\phi_q(\boldsymbol{b}_i;\boldsymbol{\theta}^{(h)})d\boldsymbol{b}_i$ and $\ell(\boldsymbol{b}_i) = \log f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\phi^{(h)}) + \log\phi_q(\boldsymbol{b}_i;\boldsymbol{\theta}^{(h)})$. Let $\ell^{(1)}(\boldsymbol{b}_i)$ denote the vector of first derivatives of $\ell(\boldsymbol{b}_i)$ and $\ell^{(2)}(\boldsymbol{b}_i)$ the second derivative matrix of $\ell(\boldsymbol{b}_i)$. Suppose that $\tilde{\boldsymbol{b}}_i$ is the maximizer of $\ell(\boldsymbol{b}_i)$ satisfying the equation $\ell^{(1)}(\boldsymbol{b}_i) = \boldsymbol{0}$. When the random effects are normal, Booth and Hobert (1998) use a Laplace approximation to show that

$$
\begin{aligned}
E(\boldsymbol{b}_i|\boldsymbol{y}_i;\boldsymbol{\delta}^{(h)}) &\approx \tilde{\boldsymbol{b}}_i \\
\mathrm{var}(\boldsymbol{b}_i|\boldsymbol{y}_i;\boldsymbol{\delta}^{(h)}) &\approx -\ell_i^{(2)}(\tilde{\boldsymbol{b}}_i)^{-1}.
\end{aligned}
$$

This conclusion is very convenient for programming purposes.

2.6 REML Estimation Method

2.6.1 REML Estimation for Classical Linear Models

Suppose we want to estimate the residual variance σ^2 in the classical linear regression model $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{Y} = (Y_1,\ldots,Y_n)$ and $\boldsymbol{X}$ a $(n \times p)$ full rank known design matrix. It is

assumed that all elements in $\boldsymbol{\varepsilon}$ are independently normally distributed with mean zero and variance σ^2. Then the ML estimator of σ^2 is

$$\hat{\sigma}^2 = (\boldsymbol{Y} - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y})'(\boldsymbol{Y} - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y})/n \tag{2.6.1}$$

which is well known to be biased by a factor of n/(n-p). The REML estimator is $\frac{n}{n-p}\hat{\sigma}^2$ which is clearly unbiased.

2.6.2 REML Estimation for Linear Mixed Models

For normal theory linear mixed models, REML is generally regarded as superior to ML (Diggle et al., 1994). For the linear mixed model for clustered data, REML is described by Verbeke and Molenberghs (2000). We briefly summarize their discussion.

In the LMM, we assume

$$\boldsymbol{Y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon}_i \tag{2.6.2}$$

with

$\boldsymbol{\varepsilon}_i \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$, $i = 1, \ldots, K$

$\boldsymbol{b}_i \sim N(\boldsymbol{0}, \boldsymbol{D})$,

$\boldsymbol{\varepsilon}_i$ and $\boldsymbol{b}_i$ are independent,

$\boldsymbol{\varepsilon}_i$ are independent,

$\boldsymbol{b}_i$ are independent. Hence the marginal model is

$$\boldsymbol{Y}_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta}, \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i' + \boldsymbol{\Sigma}_i),$$

where $\boldsymbol{D}$ is the variance-covariance matrix for random effect $\boldsymbol{b}_i$ and $\boldsymbol{\Sigma}_i$ is the intra-cluster variance-covariance matrix for $\boldsymbol{\varepsilon}_i$. Let $\boldsymbol{\theta}$ be the vector of variance-component parameters, which consists of all unknown parameters in $\boldsymbol{D}$ and $\boldsymbol{\Sigma}_i$, where $i = 1, \ldots, K$; further let $\boldsymbol{V}_i = \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i' + \boldsymbol{\Sigma}_i$ and denote the total parameter vector to be $\boldsymbol{\delta} = (\boldsymbol{\beta}^T, \boldsymbol{\theta}^T)^T$. Then the loglikelihood function is

$$\ell_{ML}(\boldsymbol{\delta}) = \sum_{i=1}^{K}\left\{-\frac{n_i}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{V}_i(\boldsymbol{\theta})| - \frac{1}{2}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T\boldsymbol{V}_i^{-1}(\boldsymbol{\theta})(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})\right\}. \tag{2.6.3}$$

The ML estimators are given by

$$\hat{\boldsymbol{\beta}}(\hat{\boldsymbol{\theta}}) = \left(\sum_{i=1}^{K} \boldsymbol{X}_i' \boldsymbol{V}_i^{-1}(\hat{\boldsymbol{\theta}}) \boldsymbol{X}_i\right)^{-1} \sum_{i=1}^{K} \boldsymbol{X}_i' \boldsymbol{V}_i^{-1}(\hat{\boldsymbol{\theta}}) \boldsymbol{Y}_i \tag{2.6.4}$$

and $\hat{\theta}_s$ the solution of

$$\sum_{i=1}^{K} tr\left\{\boldsymbol{V}_i^{-1}\left[(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta})^T(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}) - \boldsymbol{V}_i\right]\boldsymbol{V}_i^{-1}\frac{\partial \boldsymbol{V}_i}{\partial \theta_s}\right\} \quad s = 1, \ldots, dim(\boldsymbol{\theta})$$

(cf. Jennrich and Schluchter, 1986).

However, it is well known that $\hat{\boldsymbol{\theta}}$ is, in general, a biased estimator. Therefore, a bias corrected estimator $\tilde{\boldsymbol{\theta}}$ is often used based on maximizing a "concentrated" or restricted likelihood. The so-called restricted ML estimator of $\boldsymbol{\theta}$ maximizes the loglikelihood function of a set of error contrasts $\boldsymbol{U} = \boldsymbol{A}'\boldsymbol{Y}$ where $\boldsymbol{A}$ is any $n \times (n-p)$ (n is total sample size) matrix with $n - p$ linearly independent columns orthogonal to the columns of the $\boldsymbol{X}$ matrix. The distribution of $\boldsymbol{U}$ has mean zero vector and covariance matrix $\boldsymbol{A}'\boldsymbol{V}(\boldsymbol{\theta})\boldsymbol{A}$. Harville (1974) has showed that this objective function and the resulting REML estimator $\tilde{\boldsymbol{\theta}}$ does not depend on the particular choice of error contrasts (i.e., the choice of $\boldsymbol{A}$). The objective function that maximized to obtain the REML estimator of $\boldsymbol{\theta}$ is

$$\ell_{REML}(\boldsymbol{\theta}) = -\frac{1}{2}\log\left|\sum_{i=1}^{K} \boldsymbol{X}_i' \boldsymbol{V}_i^{-1}(\boldsymbol{\theta}) \boldsymbol{X}_i\right| + p\ell_{ML}(\boldsymbol{\theta}), \tag{2.6.5}$$

where $p\ell_{ML}(\boldsymbol{\theta})$ is the profile loglikelihood function given by (2.6.3), but where $\boldsymbol{\beta}$ has been replaced by (2.6.4).

Notice that $\ell_{REML}(\boldsymbol{\theta})$ differs from $p\ell_{ML}(\boldsymbol{\theta})$ only by the additional term $-\frac{1}{2}\log\left|\sum_{i=1}^{K} \boldsymbol{X}_i' \boldsymbol{V}_i^{-1}(\boldsymbol{\theta}) \boldsymbol{X}_i\right|$. This term serves as an adjustment or penalty for the estimation of $\boldsymbol{\beta}$. Hence REML estimation is sometimes called a penalized likelihood method of estimation.

Note for the classical linear regression model, the fixed effect parameter estimator $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$ is independent of the residual variance σ^2 and hence does not change if REML estimates are used for variance components instead of ML estimates (2.6.1). This is not true for linear mixed models because of the dependence of $\hat{\boldsymbol{\beta}}$ on $\boldsymbol{\theta}$ (see formula 2.6.4).

That is to say, in the linear mixed-effects model context, although REML estimation is constructed for the variance components in the model, the corresponding estimator for fixed effects is no longer the same as that from ML estimation.

2.6.3 Approximate REML Estimation for GLMMs

The REML method has been extended to GLMMs by several authors (e.g. McGilchrist, 1994; Breslow and Clayton, 1993) and most of them apply REML to a linearized version of the nonlinear models. Hence we give the name approximate REML to these methods. We briefly summarize these approaches.

The PQL method presented by Breslow and Clayton (1993) is based on the quasi-likelihood

$$\prod_{i=1}^{K}\left\{|\boldsymbol{D}|^{-1/2}\int\exp\left[-\frac{1}{2\phi}\sum_{j=1}^{n_i}d_{ij}(y_{ij};\mu_{ij}(\boldsymbol{b}_i))-\frac{1}{2}\boldsymbol{b}_i^T\boldsymbol{D}^{-1}\boldsymbol{b}_i\right]d\boldsymbol{b}_i\right\},$$

where $d_{ij}(y_{ij};\mu_{ij}(\boldsymbol{b}_i))=-2\int_y^{\mu}\frac{y-u}{a_{ij}v(u)}du$ denotes the deviance and $\mu_{ij}(\boldsymbol{b}_i)=E(y_{ij}|\boldsymbol{b}_i)$. They then use Laplace's method to approximate the integral which leads to

$$\sum_{i=1}^{K}\left\{-\frac{1}{2}\log|\boldsymbol{I}+\boldsymbol{Z}_i^T\boldsymbol{W}_i\boldsymbol{Z}_i|-\frac{1}{2\phi}\sum_{j=1}^{n_i}d_{ij}(y_{ij};\mu_{ij}(\boldsymbol{b}_i))-\frac{1}{2}\boldsymbol{b}_i^T\boldsymbol{D}^{-1}\boldsymbol{b}_i\right\}, \tag{2.6.6}$$

where $\boldsymbol{W}_i$ is the $n_i\times n_i$ diagonal matrix with diagonal terms $w_{ij}=\{\phi a_{ij}v(\mu_{ij}(\boldsymbol{b}_i))[g'(\mu_{ij}(\boldsymbol{b}_i))]^2\}^{-1}$, which can be thought as the GLM iterated weights. The first term in (2.6.6) is omitted by assuming those weights vary slowly as a function of mean. What is left to maximize is Green's (1987) PQL

$$-\frac{1}{2\phi}\sum_{j=1}^{n_i}d_{ij}(y_{ij};\mu_{ij}(\boldsymbol{b}_i))-\frac{1}{2}\boldsymbol{b}_i^T\boldsymbol{D}^{-1}\boldsymbol{b}_i. \tag{2.6.7}$$

To solve for $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}_i}$ for given $\boldsymbol{\theta}$, the Fisher scoring algorithm is employed, which leads to iteratively solving the system

$$\begin{pmatrix}\boldsymbol{X}_i^T\boldsymbol{W}_i\boldsymbol{X}_i & \boldsymbol{X}_i^T\boldsymbol{W}_i\boldsymbol{Z}_i\boldsymbol{D}\\ \boldsymbol{Z}_i^T\boldsymbol{W}_i\boldsymbol{X}_i & \boldsymbol{I}+\boldsymbol{Z}_i^T\boldsymbol{W}_i\boldsymbol{Z}_i\boldsymbol{D}\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{v}\end{pmatrix}=\begin{pmatrix}\boldsymbol{X}_i^T\boldsymbol{W}_i\boldsymbol{Y}_i\\ \boldsymbol{Z}_i^T\boldsymbol{W}_i\boldsymbol{Y}_i\end{pmatrix}, \tag{2.6.8}$$

where $\boldsymbol{b}_i = \boldsymbol{D}\boldsymbol{v}$. The equation (2.6.8) is Henderson's mixed model equation from the normal theory model (2.6.4) but with $\boldsymbol{b}_i \sim N(\boldsymbol{0}, \boldsymbol{D})$, $\boldsymbol{\varepsilon}_i \sim N(\boldsymbol{0}, \boldsymbol{W}^{-1})$ and with $\boldsymbol{Y}_i$ the working dependent variable having element $Y_{ij} = \eta_{ij}(\boldsymbol{b}_i) + (y_{ij} - \mu_{ij}(\boldsymbol{b}_i))g'(\mu_{ij}(\boldsymbol{b}_i))$. Hence this is equivalent to best linear unbiased estimation of $\boldsymbol{\beta}$ and best linear unbiased prediction of $\boldsymbol{b}_i$ based on the linearized mixed models. Denoting the estimates of $\boldsymbol{\beta}$, $\boldsymbol{b}_i$ by $\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})$ and $\hat{\boldsymbol{b}}_i(\boldsymbol{\theta})$, and plugging back into the approximate quasi-loglikelihood (2.6.6) produces an approximate profile quasi-loglikelihood function for the variance parameter $\boldsymbol{\theta}$. Further replacing $d_{ij}(y_{ij}; \mu_{ij}(\boldsymbol{b}_i))$ by the Pearson chi-squared statistic $\sum(y_{ij} - \mu_{ij}(\boldsymbol{b}_i))^2/a_{ij}v(\mu_{ij}(\boldsymbol{b}_i))$, the approximate profile quasi-loglikelihood function is

$$q\ell(\boldsymbol{\theta}) \approx \sum_{i=1}^{K} \left\{ -\frac{1}{2}\log|\boldsymbol{V}_i(\boldsymbol{\theta})| - \frac{1}{2}(\boldsymbol{Y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}))^T \boldsymbol{V}_i^{-1}(\boldsymbol{\theta})(\boldsymbol{Y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})) \right\} \tag{2.6.9}$$

where $\boldsymbol{V}_i(\boldsymbol{\theta}) = \boldsymbol{W}_i^{-1} + \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i^T$. Note (2.6.9) is the kernel of (2.6.3). Hence the REML estimate of $\boldsymbol{\theta}$ can be obtained from the corresponding version of (2.6.5), which is

$$q\ell(\boldsymbol{\theta})_{REML} = q\ell(\boldsymbol{\theta}) - \frac{1}{2}\log\left|\sum_{i=1}^{K} \boldsymbol{X}_i'\boldsymbol{V}_i^{-1}(\boldsymbol{\theta})\boldsymbol{X}_i\right|. \tag{2.6.10}$$

As described by Breslow and Clayton, "Implementation involves repeated calls to normal theory procedure for REML estimation in variance components problems."

Another approximate REML estimation in GLMM setting is proposed by McGilchrist (1994), and applied by Yau and Lee (2001) for ZI-Poisson data. For normal error model, the BLUP procedure to obtain estimates of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and a predictor of $\boldsymbol{b}$ consists of maximizing the joint loglikelihood of $\boldsymbol{y}$ and $\boldsymbol{b}$, which can be expressed as

$$\ell(\boldsymbol{y}, \boldsymbol{b}) = \ell(\boldsymbol{y}|\boldsymbol{b}) + \ell(\boldsymbol{b}), \qquad or \qquad \ell = \ell_1 + \ell_2, \tag{2.6.11}$$

where $\ell(\boldsymbol{y}|\boldsymbol{b})$ is the (normal) log density of $\boldsymbol{y}$ conditional on the random effects $\boldsymbol{b}$. This term involves both $\boldsymbol{\beta}$ and $\boldsymbol{b}$, while the second term $\ell(\boldsymbol{b})$ involves $\boldsymbol{b}$ only. Harville (1977) and other researchers showed how to develop ML and REML estimators of variance components from BLUP estimators under normal error model. The formulas are summarized as (3.1) and (3.2) in McGilchrist (1994). In the GLMM context, the normal error assumption no longer

holds. That is, $\ell(\boldsymbol{y}|\boldsymbol{b})$ or ℓ_1 in (2.6.11) has some not-necessarily normal form. In this case McGilchrist states that "provided that $[\ell(\boldsymbol{y}, \boldsymbol{b})]$ is approximately quadratic in $\boldsymbol{\beta}$ and $[\boldsymbol{b}]$, then we may consider the BLUP estimation as having been derived from the very approximate asymptotic [normal] distribution of $\hat{\boldsymbol{\beta}}$ and $[\hat{\boldsymbol{b}}]$ "with mean equal to $\boldsymbol{\beta}$ and $\boldsymbol{b}$ and variance matrix given by the information matrix for $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}$, which is

$$\begin{pmatrix} \boldsymbol{X}^T \\ \boldsymbol{Z}^T \end{pmatrix} \boldsymbol{B}(\boldsymbol{X}, \boldsymbol{Z}),$$

where $\boldsymbol{B} = -E(\partial^2 \ell_1 / \partial\eta\partial\eta^T)$ ". That means, the objective function (2.6.11) changes to

$$\ell^* = \ell_1^* + \ell_2$$

where

$$\begin{aligned} \ell_1^* &= constant - \frac{1}{2} \begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{b}} - \boldsymbol{b} \end{pmatrix} \begin{pmatrix} \boldsymbol{X}^T \\ \boldsymbol{Z}^T \end{pmatrix} \boldsymbol{B}(\boldsymbol{X}, \boldsymbol{Z}) \begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{b}} - \boldsymbol{b} \end{pmatrix} \\ &= constant - \frac{1}{2} (\boldsymbol{y}^* - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b})^T \boldsymbol{B} (\boldsymbol{y}^* - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b}) \end{aligned}$$

and the working depend variable becomes $\boldsymbol{y}^* = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{Z}\hat{\boldsymbol{b}}$ (see McGilchrist, 1994 for the details). Hence the nonlinear problem changes to a normal theory linear model BLUP estimation problem for $\ell^* = \ell_1^* + \ell_2$. Corresponding REML estimation of variance component can be derived by formula (3.2) in McGilchrist (1994) as under normal error model.

The relationship between this method and Breslow and Clayton's method has been stated in McGilchrist (1994): "the approach is similar in principle to penalized likelihood approaches and in basic aims has elements in common with Breslow and Clayton [(1993)]."

In summary, the REML methods employed by Breslow and Clayton and McGilchrist are not nuisance parameter elimination technique. Therefore, the natural questions to be asked are, How accurate are Breslow and Clayton's approximations which lead to their REML approach? (Even Breslow and Clayton stated in their paper that "Our "derivation" of the penalized quasi-likelilhood [2.6.7] and modified profile quasi-likelihood [(2.6.10)] involved several ad hoc adjustments and approximations for which no formal justification was given".)

How much information is lost? Is it possible to work on the loglikelihood from the nonlinear model directly and still apply a REML-like method to get better estimate of variance component? Liao and Lipsitz (2002) proposed a REML-type estimator which we think follows the original idea of REML method (eliminating the effect of estimating fixed effect parameters) and is based on the loglikelihood from nonlinear model directly. We will extended this method to ZI-mixed effect models in Chapter 4.

2.7 Some Useful Tools for Computation

2.7.1 Unconstrained Cholesky Parameterization

Let Σ denote a symmetric positive definite $n \times n$ variance-covariance matrix corresponding to a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$. Since it is symmetric, only $n(n+1)/2$ parameters are needed to represent it. Since Σ is positive definite, it can be factored as

$$\Sigma = \boldsymbol{L}^T \boldsymbol{L}, \tag{2.7.1}$$

where $\boldsymbol{L}$ is an $n \times n$ upper triangular matrix. Setting $\boldsymbol{\theta}$ to be the upper triangular elements of $\boldsymbol{L}$ gives the unconstrained Cholesky parameterization (see Pinheiro and Bates, 1996 for more details).

For example, a symmetric positive definite matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 5 & 5 \\ 1 & 5 & 14 \end{pmatrix}$$

can be factored as

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix}$$

By convention, we stack the elements of the upper triangular part of $\boldsymbol{L}$ columnwise to get $\boldsymbol{\theta} = (1, 1, 2, 1, 2, 3)^T$.

Lindstrom and Bates (1988) reported this parameterization dramatically improved the convergence properties of the optimization algorithm for fitting nonlinear mixed models when compared to a constrained estimation approach.

2.7.2 Newton-Raphson Algorithm

Suppose we want to solve $f(x) = 0$, we need to find x^* satisfying $f(x^*) = 0$. We require $|f'(x^*)| > 0$. By Taylor expansion of $f(x^*)$, we get

$$0 = f(x^*) = f(x) + f'(x)(x^* - x) + \dots$$

This implies that

$$x^* \approx x - \frac{f(x)}{f'(x)}$$

for x close to x^*. This suggests the iteration

$$x^{(m+1)} = x^{(m)} - \frac{f(x^{(m)})}{f'(x^{(m)})}.$$

In multiple dimensions, the iteration becomes

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{x}^{(m)} - \left[\frac{\partial}{\partial \boldsymbol{x}^{(m)T}} \boldsymbol{f}(\boldsymbol{x}^{(m)})\right]^{-1} \boldsymbol{f}(\boldsymbol{x}^{(m)}).$$

2.7.3 Finite Difference Approximations of Derivatives

To approximate a first-order derivatives, by forward difference approximations (Press et al, 1992; Dennis and Schnabel, 1983), we use

$$\frac{\partial f}{\partial \theta_i} \approx \frac{f(\theta + h_i e_i) - f(\theta)}{h_i},$$

where $h_j = \epsilon^{\frac{1}{2}}(1 + |\theta_j|)$, ϵ is the machine precision and e_i is an indicator vector with 1 in the ith position and 0's elsewhere. Similarly, to approximate a second-order derivatives, we use

$$\frac{\partial^2 f}{\partial \theta_i \theta_j} \approx \frac{f(\theta + h_i e_i + h_j e_j) - f(\theta + h_i e_i) - f(\theta + h_j e_j) + f(\theta)}{h_i h_j},$$

where $h_j = \epsilon^{\frac{1}{3}}(1 + |\theta_j|)$.

If we use central difference approximations of derivatives, for first-order derivatives, it can be expressed as

$$\frac{\partial f}{\partial \theta_i} \approx \frac{f(\theta + h_i e_i) - f(\theta - h_i e_i)}{2h_i},$$

while for second-order derivatives, it can be expressed as

$$\frac{\partial^2 f}{\partial \theta_i \theta_j} \approx \frac{f(\theta + h_i e_i + h_j e_j) - f(\theta + h_i e_i - h_j e_j) - f(\theta - h_i e_i + h_j e_j) + f(\theta - h_i e_i - h_j e_j)}{4h_i h_j}$$

and

$$\frac{\partial^2 f}{\partial \theta_i^2} \approx \frac{-f(\theta + 2h_i e_i) + 16f(\theta + h_i e_i) - 30f(\theta) + 16f(\theta - h_i e_i) - f(\theta - 2h_i e_i)}{12h_i^2},$$

where $h_j = \epsilon^{\frac{1}{3}}(1 + |\theta_j|)$.

2.8 References

[1] Agresti, A. (1990). *Categorical Data Analysis.* New York: Wiley.

[2] Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing* **6**, 251–262.

[3] Booth, J.G. and Hobert, J.H. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society B* **62**, 265–285.

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

[5] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B* **39**, 1–38.

[6] Dennis, J.E. and Schnabel, R.B. (1983). *Numerical Methods for Uncon-strained Optimization and Nonlinear Equations.* Prentice Hall: 378P.

[7] Diebolt, J. and Robert, C.P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society B* **56**, 363–375.

[8] Dietz, E. and Böhning, D. (1995). Statistical inference based on a general model of unobserved heterogeneity. In *Statistical Modelling,* B.J. Francis, R. Hatzinger, G.U.H. Seeber, and G. Steckel-Berger (eds), 75–82. New York: Springer-Verlag.

[9] Diggle, P.J., Liang, K.-Y. and Zeger, S.L. (1994). *Analysis of Longitudinal Data.* Clarendon Press, Oxford.

[10] Durbin, J. and Koopman, S.J. (1997), Monte Carlo maximum likelihood estimation for non-Gaussian state space models. *Biometrika* **84**, 669–684.

[11] Escobar, M.D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* **90,** 577–588.

[12] Evans, M. and Swartz, T.B. (1995). Methods for approximating integrals in Statistics with special emphasis on Bayesian integration. *Statistical Science* **10(3),** 254–272.

[13] Fahrmeir, L., Tutz, G. (1994, 2001). *Multivariate Statistical Modelling Based on Generalized Linear Models.* Springer Series in Statistics.

[14] Gelfand, A.E. and Carlin, B. (1993). Maximum likelihood estimation for constrained or missing data models. *Canadian Journal of Statistics* **21,** 303-311.

[15] Gelfand, A.E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85,** 398–409.

[16] Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data, (with discussion). *Journal of Royal Statistical Society B* **54,** 657–699.

[17] Hall, D.B. (2000). Zero-inflated Poisson and binomial regression with random effects: a case study. *Biometrics* **56**, 1030–1039.

[18] Harville, D.A. (1974). Bayesian inference for variance components using only error contrasts. *Biometrika* **61,** 383-385.

[19] Harville, D.A. (1977). The use of linear-model methodology to rate high school or college football teams. *Journal of the American Statistical Association* **72,** 278–289.

[20] Hesterberg, T. C. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics* **37(2),** 185-194.

[21] Jennrich, R. I. and Schluchter, M. D. (1986). Unbalanced repeated measures models with structural covariance matrices. *Biometrics* **42(4),** 805-820.

[22] Jones, G.L. and Hobert, J.P. (2001). Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*.

[23] Lambert, D.(1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* **34,** 1-14.

[24] Liao, J.G. and Lipsitz, S.R. (2002). A type of restricted maximum likelihood estimator of variance components in generalized linear mixed models. *Biometrika* **89,** 401–409.

[25] Lin, X. and Breslow, N.E. (1996). Bias correction in generalized linear mixed models with multiple components of dispersion. *Journal of the American Statistical Association* **91,** 1007–1016.

[26] Lindsay, B.G. (1995). Mixture models: theory, geometry and applications. *The Institute of Mathematical Statistics and the American Statistical Association*.

[27] Lindstrom, M.J. and Bates, D.M.(1988). Newton-Raphson and EM algorithms for linear mixed effects models for repeated measures data. *Journal of the American Statistical Association* **83,** 1014–1022.

[28] McCulloch, C.E. (1994). Maximum likelihood variance components estimation for binary data. *Journal of the American Statistical Association* **89**, 330-335.

[29] McCulloch, C.E. (1997). Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Association* **92**, 162-170.

[30] McGilchrist, C.A. (1994). Estimation in generalized mixed models. *Journal of the Royal Statistical Society, B* **56(1)**, 61–69.

[31] McLachlan, G.J. and Krishnan, T. (1997). *The EM Algorithm and Extensions.* New York: Wiley.

[32] McLachlan, G.J. and Peel, D. (2000). *Finite Mixture Models.* New York: Wiley.

[33] Nelder, J.A. and Wedderburn, R.W.M. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society, Series A* **135,** 370–384.

[34] Olsen,M.K. and Schafer, J.L.. (2001). A two-part random-effects model for semicontinuous longitudinal data. *Journal of the American Statistical Association* **96,** 730-745.

Pinheiro, J.C. and Bates, D. M. (1996). Unconstrained parameterizations for variance-covariance matrices. *Statistics and Computing* **6,** 289–296.

[35] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). *Numerical recipes in C–the art of scientific computing.* Cambridage University Press.

[36] Stiratelli, R., Laird, N. M. and Ware, J. H. (1984). Random-effects model for several observations with binary response. *Biometrics* **40**, 961–971.

[37] Tanner, M.A. (1993). *Tools for Statistical Inference.* Springer Velag.

[38] van DUIJN, M.A.J. and Bockenholt, U. (1995). Mixture models for the analysis of repeated count data. *Applied Statistics*, **44,** 473–485.

[39] Verbeke, G. and Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data.* Springer: New York.

[40] Wang, P.,Cockburn,I.M., and Puterman, M.L. (1998). Analysis of patent data— A mixed-Poisson-regression-model approach. *Journal of Business and Economic Statistics* **16,** 27–41.

[41] Wang, P. and Puterman, M.L. (1998). Mixed logistic regression models. *Journal of Agricultural, Biological, and Environmental Statistics* **3,** 175–200.

[42] Wolfinger, R.D. and Lin,X. (1997). Two Taylor-series approximation methods for non-linear mixed models. *Computional Statistics and Data Analysis* **25,** 465–490.

[43] Yau, K.W. and Lee, A.H. (2001). Zero-inflated poisson regression with random effects to evaluate an occupational injury prevention programme. *Statistics in medicine* **20,** 2907–2920.

[44] Zeger, S.L. and Liang, K.-Y. and Albert, P.S. (1988). Models for longitudinal data: a generalized estimating equation approach. *Biometrics* **44,** 1049–60.

Chapter 3

Mixtures of Generalized Linear Mixed-effects Models for Cluster-Correlated Data

3.1 Introduction

Finite mixture models with regression structure have a long and extensive literature and have been used commonly in fields such as epidemiology, medicine, genetics, economics, engineering, marketing and in the physical and social sciences. Much of this work has focused on mixtures of normal distributions; see, for example, Everitt and Hand (1981), McLachlan and Basford (1988), Hasselblad (1966) and Aitkin and Wilson (1980). It has only been relatively recently that regression models based on mixtures of non-normals have been given much attention. Some of this work has included regression structure in the linear predictor only (Hasselblad, 1969; Jansen, 1993; Dietz and Bohning, 1997), while other authors have considered covariates in both the linear predictor and the mixing probability (Thompson, Smith and Boyle, 1998; Wang and Puterman, 1998; Wang, Cockburn and Puterman, 1998; McLachlan and Peel, 2000). Of course, models without covariates occur as a special case, and such models have been considered by Titterington, et al. (1985), Leroux and Puterman (1992), and Lindsay (1995). A special case of the two-component mixture occurs when one component is a degenerate distribution with point mass of one at zero. Such models are known as zero-inflated regression models and include zero-inflated Poisson (ZIP; Lambert, 1992), zero-inflated negative binomial, zero-inflated binomial (ZIB; Hall, 2000) and others (see Ridout, et al., 1998 for a review). Finite mixture models are known as mixture-of-experts (ME) models in the neural network field, where they have an extensive literature that dates back at least to Jacobs et al. (1991); see also Jiang and Tanner (1999).

Recently, many researchers have incorporated random effects into a wide variety of regression models to account for correlated responses and multiple sources of variance. In a mixture model context, van Duijn and Bockenholt (1995) presented a latent class-Poisson model for analyzing overdispersed repeated count data. Hall (2000) added random effects to ZIP and ZIB models (see also Yau and Lee, 2001). Zero-inflated regression models for continuous data have also been considered by Olsen and Schafer (2001) and Berk and Lachenbruch (2002). In these papers, random effects are included to account for within-cluster correlation. Rosen, Jiang and Tanner (2000) extend ME models to the clustered data case by incorporating generalized estimating equations in the fitting algorithm. In this paper, we formulate a class of regression models based on a two-component mixture of generalized linear mixed effect models (two-component GLMMs). This class can be viewed as an extension of finite mixtures of generalized linear models (Jansen, 1993) obtained by adding random effects to each component. Generalized linear models (GLMs), finite mixtures of GLMs, ZIP, ZIB and many other models are special cases of this broad class.

The difficulty of parameter estimation in mixture models is well known. A major advance came with the publication of the seminal paper of Dempster, Laird, and Rubin (1977) on the EM algorithm. With the EM algorithm, finite mixture models can be fit by iteratively fitting weighted versions of the component models. So, for example, a K-component finite mixture of GLMs can be fit via maximum likelihood by fitting K weighted GLMs, updating the weights, and iterating to convergence. Mixture models with random effects pose an additional challenge to maximum likelihood (ML) estimation since the marginal likelihood involves an integral that cannot be evaluated in closed form. This challenge is similar to that found with ordinary (non-mixture) GLMMs and other nonlinear mixed models.

In the estimation of GLMMs, several approaches have been considered to evaluate the loglikelihood. Various authors have considered analytic approximations such as the Laplace approximation (e.g., Wolfinger, 1993; Steele, 1996) to motivate fitting algorithms or estimating equations. (e.g., Breslow and Clayton, 1993; Wolfinger and O'connell, 1993; Wolfinger

and Lin, 1997; Lindstrom and Bates, 1990). In addition, some authors have cast GLMMs in a Bayesian framework and utilized Bayesian computational techniques such as importance sampling (Ii and Raghunathan, 1991) and Gibbs sampling (Besag, York, and Mollie, 1991; Zeger and Karim, 1991). A third approach, which we pursue in this paper, is to evaluate the integral numerically via ordinary Gaussian (Gaussian-Hermite) quadrature, or adaptive Gaussian quadrature (Pinheiro and Bates, 1995; Liu and Pierce, 1994). We also consider a nonparametric quadrature approach that has been used in a GLMM context by Hinde and Wood (1987) and Aitkin (1999), where we drop the assumption of normality on the random effects.

The paper is organized as follows: we formulate the two-component mixture of GLMMs in section 3.2. In section 3.3, we outline the EM algorithm and consider various methods of handling the required integration with respect to the missing data. Section 3.4 discusses the computation of appropriate standard errors for parameter estimators when using the EM algorithm. The model class and estimation methods are illustrated with two real data examples in section 3.5. Finally, we give a brief discussion in section 3.6.

3.2 Two-Component Mixture of GLMMs

Suppose we observe an N-dimensional response vector $\boldsymbol{y}$ containing data from K independent clusters, so that $\boldsymbol{y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_K^T)^T$, where $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{it_i})^T$. We assume that, conditional on a q-dimensional vector of random effects $\boldsymbol{b}_i$, the random variable Y_{ij} associated with observation y_{ij} follows a two-component mixture distribution

$$Y_{ij}|\boldsymbol{b}_i \sim \begin{cases} F_1(y_{ij}|\boldsymbol{b}_i; \zeta_{1ij}, \sigma_1), & \text{with probability } p_{ij}; \\ F_2(y_{ij}|\boldsymbol{b}_i; \zeta_{2ij}, \sigma_2), & \text{with probability } 1 - p_{ij}. \end{cases}$$

Here, F_1 and F_2 are assumed to be exponential dispersion family distributions, with densities

$$f_1(y_{ij}|\boldsymbol{b}_i; \zeta_{1ij}, \sigma_1) = h_1(y_{ij}, \sigma_1) \exp[\{\zeta_{1ij} y_{ij} - \kappa_1(\zeta_{1ij})\} w_{ij}/\sigma_1],$$

$$f_2(y_{ij}|\boldsymbol{b}_i; \zeta_{2ij}, \sigma_2) = h_2(y_{ij}, \sigma_2) \exp[\{\zeta_{2ij} y_{ij} - \kappa_2(\zeta_{2ij})\} w_{ij}/\sigma_2],$$

respectively, where the w_{ij}'s are known constants (e.g., binomial denominators). The functions κ_1 and κ_2 are cumulant generating functions, so F_1 and F_2 have (conditional) means $\mu_{1ij} = \kappa_1'(\zeta_{1ij})$ and $\mu_{2ij} = \kappa_2'(\zeta_{2ij})$ and (conditional) variances $v_1(\mu_{1ij})\sigma_1/w_{ij}$ and $v_2(\mu_{2ij})\sigma_2/w_{ij}$ where $v_\ell(\mu) = \kappa_\ell''(\mu)$, $\ell = 1, 2$, are (conditional) variance functions.

We assume the canonical parameters $\boldsymbol{\zeta}_{1i} = (\zeta_{1i1}, \ldots, \zeta_{1it_i})^T$ and $\boldsymbol{\zeta}_{2i} = (\zeta_{2i1}, \ldots, \zeta_{2it_i})^T$ are related to covariates and cluster-specific random effects through GLM-type specifications. That is, for canonical link functions we have

$$\boldsymbol{\zeta}_{1i}(\boldsymbol{\mu}_{1i}) = \boldsymbol{\eta}_{1i} = \boldsymbol{X}_i\boldsymbol{\alpha} + \boldsymbol{U}_i\boldsymbol{D}_1^{T/2}\boldsymbol{b}_i, \quad \text{or} \quad \boldsymbol{\mu}_{1i} = \boldsymbol{\zeta}_{1i}^{-1}(\boldsymbol{\eta}_{1i}),$$

$$\boldsymbol{\zeta}_{2i}(\boldsymbol{\mu}_{2i}) = \boldsymbol{\eta}_{2i} = \boldsymbol{Z}_i\boldsymbol{\beta} + \boldsymbol{U}_i\boldsymbol{D}_2^{T/2}\boldsymbol{b}_i, \quad \text{or} \quad \boldsymbol{\mu}_{2i} = \boldsymbol{\zeta}_{2i}^{-1}(\boldsymbol{\eta}_{2i}).$$

Here, $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ are $t_i \times r_1$ and $t_i \times r_2$ design matrices, respectively, for fixed effects parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$; $\boldsymbol{U}_i$ is a $t_i \times q$ design matrix for the random effects $\boldsymbol{b}_i$; $\boldsymbol{b}_1 \ldots, \boldsymbol{b}_K$ are assumed to be independent, each with mean $\boldsymbol{0}$ and variance $\boldsymbol{I}_q$; and $\boldsymbol{D}_\ell^{T/2}$, $\ell = 1, 2$ are lower triangular scale matrices for the variance and covariance components associated with $\boldsymbol{b}_i$. That is, $\text{var}(\boldsymbol{\eta}_\ell) = \boldsymbol{U}_i\boldsymbol{D}_\ell\boldsymbol{U}_i^T$, $\ell = 1, 2$, where $\boldsymbol{D}_\ell$ contains variance components along the diagonal, and covariance components on the off-diagonal. We assume that $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ have the same structure, but are parameterized by vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ that have the same dimension but are allowed to differ. We adopt an unconstrained Cholesky parameterization (Pinheiro and Bates, 1996) where the elements of $\boldsymbol{\theta}_\ell$ are the nonzero entries in the upper triangular Cholesky factor $\boldsymbol{D}_\ell^{1/2}$, $\ell = 1, 2$. That is, $\boldsymbol{\theta}_\ell = \text{vech}(\boldsymbol{D}_\ell^{1/2})$, $\ell = 1, 2$, where vech stacks the columns of its matrix argument including only those elements on and above the diagonal. (Note that our definition of vech differs from the usual usage in which the elements on and below the diagonal are stacked.) Although canonical links are convenient, they are not necessary. In general, we allow known links g_1 and g_2 so that $\mu_{1ij} = g_1^{-1}(\eta_{1ij})$, $\mu_{2ij} = g_2^{-1}(\eta_{2ij})$. In addition, we assume that the mixing probabilities $\boldsymbol{p}_i = (p_{i1}, \ldots, p_{it_i})^T$, $i = 1, \ldots, K$, each following a regression model of the form $g_p(\boldsymbol{p}_i) = \boldsymbol{W}_i\boldsymbol{\gamma}$, involving a known link function g_p, unknown $s-$dimensional regression parameter $\boldsymbol{\gamma}$, and $t_i \times s$ design matrix $\boldsymbol{W}_i$. Typically,

g_p will be taken to be the logit link, but the probit, complementary-log-log, or other link function can be chosen here.

Let $\tilde{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}^T, \boldsymbol{\theta}_1^T)^T$ and $\tilde{\boldsymbol{\beta}} = (\boldsymbol{\beta}^T, \boldsymbol{\theta}_2^T)^T$, and denote the combined vector of model parameters as $\boldsymbol{\delta} = (\tilde{\boldsymbol{\alpha}}^T, \tilde{\boldsymbol{\beta}}^T, \boldsymbol{\gamma}^T, \sigma_1, \sigma_2)^T$. If we assume $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K$ are independent $N(\boldsymbol{0}, \boldsymbol{I})$ random vectors, then the loglikelihood for $\boldsymbol{\delta}$ based on $\boldsymbol{y}$ is given by

$$\ell(\boldsymbol{\delta}; \boldsymbol{y}) = \sum_{i=1}^{K} \log \left\{ \int \prod_{j=1}^{t_i} f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}) \phi_q(\boldsymbol{b}_i) d\boldsymbol{b}_i \right\}, \tag{3.2.1}$$

where

$$f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}) = \{p_{ij}(\boldsymbol{\gamma})\} f_1(y_{ij}|\boldsymbol{b}_i; \tilde{\boldsymbol{\alpha}}) + \{1 - p_{ij}(\boldsymbol{\gamma})\} f_2(y_{ij}|\boldsymbol{b}_i; \tilde{\boldsymbol{\beta}}),$$

$\phi_q(\cdot)$ denotes the $q-$dimensional standard normal density function, and the integral is $q-$dimensional over $(-\infty, \infty) \times \cdots \times (-\infty, \infty)$ (q times).

3.3 Fitting The Two-Component Mixture Model via The EM Algorithm

The complications of parameter estimation in mixture models are simplified considerably by applying the EM algorithm. Let the Bernoulli random variable u_{ij}, $i = 1, \ldots, K, j = 1, \ldots, t_i$ denote the component membership; u_{ij} equals one if Y_{ij} is drawn from distribution F_1 and equals zero if Y_{ij} is drawn from F_2. Then the "complete" data for the EM algorithm are $(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b})$. Among them, $(\boldsymbol{u}, \boldsymbol{b})$ play the role of missing data, where $\boldsymbol{u} = (u_{11}, \ldots, u_{Kt_K})^T$. Based on the complete data $(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b})$, the loglikelihood is given by

$$\log f(\boldsymbol{b}) + \log f(\boldsymbol{u}|\boldsymbol{b}; \boldsymbol{\delta}) + \log f(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta}), \tag{3.3.1}$$

which has kernel

$$\begin{aligned} \ell^c(\boldsymbol{\delta}; \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}) \;=\; & \sum_{i=1}^{K} \sum_{j=1}^{t_i} [u_{ij} \log p_{ij}(\boldsymbol{\gamma}) + (1 - u_{ij}) \log\{1 - p_{ij}(\boldsymbol{\gamma})\}] \\ & + \sum_{i=1}^{K} \sum_{j=1}^{t_i} u_{ij} (\log h_1(y_{ij}, \sigma_1) + w_{ij}[\zeta_{1ij}(\tilde{\boldsymbol{\alpha}}) y_{ij} - \kappa_1\{\zeta_{1ij}(\tilde{\boldsymbol{\alpha}})\}]/\sigma_1) \\ & + \sum_{i=1}^{K} \sum_{j=1}^{t_i} (1 - u_{ij}) (\log h_2(y_{ij}, \sigma_2) + w_{ij}[\zeta_{2ij}(\tilde{\boldsymbol{\beta}}) y_{ij} - \kappa_2\{\zeta_{2ij}(\tilde{\boldsymbol{\beta}})\}]/\sigma_2), \end{aligned} \tag{3.3.2}$$

where, $\boldsymbol{b} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K)^T$. Based on this complete data loglikelihood, the EM algorithm is applied both to ML estimation in section 3.3.1 and to nonparametric maximum likelihood estimation (NPML) in section 3.3.2.

3.3.1 ML Estimation for Normal Random Effects

Given a starting value for the parameter vector $\boldsymbol{\delta}$, the EM algorithm proceeds iteratively to obtain ML estimates, alternating between an expectation step and a maximization step. Convergence is obtained when the change in successive values of parameter estimates is small relative to a convergence criterion ϵ.

E-step

In the $(h+1)^{th}$ iteration of EM algorithm, we compute

$$Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) = E\{\log f(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta})|\boldsymbol{y}, \boldsymbol{\delta}^{(h)}\}$$

in the E-step, where the expectation is with respect to the joint distribution of $\boldsymbol{u}, \boldsymbol{b}$ given $\boldsymbol{y}$ and $\boldsymbol{\delta}^{(h)}$. This conditional expectation can be taken in two stages by writing

$$Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) = E[E\{\log f(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta})|\boldsymbol{y}, \boldsymbol{b}, \boldsymbol{\delta}^{(h)}\}|\boldsymbol{y}, \boldsymbol{\delta}^{(h)}],$$

where the inner expectation is with respect to $\boldsymbol{u}$ only. Since $\log f(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta})$ is linear with respect to $\boldsymbol{u}$, this inner expectation can be taken simply by substituting $\boldsymbol{u}^{(h)} = E(\boldsymbol{u}|\boldsymbol{y}, \boldsymbol{b}, \boldsymbol{\delta}^{(h)})$ for $\boldsymbol{u}$. The vector $\boldsymbol{u}^{(h)}$ is easily computed, with elements

$$\begin{aligned} u_{ij}^{(h)}(\boldsymbol{b}_i) &= E(u_{ij}|\boldsymbol{y}, \boldsymbol{b}_i, \boldsymbol{\delta}^{(h)}) \\ &= \left[1 + \frac{1 - p_{ij}(\boldsymbol{\gamma}^{(h)})}{p_{ij}(\boldsymbol{\gamma}^{(h)})} \frac{f_2\{y_{ij}|\boldsymbol{b}_i; \zeta_{2ij}(\tilde{\boldsymbol{\beta}}^{(h)}), \sigma_2^{(h)}\}}{f_1\{y_{ij}|\boldsymbol{b}_i; \zeta_{1ij}(\tilde{\boldsymbol{\alpha}}^{(h)}), \sigma_1^{(h)}\}}\right]^{-1}. \end{aligned} \tag{3.3.3}$$

Here, the superscript (h) indicates evaluation at the value obtained in the h^{th} step of the algorithm. Note that $\boldsymbol{u}^{(h)}$ is a function of $\boldsymbol{b}_i$, so we have indicated that dependence in the notation $u_{ij}^{(h)}(\boldsymbol{b}_i)$. Taking the outer expectation and dropping terms not involving $\boldsymbol{\delta}$, we

obtain

$$\begin{aligned} Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) &= E\{\log f(\boldsymbol{y},\boldsymbol{u}^{(h)},\boldsymbol{b};\boldsymbol{\delta})|\boldsymbol{y},\boldsymbol{\delta}^{(h)}\} \\ &= \frac{\sum_{i=1}^{K}\sum_{j=1}^{t_i}\int \ell^c(\boldsymbol{\delta};y_{ij},u_{ij}^{(h)}(\boldsymbol{b}_i))f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)})\phi_q(\boldsymbol{b}_i)d\boldsymbol{b}_i}{\int f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)})\phi_q(\boldsymbol{b}_i)d\boldsymbol{b}_i}. \end{aligned} \tag{3.3.4}$$

The integrals in (3.3.4) are now with respect to the random effects $\boldsymbol{b}$ only. We consider two numerical approximation methods to evaluate this integral: ordinary Gaussian quadrature (OGQ) and adaptive Gaussian quadrature (AGQ).

(1) Ordinary Gaussian Quadrature

Several authors (e.g., Hinde, 1982; Anderson and Aitkin, 1985; Hedeker and Gibbons, 1994) have dealt with a similar challenge in the GLMM context by employing OGQ to integrate with respect to Gaussian random effects to obtain the marginal loglikelihood of the model. Pinheiro and Bates (1995) describe this method and several others for approximating the loglikelihood in nonlinear mixed-effects models. It is also a natural and convenient approach to employ here. We follow the notation of these authors (Pinheiro and Bates, §2.4) in our presentation.

Let b_ℓ^* and π_ℓ, $\ell = 1,\ldots,m$, denote respectively the abscissas and weights for $m-$point OGQ (see, e.g., Abramowitz and Stegun, 1972, for tables of these values), and define $g_i^{(h)}$ as

$$g_i^{(h)} \equiv \sum_{\ell_1}^{m}\cdots\sum_{\ell_q}^{m} f(\boldsymbol{y}_i|\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q},\boldsymbol{\delta}^{(h)})\pi_{\ell_1}\ldots\pi_{\ell_q} \approx \int f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)})\phi_q(\boldsymbol{b}_i)d\boldsymbol{b}_i,$$

where $\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q} = (b^*_{\ell_1},\ldots,b^*_{\ell_q})^T$. Then $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ in formula (3.4) is approximated by

$$\begin{aligned} &\sum_{i,j}\Big(\sum_{\ell_1,\ldots,\ell_q}^{m} w^{(h)}_{i\ell_1,\ldots,\ell_q}[u_{ij}^{(h)}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})\log p_{ij}(\boldsymbol{\gamma}) + \{1-u_{ij}^{(h)}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})\}\log\{1-p_{ij}(\boldsymbol{\gamma})\}] \\ +\ &\sum_{\ell_1,\ldots,\ell_q}^{m} w^{(h)}_{i\ell_1,\ldots,\ell_q}u_{ij}^{(h)}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})\left[\log h_1(y_{ij},\sigma_1) + \frac{w_{ij}}{\sigma_1}\{\zeta^*_{1ij}y_{ij} - \kappa_1(\zeta^*_{1ij})\}\right] \\ +\ &\sum_{\ell_1,\ldots,\ell_q}^{m} w^{(h)}_{i\ell_1,\ldots,\ell_q}\{1-u_{ij}^{(h)}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})\}\left[\log h_2(y_{ij},\sigma_2) + \frac{w_{ij}}{\sigma_2}\{\zeta^*_{2ij}y_{ij} - \kappa_2(\zeta^*_{2ij})\}\right]\Big), \end{aligned} \tag{3.3.5}$$

where $w^{(h)}_{i\ell_1,\ldots,\ell_q} = f(\boldsymbol{y}_i|\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q},\boldsymbol{\delta}^{(h)})\pi_{\ell_1}\ldots\pi_{\ell_q}/g_i^{(h)}$ are weights evaluated at $\boldsymbol{\delta}^{(h)}$ and $\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q}$, and ζ^*_{kij}, $k = 1,2$, are the canonical parameters evaluated at $\boldsymbol{\delta}$ and $\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q}$.

Ordinary Gaussian quadrature is easy to understand and to apply. However, the number of quadrature points m necessary for a particular application must be established, and can be quite high. Recently, several authors (Albert and Follmann, 2000; Lesaffre and Spiessens, 2001; Rabe-Hesketh, 2002) have pointed out that OGQ can perform poorly for too few quadrature points even in quite simple models. This is consistent with our experience fitting two-component GLMMs with OGQ. We encountered all of the problems described by Lesaffre and Spiessens (2001): dependence of the computed loglikelihood and its derivatives on m; numerical instability for large values of m; and erroneous multimodality of the likelihood surface for m too small, generating spurious local maxima. Essentially, the problem with OGQ is that the integrand is evaluated on a fixed grid of points, regardless of its behavior over the range of integration. There may be, and in our experience often are, regions of this range in which the integrand behaves badly (not like a low-order polynomial) (Thisted, 1988, §5.4) which may be under-represented or even completely missed (see, e.g., Albert and Follmann, 2000) in the OGQ rule. In such cases, it is advantageous to customize the quadrature to the shape of the integrand, concentrating quadrature points in the regions of this "bad behavior". This is the idea behind adaptive Gaussian quadrature.

(2) Adaptive Gaussian quadrature

Adaptive Gaussian quadrature has been described by Liu and Pierce (1994) and Pinheiro and Bates (1995). In this procedure, the grid of abscissas is centered at the conditional modes of the integrand, rather than at $\mathbf{0}$ as in OGQ, and rescaled according to the curvature of the integrand. According to Liu and Pierce, "the requirement for effective results [with AGQ] is that the ratio of [the integrand] to some Gaussian curve be a moderately smooth function. This arises frequently, for example when [the integrand] is a likelihood function, the product of a likelihood function and a Gaussian density, and the product of several likelihood functions, etc." (Liu and Pierce, 1994, p.625) The integrands in (3.3.4) both satisfy this requirement.

Following the notation of Liu and Pierce (1994), let $\hat{\boldsymbol{b}}_i^1$, $\hat{\boldsymbol{b}}_i^2$, respectively, denote the modes of the integrands

$$g_1(\boldsymbol{b}_i) \equiv \sum_{j=1}^{t_i} \ell^c(\boldsymbol{\delta}; y_{ij}, u_{ij}^{(h)}(\boldsymbol{b}_i)) f(\boldsymbol{y}_i|\boldsymbol{b}_i; \boldsymbol{\delta}^{(h)}) \phi_q(\boldsymbol{b}_i)$$

and

$$g_2(\boldsymbol{b}_i) \equiv f(\boldsymbol{y}_i|\boldsymbol{b}_i; \boldsymbol{\delta}^{(h)}) \phi_q(\boldsymbol{b}_i)$$

from equation (3.3.4). In addition, let $\hat{\boldsymbol{\Gamma}}_{1i}$, $\hat{\boldsymbol{\Gamma}}_{2i}$ be the Hessian matrices of $\log g_1(\boldsymbol{b}_i)$ and $\log g_2(\boldsymbol{b}_i)$ evaluated at $\hat{\boldsymbol{b}}_i^1$, $\hat{\boldsymbol{b}}_i^2$, and let $\boldsymbol{\pi}_{\ell_1,\ldots,\ell_q} = (\pi_{\ell_1}, \ldots, \pi_{\ell_q})^T$ and $\boldsymbol{z}_{\ell_1,\ldots,\ell_q} = (z_{\ell_1}, \ldots, z_{\ell_q})^T$, where $\pi_1, \ldots, \pi_m$ and $z_1, \ldots, z_m$ are m-point OGQ weights and abscissas, respectively. Then the quadrature points under AGQ are shifted and rescaled versions of $\boldsymbol{z}_{\ell_1,\ldots,\ell_q}$, as follows:

$$\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*} = (b_{i\ell_1}^{1*}, \ldots, b_{i\ell_q}^{1*})^T = \hat{\boldsymbol{b}}_i^1 + 2^{q/2} \hat{\boldsymbol{\Gamma}}_{1i}^{-1/2} \boldsymbol{z}_{\ell_1,\ldots,\ell_q}$$

and

$$\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{2*} = (b_{i\ell_1}^{2*}, \ldots, b_{i\ell_q}^{2*})^T = \hat{\boldsymbol{b}}_i^2 + 2^{q/2} \hat{\boldsymbol{\Gamma}}_{2i}^{-1/2} \boldsymbol{z}_{\ell_1,\ldots,\ell_q},$$

for $g_1(\boldsymbol{b}_i)$ and $g_2(\boldsymbol{b}_i)$, respectively. The corresponding AGQ weights are $\boldsymbol{w}_{\ell_1,\ldots,\ell_q}^* = (w_{\ell_1}^*, \ldots, w_{\ell_q}^*)^T$, where $w_i^* = \pi_i \exp(z_i^2)$. Hence, at the E step, $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ is approximated by

$$\begin{aligned}
&\sum_{i,j} \Big(\sum_{\ell_1,\ldots,\ell_q}^{m} w_{i\ell_1,\ldots,\ell_q}^{(h)} [u_{ij}^{(h)}(\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*}) \log p_{ij}(\boldsymbol{\gamma}) + \{1 - u_{ij}^{(h)}(\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*})\} \log\{1 - p_{ij}(\boldsymbol{\gamma})\}] \\
&+ \sum_{\ell_1,\ldots,\ell_q}^{m} w_{i\ell_1,\ldots,\ell_q}^{(h)} u_{ij}^{(h)}(\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*}) \left[\log h_1(y_{ij}, \sigma_1) + \frac{w_{ij}}{\sigma_1} \{\zeta_{1ij}^* y_{ij} - \kappa_1(\zeta_{1ij}^*)\} \right] \\
&+ \sum_{\ell_1,\ldots,\ell_q}^{m} w_{i\ell_1,\ldots,\ell_q}^{(h)} \{1 - u_{ij}^{(h)}(\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*})\} \left[\log h_2(y_{ij}, \sigma_2) + \frac{w_{ij}}{\sigma_2} \{\zeta_{2ij}^* y_{ij} - \kappa_2(\zeta_{2ij}^*)\} \right] \Big),
\end{aligned} \tag{3.3.6}$$

where

$$w_{i,\ell_1,\ldots,\ell_q}^{(h)} = \frac{|\hat{\boldsymbol{\Gamma}}_{1i}|^{-1/2} f(\boldsymbol{y}_i|\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*}; \boldsymbol{\delta}^{(h)}) \phi_q(\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{1*}) \prod_{n=1}^q w_{\ell_n}^*}{|\hat{\boldsymbol{\Gamma}}_{2i}|^{-1/2} \sum_{\ell_1,\ldots,\ell_q}^m \left[f(\boldsymbol{y}_i|\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{2*}; \boldsymbol{\delta}^{(h)}) \phi_q(\boldsymbol{b}_{i\ell_1,\ldots,\ell_q}^{2*}) \prod_{n=1}^q w_{\ell_n}^* \right]}$$

are weights that do not involve $\boldsymbol{\delta}$, the parameter vector with respect to which $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ is maximized in the M step.

***M*-step**

In the $(h+1)^{th}$ iteration of the algorithm, the M-step maximizes the approximation to $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ given by either (3.3.5) or (3.3.6) with respect to $\boldsymbol{\delta}$. Whichever approximation is used, $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ has a relatively simple form which allows it to be maximized in a straightforward way. Using either OGQ (3.3.5) or AGQ (3.3.6), the approximation can be seen to be a sum of three terms: the first a weighted binomial loglikelihood involving $\boldsymbol{\gamma}$ only; the second a weighted exponential dispersion family loglikelihood involving only $\boldsymbol{\alpha}$, $\boldsymbol{\theta}_1$ and σ_1; and the third a weighted exponential dispersion family loglikelihood involving only $\boldsymbol{\beta}$, $\boldsymbol{\theta}_2$ and σ_2. Therefore, the M step for $\boldsymbol{\delta}$ can be done in three stages by separately maximizing the three terms in $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$. For each term, this can be done by fitting a weighted version of a standard GLM.

M Step for $\boldsymbol{\gamma}$. Maximization of $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ with respect to $\boldsymbol{\gamma}$ can be accomplished by fitting a weighted binomial regression of the $u_{ij}^{(h)}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})$'s on $\boldsymbol{W}_i \otimes \mathbf{1}_{m^q}$ with weights $w^{(h)}_{i\ell_1,\ldots,\ell_q}$. Here $\mathbf{1}_k$ is the $k \times 1$ vector of ones. For instance, for g_p taken to be the logit link, we would perform a weighted logistic regression with a $Nm^q \times 1$ response vector formed by stacking the $u_{ij}^{(h)}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})$'s in such a way so that the indices $i, j, \ell_1, \ldots, \ell_q$ cycle through their values most quickly from right to left. The design matrix for this regression is the matrix formed by repeating each row of $\boldsymbol{W} = (\boldsymbol{W}_1^T, \ldots, \boldsymbol{W}_K^T)^T$ m^q times, and the weight for the $(i, j, \ell_1, \ldots, \ell_q)^{th}$ response is given by $w^{(h)}_{i\ell_1,\ldots,\ell_q}$ (constant over j).

M Step for $\tilde{\boldsymbol{\alpha}}, \sigma_1$. Maximization of $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ with respect to $\tilde{\boldsymbol{\alpha}}$ and σ_1 can be done simultaneously by again fitting a weighted GLM. Let $\boldsymbol{X}^* = [(\boldsymbol{X} \otimes \mathbf{1}_{m^q}), \boldsymbol{U}^*]$ where $\boldsymbol{U}^*$ is the $Nm^q \times q(q+1)/2$ matrix with $(i, j, \ell_1, \ldots, \ell_q)^{th}$ row equal to $\{\text{vech}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q}\boldsymbol{U}_{ij})\}^T$, where $\boldsymbol{U}_{ij}$ is the j^{th} row of the random effects' design matrix $\boldsymbol{U}_i$. Then maximization with respect to $\tilde{\boldsymbol{\alpha}}$ and σ_1 can be accomplished by fitting a weighted GLM with mean $g_1^{-1}(\boldsymbol{X}^*\tilde{\boldsymbol{\alpha}})$, response vector $\boldsymbol{y} \otimes \mathbf{1}_{m^q}$ and weight $w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})$ corresponding to the $(i, j, \ell_1, \ldots, \ell_q)^{th}$ element of the response vector.

M Step for $\tilde{\boldsymbol{\beta}}, \sigma_2$. Maximization with respect to $\tilde{\boldsymbol{\beta}}$ and σ_2 can be done by maximizing the third term of $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$. This step proceeds in the same manner as the M step for $\tilde{\boldsymbol{\alpha}}$

and σ_1. Again we fit a weighted GLM based on an expanded data set. The design matrix in this regression is $\boldsymbol{Z}^* = [(\boldsymbol{Z} \otimes \mathbf{1}_{m^q}), \boldsymbol{U}^*]$, the mean function is $g_2^{-1}(\boldsymbol{Z}^*\tilde{\boldsymbol{\beta}})$, the response vector is $\boldsymbol{y} \otimes \mathbf{1}_{m^q}$, and the weight associated with the $(i, j, \ell_1, \ldots, \ell_q)^{th}$ response is $w^{(h)}_{i\ell_1,\ldots,\ell_q}\{1 - u^{(h)}_{ij}(\boldsymbol{b}^*_{\ell_1,\ldots,\ell_q})\}$.

3.3.2 NPML Estimation

One limitation of the modeling approach described above is the normality assumption on the random effects. The effects on parameter estimation of misspecification of the random effects distribution in GLMMs have been studied by Neuhaus, Hauck, and Kalbfleisch (1992) and, more recently, Heagerty and Kurland (2001). The results of these authors indicate that regression parameter estimators are asymptotically biased in this situation, although the size of the bias is typically small unless the mixing distribution assumptions are grossly violated. In situations in which little is known about the mixing distribution, or if it is believed to be highly skewed or otherwise non-normal, an alternative approach is to estimate the random effects' distribution nonparametrically. This approach, known as NPML, has been developed by many authors (e.g., Hinde and Wood, 1987; Follmann and Lambert, 1989; Aitkin, 1996, 1999) in simpler contexts; we follow Aitkin (1999) and adapt his methods to the two-component GLMM setting.

Aitkin's (1999) approach to NPML estimation can be seen as a modification of OGQ in which the quadrature weights (i.e., mass points) and abscissas are estimated from the data rather than taken as fixed constants. This can be done as part of the EM algorithm as outlined in section 3.3.1 by incorporating the abscissas and masses as parameters of the complete data loglikelihood. The procedure is most easily described for one dimensional random effects, so for the moment assume a random intercept model with $q = 1$. In our two-component mixture of GLMMs, the two GLMMs have linear predictors $\eta_{1ij} = \boldsymbol{x}_{ij}^T\boldsymbol{\alpha} + \theta_1 b_i$ and $\eta_{2ij} = \boldsymbol{z}_{ij}^T\boldsymbol{\beta} + \theta_2 b_i$, respectively. In each component, there is mixing over the continuous distribution of $\theta_k b_i$, $k = 1, 2$. In NPML we replace these continuous distributions with discrete ones with masses

at the unknown values $\boldsymbol{b}_k^* = (b_{k1}^*, b_{k2}^*, \ldots, b_{km}^*)^T$, $k = 1, 2$. Thus for each observation i, j, we obtain m linear predictors in each component: $\eta_{1ij\ell} = \boldsymbol{x}_{ij}^T\boldsymbol{\alpha} + b_{1\ell}^*$, $\ell = 1, \ldots, m$, and $\eta_{2ij\ell} = \boldsymbol{z}_{ij}^T\boldsymbol{\beta} + b_{2\ell}^*$, $\ell = 1, \ldots, m$, with unknown masses $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)^T$. The parameters $\boldsymbol{b}_1^*$, $\boldsymbol{b}_2^*$, and $\boldsymbol{\pi}$ describing the mixing distribution are regarded as nuisance parameters, with interest centered on the regression paramaters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

To describe the EM algorithm for NPML, redefine $\boldsymbol{\delta} \equiv (\boldsymbol{\alpha}, \sigma_1, b_{11}^*, \ldots, b_{1m}^*, \boldsymbol{\beta}, \sigma_2, b_{21}^*, \ldots, b_{2m}^*, \boldsymbol{\gamma})^T$. Then the E step yields $Q(\boldsymbol{\delta}, \boldsymbol{\pi}|\boldsymbol{\delta}^{(h)}, \boldsymbol{\pi}^{(h)})$, given by

$$\begin{aligned}
&\sum_{i,j}\Big(\sum_{\ell=1}^{m} w_{i\ell}^{(h)}[u_{ij}^{(h)}(\boldsymbol{b}_1^{*(h)}, \boldsymbol{b}_2^{*(h)})\log p_{ij}(\boldsymbol{\gamma}) + \{1 - u_{ij}^{(h)}(\boldsymbol{b}_1^{*(h)}, \boldsymbol{b}_2^{*(h)})\}\log\{1 - p_{ij}(\boldsymbol{\gamma})\}] \\
&+\sum_{\ell=1}^{m} w_{i\ell}^{(h)} u_{ij}^{(h)}(\boldsymbol{b}_1^{*(h)}, \boldsymbol{b}_2^{*(h)})\left[\log h_1(y_{ij}, \sigma_1) + \frac{w_{ij}}{\sigma_1}\{\zeta_{1ij\ell}^* y_{ij} - \kappa_1(\zeta_{1ij\ell}^*)\}\right] \\
&+\sum_{\ell=1}^{m} w_{i\ell}^{(h)}\{1 - u_{ij}^{(h)}(\boldsymbol{b}_1^{*(h)}, \boldsymbol{b}_2^{*(h)})\}\left[\log h_2(y_{ij}, \sigma_2) + \frac{w_{ij}}{\sigma_2}\{\zeta_{2ij\ell}^* y_{ij} - \kappa_2(\zeta_{2ij\ell}^*)\}\right] \\
&+\sum_{\ell=1}^{m} w_{i\ell}^{(h)}\log(\pi_\ell)) \qquad (3.3.7)
\end{aligned}$$

(cf. equation (3.3.5)), where

$$w_{i\ell}^{(h)} = \frac{f(\boldsymbol{y}_i|b_{1\ell}^{*(h)}, b_{2\ell}^{*(h)}; \boldsymbol{\delta}^{(h)})\pi_\ell^{(h)}}{\sum_{k=1}^m f(\boldsymbol{y}_i|b_{1k}^{*(h)}, b_{2k}^{*(h)}; \boldsymbol{\delta}^{(h)})\pi_k^{(h)}},$$

and the canonical parameters $\boldsymbol{\zeta}_{1ijl}^*$ and $\boldsymbol{\zeta}_{2ijl}^*$ are evaluated at the linear predictors

$$\begin{aligned}
\boldsymbol{\eta}_{1ijl}^* &= \boldsymbol{x}_{ij}^T\boldsymbol{\alpha} + b_{1\ell}^*, \\
\boldsymbol{\eta}_{2ijl}^* &= \boldsymbol{z}_{ij}^T\boldsymbol{\beta} + b_{2\ell}^*.
\end{aligned}$$

Comparing the above expression for $Q(\boldsymbol{\delta}, \boldsymbol{\pi}|\boldsymbol{\delta}^{(h)}, \boldsymbol{\pi}^{(h)})$ to (3.3.5), the corresponding quantity in OGQ, we see that we have much the same form, with just an extra term for $\boldsymbol{\pi}$ in (3.3.7). Therefore, the M step proceeds in much the same manner as described previously.

M Step for $\boldsymbol{\gamma}$. This can be done by fitting a weighted binomial regression of the $u_{ij}^{(h)}(\boldsymbol{b}_1^{*(h)}, \boldsymbol{b}_2^{*(h)})$'s on $\boldsymbol{W}_i \otimes \mathbf{1}_m$ with weights $w_{i\ell}$.

M Step for $\boldsymbol{\alpha}, \sigma_1, \boldsymbol{b}_1^*$. Maximization of $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}, \boldsymbol{\pi}^{(h)})$ with respect to $\boldsymbol{\alpha}$, σ_1 and $\boldsymbol{b}_1^*$ can be done simultaneously by again fitting a weighted GLM. Let $\boldsymbol{X}^* = [(\boldsymbol{X} \otimes \mathbf{1}_m), \boldsymbol{I}_n \otimes \mathbf{1}_N]$. Then

maximization with respect to $\boldsymbol{\alpha}$, σ_1 and $\boldsymbol{b}_1^*$ consists of fitting a weighted GLM with mean $g_1^{-1}\{\boldsymbol{X}^*(\boldsymbol{\alpha}^T, \boldsymbol{b}_1^{*T})^T\}$, response vector $\boldsymbol{y} \otimes \mathbf{1}_m$ and weight $w_{i\ell}^{(h)} u_{ij}^{(h)}(b_{1\ell}^*, b_{2\ell}^*)$ corresponding to the $(i, j, \ell)^{th}$ element of the response vector.

M Step for $\boldsymbol{\beta}, \sigma_2, \boldsymbol{b}_2^*$. Maximization with respect to $\boldsymbol{\beta}$, σ_2 and $\boldsymbol{b}_2^*$ can be done by maximizing the third term of $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}, \boldsymbol{\pi}^{(h)})$. Again we fit a weighted GLM based on an expanded data set. The design matrix in this regression is $\boldsymbol{Z}^* = [(\boldsymbol{Z} \otimes \mathbf{1}_m), \boldsymbol{I}_n \otimes \mathbf{1}_N]$, the mean function is $g_2^{-1}(\boldsymbol{Z}^*[\boldsymbol{\beta}, \boldsymbol{b}_2^*])$, the response vector is $\boldsymbol{y} \otimes \mathbf{1}_m$ and the weight associated with the $(i, j, \ell)^{th}$ response is $w_{i\ell}^{(h)}\{1 - u_{ij}^{(h)}(b_{1\ell}^*, b_{2\ell}^*)\}$.

M Step for $\boldsymbol{\pi}$. Maximization with respect to $\boldsymbol{\pi}$ can be done by maximizing the fourth term of (3.3.7). This maximization yields the closed-form solution

$$\pi_\ell^{(h+1)} = \sum_{i=1}^{K} t_i w_{i\ell}^{(h)} / \sum_{i=1}^{K} t_i, \qquad \ell = 1, \ldots, m.$$

Extension to more than 1 dimensional random effects is straight-forward. For example, suppose we have two random effects z and u; we can estimate the joint distribution of z and u nonparametrically. Suppose the number of quadrature points is 2; then we have z_i and u_j, where $i, j = 1, 2$. The discrete mass points are $(z_1, u_1), (z_1, u_2), (z_2, u_1), (z_2, u_2)$, extending the data to be 4 times their original length. We can then estimate $k = 4$ components in the (z, u) plane together with their masses π_k, $k = 1, \ldots, 4$.

3.4 Computation of Information Matrix

A common criticism of the EM algorithm is that, unlike Newton-Raphson and other gradient methods, it does not produce a variance-covariance matrix for parameter estimators as a by-product of estimation. Hence, many authors have considered how to obtain a variance-covariance matrix, or at least standard errors, with minimal extra effort when using the EM algorithm. In this paper we employ a method of calculating the observed information matrix presented by Oakes (1999) for the OGQ and AGQ approaches and a method of calculating the expected information matrix introduced by Friedl and Kauermann (2000) for the NPML approach.

Following Oakes (1999), we can obtain the observed information matrix from $Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$, the conditional expectation of the complete data loglikelihood given the observed data. The relationship can be expressed as:

$$\frac{\partial^2 \ell(\boldsymbol{\delta}^{(h)};\boldsymbol{y})}{\partial \boldsymbol{\delta}^{(h)} \partial \boldsymbol{\delta}^{(h)T}} = \left\{ \frac{\partial^2 Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^T} + \frac{\partial^2 Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^{(h)T}} \right\} \Big|_{\boldsymbol{\delta}=\boldsymbol{\delta}^{(h)}}. \tag{3.4.1}$$

This relationship is valid for all $\boldsymbol{\delta}$, hence it is valid at the ML estimator $\boldsymbol{\delta}^{(h)} = \hat{\boldsymbol{\delta}}$ also.

The first term of (3.4.1) is block-diagonal, with components that are given by the negative information matrices associated with the GLM fits conducted in the M step. That is, $\{\frac{\partial^2 Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^T}\}\big|_{\boldsymbol{\delta}=\boldsymbol{\delta}^{(h)}}$ is automatically obtained as part of the M step from the variance-covariance matrices output by the GLM fitting routine.

Based on equation (3.3.5), the second term $\frac{\partial^2 Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^{(h)T}}$ can be written as

$$\begin{aligned}
&\sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial}{\partial \boldsymbol{\delta}} \left\{ \log p_{ij}(\boldsymbol{\gamma}) - \log\{1 - p_{ij}(\boldsymbol{\gamma})\} \right\} \frac{\partial w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}))}{\partial \boldsymbol{\delta}^{(h)T}} \\
&+ \sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial}{\partial \boldsymbol{\delta}} \left\{ \log f_{1ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}; \tilde{\boldsymbol{\alpha}}) \right\} \frac{\partial w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}))}{\partial \boldsymbol{\delta}^{(h)T}} \\
&- \sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial}{\partial \boldsymbol{\delta}} \left\{ \log f_{2ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}; \tilde{\boldsymbol{\beta}}) \right\} \frac{\partial w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}))}{\partial \boldsymbol{\delta}^{(h)T}} \\
&+ \sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial}{\partial \boldsymbol{\delta}} \left\{ \log\{1 - p_{ij}(\boldsymbol{\gamma})\} + \log f_{2ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}; \tilde{\boldsymbol{\beta}}) \right\} \frac{\partial w^{(h)}_{i\ell_1,\ldots,\ell_q})}{\partial \boldsymbol{\delta}^{(h)T}}
\end{aligned} \tag{3.4.2}$$

for the OGQ approach. Based on equation (3.3.6), the second term of (3.4.1) can be written as

$$\begin{aligned}
&\sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial \log p_{ij}(\boldsymbol{\gamma})}{\partial \boldsymbol{\delta}} \frac{\partial (w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}))}{\boldsymbol{\delta}^{(h)T}} \\
&+ \sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial \log\{1 - p_{ij}(\boldsymbol{\gamma})\}}{\partial \boldsymbol{\delta}} \frac{\partial (w^{(h)}_{i\ell_1,\ldots,\ell_q}\{1 - u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q})\})}{\boldsymbol{\delta}^{(h)T}} \\
&+ \sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial \log f_{1ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}; \tilde{\boldsymbol{\alpha}})}{\partial \boldsymbol{\delta}} \frac{\partial (w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}))}{\boldsymbol{\delta}^{(h)T}} \\
&+ \sum_{i,j,\ell_1,\ldots,\ell_q} \frac{\partial^2 \log f_{1ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}; \tilde{\boldsymbol{\alpha}})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^{(h)T}} (w^{(h)}_{i\ell_1,\ldots,\ell_q} u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q}))
\end{aligned}$$

$$+\sum_{i,j,\ell_1,\ldots,\ell_q}\frac{\partial \log f_{2ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q};\tilde{\boldsymbol{\beta}})}{\partial\boldsymbol{\delta}}\frac{\partial(w^{(h)}_{i\ell_1,\ldots,\ell_q}\{1-u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q})\})}{\boldsymbol{\delta}^{(h)T}}$$
$$+\sum_{i,j,\ell_1,\ldots,\ell_q}\frac{\partial^2 \log f_{2ij\ell_1,\ldots,\ell_q}(y_{ij\ell_1,\ldots,\ell_q}|\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q};\tilde{\boldsymbol{\beta}})}{\partial\boldsymbol{\delta}\partial\boldsymbol{\delta}^{(h)T}}(w^{(h)}_{i\ell_1,\ldots,\ell_q}\{1-u^{(h)}_{ij\ell_1,\ldots,\ell_q}(\boldsymbol{b}^{1*}_{1\ell_1,\ldots,\ell_q})\}) \tag{3.4.3}$$

for the AGQ approach. Although (3.4.2) and (3.4.3) seem complicated, there is not much extra work involved in calculating them. Every term in these two formula used to obtain the derivatives has been calculated when fitting the model. At convergence, numerical differentiation can be applied to these terms to get a variance-covariance matrix.

Friedl and Kauermann's (2000) paper provides a way to obtain an approximation to the expected information matrix for normally distributed random effects and in the NPML context as well. The idea can be more easily explained under the assumption of normal random effects, though extension to NPML is straightforward. This method is based on using OGQ and embedding the EM algorithm into the estimating equation context by defining $g_{\boldsymbol{\delta}}(\boldsymbol{\delta}) = \partial Q_m(\tilde{\boldsymbol{\delta}}|\boldsymbol{\delta})/\partial\tilde{\boldsymbol{\delta}}|_{\tilde{\boldsymbol{\delta}}=\boldsymbol{\delta}}$ (cf. Friedl and Kauermann, 2000, p.763), where Q_m means the m point OGQ approximation to Q. EM estimates are solutions to the estimating equation $g_{\boldsymbol{\delta}}(\boldsymbol{\delta}) = \mathbf{0}$, while the true parameters solve $E_m\{g_{\boldsymbol{\delta}}(\boldsymbol{\delta})\} = \mathbf{0}$. Since $E_m\Big(-\frac{\partial g_{\boldsymbol{\delta}}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}^T}\Big) = E_m\Big(g_{\boldsymbol{\delta}}(\boldsymbol{\delta})g^T_{\boldsymbol{\delta}}(\boldsymbol{\delta})\Big)$ in the GLM setting, $g_{\boldsymbol{\delta}}(\boldsymbol{\delta})$ behaves like a score equation. Thus, the variance-covariance matrix for the MLE's can be obtained by inverting $E_m\Big(-\frac{\partial g_{\boldsymbol{\delta}}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}^T}\Big)$.

For NPML, one more estimating equation is needed for the quadrature weights ($\pi_\ell, \ell = 1,\ldots,m$). Denote the combined estimating function as $g(\boldsymbol{\delta},\boldsymbol{\pi}) = (g_{\boldsymbol{\delta}}(\boldsymbol{\delta},\boldsymbol{\pi})^T, g_{\boldsymbol{\pi}}(\boldsymbol{\delta},\boldsymbol{\pi})^T)^T$. Then the variance-covariance matrix of the NPML estimators is obtained in the same manner as described for ML estimates, by working with $g(\boldsymbol{\delta},\boldsymbol{\pi})$ rather than $g_{\boldsymbol{\delta}}(\boldsymbol{\delta})$.

When applying this method to our setting, the estimating function $g_{\boldsymbol{\delta}}(\boldsymbol{\delta},\boldsymbol{\pi})$ can be obtained as follows. Define $Q_m(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) = \sum_{i,j,\ell} w^{(h)}_{i\ell}\log\{f(y_{ij},u^{(h)}_{ij}(\boldsymbol{b}_\ell)|\boldsymbol{b}_\ell;\boldsymbol{\delta})\pi_\ell\}$; then, $\frac{\partial Q_m(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial\boldsymbol{\delta}}$ is

$$\sum_{i,j,\ell} w^{(h)}_{i\ell}\frac{\partial}{\partial\boldsymbol{\delta}}\log\{f(y_{ij},u^{(h)}_{ij}(\boldsymbol{b}_\ell)|\boldsymbol{b}_\ell;\boldsymbol{\delta})\}$$

$$
\begin{aligned}
&= \sum_{i,j,\ell} w_{i\ell}^{(h)} \Big[u_{ij}^{(h)} \frac{\frac{\partial}{\partial \boldsymbol{\delta}} f_1(y_{ij}|b_\ell;\boldsymbol{\delta})}{f_1(y_{ij}|b_\ell;\boldsymbol{\delta})} + (1-u_{ij}^{(h)}) \frac{\frac{\partial}{\partial \boldsymbol{\delta}} f_2(y_{ij}|b_\ell;\boldsymbol{\delta})}{f_2(y_{ij}|b_\ell;\boldsymbol{\delta})} \Big] \\
&+ \sum_{i,j,\ell} w_{i\ell}^{(h)} \Big[u_{ij}^{(h)} \frac{\frac{\partial}{\partial \boldsymbol{\delta}} p_{ij}(\boldsymbol{\delta})}{p_{ij}(\boldsymbol{\delta})} - (1-u_{ij}^{(h)}) \frac{\frac{\partial}{\partial \boldsymbol{\delta}} p_{ij}(\boldsymbol{\delta})}{1-p_{ij}(\boldsymbol{\delta})} \Big].
\end{aligned}
$$

Letting $\boldsymbol{\delta} = \boldsymbol{\delta}^{(h)}$, the above formula becomes

$$
\begin{aligned}
&\sum_{i,j,\ell} w_{i\ell}^{(h)} \Big[u_{ij}^{(h)} \frac{\frac{\partial}{\partial \boldsymbol{\delta}^{(h)}} f_1(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)})}{f_1(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)})} + (1-u_{ij}^{(h)}) \frac{\frac{\partial}{\partial \boldsymbol{\delta}^{(h)}} f_2(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)})}{f_2(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)})} \Big] \\
+ &\sum_{i,j,\ell} w_{i\ell}^{(h)} \Big[u_{ij}^{(h)} \frac{\frac{\partial}{\partial \boldsymbol{\delta}^{(h)}} p_{ij}(\boldsymbol{\delta})}{p_{ij}(\boldsymbol{\delta}^{(h)})} - (1-u_{ij}^{(h)}) \frac{\frac{\partial}{\partial \boldsymbol{\delta}^{(h)}} p_{ij}(\boldsymbol{\delta}^{(h)})}{1-p_{ij}(\boldsymbol{\delta}^{(h)})} \Big].
\end{aligned}
$$

Since $u_{ij}^{(h)} = p_{ij}(\boldsymbol{\delta}^{(h)}) f_1(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)}) / f(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)})$, we obtain

$$
\frac{\partial Q_m(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})}{\partial \boldsymbol{\delta}}\Big|_{\boldsymbol{\delta}=\boldsymbol{\delta}^{(h)}} = \sum_{i,j,\ell} w_{i\ell}^{(h)} \frac{\partial}{\partial \boldsymbol{\delta}^{(h)}} \log f(y_{ij}|b_\ell;\boldsymbol{\delta}^{(h)}).
$$

In Friedl and Kauermann's (2000) notation, we write $g_{\boldsymbol{\delta}}(\boldsymbol{\delta},\boldsymbol{\pi}) = \sum_{i,j,\ell} w_{i\ell} \frac{\partial}{\partial \boldsymbol{\delta}} \log f(y_{ij}|b_\ell;\boldsymbol{\delta})$. This corresponds to formula (10) and the definition of $g_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{\vartheta})$ in their paper. In addition, the estimating function for $\boldsymbol{\pi}$, $g_{\boldsymbol{\pi}}(\boldsymbol{\delta},\boldsymbol{\pi})$, for our setting is exactly the same as that in their paper (equation (11)). Based on these two estimating functions, the steps leading to the expected information matrix follow exactly as in section 3 of Friedl and Kauermann's paper.

3.5 Examples

3.5.1 Measles Data

As an illustration of our methodology, we analyze annual measles data that were collected for each of 15 counties in Texas between 1985 and 1991. For each county, the annual number of preschoolers with measles was recorded as well as two variables related to measles incidence: immunization rate and density of preschoolers per county. These data are given in Sherman and le Cessie (1997) and analyzed by these authors as well. They employed a bootstrap method for dependent data to get bootstrap replicates from 15 counties. For each bootstrap resample, the parameters were estimated by maximizing the independence likelihood using

GLM methodology with Poisson response variable and the natural logarithm of the number of children as the offset. The clustered data structure and the bimodal shape of Figure 1 in their paper (see, Sherman and le Cessie, 1997, p.914) motivated us to consider a two-component GLMM for these data. In addition, from the plot of measles incidence for each county in Figure 3.1, there appears to be a mix of high and low incidences across the years. Intuitively, we can think of these high and low counts as corresponding to epidemic and non-epidemic years. This structure suggests that a two component model may fit well. In addition, such a model will allow us to separately quantify covariate effects in epidemic and non-epidemic years.

Let y_{ij} be the number of cases in county $i, (i = 1, \ldots, 15)$ in year $j, (j = 1, \ldots, 7)$, and let b_i be a 1-dimensional random county effect for county i. Then the two-component GLMM for the measles data can be expressed as

$$\begin{aligned} Y_{ij}|b_i &\sim p_{ij}\text{Poisson}(\lambda_{1ij}|b_i) + (1-p_{ij})\text{Poisson}(\lambda_{2ij}|b_2), \\ \log(\lambda_{1ij}) &= \alpha_0 + \alpha_1 \text{rate}_{ij} + \sigma_1 b_i + \log(n_{ij}), \\ \log(\lambda_{2ij}) &= \beta_0 + \beta_1 \text{rate}_{ij} + \sigma_2 b_i + \log(n_{ij}), \end{aligned} \tag{3.5.1}$$

where α_0, β_0 are fixed intercepts and α_1, β_1 are fixed effects of immunization rate for the two components respectively. In addition, $\log(n_{ij})$ represents an offset corresponding to the natural logarithm of the number of children in the ith county during the jth year, and λ_{1ij}, λ_{2ij} are (conditional) means for each Poisson component.

We fit models with (3.5.1) and two different choices of linear predictor for $\text{logit}(p_{ij})$: one with a constant mixing probability $\text{logit}(p_{ij}) = \gamma_0$, and the other assumed $\text{logit}(p_{ij})$ is linearly related to immunization rate; see models 4 and 5 in Table 3.1. Based on AIC, we selected the model with rate as a covariate in the linear predictor for $\text{logit}(p)$. This model yielded a maximum log likelihood of -1069.3 and a AIC of 2154.5.

For comparison, we fit a two-component GLM with the same linear predictors for each component as in (3.11), but with no random county effect. The results are in Table 3.1 (see models 2 and 3). The model with no covariate in the linear predictor for $\text{logit}(p_{ij})$ gave a

maximum log likelihood of -1356.2 and a AIC of 2720.3, whereas, the model with a covariate for logit(p_{ij}) yielded a maximum log likelihood of -1356.0 and a AIC of 2724.0. There is no significant difference in fit between these two models according to a likelihood ratio test, but when compared with two-component GLMMs, the inclusion of a random county effect b_i improved the fit significantly.

We also fit a non-mixture GLMM with the same linear predictor as in (3.5.1) to these data. Such a model may be a first choice to account for intra-county correlation. From Table 3.1, model 1, we can see this model clearly fit the data badly compared with the other models giving a maximum log likelihood of -5087.0 and a AIC of 10180.0.

To further investigate the suitability of the models we fit in this example, we follow the approach of Vieira et al. (2000) who suggested the use of half-normal plots as goodness-of-fit tools. Half-normal plots for the GLMM (model 1), two-component GLM (model 3) and two-component GLMM (model 5) appear in Figure 3.2 (a-c). The plots display the absolute values of the Pearson residuals versus half-normal scores, with simulated envelopes based on the assumed model evaluated at the estimated parameter values. A suitable model is indicated by the observed values falling within the simulated envelope. The Pearson residuals are defined as $[y_{ij} - \widehat{E(Y_{ij})}]/\sqrt{\widehat{\text{var}(Y_{ij})}}$, where $E(Y_{ij}) = E\{E(Y_{ij}|b_i)\}$, $\text{var}(Y_{ij}) = E(Y_{ij}^2) - \{E(Y_{ij})\}^2 = E\{E(Y_{ij}^2|b_i)\} - \{E(Y_{ij})\}^2$ for the mixed models. The marginal expectations here were evaluated using 20-point OGQ and the hats indicate evaluation at the final parameter estimates, which were obtained using 11-point AGQ. For the two-component GLM, $E(Y_{ij}) = p_{ij}\lambda_{1ij} + (1 - p_{ij})\lambda_{2ij}$, $\text{var}(Y_{ij}) = p_{ij}(\lambda_{1ij} + \lambda_{1ij}^2) + (1 - p_{ij})(\lambda_{2ij} + \lambda_{2ij}^2) - \{E(Y_{ij})\}^2$, where λ_{1ij} and λ_{2ij} are means for each Poisson component.

Figure 3.2(a) clearly indicates that the one component GLMM model is inadequate for the measles data since almost all points fall outside of the simulated envelope. Figure 3.2(b) shows that the two-component GLM improves the fit, but in the left half of the plot there are still many points outside the envelope. In Figure 3.2(c) nearly all points are along the simulated means, confirming that the two-component GLMM fits these data best.

The fitting results above are consistent with expectations based on a preliminary examination of the data and some consideration of the epidemiology of this disease. Because of the mixture of large and small incidences of measles and the epidemic phenomenon, we expected that two components would be necessary to model these data. This is borne out by the vast improvement in fit from a one to a two component GLM. Of course, the data are clustered as well, so the within county correlation must be accounted for somehow. We have chosen to account for this correlation through a random county effect, and this approach improves the fit compared with the fixed effect two component GLM. Clearly, there are other valid approaches for accounting for within-cluster correlation. Alternatives include marginal models (Rosen et al. , 2000) and transition models (Park and Basawa, 2002). As in the non-mixture case, which approach is most appropriate for accounting for the correlation will depend upon the application.

As mentioned earlier, fitting two-component GLMMs involves evaluation of the integral in the E step of the EM algorithm. The most straightforward method is OGQ. We now illustrate the limitations of this approach. We fit model (3.5.1) with logistic regression for p:

$$\text{logit}(p_{ij}) = \gamma_0 + \gamma_1 \text{rate}_{ij}. \tag{3.5.2}$$

Table 3.2 illustrates the effects of the number of quadrature points on the loglikelihood, parameter estimates of the immunization rate effect and their standard errors. The results are obtained for the number of quadrature points m ranging from 5 to 35. Standard errors are calculated from diagonal elements of the observed information matrix (inverse negative Hessian). P-values are obtained using Wald tests. From Table 3.2 it is clear that the loglikelihood, parameter estimates, and standard errors vary considerably with m. In addition, these values have not yet settled down and become close to the more accurate AGQ values. We can see Table 3.3 for m as large as 35. The closest value of the loglikelihood to those obtained with AGQ occurs for $m = 11$, but the values of $\hat{\alpha}_1$ and $\hat{\beta}_1$ are quite different than for AGQ. More importantly, perhaps, the p-values for these immunization rates are considerably dif-

ferent from those based on AGQ. In contrast, from Table 3.3, we see parameter estimates, standard errors, and loglikelihood values for AGQ, show relatively little dependence on m.

Another way that dependence of OGQ on the number of quadrature points can be seen is via plots of the loglikelihood surface. We calculated the marginal loglikelihood for model 5 on a grid of parameter values centered at the ML estimates obtained from AGQ with 11-points. We changed the coefficient of rate for component 1 (α_1) from -0.135 to -0.09 by 0.0075, and the coefficient of rate for component 2 (β_1) from -0.065 to -0.04 by 0.004 and kept other parameter values unchanged. Figure 3.3 shows the OGQ results for quadrature points ranging from 5 to 21 by 2 (cf. Lesaffre and Spiessens, 2001, Figure 5). Figure 3.4(a-c) shows the surface plots for m=5, 9, 15 when using OGQ. In both figures, it is clear that the marginal loglikelihood changes dramatically due to numerical inaccuracy. In addition, we see that multimodality of the loglikelihood surface does occur, allowing the maximization procedure to converge to a local maximum. In contrast, for AGQ, the maximized loglikelihood and loglikelihood surface show little dependence on m. In Figure 3.3, note that the loglikelihood for m=25, 27, 29 are actually for m=5, 7, 9 with AGQ. This figure shows that AGQ can obtain the true loglikelihood for small m, whereas a much higher value of m is necessary for OGQ. We also plotted the loglikelihood surface plots for AGQ using the same grid in Figure 3.4, d-f). As expected, the surfaces do not change nearly as much as for OGQ. Hence results from the AGQ method appear to be reliable, and we recommend this method over OGQ in general.

Dropping the normality assumption on the random effect, we used the NPML method to fit the model based on (3.5.1) and (3.5.2) (model 5). We followed the strategy described by Friedl and Kauermann (2000) and started the fitting procedure from a large value of m ($m = 12$), and then reduced m systematically until all quadrature points are different and no quadrature weights are very small (less than 0.01). For the measles data, we stopped the fitting procedure at m=7. For m=7, we have α_1=-0.0944, β_1=-0.14295, γ_0=0.3029 and γ_1=0.009392 with loglikelihood equal to -957.03.

3.5.2 Whitefly Data

Our second example involves data from a horticulture experiment to investigate the efficacy of several different means of applying pesticide to control whiteflies on greenhouse-raised poinsettia plants. The data arise from a randomized complete block design with repeated measures taken over 12 weeks. Eighteen experimental units were formed from 54 plants, with units consisting of 3 plants each. These units were randomized to six treatments in 3 blocks. The response variable of interest here is the number of surviving whiteflies out of the total number placed on the plant two weeks previously. These data are discussed in more detail in van Iersel, Oetting, and Hall (2000). In that paper, ZIB regression models were used to analyze these data, with random effects at the plant level to account for correlation among the repeated measures on a given plant. We return to this problem to investigate whether a two-component mixture of GLMMs can improve upon the fit of a ZIB-mixed model for these data.

Let $y_{ijk\ell}$ be the number of live adult whiteflies on plant k ($k = 1, \ldots, 54$) in treatment i ($i = 1, \ldots, 6$) in block j ($j = 1, \ldots, 3$) measured at time ℓ ($\ell = 1, \ldots, 12$). Let $n_{ijk\ell}$ be the total number of whiteflies placed on the leaf of plant k in treatment i in block j measured at time ℓ. Further let α_i be the ith treatment effect, β_j be the jth block effect, τ_ℓ be the ℓth week effect, and b_k be a 1-dimensional random plant effect for plant k. For simplicity, we consider a model containing only main effects (treatment, block and week). The two-component GLMM for these data with main effects can be expressed as

$$\begin{aligned} Y_{ijk\ell}|b_k &\sim p_{ijk\ell}\text{Binomial}(n_{ijk\ell}, \pi_{1ijk\ell}|b_k) + (1 - p_{ijk\ell})\text{Binomial}(n_{ijk\ell}, \pi_{2ijk\ell}|b_k), \\ \text{logit}(\pi_{1ijk\ell}) &= \mu_1 + \alpha_{1i}\text{treatment}_i + \beta_{1j}\text{block}_j + \tau_{1\ell}\text{week}_\ell + \sigma_1 b_k, \\ \text{logit}(\pi_{2ijk\ell}) &= \mu_2 + \alpha_{2i}\text{treatment}_i + \beta_{2j}\text{block}_j + \tau_{2\ell}\text{week}_\ell + \sigma_2 b_k. \end{aligned} \tag{3.5.3}$$

We fit model (3.5.3) with the linear predictor for the mixing probability $p_{ijk\ell}$ specified as

$$\text{logit}(p_{ijk\ell}) = \mu_3 + \alpha_{3i}\text{treatment}_i + \beta_{3j}\text{block}_j + \tau_{3\ell}\text{week}_\ell. \tag{3.5.4}$$

The results are in Table 3.4. This model yields a maximum loglikelihood of -803.48 and a AIC of 1724.96. We also fit a one-component GLM, a one-component GLMM, a ZIB model, a ZIB mixed model, and a two-component GLM. The linear predictors for the components and mixing probability contain the main effects (treatment, block and week) with or without plant random effects. That is, each of these models was chosen to be the closest and most comparable model to that given in (3.5.3) and (3.5.4) in its model class. The fitting results are shown in Table 3.4 also.

From Table 3.4, we find that the two component models are better than the corresponding one component models. In addition, models with random plant effects are better than the corresponding models without random effects. From these results, it is clear that both random effects and a second component are necessary here. In addition, a non-degenerate (non-zero) second component also improves the fit over a ZIB model. That is, the two-component GLMM fits best.

In this example the reported results are all based on 5-point AGQ. As in the previous example, we examined the performance of OGQ and AGQ for different values of m. For brevity, we omit the details of this comparison, but the results are much the same as before. Parameter estimates, standard errors and loglikelihoods were highly dependent on m for OGQ, but not for AGQ. It appears that AGQ is necessary to achieve sufficient numerical accuracy when fitting these models.

3.6 Discussion

In this paper we have formulated a class of two component mixtures of GLMMs for clustered data and described how the EM algorithm, combined with quadrature methods, can be used to fit these models using ML estimation. Extension of this model class to more than two components is possible, and is, in principle, straightforward. However, the complexity of the model, its notation, and its fitting algorithm will grow rapidly with the number of components, and it is not clear that the practical value of such models justifies consideration

of cases beyond two or three components. We envision that these finite mixture extensions of GLMMs will have application primarily in problems where there is some readily identified heterogeneity in the population so that the data represent a small number (two or three) of subpopulations that cannot be directly identified. For example, disease counts from epidemic and non-epidemic years, weekly epileptic seizure counts from patients who have "good weeks" and "bad weeks", arrhythmia counts from a sample of clinically normal patients that is contaminated with abnormal patients (e.g., patients with an undetected genetic defect for cardiomyopathy), etc.

Our model class allows for correlation due to clustering to be accounted for through the inclusion of cluster-specific random effects. Extension to multilevel models (multiple nested levels of clustering), crossed random effects, and other more general random effects structures is an important areas of future research. One attractive approach for this extension is to use a Monte Carlo EM algorithm (McCulloch, 1997) in place of our EM algorithm with quadrature. The main challenge to implementing the Monte Carlo EM in this context is sampling from the conditional distribution of random effects given the observed data. We have had some success with this approach, but have found that the computing time is prohibitively long for practical use. Another possibility is to use approximate ML or estimating equation methods such as penalized quasi-likelihood (Breslow and Clayton, 1993) and its extensions. These approaches have some drawbacks in terms of bias in the parameter estimators for highly non-normal data (e.g. binary data); but they can be applied to general random effects structures and work reasonably well in some problems. Such methods can be applied to a finite mixture of GLMMs with general random effects structures by replacing the M-step in the EM algorithm with the solution of an estimating equation. This idea is due to Rosen et al. (2000) and leads to an ES (expectation-solution) algorithm to produce approximate MLEs rather than an EM algorithm to produce MLEs. Elaboration of these ideas will appear elsewhere.

Table 3.1: Comparison of different models for the measles data

Model	Method	$\boldsymbol{W}_{ij}^T\boldsymbol{\gamma}$	-2 Log likelihood	AIC
1	GLMM	none	10174.0	10180.0
2	Two-component GLM	none	2712.3	2720.3
3	”	$\gamma_0 + \gamma_1 rate$	2712.0	2724.0
4	Two-component GLMM	none	2166.0	2178.0
5	”	$\gamma_0 + \gamma_1 rate$	2138.5	2154.5
5*	NPML	$\gamma_0 + \gamma_1 rate$	1914.1	1964.1

* without normality assumption on the random effects

Table 3.2: Fitting Results From Ordinary Gaussian Quadrature

		Component 1			Component 2		
Q. Points	Loglikelihood	α_1	std. error	p-value	β_1	std. error	p-value
m=5	-1112.93	-0.0684	0.0026	<0.0001	-0.0353	0.0095	0.0022
m=9	-1093.49	-0.0628	0.0031	<0.0001	-0.0239	0.0117	0.0603
m=11	-1069.69	-0.1006	0.0046	<0.0001	-0.0439	0.0104	0.0009
m=15	-1082.27	-0.0809	0.0029	<0.0001	-0.0342	0.0101	0.0043
m=19	-1079.21	-0.0806	0.0030	<0.0001	-0.0336	0.0102	0.0054
m=21	-1078.15	-0.0804	0.0031	<0.0001	-0.0332	0.0103	0.0061
m=25	-1082.84	-0.0804	0.0030	<0.0001	-0.0328	0.0101	0.0057
m=30	-1081.35	-0.0806	0.0030	<0.0001	-0.0336	0.0101	0.0051
m=35	-1077.68	-0.0796	0.0029	<0.0001	-0.0316	0.0103	0.0080

Table 3.3: Fitting Results From Adaptive Gaussian Quadrature

		Component 1			Component 2		
Q. Points	Loglikelihood	α_1	std. error	p-value	β_1	std. error	p-value
m=5	-1069.26	-0.1170	0.0262	0.00053	-0.0537	0.02297	0.0348
m=7	-1069.24	-0.1187	0.02667	0.00055	-0.0553	0.02301	0.0307
m=9	-1069.235	-0.1193	0.02646	0.00049	-0.05592	0.02262	0.0269
m=11	-1069.233	-0.1193	0.02590	0.00041	-0.05596	0.02213	0.0241
m=15	-1069.23	-0.1191	0.02591	0.00042	-0.05573	0.02210	0.0244
m=19	-1069.23	-0.1195	0.02615	0.00044	-0.05604	0.02226	0.0246
m=21	1069.23	-0.1194	0.02617	0.00044	-0.05601	0.02228	0.0248

Table 3.4: Comparison of different models

Model	Method	-2 Log likelihood	AIC
1	GLM	2593.57	2631.6
2	GLMM	2408.97	2449.0
3	ZIB	1928.69	2004.7
4	ZIB mixed	1883.33	1961.3
5	Two-component GLM	1628.17	1742.2
6	Two-component GLMM	1606.96	1724.96

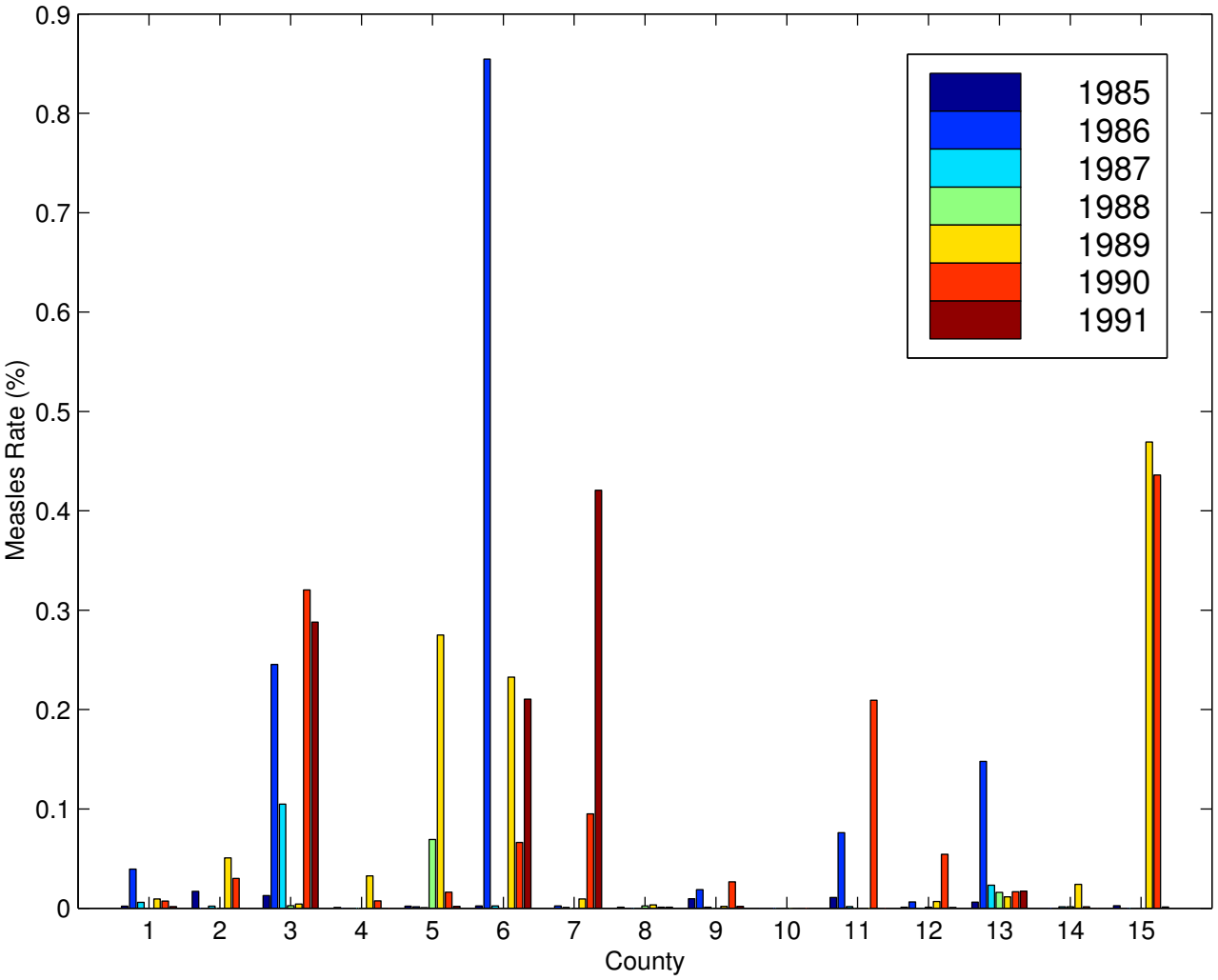

Figure 3.1: Texas measles data. Years are grouped together for each county 1985-1991 from left to right.

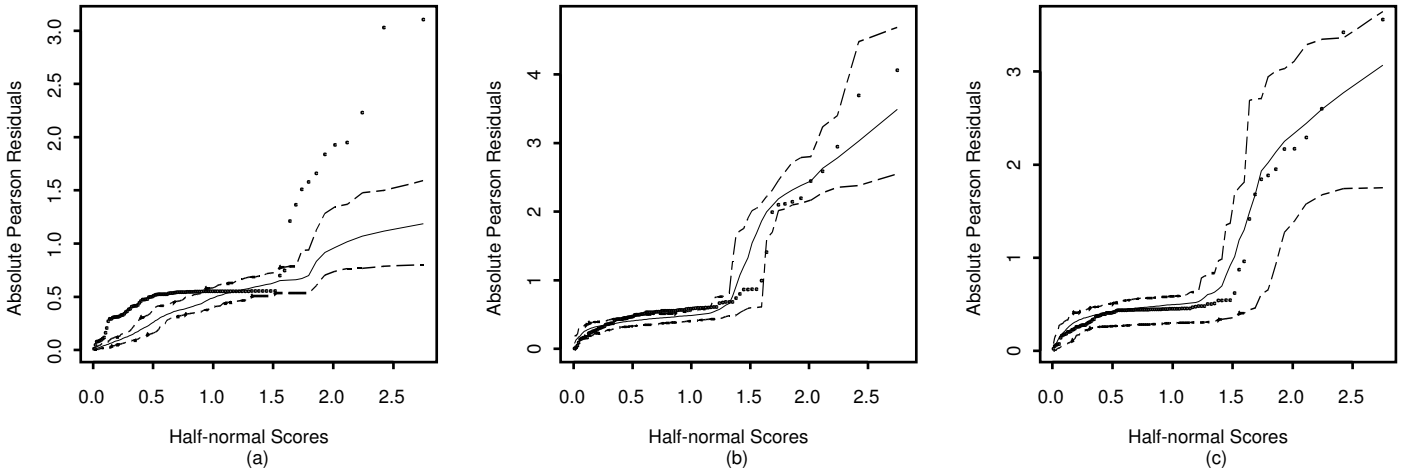

Figure 3.2: Half-normal plot for assessing goodness of fit of models 1 (Figure a), 3 (Figure b) and 5 (Figure c). Theses three models are a GLMM, two-component GLM, and two-component GLMM, respectively.

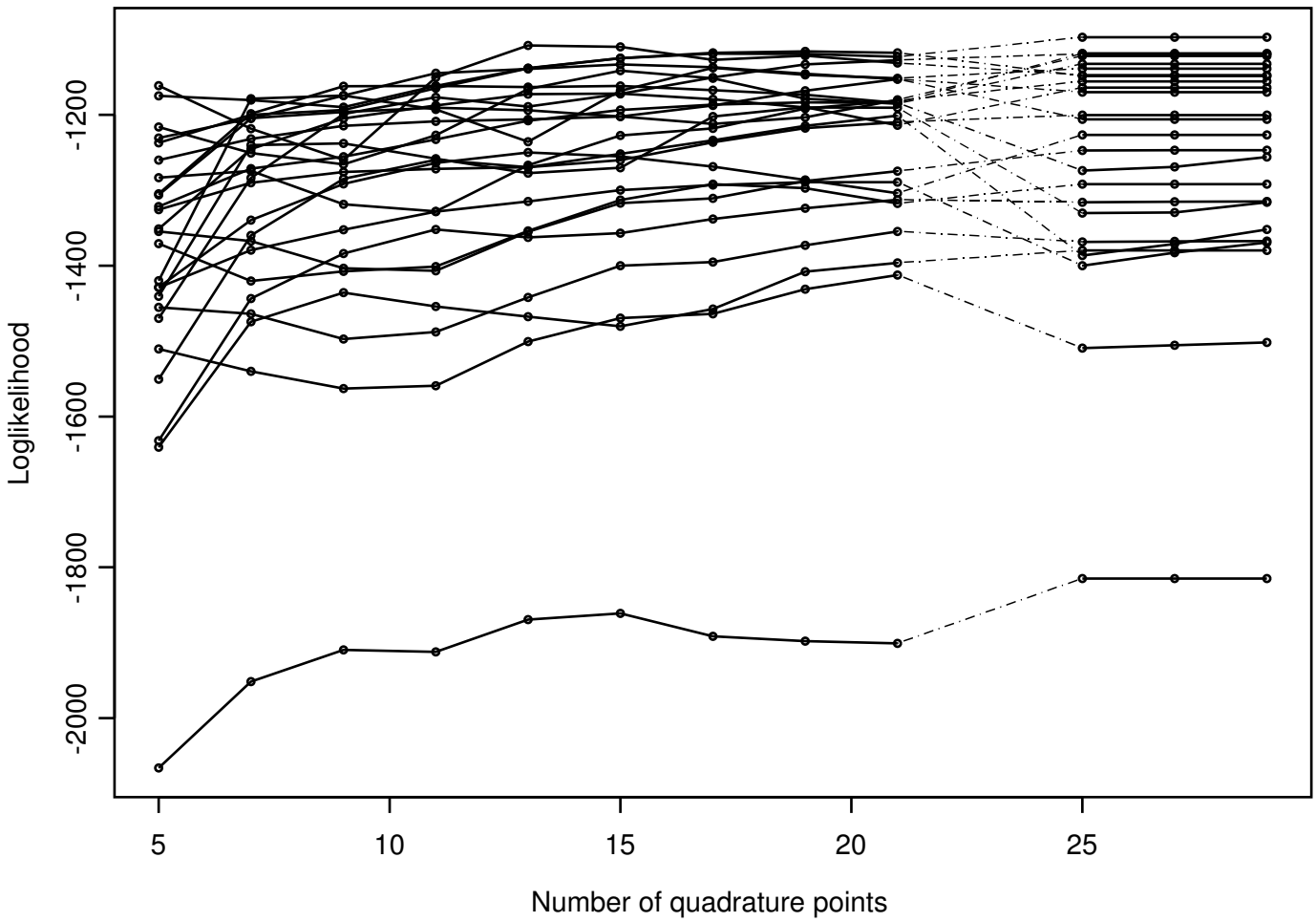

Figure 3.3: Loglikelihood as a function of the number of quadrature points m from 5 to 21 for ordinary Gaussian quadrature and $m + 20$ for adaptive Gaussian quadrature.

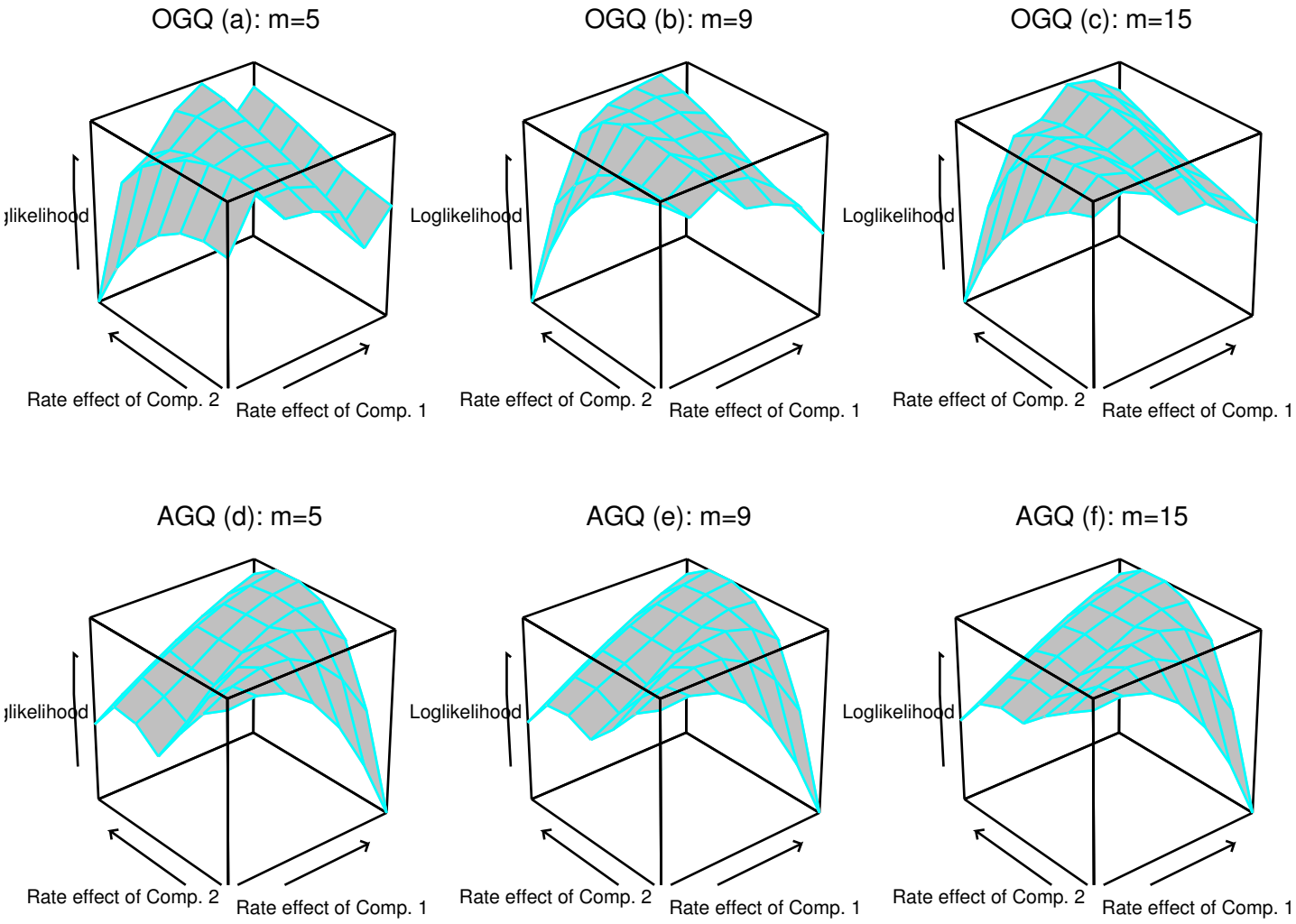

Figure 3.4: Surface plots for OGQ and AGQ approaches based on the measles data.

3.7 References

[1] Anderson, D. A. and Aitkin, M. (1985). Variance component models with binary response: interviewer variability. *Journal of the Royal Statistical Society. Series B (Methodological)* **47**, 203–210.

[2] Aitkin, M. and Wilson, G.T. (1980). Mixture models, outliers, and the EM algorithm . *Technometrics* **22**, 325–331.

[3] Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing* **6**, 251–262.

[4] Aitkin, M. (1999). A general maximum likelihood analysis of variance components in generalized linear models. *Biometrics* **55**, 117–128.

[5] Albert, P.S. and Follmann, D.A. (2000). Modelling repeated count data subject to informative dropout. *Biometrics* **56**, 667–677.

[6] Berk, K.N. and Lachenbruch, P.A. (2002). Repeated measures with zeros.

[7] Besag, J., York, J. and Mollie, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Ann. Inst. Statist. Math.* **43**, 1–59.

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

[9] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B* **39**, 1–38.

[10] Dietz, E. and Böhning, D. (1997). The use of two-component mixture models with one completely or partly known component. *Computational Statistics* **12**, 219–234.

[11] Everitt, B.S. and Hand, D.J. (1981). *Finite Mixture Distributions.* London: Chapman and Hall.

[12] Follmann, D. and Lambert, D. (1989). Generalizing logistic regression nonparametrically. *Journal of the American Statistical Association* **84,** 295–300.

[13] Friedl, H. and Kauermann, G. (2000). Standard errors for EM estimates in generalized linear models with random effects. *Biometrics* **56,** 761–767.

[14] Hall, D.B. (2000). Zero-inflated Poisson and binomial regression with random effects: a case study. *Biometrics* **56**, 1030–1039.

[15] Hasselblad, Victor(1966). Estimation of parameters for a mixture of normal distributions. *Technometrics* **8**, 431–444.

[16] Hasselblad, Victor(1969). Estimation of finite mixtures of distributions from the exponential family. *Journal of the American Statistical Association* **64,** 1459-1471.

[17] Heagerty, P.J. and Kurland, B.F. (2001). Misspecified maximum likelihood estimates and generalized linear mixed models. *Biometrika* **88**, 973–985.

[18] Hedeker, D. and Gibbons, R.D. (1994). A random-effects ordinal regression model for multilevel analysis. *Biometrics* **50**, 933–944.

[19] Hinde, J.(1982). Compound poisson regression models. In GLIM 82: *Proceedings of the international conference on generalized linear models.* R. Gilchrist(ed), 109–121. New York: Springer-Verlag.

[20] Hinde, J.P. and Wood, A.T.A. (1987). Binomial variance component models with a non-parametric assumption concerning random effects. In *Longitudinal Data Analysis,* R. Crouchley (ed.) Avebury, Aldershot, Hants.

[21] Ii, Y. and Raghunathan, T.E. (1991). Bayesian Analysis of series of 2×2 tables. unpublished manuscript submitted to Statistics in Medicine.

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

[23] Jansen, R.c.(1993). Maximum likelihood in a generalized linear finite mixture model by using the EM algorithm . *Biometrics* **49,** 227–231.

[24] Jiang, W and Tanner,M.A.(1999). On the identifiability of mixtures-of-experts. *Neural Networks* **12,** 1253–1258.

[25] Leroux, B.G. and Puterman, M.L. (1992). Maximum-penalized-likelihood estimation for independent and markov-dependent mixture models. *Biometrics* **48,** 545–558.

[26] Lesaffre, E. and Spiessens, B.(2001). On the effect of the number of quadrature points in a logistic random-effects model: an example. *Applied Statistics* **50,** 325–335.

[27] Lindsay, B.G. (1995). Mixture models: theory, geometry and applications. *The Institute of Mathematical Statistics and the American Statistical Association*.

[28] Lindstrom, M.J. and Bates, D.M.(1990). Nonlinear mixed effects models for repeated measurement data. *Biometrics* **46,** 673–687.

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

[30] McCulloch, C.E. (1997). Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Association* **92**, 162-170.

[31] McLachlan, G.J. and Basford, K.E. (1988). *Mixture Models: Inference and Applications to Clustering.* New York: Marcel Dekker.

[32] McLachlan, G.J. and Peel, D. (2000). *Finite Mixture Models.* New York: Wiley.

[33] Neuhaus, J.M., Hauck, W.W. and Kalbfleisch, J.D. (1992). The effects of mixture distribution misspecification when fitting mixed effects logistic models. *Biometrika* **79,** 755–762.

[34] Oakes, D. (1999). Direct calculation of the information matrix via the EM algorithm. *Journal of the Royal Statistical Society, Series B* **61,** 479–482.

[35] Olsen,M.K. and Schafer, J.L.. (2001). A two-part random-effects model for semicontinuous longitudinal data. *Journal of the American Statistical Association* **96,** 730-745.

[36] Park, J.G. and Basawa, I. V. (2002). Estimation for mixtures of Markov processes. *Statistics and Probability Letters* **59,** 235-244.

[37] Pinheiro, J.C. and Bates, D. M. (1995). Approximations to the loglikelihood function in the nonlinear mixed effects model. *Journal of Computational and Graphical Statistics* **4,** 12–35.

[38] Pinheiro, J.C. and Bates, D. M. (1996). Unconstrained parameterizations for variance-covariance matrices. *Statistics and Computing* **6,** 289–296.

[39] Rabe-Hesketh, S., and Skrondal, A and Pickles, A.(2002). Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal* **2,** 1–21.

[40] Rabe-Hesketh, S., Skrondal, A and Pickles, A.(2002). Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects.

[41] Ridout, M., Demétrio, C.G.B., and Hinde, J. (2001). A score test for testing a zero-inflated Poisson regression model against zero-inflated negative binomial alternatives. *Biometrics* In press.

[42] Rosen, O, Jiang, W.X. and Tanner, M.A. (2000). Mixtures of marginal models. *Biometrika* **87**, 391–404.

[43] Sherman, M. and le Cessie, S. (1997). A comparison between bootstrap methods and generalized estimating equations for correlated outcomes in generalized linear models. *Commun. Statist.-Simula.* **26**, 901–925.

[44] Steele, B.M. (1996). A modified EM algorithm for estimation in generalized mixed models. *Biometrics* **52**, 1295–1310.

[45] Thisted, R.A. (1988). *Elements of Statistical Computing: Numerical Computation.* Chapman and Hall, New York.

[46] Thompson, T.J., Smith, P.J. and Boyle, J.P. . (1998). Finite mixture models with concomitant information: assessing diagnostic criteria for diabetes . *Applied Statistics* **47,**, part 3 393–404.

[47] Titterington, D.M., Smith,A.F.M. and Makov, U.E.(1985). *Statistical Analysis of Finite Mixture Distributions. New York: Wiley.*

[48] van DUIJN, M.A.J. and Bockenholt, U. (1995). Mixture models for the analysis of repeated count data. *Applied Statistics*, **44,** 473–485.

[49] van Iersel, M., Oetting, R. and Hall, D.B. (2000). Imidicloprid applications by subirrigation for control of silverleaf whitefly on poinsettia. *Journal of Economic Entomology* **93,** 813–819.

[50] Vieira, A.M.C., Hinde, J.P., and Demétrio, C.G.B. (2000). Zero-inflated proportion data models applied to a biological control assay. *Journal of Applied Statistics* **27,** 373–389.

[51] Wang, P.,Cockburn,I.M., and Puterman, M.L. (1998). Analysis of patent data— A mixed-Poisson-regression-model approach. *Journal of Business and Economic Statistics* **16,** 27–41.

[52] Wang, P. and Puterman, M.L. (1998). Mixed logistic regression models. *Journal of Agricultural, Biological, and Environmental Statistics* **3,** 175–200.

[53] Wolfinger, R.D.(1993). Laplace's approximation for nonlinear mixed models. *Biometrika* **80,** 791–795.

[54] Wolfinger, R.D. and Lin,X. (1997). Two Taylor-series approximation methods for nonlinear mixed models. *Computional Statistics and Data Analysis* **25,** 465–490.

[55] Wolfinger, R.D. and O'Connell, M.(1993). Generalized linear models: A pseudo-likelihood approach. *Journal of statistical computation and simulation.* **48,** 233–243.

[56] Yau, K.W. and Lee, A.H. (2001). Zero-inflated poisson regression with random effects to evaluate an occupational injury prevention programme. *Statistics in medicine* **20,** 2907–2920.

[57] Zeger, S.L. and Karim, M.R. (1991). Generalized linear models with random effects; A Gibbs sampling approach. *Journal of the American Statistical Association* **86,** 79–86.

Chapter 4

Restricted Maximum Likelihood Method for ZI-mixed Effect Models

4.1 Introduction

Restricted Maximum Likelihood Method (REML) for linear models was originally formulated by Patterson and Thompson (1971) for estimating intra-block and inter-block weights in the analysis of incomplete block designs with block sizes not necessarily equal. They proposed a set of error contrasts whose likelihood function depends only on the variance components and not the regression parameter of the model. Their proposal was to maximize the likelihood function of those error contrasts rather than the likelihood of the data. This idea was generalized to the context of the linear mixed model by Corbeil and Searle (1976a). The purpose of REML in linear mixed effects models is to estimate variance components using a general likelihood-based methodology that leads to estimators with less bias than ML estimators. In special cases such as balanced ANOVA models, REML estimation leads to the classical unbiased ANOVA-type estimators which can be thought of bias-corrected MLEs where adjustments have been made to account for degrees of freedom lost in estimating regression parameters.

The merits of REML and ML estimators for variance-covariance components have been discussed by several authors (Patterson and Thompson, 1971; Harville, 1977; Diggle et al., 1994; Verbeke and Molenberghs, 2000). ML and REML estimation methods are both likelihood-based, and the estimators have useful properties such as consistency, asymptotic normality and efficiency. But in general, ML estimators do not adjust for the loss of degrees of freedom resulting from the estimation of the model's fixed effects and produce biased

estimators of the variance-covariance parameters. According to Diggle et al., "... the distinction between the maximum likelihood and REML estimation is important only when p [number of fixed effect parameters] is relatively large" and "In summary, maximum likelihood and REML estimators will often give very similar results. However, when they do differ substantially, REML estimators should be less biased." (Diggle et al., 1994, p.69).

The REML method has been extended to GLMMs by several authors. In the GLMM context, Drum and McCullagh (1993) apply REML to logistic mixed effect models. Breslow and Clayton (1993) proposed a REML-type adjustments for penalized quasi-likelihood. Another approximate REML estimation in GLMM setting is proposed by McGilchrist (1994). His method is based on hierarchical or h-likelihood (Lee and Nelder, 1996). However, the latter two approaches only apply to approximate ML methods for GLMMs, where the model is approximated by successive LMMs. These approaches are quite similar, as noted by McGilchrist (1994), who writes, "the approach [McGilchrist (1994)] is similar in principle to penalized likelihood approaches and in basic aims has elements in common with Breslow and Clayton [(1993)]." Actually, the REML methods employed by Breslow and Clayton and McGilchrist are not nuisance parameter elimination technique. Therefore, the natural questions to be asked are, How accurate are Breslow and Clayton's approximations which lead to their REML approach? How much information is lost? Is it possible to work on the loglikelihood from the nonlinear model directly and still apply REML method to get better estimate of variance component? Liao and Lipsitz (2002) proposed a REML-type estimator for GLMM model based upon correcting the bias in the profile score function of the variance components, an idea by McCullagh and Tibshirani (1990). The idea is more in the spirit of REML estimation in LMM because, like REML but unlike the other methods, Liao and Lipsitz's approach is based upon reducing or eliminating the effect of the nuisance parameters (the fixed effects) in the estimator of the variance component.

Currently, variance-covariance parameters in ZI-mixed effect models have been estimated by ML approach (Hall, 2000) and approximate REML approach (Yau and Lee, 2001) based

on McGilchrist (1994). We think Liao and Lipsitz approach to REML can be extended to ZI-mixed effects models, and expect it to improve upon ML estimation and Yau and Lee's approximate REML in that context.

Section 4.2 presents the proposed REML-like estimator for ZI-mixed effect models. The fitting algorithm is provided in section 4.3. In section 4.4, a simulation study is performed to compare ML and REML estimators for variance components. Standard errors are discussed in section 4.5. In section 4.6, a real data analysis is given and at the end, a discussion is given in section 4.7.

4.2 REML Estimator for ZI-Mixed Effect Models

A special case of the two-component mixture occurs when one component is a degenerate distribution with point mass of one at zero. Such models are known as zero-inflated regression models and include zero-inflated Poisson (ZIP; Lambert, 1992), negative binomial, binomial (ZIB; Hall, 2000) and others (see Ridout, et al., 1998 for a review). Recently, Hall (2000) and Yau and Lee (2001) considered ZI-Poisson model with cluster-specific random effects. Hall (2000) also considered ZI-Binomial model with random effects. Zero-inflated regression models with random effects for continuous data have been considered by Olsen and Schafer (2001) and Berk and Lachenbruch (2002).

4.2.1 Formulation of ZI-Mixed Effect Models

The ZI-mixed effect models can be expressed as

$$Y_{ij}|\boldsymbol{b}_i \sim \begin{cases} 0, & \text{with probability } p_{ij}; \\ F_1(y_{ij}|\boldsymbol{b}_i;\zeta_{ij},\sigma), & \text{with probability } 1-p_{ij}. \end{cases}$$

Here, F_1 is assumed to be an exponential dispersion family distribution, with density

$$f_1(y_{ij}|\boldsymbol{b}_i;\zeta_{ij},\sigma) = h(y_{ij},\sigma)\exp[\{\zeta_{ij}y_{ij} - \kappa(\zeta_{ij})\}w_{ij}/\sigma].$$

Usually, F_1 is Poisson or binomial, but other distributions can and have been considered in the literature (e.g. negative binomial). As before, the w_{ij}'s are known constants (e.g., binomial

denominators), σ is the dispersion parameter. The function κ is a cumulant generating function, so F has means $\mu_{ij} = \kappa'(\zeta_{ij})$ and variances $v(\mu_{ij})\sigma/w_{ij}$ where $v(\mu) = \kappa''(\mu)$, is the variance function.

We assume the canonical parameters $\boldsymbol{\zeta}_i = (\zeta_{i1}, \ldots, \zeta_{it_i})^T$ are related to covariates and cluster-specific random effects through GLM-type specifications. That is, for canonical link function we have

$$\boldsymbol{\zeta}_i(\boldsymbol{\mu}_i) = \boldsymbol{\eta}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{U}_i\boldsymbol{b}_i, \quad \text{or} \quad \boldsymbol{\mu}_i = \boldsymbol{\zeta}_i^{-1}(\boldsymbol{\eta}_i).$$

Here, $\boldsymbol{X}_i$ is a $t_i \times r$ design matrix for fixed effects parameters $\boldsymbol{\beta}$; $\boldsymbol{U}_i$ is a $t_i \times q$ design matrix for the random effects $\boldsymbol{b}_i$; $\boldsymbol{b}_1 \ldots, \boldsymbol{b}_K$ are assumed to be independent, each with mean $\mathbf{0}$ and variance-covariance matrix $\boldsymbol{V}_q$. We assume $\boldsymbol{V}_q$ is parameterized by the vector $\boldsymbol{\theta}$. In addition, we assume the structure for mixing probabilities is $g_p(\boldsymbol{p}_i) = \boldsymbol{W}_i\boldsymbol{\gamma}$, where $\boldsymbol{p}_i = (p_{i1}, \ldots, p_{it_i})^T$, $i = 1, \ldots, K$, $\boldsymbol{\gamma}$ is an unknown $s-$dimensional regression parameter, $\boldsymbol{W}_i$ is a $t_i \times s$ design matrix, and g_p is a known link function. Typically, g_p will be taken to be the logit link, but the probit, complementary-log-log, or other link function can be chosen here.

Let $\boldsymbol{\delta} = (\boldsymbol{\beta}^T, \sigma, \boldsymbol{\gamma}^T)^T$ denote the fixed effect parameter vector, $\boldsymbol{\theta}$ denote the variance component parameter vector of random effects, and $\boldsymbol{\delta}^c = (\boldsymbol{\beta}^T, \sigma, \boldsymbol{\gamma}^T, \boldsymbol{\theta}^T)^T$ denote the combined vector of model parameters. If we assume $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K$ are independent $N(\mathbf{0}, \boldsymbol{V}_q(\boldsymbol{\theta}))$ random vectors, then the loglikelihood for $\boldsymbol{\delta}^c$ based on $\boldsymbol{y}$ is given by

$$\ell(\boldsymbol{\delta}^c; \boldsymbol{y}) = \sum_{i=1}^{K} \log \left\{ \int \prod_{j=1}^{t_i} f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}) \phi_q(\boldsymbol{b}_i; \boldsymbol{\theta}) d\boldsymbol{b}_i \right\}, \tag{4.2.1}$$

where $f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}) = \{p_{ij}(\boldsymbol{\gamma}) + (1 - p_{ij}(\boldsymbol{\gamma}))f_1(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\beta}, \sigma)\}^{z_{ij}} \{(1 - p_{ij}(\boldsymbol{\gamma}))f_1(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\beta}, \sigma)\}^{1-z_{ij}}$, and $z_{ij} = 1$ if $y_{ij} = 0$, otherwise $z_{ij} = 0$. In addition, $\phi_q(\cdot)$ denotes the $q-$dimensional normal density function, and the integral is $q-$dimensional over $(-\infty, \infty) \times \cdots \times (-\infty, \infty)$ (q times).

4.2.2 Complete Data Loglikelihood for EM Algorithm

The EM algorithm is convenient for fitting ZI-mixed effects models (Hall, 2000). Define the Bernoulli random variable $U_{ij} = 1$ if Y_{ij} is drawn from zero state, $U_{ij} = 0$ if Y_{ij} is drawn

from distribution F_1, where $i = 1, \ldots, K, j = 1, \ldots, t_i$. Then the "complete" data for the EM algorithm are $(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b})$, where $\boldsymbol{u} = (u_{11}, \ldots, u_{Kt_K})^T$ contain the realizations of the U_{ij}'s. Here, $(\boldsymbol{u}, \boldsymbol{b})$ play the role of missing data. Based on $(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b})$, the complete data loglikelihood is given by

$$\begin{aligned}
\ell^c(\boldsymbol{\delta}^c; \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}) &= \log f(\boldsymbol{y}|\boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta}^c) + \log f(\boldsymbol{u}|\boldsymbol{b}; \boldsymbol{\delta}^c) + \log \phi_q(\boldsymbol{b}) \\
&= \sum_{i=1}^{K} \sum_{j=1}^{t_i} (1 - u_{ij}) \left(\log f_1(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\beta}, \sigma)\right) \\
&+ \sum_{i=1}^{K} \sum_{j=1}^{t_i} \{u_{ij} \log p_{ij}(\boldsymbol{\gamma}) + (1 - u_{ij}) \log[1 - p_{ij}(\boldsymbol{\gamma})]\} \\
&+ \sum_{i=1}^{K} \log \phi_q(\boldsymbol{b}_i; \boldsymbol{\theta}). \qquad (4.2.2)
\end{aligned}$$

In the current literature, Hall (2000) obtains the parameter estimates of ZIP and ZIB mixed models by maximum likelihood method via the EM algorithm. Given $\boldsymbol{\theta}$, Yau and Lee (2001) use the Newton-Raphson algorithm to iteratively maximize the joint loglikelihood of $\boldsymbol{y}$ and $\boldsymbol{b}$, while the estimate of $\boldsymbol{\theta}$ is obtained by modifying REML estimation equation (3.2) and the REML information matrix of McGilchrist (1994). Alternatively, the REML estimation method proposed by Liao and Lipsitz (2002) uses the Monte Carlo EM algorithm (MCEM) to get fixed effect parameter estimates given the variance component parameter $\boldsymbol{\theta}$, and then $\boldsymbol{\theta}$ is obtained by iteratively solving a bias-corrected profile score function. Our estimation approach for ZI-mixed model is based on their work.

4.2.3 Definition of REML estimator of Variance Components

Following Liao and Lipsitz (2002) and our ZI-mixed effect model description above, let $\boldsymbol{\delta} = (\boldsymbol{\beta}^T, \sigma, \boldsymbol{\gamma}^T)^T$ be the vector of fixed effect parameters, let $\boldsymbol{\delta}^c = (\boldsymbol{\delta}^T, \boldsymbol{\theta}^T)^T$, let $(\hat{\boldsymbol{\theta}}_{MLE}, \hat{\boldsymbol{\delta}}_{MLE})$ be the ML estimator of variance component and fixed effects parameters respectively, let $\hat{\boldsymbol{\theta}}^y_{\boldsymbol{\delta}}$ be the ML estimator of $\boldsymbol{\theta}$ for known $\boldsymbol{\delta}$. The "y" superscript here denotes that this quantity is based on the observed data y rather than generated data as described below. In order to

develop a REML estimator $\hat{\boldsymbol{\theta}}_{REML}$ that has smaller bias than $\hat{\boldsymbol{\theta}}_{MLE}$, we need to compare the profile score function for $\hat{\boldsymbol{\theta}}_{MLE}$ with the score function for $\hat{\boldsymbol{\theta}}^{y}_{\boldsymbol{\delta}}$.

Score Function for $\hat{\boldsymbol{\theta}}^{y}_{\boldsymbol{\delta}}$:

The observed data loglikelihood of the ZI-mixed effect model is given in equation (4.2.1). Suppose the fixed effects parameters $\boldsymbol{\delta}$ are known, then the maximum likelihood estimate of the variance component parameter $\boldsymbol{\theta}$ is obtained by solving $\frac{\partial \ell(\boldsymbol{\delta}^c;\boldsymbol{y})}{\partial \boldsymbol{\theta}} = \mathbf{0}$, where

$$\frac{\partial \ell(\boldsymbol{\delta}^c;\boldsymbol{y})}{\partial \boldsymbol{\theta}} = -\frac{1}{2}\sum_{i=1}^{K}\frac{\partial}{\partial \boldsymbol{\theta}}\left\{\log|\boldsymbol{V}_{\boldsymbol{\theta}}| + \operatorname{tr}\left[\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta})\right]\right\}. \tag{4.2.3}$$

This quantity can be obtained from the EM algorithm based on the complete data loglikelihood (4.2.2). Notice that only the last term in equation (4.2.2) involves $\boldsymbol{\theta}$. Hence, given $\boldsymbol{\delta}$ known, maximizing $Q(\boldsymbol{\delta}^c|\boldsymbol{\delta}^{c(h)})$ is equivalent to maximizing the expectation of the last term of equation (4.2.2) with respect to $\boldsymbol{\theta}$. Thus, the score function for $\boldsymbol{\theta}$ based on the EM algorithm is

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(h)})}{\partial \boldsymbol{\theta}} = -\frac{1}{2}\sum_{i=1}^{K}\frac{\partial}{\partial \boldsymbol{\theta}}\left\{\log|\boldsymbol{V}_{\boldsymbol{\theta}}| + \operatorname{tr}\left[\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta}^{(h)})\right]\right\}.$$

It follows that

$$\frac{\partial \ell(\boldsymbol{\delta};\boldsymbol{y})}{\partial \boldsymbol{\theta}} = \frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(h)})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^{(h)}=\boldsymbol{\theta}}.$$

Profile Score Function for $\hat{\boldsymbol{\theta}}_{MLE}$:

Usually, we don't know $\boldsymbol{\delta}$, but must estimate $\boldsymbol{\delta}$ by the ML estimator $\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}}$ for fixed $\boldsymbol{\theta}$. We name the score function obtained by plugging the ML estimator $\hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}$ into the observed data loglikelihood as the profile score function of the variance component $\boldsymbol{\theta}$. For the ZI-mixed model, the profile score function for $\boldsymbol{\theta}$ is

$$ps(\boldsymbol{\theta};\boldsymbol{y}) = \frac{\partial \ell(\boldsymbol{\delta}^c;\boldsymbol{y})}{\partial \boldsymbol{\theta}} = -\frac{1}{2}\sum_{i=1}^{K}\frac{\partial}{\partial \boldsymbol{\theta}}\left\{\log|\boldsymbol{V}_{\boldsymbol{\theta}}| + \operatorname{tr}\left[\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}},\boldsymbol{\theta})\right]\right\}. \tag{4.2.4}$$

Bias Correction:

Comparing equation (4.2.3) with (4.2.4), we notice that the only difference lies in the expectation terms in those equations. Define

$$h(\boldsymbol{\theta},\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{K}\frac{\partial}{\partial \boldsymbol{\theta}}\left\{\log|\boldsymbol{V}_{\boldsymbol{\theta}}| + \operatorname{tr}\left[\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{S}\right]\right\},$$

where $\boldsymbol{S} = E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta})$ in equation (4.2.3) and $\boldsymbol{S} = E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}},\boldsymbol{\theta})$ in equation (4.2.4). Let θ_r be the rth element of $\boldsymbol{\theta}$; then the rth element of $h(\boldsymbol{\theta},\boldsymbol{S})$ is

$$-\frac{1}{2}\sum_{i=1}^{K}\mathrm{tr}\left\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\left[\boldsymbol{S}-\boldsymbol{V}_{\boldsymbol{\theta}}\right]\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\frac{\partial\boldsymbol{V}_{\boldsymbol{\theta}}}{\partial\theta_r}\right\} \tag{4.2.5}$$

(see Jennrich and Schuchter, 1986, p809). If $\boldsymbol{S} = E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta})$, (4.2.5) is an unbiased estimating function for $\boldsymbol{\theta}$ because

$$E_{\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta}}\left\{E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta})\right\} = E(\boldsymbol{b}_i\boldsymbol{b}_i^T;\boldsymbol{\delta},\boldsymbol{\theta}) = \boldsymbol{V}_{\boldsymbol{\theta}}.$$

But if $\boldsymbol{S} = E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}},\boldsymbol{\theta})$, (4.5) is biased, with bias equal to

$$B(\boldsymbol{\theta},\boldsymbol{\delta}) \;=\; 1/K\sum_{i=1}^{K}\left\{E_{\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta}}\left[E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}},\boldsymbol{\theta})\right] - E_{\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta}}\left[E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta})\right]\right\}. \tag{4.2.6}$$

In practice, we don't know $\boldsymbol{\delta}$ in equation (4.2.6). Following Liao and Lipsitz, we substitute $\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}}$ for $\boldsymbol{\delta}$ and define our estimated bias as $B(\boldsymbol{\theta},\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$. Correcting this bias in the profile score function (4.2.4) and solving the resulting equation, we obtain the REML type estimator $\hat{\boldsymbol{\theta}}_{REML}$. In this approach, $E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}},\boldsymbol{\theta})$ in equation (4.2.4) has been replaced by $E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}},\boldsymbol{\theta}) - B(\boldsymbol{\theta},\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$.

With the replacement of $\boldsymbol{\delta}$ by $\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}}$, we no longer have an unbiased estimating function for $\boldsymbol{\theta}$. However, Liao and Lipsitz (2002) have argued that "the dependence of [$B(\boldsymbol{\theta},\boldsymbol{\delta})$] on [$\boldsymbol{\delta}$] should be weak and the difference between [$B(\boldsymbol{\theta},\boldsymbol{\delta})$] and [$B(\boldsymbol{\theta},\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$] should thus be small." We investigate the validity of this claim in the context of our problem in section 4.5.

4.3 The Algorithm for REML Estimator of Variance Components

Following Liao and Lipsitz (2002), the fitting algorithm we present involves the MCEM algorithm using importance sampling (Booth and Hobert, 1999). Based on (4.2.2), the E-step of the EM algorithm is approximated in three parts which allows the maximization to be done by solving three separate problems: (1) maximizing a weighted binomial loglikelihood

involving $\boldsymbol{\gamma}$ only; (2) maximizing a weighted exponential dispersion family loglikelihood involving $\boldsymbol{\alpha}$ and σ only; (3) and solving an estimating equation involving $\boldsymbol{\theta}$ only.

The complete algorithm for computing the REML estimator is as follows:

1. Given $\boldsymbol{\theta} = \boldsymbol{\theta}^{(h)}$ and data $\boldsymbol{y}$, (1) and (2) are used to get the ML estimator $\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta} = \boldsymbol{\theta}^{(h)}$. $E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}},\boldsymbol{\theta})$ is obtained as a by-product of the MCEM algorithm (see Appendix for the derivation of MCEM algorithm for ZI-inflated model), because we have already drawn random variates from the conditional distribution of $\boldsymbol{b}_i$ given $\boldsymbol{y}_i$.

2. Generate a random sample $\boldsymbol{Y}^{(h)}$ that has the same dimension as the observed data vector $\boldsymbol{y}$ from the ZI-mixed model with parameter $\boldsymbol{\theta} = \boldsymbol{\theta}^{(h)}$ and $\boldsymbol{\delta} = \hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}}$.

3. Using (1) and (2) and taking $\boldsymbol{\theta} = \boldsymbol{\theta}^{(h)}$ as known parameters, obtain fixed effect parameter estimator $\hat{\boldsymbol{\delta}}^{Y}_{\boldsymbol{\theta}}$. Again, $E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{Y}_i;\hat{\boldsymbol{\delta}}^{Y}_{\boldsymbol{\theta}},\boldsymbol{\theta})$ and $E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{Y}_i;\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}},\boldsymbol{\theta})$ are by-products of the MCEM algorithm, because we have already drawn random variates from the conditional distribution of $\boldsymbol{b}_i$ given $\boldsymbol{Y}_i$. Note, for simplification, $\boldsymbol{Y} = \boldsymbol{Y}^{(h)}$ and $\boldsymbol{Y}_i = \boldsymbol{Y}_i^{(h)}$ in this step.

4. As in Liao and Lipsitz (2002, p.405), the bias is calculated by

$$B_{h+1} = (1 - h^{-1})B_h + h^{-1}\left\{\frac{1}{K}\sum_{i=1}^{K}\left[E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{Y}_i;\hat{\boldsymbol{\delta}}^{Y}_{\boldsymbol{\theta}},\boldsymbol{\theta}) - E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{Y}_i;\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}},\boldsymbol{\theta})\right]\right\},$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}^{(h)}$, $\boldsymbol{Y} = \boldsymbol{Y}_h$ and $B_0 = 0$.

5. Correct the bias in the profile score function and obtain the $(h+1)th$ iteration of $\boldsymbol{\theta}$ ($\boldsymbol{\theta}^{(h+1)}$) by solving

$$h\left(\boldsymbol{\theta}, E(\boldsymbol{b}_i\boldsymbol{b}_i^T|\boldsymbol{y}_i;\hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}},\boldsymbol{\theta}) - B_{h+1}\right) = \mathbf{0}.$$

Repeat step 1-5 until convergence of $\boldsymbol{\theta}$, which yields $\hat{\boldsymbol{\theta}}_{REML}$. By conducting one more step, we obtain the REML estimates of the fixed effect parameters $\boldsymbol{\delta}$.

4.4 Inference for Fixed Effect Parameters

Inference for fixed effect parameters is a very challenging topic for GLMM models, mixtures of GLMM models and ZI-inflated models. As is well know, as we know, likelihood based-inference on the fixed effects can be done based on the Hessian matrix. In small samples,

this may not be the best way to do the inference, but it is a reasonable first choice. In our context, the 5 steps of the algorithm presented in section 4.3 performed iteratively until convergence of $\boldsymbol{\theta}$. After convergence, step 1 is performed again to obtain the REML estimate of the fixed effect parameter $\boldsymbol{\delta}$ (refer to the first two terms of formula A.3 in the Appendix. At convergence of this step, the Hessian matrix of fixed effect parameters is a by product, which is the second derivative of the first two terms of formula (A.3) in the Appendix. Usual Wald-based inference for the fixed effects can be based on the negative inverse of this matrix. Of course, as in linear mixed effect models, this asymptotic variance-covariance estimator does not take any account of the error introduced by having to estimate $\boldsymbol{\theta}$, and the resulting inferences may be poor in small samples. A worthwhile area of future research would be to improve upon asymptotic inference methods for ZI-mixed models fit with ML or our proposed REML-like procedure. It may be possible to produce small sample F tests in a manner similar to what Kenward and Roger (1997) proposed in the linear mixed model context. Inference for fixed effect can based on this.

4.5 Simulation Study

In order to compare the ML estimator ($\hat{\boldsymbol{\theta}}_{ML}$) and the proposed REML type estimator ($\hat{\boldsymbol{\theta}}_{REML}$) presented in section 4.2, a simulation study was carried out. In this study, we simulated data sets from a ZI-mixed Poisson distribution. The study design is adapted from Breslow and Clayton (1993) and also from Liao and Lipsitz (2002).

More specifically, the model we use in the simulation study is:

$$Y_{ij}|\boldsymbol{b}_i \sim \begin{cases} 0, & \text{with probability } p_{ij}; \\ \text{Poisson}(\lambda_{ij}|\boldsymbol{b}_i), & \text{with probability } 1-p_{ij}, \end{cases}$$

where

$$\log \lambda_{ij} = \beta_1 + \beta_2 x_{1ij} + \beta_3 x_{2ij} + b_{1i} + b_{2i} x_{1ij},$$

$$p_{ij} = \gamma_1 + \gamma_2 x_{1ij} + \gamma_3 x_{2ij},$$

$$\boldsymbol{b}_i = (b_{1i}, b_{2i})^T \sim N(\mathbf{0}, \boldsymbol{V}_2(\boldsymbol{\theta})),$$

and where $\boldsymbol{V}_2(\boldsymbol{\theta})$ is the diagonal matrix $\begin{bmatrix} \theta_1^2 & 0 \\ 0 & \theta_2^2 \end{bmatrix}$.

We let $i = 1, \ldots, 50$ index 50 independent clusters; $j = 1, \ldots, 10$ index 10 subjects within each cluster, and we let the true parameters be $(\beta_1, \beta_2, \beta_3) = (2, 1, -1)$, $(\gamma_1, \gamma_2, \gamma_3) = (-0.5, -0.5, 0.1)$, and $(\theta_1, \theta_2) = (0.5, 0.5)$. In addition, we assume $x_{1ij} = (j-5)/4$, $x_{2ij} = 0$ for $i = 1, \ldots, 25$ and $x_{2ij} = 1$ for $i = 26, \ldots, 50$. We fit two models: the first model (model 1) is the model we use to generate the data. The second model is an overspecified model with 3 extra covariates x_{3ij}, x_{4ij}, x_{5ij}. We assume these covariates are independent and are generated from separate standard normal distributions.

Two hundred and forty data sets were generated. Due to computing time demands, 240 data sets are use to compute $\hat{\boldsymbol{\theta}}_{\boldsymbol{\delta}}^{y}$, but only 100 data sets are used to compute $\hat{\boldsymbol{\theta}}_{ML}$, $\hat{\boldsymbol{\theta}}_{REML}$ and $\hat{\boldsymbol{\theta}}_{BLUP}$. Here, $\hat{\boldsymbol{\theta}}_{BLUP}$ represents the estimator obtained from the approximate REML method proposed by McGilchrist (1994) and applied by Yau and Lee (2001).

The simulation results are presented in Table 4.1 and Table 4.2. In Table 4.1, we compare ML and REML estimates of $\boldsymbol{\theta}$ with the true parameter values of $\boldsymbol{\theta}$ when the fixed effects parameters are known. In Table 4.2, we compare ML and REML estimates of $\boldsymbol{\theta}$ with the estimated $\boldsymbol{\theta}$ ($\hat{\boldsymbol{\theta}}_{\boldsymbol{\delta}}^{y}$). From Table 4.1, several conclusions can be drawn:

(1) $\hat{\boldsymbol{\theta}}_{\boldsymbol{\delta}}^{y}$ are essentially unbiased and the standard deviation is small. Note that the estimated $\hat{\boldsymbol{\theta}}_{\boldsymbol{\delta}}^{y}$ are the same for models 1 and 2 since they are estimated assuming fixed effect parameters are known.

(2) As expected, the ML estimator of the variance component parameters are biased downward. The bias is very severe for θ_1 for both of the models. In addition, when the number of fixed effect parameters increase, the bias becomes more severe for model 2 than for model 1.

(3) As expected, the REML method presented in section 4.3 effectively reduces the bias. On average, it reduces about 38 percent of the bias relative to the ML estimator of θ_1, 18 percent of the bias relative to the ML estimator of θ_2 for model 1, and 40 percent and 21

Table 4.1: Simulation results for Zero-Inflated Poisson data with two dimensional random effects: compare with the true parameter value of $\boldsymbol{\theta}$

Statistics	Parameter	Simulation size	Model 1		Model 2	
			Mean	Standard deviation	Mean	Standard deviation
$\hat{\boldsymbol{\theta}}^y_{\boldsymbol{\delta}}$	θ_1	240	0.49097	0.06254	0.49097	0.06254
	θ_2	240	0.48925	0.067157	0.48925	0.06715
$\hat{\boldsymbol{\theta}}_{ML}$-0.5	θ_1	100	-0.02641	0.06733	-0.02645	0.06691
	θ_2	100	-0.02454	0.06886	-0.02557	0.06972
$\hat{\boldsymbol{\theta}}_{REML}$-0.5	θ_1	100	-0.01630	0.06748	-0.01586	0.06716
	θ_2	100	-0.02009	0.06959	-0.02019	0.07019
$\hat{\boldsymbol{\theta}}_{BLUP}$-0.5	θ_1	100	-0.02493	0.06792	-0.02469	0.06748
	θ_2	100	-0.02411	0.06903	-0.02461	0.06983

percent for θ_1 and θ_2, respectively for model 2. We would expect that as the number of fixed effects increases, the practical effect of reducing the bias via REML estimation will increase.

(4) Comparing the approximate REML estimation method of Yau and Lee (2001) with ML, the estimators from the former method have smaller bias, but not nearly as small as our proposed method has. On average, the Yau and Lee approach reduces the bias by about 5.6 percent relative to ML for θ_1 and by 1.8 percent for θ_2 in model 1, and by 6.7 and 3.8 percent for θ_1 and θ_2, respectively, for model 2.

Similar conclusions can be drawn from Table 4.2.

In order to verify that the dependence of the bias defined in (4.2.6) on $\boldsymbol{\delta}$ is weak, and the difference between $B(\boldsymbol{\theta}, \boldsymbol{\delta})$ and $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$ is small, we add one more step in our simulation work to compute the value of $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$ for 100 simulated data sets. We think small variability in $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$ resulting from the variation of $\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}}$ from those simulated data sets will confirm our belief. In another words, we are trying to demonstrate that $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$ has no or little dependence on $\hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}}$. The results are shown in Table 4.3.

Table 4.2: Simulation results for Zero-Inflated Poisson data with two dimensional random effects

Statistics	Parameter	Simulation size	Model 1		Model 2	
			Mean	Standard deviation	Mean	Standard deviation
$\hat{\boldsymbol{\theta}}^{y}_{\boldsymbol{\delta}}$	θ_1	240	0.49097	0.06254	0.49097	0.06254
	θ_2	240	0.48925	0.067157	0.48925	0.06715
$\hat{\boldsymbol{\theta}}_{ML}$-$\hat{\boldsymbol{\theta}}^{y}_{\boldsymbol{\delta}}$	θ_1	100	-0.01477	0.01993	-0.01482	0.02083
	θ_2	100	-0.00710	0.01199	-0.00813	0.01228
$\hat{\boldsymbol{\theta}}_{REML}$-$\hat{\boldsymbol{\theta}}^{y}_{\boldsymbol{\delta}}$	θ_1	100	-0.00466	0.01989	-0.00423	0.02094
	θ_2	100	-0.00266	0.01327	-0.00276	0.01346
$\hat{\boldsymbol{\theta}}_{BLUP}$-$\hat{\boldsymbol{\theta}}^{y}_{\boldsymbol{\delta}}$	θ_1	100	-0.01329	0.02170	-0.01306	0.02256
	θ_2	100	-0.00668	0.01287	-0.00718	0.01343

Table 4.3: Calculation of $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}})$ using model 1 and model 2

Elements of $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^{y}_{\boldsymbol{\theta}})$	Model 1		Model 2	
	Mean	Standard deviation	Mean	Standard deviation
(1,1)	-0.00828	0.01183	-0.00835	0.01216
(1,2), (2,1)	-0.00179	0.00804	-0.00194	0.00801
(2,2)	-0.00362	0.00646	-0.00406	0.00649

From Table 4.3 we see that the means of $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$ (the average magnitude of bias correction) are reasonably small comparing to the magnitude of the variance component parameter estimates. But compared with the means of $B(\boldsymbol{\theta}, \hat{\boldsymbol{\delta}}^y_{\boldsymbol{\theta}})$, the corresponding standard deviations are not as small as expected. The reason for large standard deviations relative to the means may be due to inherent variability in the random generation of samples from the model relative to the computational accuracy of the numerical expectations taken in the E step. Despite this somewhat surprising finding, the simulation results of Tables 4.1 and 4.2 suggest that the proposed REML approach works very well.

The algorithm to implement the proposed method of estimation is not difficult to implement, but computing time can be long. We implemented all four estimation methods using FORTRAN90 programs. In the simulation study it took approximately one and half hours to fit model 1 to a single data set using the proposed REML approach.

4.6 Example–Whitefly Data

As mentioned in Chapter 3, the whitefly data are discussed in more detail in van Iersel, Oetting, and Hall (2000). In that paper, ZIB regression models were used to analyze the data, with random effects at the plant level to account for correlation among the repeated measures on a given plant. We return to this problem to fit a ZIB-mixed model for these data using the REML method we have developed here.

Let $y_{ijk\ell}$ be the number of live adult whiteflies on plant k ($k = 1, \ldots, 54$) in treatment i ($i = 1, \ldots, 6$) in block j ($j = 1, \ldots, 3$) measured at time ℓ ($\ell = 1, \ldots, 12$). Let $n_{ijk\ell}$ be the total number of whiteflies placed on the leaf of plant k in treatment i in block j measured at time ℓ. Further let β_{2i} be the ith treatment effect, β_{3j} be the jth block effect, $\beta_{4\ell}$ be the ℓth week effect, and b_k be a 1-dimensional random plant effect for plant k. For simplicity, we consider a model containing only main effects (treatment, block and week). The ZI-binomial model for these data with main effects can be expressed as

$$Y_{ijk\ell}|b_k \sim \{p_{ijk\ell} + (1 - p_{ijk\ell})(1 - \pi_{ijk\ell})^{n_{ijk\ell}}\}^{u_{ijk\ell}} \{(1 - p_{ijk\ell})\text{Binomial}(n_{ijk\ell}, \pi_{ijk\ell}|b_k)\}^{(1-u_{ijk\ell})},$$

where

$$u_{ijk\ell} = \begin{cases} 1, & \text{if } y_{ijk\ell} = 0 \\ 0, & \text{if } y_{ijk\ell} = 1, \end{cases}$$

and

$$\begin{aligned} \text{logit}(\pi_{ijk\ell}) &= \beta_1 + \beta_{2i}\text{treatment}_i + \beta_{3j}\text{block}_j + \beta_{4\ell}\text{week}_\ell + \sigma b_k \\ \text{logit}(p_{ijk\ell}) &= \gamma_1 + \gamma_{2i}\text{treatment}_i + \gamma_{3j}\text{block}_j + \gamma_{4\ell}\text{week}_\ell. \end{aligned}$$

The estimates of the fixed effect parameters and the variance component parameter σ are given in Table 4.4. It can be seen that the fixed effect parameter estimates are different but similar in magnitude for the ML and REML estimation methods. In addition, the variance component parameter σ has been adjusted upward a small amount by the REML method.

4.7 Discussion

Although this REML method is computationally intensive, the improvement in the estimator is impressive. Encouraged by this and since the ZI-mixed effect models are special cases of two-component GLMMs, a natural extension of REML to two-component GLMMs context appears worthwhile. We did some work on this topic, but it turned out to be difficult to apply REML in this context due to the shared random effects in both components (further discussions of this issue are included in Chapter 5). In addition, the proposed method of inference is based on large sample asymptotic properties, and improved methods for finite samples are desirable. In this research area, Kenward and Roger (1997) proposed a scaled Wald statistic used for small sample inference for fixed effects from restricted maximum likelihood method. Kackar and Harville (1984) investigated how much increase of the mean squared errors when we use estimators of fixed and random effects instead of the true values of them and proposed a general approximation of it. We think it is desirable to extend these efforts to the inference problems of the GLMM and ZI-mixed model cases.

Table 4.4: REML and ML estimates for Whitefly data

Parameter	REML estimate	ML estimate	Parameter	REML estimate	ML estimate
β_1	-0.6957	-0.5733	γ_1	-0.3502	-0.4261
β_{21}	-0.9306	-1.0577	γ_{21}	0.0445	0.0796
β_{22}	-0.4187	-0.6270	γ_{22}	0.3277	0.4118
β_{23}	-0.8351	-1.0894	γ_{23}	0.5154	0.4552
β_{24}	-0.3997	-0.5479	γ_{24}	0.5487	0.6156
β_{25}	2.8065	2.6677	γ_{25}	-3.4491	-3.3269
β_{31}	0.3664	0.3960	γ_{31}	0.0328	0.0504
β_{32}	0.3301	0.2544	γ_{32}	0.1302	0.1584
β_{41}	0.1849	0.1636	γ_{41}	-0.4537	-0.5168
β_{42}	-0.2557	-0.2342	γ_{42}	-0.3491	-0.3754 *
β_{43}	0.6186	0.5705	γ_{43}	0.5167	0.5711
β_{44}	-0.1455	-0.2271	γ_{44}	1.2075	1.2579
β_{45}	0.0632	-0.1457*	γ_{45}	1.1518	1.1664
β_{46}	-0.2346	-0.4278	γ_{46}	0.3859	0.3691
β_{47}	0.1119	-0.0111*	γ_{47}	0.4467	0.4247
β_{48}	-0.4038	-0.4898	γ_{48}	1.0452	1.0606
β_{49}	-0.0492	-0.1609*	γ_{49}	0.4198	0.4266
$\beta_{4,10}$	-0.0075	-0.0322*	$\gamma_{4,10}$	0.7942	0.8028
$\beta_{4,11}$	0.6303	0.6855	$\gamma_{4,11}$	-0.3637	-0.3339
σ	0.5622	0.5417			

not significantly different from 0 at $\alpha = 0.5$

4.8 References

[1] Berk, K.N. and Lachenbruch, P.A. (2002). Repeated measures with zeros.

[2] Booth, J.G. and Hobert, J.H. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society B* **62**, 265–285.

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

[4] Corbeil, R. R. and Searle, S. R. (1976). Restricted maximum likelihood (REML) estimation of variance components in the mixed model. *Technometrics* **18**, 31–38.

[5] Diggle, P.J., Liang, K.-Y. and Zeger, S.L. (1994). *Analysis of Longitudinal Data.* Clarendon Press, Oxford.

[6] Drum, M.L. and McCullagh, P. (1993). REML estimation with exact covariance in the logistic mixed model. *Biometrics* **49**, 677–689.

[7] Hall, D.B. (2000). Zero-inflated Poisson and binomial regression with random effects: a case study. *Biometrics* **56**, 1030–1039.

[8] Harville, D.A. (1977). The use of linear-model methodology to rate high school or college football teams. *Journal of the American Statistical Association* **72,** 278–289.

[9] Kackar, R.N., and Harville, D.A. (1984). Approximations for standard errors of estimators for fixed and random effects in mixed models. *Journal of the American Statistical Association* **79,** 853-862.

[10] Kenward, M.G. and Roger, J.H. (1997). Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics* **53,** 983-997.

[11] Lambert, D.(1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* **34,** 1-14.

[12] Lee, Y. and Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B* **58,** 619–678.

[13] Liao, J.G. and Lipsitz, S.R. (2002). A type of restricted maximum likelihood estimator of variance components in generalized linear mixed models. *Biometrika* **89,** 401–409.

[14] McCullagh, P. and Tibshirani, R. (1990). A simple method for the adjustment of profile likelihood. *Journal of Royal Statistical Society B* **52**, 325–344.

[15] McGilchrist, C.A. (1994). Estimation in generalized mixed models. *Journal of the Royal Statistical Society, B* **56(1)**, 61–69.

[16] Olsen,M.K. and Schafer, J.L.. (2001). A two-part random-effects model for semicontinuous longitudinal data. *Journal of the American Statistical Association* **96,** 730-745.

[17] Patterson, H.D. and Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika* **58,** 545–554.

[18] Ridout, M., Demétrio, C.G.B., and Hinde, J. (1998). Models for count data with many zeros. Invited Paper, *The XIXth International Biometric Conference.* Cape Town, South Africa, 179–192.

[19] Verbeke, G. and Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data.* Springer: New York.

[20] Yau, K.W. and Lee, A.H. (2001). Zero-inflated poisson regression with random effects to evaluate an occupational injury prevention programme. *Statistics in medicine* **20,** 2907–2920.

Chapter 5

Some Review and Future Research

This dissertation focuses on two component mixtures of GLMMs and the special case of zero-inflated mixed effect models. First, we consider ML estimation with the EM algorithm used to facilitate the computations. Then a REML-like estimation method is developed for ZI-inflated mixed effect models. Actually, our original purpose was to apply this REML method to two component mixture of GLMMs. We believed that as long as we can separate the loglikelihood into two parts with one part involving variance component parameters the other part involving fixed effect parameters, we can apply the REML method in Chapter 4. However, we encountered several problems.

First problem comes with assuming that, for two component GLMMs, both components share the same random effects but with different magnitude. For univariate random effects, this can be expressed as $\sigma_1 \boldsymbol{b}_i$ and $\sigma_2 \boldsymbol{b}_i$, and $\boldsymbol{b}_i \sim N(\boldsymbol{0}, \boldsymbol{I})$. This is the definition used in chapter 3. As shown above, with this definition, the observed data loglikelihood is (3.2.1) and the complete data loglikelihood is given by (3.2.2). The first term of (3.2.2) is $\log f(\boldsymbol{b})$, which involves no parameters. The last two terms ($\log f(\boldsymbol{u}|\boldsymbol{b};\boldsymbol{\delta})$ and $\log f(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{b};\boldsymbol{\delta})$) involve both variance component parameter $\boldsymbol{\theta}$ and fixed effect parameter $\boldsymbol{\delta}$. Suppose $\boldsymbol{\delta}$ are known, then the score function for $\boldsymbol{\theta}$ is

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\delta}^c; \boldsymbol{y}) &= \sum_{i=1}^{K} \frac{\partial}{\partial \boldsymbol{\theta}} \log \left\{ \int \prod_{j=1}^{t_i} f(y_{ij}|\boldsymbol{b}_i;\boldsymbol{\delta}^c) \phi_q(\boldsymbol{b}_i) d\boldsymbol{b}_i \right\} \\
&= \sum_{i=1}^{K} \int \left\{ \sum_{j=1}^{t_i} \frac{\partial}{\partial \boldsymbol{\theta}} \log f(y_{ij}|\boldsymbol{b}_i;\boldsymbol{\delta}^c) \right\} f(\boldsymbol{b}_i|\boldsymbol{y}_i;\boldsymbol{\delta}^c) d\boldsymbol{b}_i \\
&= \sum_{i=1}^{K} \sum_{j=1}^{t_i} E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log f(y_{ij}|\boldsymbol{b}_i;\boldsymbol{\delta},\boldsymbol{\theta}) | \boldsymbol{y}_i;\boldsymbol{\delta},\boldsymbol{\theta} \right], \qquad (5.0.1)
\end{aligned}$$

where $\boldsymbol{\delta}^c = (\boldsymbol{\delta}^T, \boldsymbol{\theta}^T)^T)$ and $f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}^c) = \{p_{ij}(\boldsymbol{\gamma})\}f_1(y_{ij}|\boldsymbol{b}_i; \tilde{\boldsymbol{\alpha}}) + \{1 - p_{ij}(\boldsymbol{\gamma})\}f_2(y_{ij}|\boldsymbol{b}_i; \tilde{\boldsymbol{\beta}})$. Correspondingly, supposing $\boldsymbol{\delta}$ has been estimated by the ML estimator $\hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}$ for fixed $\boldsymbol{\theta}$, the profile score function for $\boldsymbol{\theta}$ is

$$p\ell(\boldsymbol{\theta}, \boldsymbol{y}) = \sum_{i=1}^{K}\sum_{j=1}^{t_i} E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log f(y_{ij}|\boldsymbol{b}_i; \hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}, \boldsymbol{\theta})|\boldsymbol{y}_i; \hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}\right], \tag{5.0.2}$$

where $f(y_{ij}|\boldsymbol{b}_i; \hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \{p_{ij}(\hat{\boldsymbol{\gamma}})\}f_1(y_{ij}|\boldsymbol{b}_i; \hat{\tilde{\boldsymbol{\alpha}}}) + \{1 - p_{ij}(\hat{\boldsymbol{\gamma}})\}f_2(y_{ij}|\boldsymbol{b}_i; \hat{\tilde{\boldsymbol{\beta}}})$. Then the bias of estimating $\boldsymbol{\theta}$ by not knowing the fixed effect parameters can be expressed as

$$E_{y_{ij}}\left\{E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log f(y_{ij}|\boldsymbol{b}_i; \hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}, \boldsymbol{\theta})|\boldsymbol{y}_i; \hat{\boldsymbol{\delta}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}\right] - E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}, \boldsymbol{\theta})|\boldsymbol{y}_i; \boldsymbol{\delta}, \boldsymbol{\theta}\right]\right\} \tag{5.0.3}$$

for $i = 1, \ldots, K$ and $j = 1, \ldots, t_i$. Since the variance component parameters are included in each linear predictor, it is not possible to separate them from the conditional density of the observed data. That means the formula (5.0.3) cannot be further simplified to involve only the variance component parameters for us to calculate their bias easily. So, the bias correction idea is very difficult to implement, and the REML method in Chapter 4 can not be extended in this area.

Another finding for the two component GLMMs sharing the same random effects was that

$$\frac{\partial \ell(\boldsymbol{\delta}^c; \boldsymbol{y})}{\partial \boldsymbol{\theta}} \neq \frac{\partial Q(\boldsymbol{\delta}^c|\boldsymbol{\delta}^{c(h)})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\delta}^{c(h)}=\boldsymbol{\delta}^c},$$

where

$$\begin{aligned}
\frac{\partial Q(\boldsymbol{\delta}^c|\boldsymbol{\delta}^{c(h)})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\delta}^{c(h)}=\boldsymbol{\delta}^c} &= \sum_{i=1}^{K}\sum_{j=1}^{t_i} u_{ij} E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log f_1(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}, \boldsymbol{\theta})|\boldsymbol{y}_{ij}; \boldsymbol{\delta}, \boldsymbol{\theta}\right] \\
&+ \sum_{j=1}^{t_i}(1 - u_{ij}) E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log f_2(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}, \boldsymbol{\theta})|\boldsymbol{y}_{ij}; \boldsymbol{\delta}, \boldsymbol{\theta}\right] \\
u_{ij} &= \frac{p_{ij}(\boldsymbol{\gamma}) f_1(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}, \boldsymbol{\theta})}{f(y_{ij}|\boldsymbol{b}_i; \boldsymbol{\delta}, \boldsymbol{\theta})},
\end{aligned}$$

which also prevents us from using REML for two component GLMMs. In summary, the reason for these two difficulties is we can't separate variance component parameters $\boldsymbol{\theta}$ from the component loglikelihood.

To get around this problem, we considered using separate, but possibly correlated, random effects in each component. This approach is problematic for two reasons. First, we think this assumption is not reasonable or practical. How can the random factor affect one component but not the other? For example, in the measles data set, it is natural to think both components have a random county effect. Same thought is given for random plant factor in whitefly data. It is more understandable to assume different random effects for mixing probability and component. But this is a topic which is not addressed in this research. Second, supposing that the assumption of separate random effects in each component were deemed reasonable, denote one set as $\boldsymbol{b}_{1i}$, the other set as $\boldsymbol{b}_{2i}$. Then, depending upon whether we assume these random effects to be independent or dependent, their joint distribution can be expressed as

$$\begin{pmatrix} \boldsymbol{b}_{1i} \\ \boldsymbol{b}_{2i} \end{pmatrix} \sim N(\mathbf{0}, \begin{pmatrix} \boldsymbol{V}_{\boldsymbol{\theta}_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{V}_{\boldsymbol{\theta}_2} \end{pmatrix})$$

or

$$\begin{pmatrix} \boldsymbol{b}_{1i} \\ \boldsymbol{b}_{2i} \end{pmatrix} \sim N(\mathbf{0}, \begin{pmatrix} \boldsymbol{V}_{\boldsymbol{\theta}_1} & \mathrm{Cov}_{\boldsymbol{\theta}_3} \\ \mathrm{Cov}^T_{\boldsymbol{\theta}_3} & \boldsymbol{V}_{\boldsymbol{\theta}_2} \end{pmatrix}),$$

where $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$ denote the unknown variance component parameter vectors for $\boldsymbol{b}_{1i}$ and $\boldsymbol{b}_{2i}$, and $\boldsymbol{\theta}_3$ denote the correlation parameters. Then similar to Section 3.2, the link functions of two component GLMMs can be written as

$$\boldsymbol{\zeta}_{1i}(\boldsymbol{\mu}_{1i}) = \boldsymbol{\eta}_{1i} = \boldsymbol{X}_i\boldsymbol{\alpha} + \boldsymbol{U}_{1i}\boldsymbol{b}_{1i}$$

$$\boldsymbol{\zeta}_{2i}(\boldsymbol{\mu}_{2i}) = \boldsymbol{\eta}_{2i} = \boldsymbol{Z}_i\boldsymbol{\beta} + \boldsymbol{U}_{2i}\boldsymbol{b}_{2i},$$

which involve no variance-covariance parameters. In another words, the loglikelihood can now be separated into two parts with one part involving only fixed effect parameters and the other part involving only variance-covariance parameters. The REML method can now be easily extended here just as in Chapter 4. We tried this approach. However, problems were encountered with singular matrices in the fitting algorithm. We suspect that this is due to the unreasonableness of the modelling assumption. In practice, we expect that random

effects acting upon each component may have different variances but will often be perfectly correlated. This leads to a singular normal random effects distribution, which we believe caused the problems encountered in trying to fit the model.

For estimating the variance-covariance matrix of parameter estimates in two component GLMMs, we had thought about using results obtained for the ES (Expectation-Solution) algorithm proposed by Rosen, Jiang and Tanner (2000). This algorithm is used to fit mixture of experts model for independent or correlated outcome data. It is essentially the same as EM, but with the M step replaced by the solution of an estimating equation, which does not necessarily correspond to a maximization problem. We wanted to see if this approach provide us a better way to calculate the variance-covariance matrix of parameters. Following the notation used by Rosen, Jiang and Tanner (2000), if we define

$$q(.) = \frac{\partial}{\partial \boldsymbol{\delta}} \log f(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta}),$$

then

$$S(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) = E\{\frac{\partial}{\partial \boldsymbol{\delta}} \log f(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}; \boldsymbol{\delta})|\boldsymbol{y}, \boldsymbol{\delta}^{(h)}\}. \tag{5.0.4}$$

It's not hard to show that $S(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)})$ is an unbiased estimating equation and satisfies the proposition (a) and (b) in their paper (p.401). Based on these properties, we derive

$$\begin{aligned}
\hat{\phi} &= \nabla S(\hat{\boldsymbol{\delta}}|\hat{\boldsymbol{\delta}}) = E\{\frac{\partial^2}{\partial \hat{\boldsymbol{\delta}}^2} \log f(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{b}; \hat{\boldsymbol{\delta}})|\boldsymbol{y}, \hat{\boldsymbol{\delta}}\} \\
\hat{\upsilon} &= \sum_{i=1}^{K} \left\{ \left[E(\frac{\partial}{\partial \hat{\boldsymbol{\delta}}} \log f(\boldsymbol{y}_i, \boldsymbol{u}_i, \boldsymbol{b}_i; \hat{\boldsymbol{\delta}})|\boldsymbol{y}_i, \hat{\boldsymbol{\delta}}) \right] \left[E(\frac{\partial}{\partial \hat{\boldsymbol{\delta}}} \log f(\boldsymbol{y}_i, \boldsymbol{u}_i, \boldsymbol{b}_i; \hat{\boldsymbol{\delta}})|\boldsymbol{y}_i, \hat{\boldsymbol{\delta}}) \right]^T \right\}
\end{aligned}$$

and according to their paper, the asymptotic variance of $\hat{\boldsymbol{\delta}}$ can be estimated by $\hat{avar}(\hat{\boldsymbol{\delta}}) = \hat{\phi}^{-1}\hat{\upsilon}\hat{\phi}^{-T}$. This is robust variance estimator. Comparing this estimator with the estimators we proposed in section 3.4, we conclude that there is no significant advantage for variance estimation because

1. The calculation of $\hat{avar}(\hat{\boldsymbol{\delta}})$ still involves problem of how to approximate integral. So it is difficult to apply this methodology unless we use OGQ.

2. $S(\boldsymbol{\delta}|\boldsymbol{\delta})$ in Rosen, Jiang and Tanner (2000) is equivalent to $g_{\boldsymbol{\delta}}(\boldsymbol{\delta}) = \partial Q_m(\tilde{\boldsymbol{\delta}}|\boldsymbol{\delta})/\partial\tilde{\boldsymbol{\delta}}|_{\tilde{\boldsymbol{\delta}}=\boldsymbol{\delta}}$ in Friedl and Kauermann (2000) (see section 3.4), as long as we use OGQ to approximate the integral. Note Q_m means the m point OGQ approximation to Q. Rosen et. al. prove $E\{S(\boldsymbol{\delta}|\boldsymbol{\delta})\} = 0$ while Friedl and Kauermann prove $Eg_{\boldsymbol{\delta}}(\boldsymbol{\delta}) = 0$.

There are many problems left unsolved for mixture modelling. Based on our experience, potential future research will focus on following topics:

Model Diagnostics

Developing model diagnostics tool is a challenging topic not only for GLMMs but also for mixture of GLMMs. For these type of models, model diagnostics and goodness of fit issues are not as well developed as methods of estimation and inference. The commonly used methods include graphical tools such as half-normal plots (Vieira et al, 2000), diagnostic measures such as deviance and deviance residuals (Yau and Lee, 2001; Dietz and Böhning, 1997; Wang and Puterman, 1998; McCullagh and Nelder, 1989), Pearson residuals (Wang and Puterman, 1998), model selection tools such as AIC and BIC (Wang, Cockburn and Puterman, 1998; McLachlan, 2000). Simulation study is another model checking method used very often to examine the properties of the model (Albert and Follmann, 2000), or the performance of the estimates (Olsen and Schafer, 2001; Wang and Puterman, 1998). Half-normal plots also involve simulation by which the simulated envelops are obtained using the estimated parameters.

Based on our experience, some of the existing tools are questionable for mixture of GLMMs. For example, when we fit a negative binomial model for measles data, we obtained lower AIC and BIC (lower is better)for this model than mixture of Poisson model. However, the residual plot from fitting the negative binomial model exhibit a clear bimodal shape. So it is hard for us to believe that a one component model is enough. It is possible that current AIC and BIC criteria or the definition of residuals may not suitable for mixture of GLMMs and need adjustment. Some work on this issue has been done by Lindsay and Roeder (1992) and we think model diagnostics are a worthy topic of future study for mixture of GLMMs.

5.1 References

[1] Albert, P.S. and Follmann, D.A. (2000). Modelling repeated count data subject to informative dropout. *Biometrics* **56**, 667–677.

[2] Dietz, E. and Böhning, D. (1997). The use of two-component mixture models with one completely or partly known component. *Computational Statistics* **12**, 219–234.

[3] Friedl, H. and Kauermann, G. (2000). Standard errors for EM estimates in generalized linear models with random effects. *Biometrics* **56,** 761–767.

[4] Lindsay, B.g. and Roeder, K. (1992). Residual diagnostics for mixture models. *Journal of the American Statistical Association.* **87**, 785–794.

[5] McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models.* Chapman and Hall: London.

[6] McLachlan, G.J. and Peel, D. (2000). *Finite Mixture Models.* New York: Wiley.

[7] Olsen,M.K. and Schafer, J.L.. (2001). A two-part random-effects model for semicontinuous longitudinal data. *Journal of the American Statistical Association* **96,** 730-745.

[8] Rosen, O, Jiang, W.X. and Tanner, M.A. (2000). Mixtures of marginal models. *Biometrika* **87**, 391–404.

[9] Vieira, A.M.C., Hinde, J.P., and Demétrio, C.G.B. (2000). Zero-inflated proportion data models applied to a biological control assay. *Journal of Applied Statistics* **27,** 373–389.

[10] Wang, P.,Cockburn,I.M., and Puterman, M.L. (1998). Analysis of patent data— A mixed-Poisson-regression-model approach. *Journal of Business and Economic Statistics* **16,** 27–41.

[11] Wang, P. and Puterman, M.L. (1998). Mixed logistic regression models. *Journal of Agricultural, Biological, and Environmental Statistics* **3,** 175–200.

[12] Yau, K.W. and Lee, A.H. (2001). Zero-inflated poisson regression with random effects to evaluate an occupational injury prevention programme. *Statistics in medicine* **20,** 2907–2920.

Appendix A

Derivation of MCEM Algorithm for ZI-inflated Models

The complete data loglikelihood is (4.2). So the EM algorithm computes

$$\begin{aligned} Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) &= E\{\ell^c(\boldsymbol{\delta};\boldsymbol{y},\boldsymbol{u},\boldsymbol{b})|\boldsymbol{y};\boldsymbol{\delta}^{(h)}\} \\ &= E\left\{E[\ell^c(\boldsymbol{\delta};\boldsymbol{y},\boldsymbol{u},\boldsymbol{b})|\boldsymbol{y},\boldsymbol{b};\boldsymbol{\delta}^{(h)}]|\boldsymbol{y};\boldsymbol{\delta}^{(h)}\right\}. \end{aligned} \tag{A.0.1}$$

The inner expectation is with respect to the distribution of $\boldsymbol{u}$ given $\boldsymbol{y}$, $\boldsymbol{b}$ and $\boldsymbol{\delta}^{(h)}$ only. Since $\ell^c(\boldsymbol{\delta};\boldsymbol{y},\boldsymbol{u},\boldsymbol{b})$ is linear with respect to $\boldsymbol{u}$, this expectation is simply $\ell^c(\boldsymbol{\delta};\boldsymbol{y},\boldsymbol{u}^{(h)}(\boldsymbol{b}),\boldsymbol{b})$ where the element of $\boldsymbol{u}^{(h)}(\boldsymbol{b})$ is given by

$$\begin{aligned} u_{ij}^{(h)}(\boldsymbol{b}_i) &= E\{u_{ij}|y_{ij},\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)}\} \\ &= Prob(u_{ij}=1|y_{ij},\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)}) \\ &= \frac{f(u_{ij}=1,y_{ij}|\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)})}{f(y_{ij}|\boldsymbol{b}_i;\boldsymbol{\delta}^{(h)})} \\ &= \frac{p_{ij}(\boldsymbol{\gamma}^{(h)})f(y_{ij}|u_{ij}=1,\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\sigma^{(h)})}{p_{ij}(\boldsymbol{\gamma}^{(h)})f(y_{ij}|u_{ij}=1,\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\sigma^{(h)})+(1-p_{ij}(\boldsymbol{\gamma}^{(h)}))f(y_{ij}|u_{ij}=0,\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\sigma^{(h)})} \\ &= \begin{cases} 0, & \text{if } y_{ij}>0; \\ \frac{p_{ij}(\boldsymbol{\gamma}^{(h)})}{p_{ij}(\boldsymbol{\gamma}^{(h)})+(1-p_{ij}(\boldsymbol{\gamma}^{(h)}))f_1(y_{ij}|\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\sigma^{(h)})}, & \text{if } y_{ij}\neq 0. \end{cases} \end{aligned}$$

Plugging this quantity into (1) yields

$$\begin{aligned} Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) &= E\{\ell^c(\boldsymbol{\delta};\boldsymbol{y},\boldsymbol{u}^{(h)}(\boldsymbol{b}),\boldsymbol{b})|\boldsymbol{y};\boldsymbol{\delta}^{(h)}\} \\ &= \sum_{i=1}^{K}\int \ell^c(\boldsymbol{\delta};\boldsymbol{y}_i,\boldsymbol{u}_i^{(h)}(\boldsymbol{b}_i),\boldsymbol{b}_i)f(\boldsymbol{b}_i|\boldsymbol{y}_i;\boldsymbol{\delta}^{(h)})d\boldsymbol{b}_i \\ &= \sum_{i=1}^{K}\int \frac{\ell^c(\boldsymbol{\delta};\boldsymbol{y}_i,\boldsymbol{u}_i^{(h)}(\boldsymbol{b}_i),\boldsymbol{b}_i)f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\sigma^{(h)})\phi_q(\boldsymbol{b}_i;\boldsymbol{\theta}^{(h)})d\boldsymbol{b}_i}{\int f(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\beta}^{(h)},\sigma^{(h)})\phi_q(\boldsymbol{b}_i;\boldsymbol{\theta}^{(h)})d\boldsymbol{b}_i}. \end{aligned} \tag{A.0.2}$$

The integrals in equation (A.2) are now with respect to the random effects $\boldsymbol{b}$ only. We consider an importance sampling approach to approximate this integral.

Suppose $I(\boldsymbol{b}_i), i = 1, \ldots, K$ are the importance samplers which have similar distributional shape as $f(\boldsymbol{y}_i|\boldsymbol{b}_i; \boldsymbol{\delta}^{(h)})\phi_q(\boldsymbol{b}_i)$. Here $\boldsymbol{b}_{i1}, \ldots, \boldsymbol{b}_{im}$ are identically independently drawn from $I(\boldsymbol{b}_i)$. Then equation (2) can be approximated by

$$\begin{aligned}
Q(\boldsymbol{\delta}|\boldsymbol{\delta}^{(h)}) &\approx \sum_{i=1}^{K} \frac{\sum_{\ell=1}^{m} w_{i\ell}^{*} \ell^{c}(\boldsymbol{\delta}; \boldsymbol{y}_i, \boldsymbol{u}_i^{(h)}(\boldsymbol{b}_{i\ell}), \boldsymbol{b}_{i\ell})}{\sum_{\ell=1}^{m} w_{i\ell}^{*}} \\
&= \sum_{i=1}^{K} \sum_{\ell=1}^{m} w_{i\ell} \ell^{c}(\boldsymbol{\delta}; \boldsymbol{y}_i, \boldsymbol{u}_i^{(h)}(\boldsymbol{b}_{i\ell}), \boldsymbol{b}_{i\ell}) \\
&= \sum_{i=1}^{K} \sum_{j=1}^{t_i} \sum_{\ell=1}^{m} w_{i\ell} \left[u_{ij}^{(h)}(\boldsymbol{b}_{i\ell}) \log p_{ij}(\boldsymbol{\gamma}) + (1 - u_{ij}^{(h)}(\boldsymbol{b}_{i\ell})) \log\{1 - p_{ij}(\boldsymbol{\gamma})\} \right] \\
&\quad + \sum_{i=1}^{K} \sum_{j=1}^{t_i} \sum_{\ell=1}^{m} w_{i\ell} (1 - u_{ij}^{(h)}(\boldsymbol{b}_{i\ell})) \log f_1(y_{ij}|\boldsymbol{b}_{i\ell}; \boldsymbol{\beta}, \sigma) \\
&\quad + \sum_{i=1}^{K} \sum_{\ell=1}^{m} w_{i\ell} \log \phi_q(\boldsymbol{b}_{i\ell}; \boldsymbol{\theta}), \qquad \text{(A.0.3)}
\end{aligned}$$

where $w_{i\ell}^{*} = f(\boldsymbol{y}_i|\boldsymbol{b}_{i\ell}; \boldsymbol{\beta}^{(h)}, \sigma^{(h)})\phi_q(\boldsymbol{b}_{i\ell}; \boldsymbol{\theta}^{(h)})/I(\boldsymbol{b}_{i\ell})$ and $w_{i\ell} = w_{i\ell}^{*}/\sum_{\ell=1}^{m} w_{i\ell}^{*}$.

Appendix B

Part of MATLAB Programs for Measles Data Example in Chapter 3[1]

B.1 Main Programs

MEASLESOGQ.M

MEASLESAGQ.M

MEASLESNPML.M

/**** Program name : MEASLESOGQ.M ******/**

/* Purpose : read in data and set initial parameters for OGQ methods*/

```
global Y ID N N2 XMAT ZMAT WMAT BINN1 BINN2 ERRDIST1 ERRDIST2
        PLINK MU1LINK; global MU2LINK;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR
        INCCONST;
global OFFSET1 OFFSET2 WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
load 'measles.dat';
countyid=measles(:,1); ID=countyid;
N=size(ID,1); N2=max(ID);
Y=measles(:,2);
rate=measles(:,3);
nkids=measles(:,4);
year=measles(:,5);
BINN1=ones(N,1); BINN2=BINN1; 1);
```

[1]For complete programs, please contact Lihua Wang at lwang@stat.uga.edu.

```
yearmat=fac(year);
OFFSET1=log(nkids); OFFSET2=log(nkids);
ID2=zeros(N,1);
for i=1:N2;
    ID2(ID==i)=[1:size(ID(ID==i),1)]';
end;
PLINK='logit'; MU1LINK='log'; MU2LINK='log';
ERRDIST1='poisson'; ERRDIST2='poisson';
XMAT=[ones(N,1), rate]; ZMAT=[ones(N,1), rate]; WMAT=[ones(N,1)];
glmlab;
alphainit=[-2.0308 -0.0684]'; betainit=[-7.6017 -0.0356]';
gammainit=[2.19 -.05]'; sigmainit=[0.3029 0.1971]';
mquad=7
diary measlesoakes.diary
[alpha,beta,gamma,sigma,converge,covoakes]=oemoakes(alphainit,betainit,
      gammainit, sigmainit,mquad)
covoakes;
cc=diag(covoakes)
sdoakes=sqrt(cc)
diary off
```

/**** Program name : MEASLESAGQ.M ******/**

/* Purpose : read in data and set initial parameters for AGQ methods*/

```
global Y ID N N2 XMAT ZMAT WMAT BINN1 BINN2 ERRDIST1 ERRDIST2
        PLINK MU1LINK;
global MU2LINK;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR INC-
CONST;
```

```
global OFFSET1 OFFSET2 WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
load 'measles.dat';
countyid=measles(:,1); ID=countyid;
N=size(ID,1); N2=max(ID);
Y=measles(:,2); rate=measles(:,3); nkids=measles(:,4); year=measles(:,5);
BINN1=ones(N,1); BINN2=BINN1; logn=log(BINN1);
yearmat=fac(year);
OFFSET1=log(nkids); OFFSET2=log(nkids);
ID2=zeros(N,1);
for i=1:N2;
    ID2(ID==i)=[1:size(ID(ID==i),1)]';
end;
PLINK='logit'; MU1LINK='log'; MU2LINK='log';
ERRDIST1='poisson'; ERRDIST2='poisson';
XMAT=[ones(N,1), rate]; ZMAT=[ones(N,1), rate]; WMAT=[ones(N,1), rate];
glmlab;
alphainit=[-2.0308 -.0684]'; betainit=[-7.6017 -.0356]';
gammainit=[2.19 -.05]'; sigmainit=[0.3029 0.1971]';
diary emagq.diary
[alpha,beta,gamma,sigma,converge]=em(alphainit,betainit,gammainit,sigmainit,7,100,7)
diary off
```

/**** Program name : MEASLESNPML.M ******/**

/* Purpose : read in data and set initial parameters for NPML methods*/

```
global Y ID N N2 XMAT ZMAT WMAT BINN1 BINN2 ERRDIST1 ERRDIST2
        PLINK MU1LINK;
global MU2LINK YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR
```

```
        INCCONST;
global OFFSET1 OFFSET2 WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
load 'measles.dat';
countyid=measles(:,1); ID=countyid;
N=size(ID,1); N2=max(ID);
Y=measles(:,2); rate=measles(:,3);
nkids=measles(:,4); year=measles(:,5);
BINN1=ones(N,1); BINN2=BINN1;
yearmat=fac(year);
OFFSET1=log(nkids); OFFSET2=log(nkids);
ID2=zeros(N,1);
for i=1:N2;
    ID2(ID==i)=[1:size(ID(ID==i),1)]';
end;
PLINK='logit'; MU1LINK='log'; MU2LINK='log';
ERRDIST1='poisson'; ERRDIST2='poisson';
XMAT=[rate]; ZMAT=[rate]; WMAT=[ones(N,1), rate];
glmlab;
alphainit=[-0.0944]'; betainit=[-0.14295]'; gammainit=[0.3029 0.009392]';
mquad=7;
diary npmlfridel.diary
[alpha,beta,gamma,quadwts,quadvals1,quadvals2,converge]=emnpmlfridel(alphainit,
        betainit,gammainit,mquad)
diary off
```

B.2 Core Subroutines

OEMOAKES.M

EM.M

EMNPMLFRIDEL.M

/**** Program name : OEMOAKES.M ******/**

/* Purpose : Fit two component GLMMs using OGQ*/

```
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR
        INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
global Y ID ID2 N N2 XMAT ZMAT WMAT BINN1 BINN2 ERRDIST1 ERRDIST2
        MU1LINK global MU2LINK PLINK OFFSET1 OFFSET2;
mquad
clear paramtrs;
[toler,maxits,illctol]=myparamtrs
converge=0;
[quadwts,quadvals]=getgaussherm(mquad);
quadwts=quadwts./sqrt(3.14159265358979);
quadvals=sqrt(2).*quadvals;
alpha=alphainit; beta=betainit; gamma=gammainit; sigma=sigmainit;
dimbeta=size(beta,1); dimgamma=size(gamma,1);
dimalpha=size(alpha,1); dimsigma=2;
dimparm=dimalpha+dimbeta+dimgamma+dimsigma;
its=0;
fail=0;
loglik = getloglik(alpha,beta,gamma,sigma,mquad,quadwts,quadvals)
while (its < maxits)
```

```
emitcount=0;
while (emitcount < 5000)
    emitcount=emitcount+1
    its=its+1
    alphaold=alpha; betaold=beta; gammaold=gamma; sigmaold=sigma;
    alp=OFFSET1+XMAT*alpha; blp=OFFSET2+ZMAT*beta;
    glp=WMAT*gamma; p=getmixp(glp);
    warning off
    newmodel;
    clear global YVAR XVARS;
    global YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR
            INCCONST;
    global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
    warning on
    NAMEXV='[';
    for i=1:size(WMAT,2)
        NAMEXV=strcat(NAMEXV,'G');
        NAMEXV=strcat(NAMEXV,int2str(i));
        if (i =size(WMAT,2))
            NAMEXV=strcat(NAMEXV,',');
        end;
    end;
    NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
    for i=1:size(WMAT,2)-1
        NAMELIST=str2mat(NAMELIST,['Gvar ',num2str(i)]);
    end;
    INCCONST=0;
```

```
fofyijcond=zeros(N,mquad);
mu1cond=zeros(N,mquad); mu2cond=zeros(N,mquad);
u=zeros(N,mquad); u1=zeros(N,1);
for l=1:mquad
    mu1cond(:,l)=getmu1(alp+sigma(1)*quadvals(l),BINN1);
    mu2cond(:,l)=getmu2(blp+sigma(2)*quadvals(l),BINN2);
    temp= getf1cond(Y,mu1cond(:,l),BINN1).*p;
    fofyijcond(:,l)= temp+(1-p).*getf2cond(Y,mu2cond(:,l),BINN2);
    cc = temp+(1-p).*getf2cond(Y,mu2cond(:,l),BINN2);
    u1(cc==0) = zeros(size(cc(cc==0),1),1);
    u1(cc =0) = temp(cc =0)./cc(cc =0);
    u(:,l)=u1;
end;
YVAR=[stackrows(u),ones(N*mquad,1)]; LINKFUNCTION=PLINK;
XVARS=kron(WMAT,ones(mquad,1)); ERRDISFUNCTION='binoml';
OFFSET=zeros(size(YVAR,1),1); WEIGHTS=zeros(size(YVAR,1),1);
count1=0; count2=0;
prodi=zeros(N2,mquad);
for i=1:N2
    prodi(i,:)=prod(fofyijcond(ID==i,:));
    ti=size(ID(ID==i),1);
    for j=1:ti;
        count2=count2+1;
        for l=1:mquad;
            count1=count1+1;
            WEIGHTS(count1)=prodi(i,l)*quadwts(l)/(prodi(i,:)*quadwts);
        end;
```

```
    end;
end;
WEIGHTS(isinf(WEIGHTS)—isnan(WEIGHTS))=0;
[gamma, fits, resids, glmcovgamma, covd, devlist]=glmfit;
gamma=gamma(:,1);
warning off
newmodel;
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
        SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:size(XMAT,2)+1
    NAMEXV=strcat(NAMEXV,'A'); NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(XMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
for i=1:size(XMAT,2)
    NAMELIST=str2mat(NAMELIST,['Avar ',num2str(i)]);
end;
if strcmp(ERRDIST1,'binoml')
    YVAR=kron([Y,BINN1],ones(mquad,1));
else
    YVAR=kron([Y],ones(mquad,1));
```

```
end;
LINKFUNCTION=MU1LINK;
XVARS=[ kron(XMAT,ones(mquad,1)), kron(ones(N,1),quadvals)];
ERRDISFUNCTION=ERRDIST1;
SCALEPAR=1; INCCONST=0;
OFFSET=kron(OFFSET1,ones(mquad,1)); WEIGHTS=zeros(size(YVAR,1),1);
count1=0; count2=0;
for i=1:N2
    ti=size(ID(ID==i),1);
    for j=1:ti;
        count2=count2+1;
        for l=1:mquad;
            count1=count1+1;
            WEIGHTS(count1)=u(count2,l)*prodi(i,l)*quadwts(l)
                        /(prodi(i,:)*quadwts);
        end;
    end;
end;
WEIGHTS(isinf(WEIGHTS)—isnan(WEIGHTS))=0;
[alphatmp fits resids glmcovalpha covd devlist]=glmfit;
alpha=alphatmp(1:size(XMAT,2),1); sigma(1)=alphatmp(end,1);
warning off
newmodel;
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
        SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
```

```
warning on
NAMEXV='[';
for i=1:size(ZMAT,2)+1
    NAMEXV=strcat(NAMEXV,'B');
    NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(ZMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
for i=1:size
    NAMELIST=str2mat(NAMELIST,['Bvar ',num2str(i)]);
end;
if strcmp(ERRDIST2,'binoml')
    YVAR=kron([Y,BINN2],ones(mquad,1));
else
    v YVAR=kron([Y],ones(mquad,1));
end;
LINKFUNCTION=MU2LINK;
XVARS=[ kron(ZMAT,ones(mquad,1)), kron(ones(N,1),quadvals)];
ERRDISFUNCTION=ERRDIST2;
SCALEPAR=1; INCCONST=0;
OFFSET=kron(OFFSET2,ones(mquad,1));
WEIGHTS=zeros(size(YVAR,1),1);
count1=0; count2=0;
for i=1:N2
    ti=size(ID(ID==i),1);
```

```
    for j=1:ti;
        count2=count2+1;
        for l=1:mquad;
            count1=count1+1;
            WEIGHTS(count1)=(1-u(count2,l))*prodi(i,l)*quadwts(l)
                /(prodi(i,:)*quadwts);
        end;
    end;
end;
WEIGHTS(isinf(WEIGHTS)—isnan(WEIGHTS))=0;
[betatmp fits resids glmcovbeta covd devlist]=glmfit;
beta=betatmp(1:size(ZMAT,2),1); sigma(2)=betatmp(end,1);
maxchange=max(abs([alpha-alphaold;beta-betaold;gamma-gammaold;
                sigma-sigmaold]))
loglik= getloglik(alpha,beta,gamma,sigma,mquad,quadwts,quadvals)
if (maxchange¡toler)
    converge=1
    its=maxits+1;
end;
if (converge==1)
    emitcount=5000;
    minus2loglik=-2*loglik
    conexpfullhess= [inv(glmcovgamma), zeros(dimgamma,dimalpha+
        dimbeta+2);zeros(dimalpha+1,dimgamma), inv(glmcovalpha),
        zeros(dimalpha+1,dimbeta+1); zeros(dimbeta+1,dimgamma+
        dimalpha+1), inv(glmcovbeta)]
    [part1, part2]=oakespart(alpha,beta,gamma,sigma,mquad,quadwts,quadvals);
```

```
        oldpart1=part1; oldpart2=part2;
        [score1, score2]=oakesdelta(oldpart1, oldpart2,alpha,beta,gamma,
            sigma,mquad,quadwts,quadvals);
        [uweight, weight]=oakesweightu(alpha,beta,gamma,sigma,
            mquad,quadwts,quadvals);
        olduw=uweight; oldw=weight;
        [uwscore1, wscore2]=oakesdeltah(olduw, oldw,alpha,beta,gamma,sigma,
            mquad,quadwts,quadvals);
        term1=zeros(dimparm, dimparm); term2=zeros(dimparm, dimparm);
        count=0;
        for i=1:N2
            ti=size(ID(ID==i),1);
            for jj=1:ti
                for l=1:mquad
                    count=count+1;
                    result1=score1(:,:,count)*uwscore1(:,:,count);
                    term1=term1+result1;
                    result2=score2(:,:,count)*wscore2(:,:,count);
                    term2=term2+result2;
                end;
            end;
        end;
        oakes2=term1+term2;
        infooakes=conexpfullhess-oakes2;
        covoakes=inv(infooakes);
    end;
end;
```

```
end;
```

/**** Program name : EM.M ******/**

/* Purpose : Fit two component GLMMs using AGQ*/

```
function [alpha,beta,gamma,sigma,converge]=em(alphainit,betainit,gammainit,
        sigmainit,mquad1,changeit,mquad2)
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR
        INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
global Y ID ID2 N N2 XMAT ZMAT WMAT BINN1 BINN2 ERRDIST1
        ERRDIST2 MU1LINK global MU2LINK PLINK OFFSET1 OFFSET2;
clear paramtrs; [toler,maxits,illctol]=paramtrs
BIG=1.0e10; SMALL=1.0e-10; converge=0;
[quadwts,quadvals]=getgaussherm(mquad1);
oquadvals=sqrt(2).*quadvals; oquadwts=quadwts./sqrt(3.14159265358979);
mquad=mquad1;
alpha=alphainit; beta=betainit; gamma=gammainit; sigma=sigmainit
dimbeta=size(beta,1); dimgamma=size(gamma,1);
dimalpha=size(alpha,1); dimsigma=2;
dimparm=dimalpha+dimbeta+dimgamma+dimsigma;
its=0; fail=0;
while (its < maxits)
    emitcount=0;
    while (emitcount < 5000)
        emitcount=emitcount+1
        its=its+1
        if (its==changeit)
```

```
        [quadwts,quadvals]=getgaussherm(mquad2);
        oquadwts=quadwts./sqrt(3.14159265358979);
        oquadvals=sqrt(2)*quadvals;
        mquad=mquad2;
    end;
    alphaold=alpha; betaold=beta; gammaold=gamma; sigmaold=sigma;
    alp=OFFSET1+XMAT*alpha; blp=OFFSET2+ZMAT*beta;
    glp=WMAT*gamma; p=getmixp(glp);
    if (its==1)
        b2vec=zeros(N2,1); b1vec=zeros(N2,1);
    end;
    blupits=0; blup1converge=0; blup2converge=0;
    blupfail=1; maxblupits=20;
    while (blupits¡maxblupits)
        blupits=blupits+1;
        b2vecold=b2vec; b1vecold=b1vec;
        if (blup2converge==0)
            b2score=getb2score(p,alp,blp,sigma,b2vec);
            b2neghessdiag= -nrdb2score(b2score,p,alp,blp,sigma,b2vec);
        end;
        if (blup1converge==0)
            b1score=getb1score(p,alp,blp,sigma,b1vec);
            b1neghessdiag= -nrdb1score(b1score,p,alp,blp,sigma,b1vec);
        end;
        if (blup2converge==0)
            b2vec=b2vec+ b2score./b2neghessdiag;
            blup2maxchange=max(abs(b2vec-b2vecold));
```

```
            if (blup2maxchange¡toler AND norm(b2score)¡toler)
                blup2converge=1
            end;
        end;
        if (blup1converge==0)
            b1vec=b1vec+ b1score./b1neghessdiag;
            blup1maxchange=max(abs(b1vec-b1vecold));
            if (blup1maxchange¡toler AND norm(b1score)¡toler)
                blup1converge=1
            end;
        end;
        if (blup1converge AND blup2converge)
            blupits=maxblupits+1;
            blupfail=0;
        end;
    end;
    blupfail=blupfail
    if (blupfail)
        b1vec=b1vec; b1neghessdiag=b1neghessdiag
        b2vec=b2vec; b2neghessdiag=b2neghessdiag
    end;
    b2score=getb2score(p,alp,blp,sigma,b2vec);
    b2neghessdiag= -nrdb2score(b2score,p,alp,blp,sigma,b2vec);
    b1score=getb1score(p,alp,blp,sigma,b1vec);
    b1neghessdiag= -nrdb1score(b1score,p,alp,blp,sigma,b1vec);
    b1neghessdiag(b1neghessdiag¡=0)=1;
    b2neghessdiag(b2neghessdiag¡=0)=1;
```

```
for l=1:mquad;
    b1vecstar(:,l)=b1vec + sqrt(2)*quadvals(l)./sqrt(b1neghessdiag);
    b2vecstar(:,l)=b2vec + sqrt(2)*quadvals(l)./sqrt(b2neghessdiag);
end;
warning off
newmodel;
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
        SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:size(WMAT,2)
    NAMEXV=strcat(NAMEXV,'G'); NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(WMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
for i=1:size(WMAT,2)-1
    NAMELIST=str2mat(NAMELIST,['Gvar ',num2str(i)]);
end;
INCCONST=0;
fofyijcond=zeros(N,mquad);
mu1cond=zeros(N,mquad); mu2cond=zeros(N,mquad);
u=zeros(N,mquad); ublup=zeros(N,1);
W=zeros(N2,mquad); Wdenom=zeros(N2,mquad);
```

```
fofyijcondi1=zeros(N2,mquad); fofyijcondi2=zeros(N2,mquad);
for l=1:mquad;
    for i=1:N2
        condalpi=alp(ID==i)+sigma(1)*b1vecstar(i,l);
        condblpi=blp(ID==i)+sigma(2)*b1vecstar(i,l);
        mu1condi=getmu1(condalpi,BINN1);
        mu2condi=getmu2(condblpi,BINN2);
        tempi= getf1cond(Y(ID==i),mu1condi,BINN1(ID==i)).*p(ID==i);
        fofyijcondi1(i,l)= prod(tempi+(1p(ID==i))
            .*getf2cond(Y(ID==i),mu2condi,BINN2(ID==i)));
        u(ID==i,l)= tempi./(tempi+(1-p(ID==i)).*getf2cond(Y(ID==i),
            mu2condi,BINN2(ID==i)));
        condalpi=alp(ID==i)+sigma(1)*b2vecstar(i,l);
        condblpi=blp(ID==i)+sigma(2)*b2vecstar(i,l);
        mu1condi=getmu1(condalpi,BINN1);
        mu2condi=getmu2(condblpi,BINN2);
        tempi= getf1cond(Y(ID==i),mu1condi,BINN1(ID==i)).*p(ID==i);
        fofyijcondi2(i,l)= prod(tempi+(1-p(ID==i)).*getf2cond(Y(ID==i),
            mu2condi,BINN2(ID==i)));
    end;
    W(:,l)= sqrt(2)*quadwts(l)*exp(quadvals(l)*quadvals(l))
        *fofyijcondi1(:,l).*normpdf(b1vecstar(:,l))./sqrt(b1neghessdiag);
    Wdenom(:,l)= sqrt(2)*quadwts(l)*exp(quadvals(l)*quadvals(l))
        *fofyijcondi2(:,l).*normpdf(b2vecstar(:,l))./sqrt(b2neghessdiag);
end;
for l=1:mquad;
    W(:,l)=W(:,l)./sum(Wdenom,2);
```

```
    W(W(:,l)==Inf,l)=zeros(sum(W(:,l)==Inf),1);
end;
YVAR=[stackrows(u),ones(N*mquad,1)];
WEIGHTS=zeros(N*mquad,1);
for i=1:N2;
    ti=size(ID(ID==i),1);
    WEIGHTS((((i-1)*ti*mquad)+1):(i*ti*mquad))=
        stackrows( kron(W(i,:),ones(ti,1)));
end;
wts=WEIGHTS;
if any(YVAR(:,1).*wts¡0)
    junk=YVAR(:,1).*wts; YVAR(junk¡0,1); wts(junk¡0)
end;
LINKFUNCTION=PLINK; XVARS=kron(WMAT,ones(mquad,1));
ERRDISFUNCTION='binoml'; OFFSET=zeros(size(YVAR,1),1);
[gamma, fits, resids, glmcovgamma, covd, devlist]=glmfit;
gamma=gamma(:,1);
warning off
newmodel; clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
SCALEPAR INC-CONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:size(XMAT,2)+1
    NAMEXV=strcat(NAMEXV,'A'); NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(XMAT,2))
```

```
            NAMEXV=strcat(NAMEXV,',');
        end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
for i=1:size(XMAT,2)
        NAMELIST=str2mat(NAMELIST,['Avar ',num2str(i)]);
end;
if strcmp(ERRDIST1,'binoml')
        YVAR=kron([Y,BINN1],ones(mquad,1));
else
        YVAR=kron([Y],ones(mquad,1));
end;
LINKFUNCTION=MU1LINK;
XVARS=[ kron(XMAT,ones(mquad,1)),zeros(N*mquad,1)];
for i=1:N2;
        ti=size(ID(ID==i),1);
        XVARS((((i-1)*ti*mquad)+1):(i*ti*mquad),dimalpha+1)=
            stackrows( kron(b1vecstar(i,:),ones(ti,1)));
end;
ERRDISFUNCTION=ERRDIST1;
SCALEPAR=1; INCCONST=0;
OFFSET=kron(OFFSET1,ones(mquad,1)); WEIGHTS=wts.*stackrows(u);
[alphatmp fits resids glmcovalpha covd devlist]=glmfit;
alpha=alphatmp(1:size(XMAT,2),1);
sigma(1)=alphatmp(end,1);
warning off
newmodel;
```

```
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
        SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:size(ZMAT,2)+1
    NAMEXV=strcat(NAMEXV,'B'); NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(ZMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
for i=1:size(ZMAT,2)
    NAMELIST=str2mat(NAMELIST,['Bvar ',num2str(i)]);
end;
if strcmp(ERRDIST2,'binoml')
    YVAR=kron([Y,BINN2],ones(mquad,1));
else
    YVAR=kron([Y],ones(mquad,1));
end;
LINKFUNCTION=MU2LINK;
XVARS=[ kron(ZMAT,ones(mquad,1)),zeros(N*mquad,1)];
for i=1:N2;
    ti=size(ID(ID==i),1);
    XVARS((((i-1)*ti*mquad)+1):(i*ti*mquad),dimbeta+1)=
        stackrows( kron(b1vecstar(i,:),ones(ti,1)));
```

```
end;
ERRDISFUNCTION=ERRDIST2;
SCALEPAR=1; INCCONST=0;
OFFSET=kron(OFFSET2,ones(mquad,1)); WEIGHTS=wts.*stackrows(1-u);
[betatmp fits resids glmcovbeta covd devlist]=glmfit;
beta=betatmp(1:size(ZMAT,2),1); sigma(2)=betatmp(end,1);
maxchange=max(abs([alpha-alphaold;beta-betaold;
    gamma-gammaold;sigma-sigmaold]))
loglik= getloglikb(alpha,beta,gamma,sigma,quadwts,
    quadvals,b2vecstar,b2neghessdiag)
if (maxchange¡toler)
    converge=1; its=maxits+1;
end;
if (converge==1)
    emitcount=5000;
    alp=OFFSET1+XMAT*alpha; blp=OFFSET2+ZMAT*beta;
    glp=WMAT*gamma; p=getmixp(glp);
    blupits=0; blup2converge=0; blupfail=1;
    maxblupits=30;
    while (blupits < maxblupits)
        blupits=blupits+1; b2vecold=b2vec
        if (blup2converge==0)
            b2score=getb2score(p,alp,blp,sigma,b2vec);
            b2neghessdiag= -nrdb2score(b2score,p,alp,blp,sigma,b2vec);
        end;
        if (blup2converge==0)
            b2vec=b2vec+ b2score./b2neghessdiag;
```

```
                    blup2maxchange=max(abs(b2vec-b2vecold));
                    if (blup2maxchange¡toler AND norm(b2score)¡toler)
                        blup2converge=1
                    end;
                end;
                if (blup2converge)
                    blupits=maxblupits+1; blupfail=0
                end;
            end;
            b2score=getb2score(p,alp,blp,sigma,b2vec);
            b2neghessdiag= -nrdb2score(b2score,p,alp,blp,sigma,b2vec);
            b2neghessdiag(b2neghessdiag¡=0)=1;
            for l=1:mquad;
                b2vecstar(:,l)=b2vec + sqrt(2)*quadvals(l)
                    ./sqrt(b2neghessdiag);
            end;
            loglik= getloglikb(alpha,beta,gamma,sigma,quadwts,quadvals,
                b2vecstar,b2neghessdiag)
            NRhess = gethessbaseloglik(loglik,b2vecstar,b2neghessdiag,
                alpha,beta,gamma,sigma,mquad,quadwts,quadvals)
            cov=inv(-NRhess)
        end;
    end;
end;
```

/**** Program name : EMNPMLFRIDEL.M ******/**

/* Purpose : Fit two component GLMMs using NPML*/

```
function [alpha,beta,gamma,quadwts,quadvals1, quadvals2,converge]=emnpmlfridel(
        alphainit ,betainit,gammainit,mquad)
    global YVAR LINKFUNCTION XVARS ERRDISFUNCTION SCALEPAR
        INCCONST;
    global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV identity;
    global Y ID ID2 N N2 XMAT ZMAT WMAT BINN1 BINN2 ERRDIST1 ERRDIST2
        MU1LINK global MU2LINK PLINK OFFSET1 OFFSET2;
    clear paramtrs;
    [toler,maxits,illctol]=myparamtrs
    converge=0;
    quadwtsinit =[ 0.07718399665045 0.136462301 0.13888795910759
        0.17504762830876 0.20555368975748 0.06733699007994 0.19952743383709]';
    quadvals1init =[ 3.53232834327429 -4.36151188124223 -5.43777170255962
        -6.27835886715718 -3.79456097829275 -2.26994797826750 -4.05724872099781]';
    quadvals2init =[ 3.27828414975511 3.35678302771616 1.73019687339642
        3.17321509199899 2.91127651348920 3.34647540589581 1.68645207260083]';
    identity=eye(mquad);
    alpha=alphainit; beta=betainit; gamma=gammainit
    quadwts=quadwtsinit; quadvals1=quadvals1init; quadvals2=quadvals2init
    dimbeta=size(beta,1); dimgamma=size(gamma,1); dimalpha=size(alpha,1);
    dimvals1=size(quadvals1,1); dimvals2=size(quadvals2,1); dimwts=size(quadwts,1)-1;
    dimparm2=dimwts;    dimparm1=dimalpha+dimbeta+dimgamma+dimvals1+dimvals2;
dimparm=dimparm1+dimparm2;
    its=0; fail=0;
    loglik= getloglik(alpha,beta,gamma,mquad,quadwts,quadvals1,quadvals2)
    while (its < maxits)
        emitcount=0;
```

```
while (emitcount < 5000)
    emitcount=emitcount+1
    its=its+1
    alphaold=alpha; betaold=beta; gammaold=gamma;
    quadwtsold=quadwts; quadvals1old=quadvals1; quadvals2old=quadvals2;
    alp=OFFSET1+XMAT*alpha; blp=OFFSET2+ZMAT*beta;
    glp=WMAT*gamma; p=getmixp(glp);
    fofyijcond=zeros(N,mquad);
    for l=1:mquad
        mu1cond=getmu1( alp + identity(l,:)*quadvals1,BINN1);
        mu2cond=getmu2( blp + identity(l,:)*quadvals2,BINN2);
        fofyijcond(:,l)= p.*getf1cond(Y,mu1cond,BINN1)+(1-p)
            .*getf2cond(Y,mu2cond,BINN2);
    end;
    w=zeros(N2,mquad);
    for i=1:N2;
        ti=size(ID(ID==i),1);
        denom = prod(fofyijcond(ID==i,:))*quadwts;
        if (denom==0)
            w(i,:)=(zeros(mquad,1))';
        else
            w(i,:) =ti*(prod(fofyijcond(ID==i,:)).*(quadwts'))/denom;
        end;
    end;
    quadwtsnew = sum(w)/N; quadwtsnew=quadwtsnew
    warning off
    newmodel;
```

```
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
    SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:size(WMAT,2)
    NAMEXV=strcat(NAMEXV,'G');
    NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(WMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Constant'];
for i=1:size(WMAT,2)-1
    NAMELIST=str2mat(NAMELIST,['Gvar ',num2str(i)]);
end;
INCCONST=0;
fofyijcond=zeros(N,mquad);
mu1cond=zeros(N,mquad);
mu2cond=zeros(N,mquad);
u=zeros(N,mquad);
u1=zeros(N,1)
for l=1:mquad
    mu1cond(:,l)=getmu1(alp+(identity(:,l))'*quadvals1,BINN1);
    mu2cond(:,l)=getmu2(blp+(identity(:,l))'*quadvals2,BINN2);
    temp = getf1cond(Y,mu1cond(:,l),BINN1).*p;
```

```
    fofyijcond(:,l)= temp+(1-p).*getf2cond(Y,mu2cond(:,l),BINN2);
    cc=(temp+(1-p).*getf2cond(Y,mu2cond(:,l),BINN2));
    u1(cc==0) = zeros(size(cc(cc==0),1),1);
    u1(cc =0) = temp(cc =0)./cc(cc =0);
    u(:,l)=u1;
end;
YVAR=[stackrows(u),ones(N*mquad,1)];
LINKFUNCTION=PLINK;
XVARS=kron(WMAT,ones(mquad,1));
ERRDISFUNCTION='binoml';
OFFSET=zeros(size(YVAR,1),1);
WEIGHTS=zeros(size(YVAR,1),1);
count1=0; count2=0; prodi=zeros(N2,mquad);
for i=1:N2
    prodi(i,:)=prod(fofyijcond(ID==i,:));
    ti=size(ID(ID==i),1);
    for j=1:ti;
        count2=count2+1;
        for l=1:mquad;
            count1=count1+1;
            aa = (prodi(i,:)*quadwts);
            if (aa==0)
                WEIGHTS(count1)=0;
            else
                WEIGHTS(count1)=prodi(i,l)*quadwts(l)/aa;
            end;
        end;
```

```
    end;
end;
[gamma, fits, resids, glmcovgamma, covd, devlist]=glmfit;
gamma=gamma(:,1);
warning off
newmodel;
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
    SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:(size(XMAT,2)+mquad)
    NAMEXV=strcat(NAMEXV,'A');
    NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(XMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Alpha'];
for i=1:mquad
    NAMELIST=str2mat(NAMELIST,['Avar ',num2str(i)]);
end;
if strcmp(ERRDIST1,'binoml')
    YVAR=kron([Y,BINN1],ones(mquad,1));
else
    YVAR=kron([Y],ones(mquad,1));
```

```
end;
LINKFUNCTION=MU1LINK;
XVARS=[ kron(XMAT,ones(mquad,1)), kron(ones(N,1),identity)];
ERRDISFUNCTION=ERRDIST1;
SCALEPAR=1; INCCONST=0;
OFFSET=kron(OFFSET1,ones(mquad,1)); WEIGHTS=zeros(size(YVAR,1),1);
count1=0; count2=0;
for i=1:N2
    ti=size(ID(ID==i),1);
    for j=1:ti;
        count2=count2+1;
        for l=1:mquad;
            count1=count1+1;
            aa = (prodi(i,:)*quadwts);
            if (aa==0)
                WEIGHTS(count1)=0;
            else
                WEIGHTS(count1)=u(count2,l)*prodi(i,l)*quadwts(l)/aa;
            end;
        end;
    end;
end;
[alphatmp fits resids glmcovalpha covd devlist]=glmfit;
alpha=alphatmp(1:size(XMAT,2),1);
quadvals1=alphatmp(size(XMAT,2)+1:end,1);
warning off
newmodel;
```

```
clear global YVAR XVARS;
global YVAR LINKFUNCTION XVARS ERRDISFUNCTION
    SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
warning on
NAMEXV='[';
for i=1:size(ZMAT,2)+1
    NAMEXV=strcat(NAMEXV,'B');
    NAMEXV=strcat(NAMEXV,int2str(i));
    if (i =size(ZMAT,2))
        NAMEXV=strcat(NAMEXV,',');
    end;
end;
NAMEXV=strcat(NAMEXV,']'); NAMEYV='[y]'; NAMELIST=['Beta'];
for i=1:mquad
    NAMELIST=str2mat(NAMELIST,['Bvar ',num2str(i)]);
end;
if strcmp(ERRDIST2,'binoml')
    YVAR=kron([Y,BINN2],ones(mquad,1));
else
    YVAR=kron([Y],ones(mquad,1));
end;
LINKFUNCTION=MU2LINK;
XVARS=[ kron(ZMAT,ones(mquad,1)), kron(ones(N,1),identity)];
ERRDISFUNCTION=ERRDIST2;
SCALEPAR=1; INCCONST=0;
OFFSET=kron(OFFSET2,ones(mquad,1));
```

```
WEIGHTS=zeros(size(YVAR,1),1);
count1=0; count2=0;
for i=1:N2
    ti=size(ID(ID==i),1);
    for j=1:ti;
        count2=count2+1;
        for l=1:mquad;
            count1=count1+1;
            aa=(prodi(i,:)*quadwts);
            if (aa==0)
                WEIGHTS(count1)=0;
            else
                WEIGHTS(count1)=(1-u(count2,l))*prodi(i,l)*quadwts(l)/aa;
            end;
        end;
    end;
end;
[betatmp fits resids glmcovbeta covd devlist]=glmfit;
beta=betatmp(1:size(ZMAT,2),1);
quadvals2=betatmp((size(ZMAT,2)+1:end),1);
if (its > 5)
    quadwts=quadwtsnew';
end;
maxchange=max(abs([alpha-alphaold;beta-betaold;gamma-gammaold]))
loglik = getloglik(alpha,beta,gamma,mquad,quadwts,quadvals1, quadvals2)
if (maxchange < toler)
    converge=1; emitcount=5000;
```

```
            its=maxits+1;
        end;
    end;
end;
FEM = fridel(alpha, beta, gamma, Y, mquad, quadvals1,
    quadvals2, quadwts,dimparm1, dimparm2)
B=100; FMCft=zeros(dimparm,dimparm);
for t = 1:mquad
    Bt=floor(1+B*quadwts(t));
    lamda1=exp(alp+identity(t,:)*quadvals1);
    lamda2=exp(blp+identity(t,:)*quadvals2);
    sum=zeros(dimparm,dimparm);
    for b = 1:Bt
        z = rand(N,1);
        poi1 = poissrnd(lamda1);
        poi2 = poissrnd( lamda2);
        yijstar = (z ¡= p).*poi1 + (z ¿ p).*poi2;
        Ftb=fridel(alpha, beta, gamma, yijstar, mquad, quadvals1, quadvals2,
            quadwts,dimparm1, dimparm2);
        sum=sum+Ftb;
    end;
    Ft = quadwts(t)*sum / Bt;
    FMCft = FMCft + Ft;
end;
Covfridel = inv(FMCft)*FEM*inv(FMCft)
```

Appendix C

Part of MATLAB Programs for Whiteflies Data Example in Chapter 4[1]

C.1 Main Programs

/**** Program name : WFLY.M ******/**

/* Purpose : read in data and set initial parameters for REML methods*/

```
global Y ID ID2 N N2 GMAT BMAT BINN;
global PLINK MULINK ERRDISTP ERRDISTMU;
load 'wflyid.dat';
plantid=wflyid(:,7); ID=plantid;
N=size(ID,1); N2=max(ID);
Y=wflyid(:,6); week=wflyid(:,2); rep=wflyid(:,3); trt=wflyid(:,4);
BINN=wflyid(:,5);
trtmat=fac(trt,6); weekmat=fac(week); repmat=fac(rep);
ID2=zeros(N,1); for i=1:N2;
    ID2(ID==i)=[1:size(ID(ID==i),1)]';
end;
GMAT=[ones(N,1), trtmat, repmat,weekmat];
BMAT=[ones(N,1), trtmat, repmat,weekmat];
PLINK='logit'; MULINK='logit';
ERRDISTP='binoml'; ERRDISTMU='binoml';
```

[1]For complete programs, please contact Lihua Wang at lwang@stat.uga.edu. In addition, the symbol of power calculation in MATLAB has been changed to "**" because of LATEX recognotion issue.

```
glmlab;
betainit = [ 0.9441 -0.365 -0.89 -0.39 -0.98 0.248 1.0319 0.452 -0.979 -1.402 -1.063
    -2.082 -1.860 -2.19 -1.64 -2.506 -1.68 -2.27 -0.848]';
gammainit =[ 0.3202 1.172 0.371 1.846 0.434 0.662 0.4100 0.347 -0.925 -1.200 -0.500
    0.225 -0.018 -0.31 -0.12 -0.195 -0.12 -0.39 -0.455]';
sigmainit =[0.1]';
diary outputlast30.diary
mquad=7;
[beta,gamma,sigma,converge]=main(Y,betainit,gammainit,sigmainit,mquad)
diary off
```

C.2 Core Subroutines

MAIN.M

EMNR.M

/**** Program name : MAIN.M ******/**

/* Purpose : Fit data with REML estimation method*/

```
function [beta,gamma,sigma,converge]=main(Y, betainit,gammainit,sigmainit,mquad)
global YVAR LINKFUNCTION ERRDISFUNCTION SCALEPAR INCCONST;
global OFFSET WEIGHTS DETAILSFILE NAMEXV NAMELIST NAMEYV;
global ID ID2 N N2 GMAT BMAT BINN ERRDISTP ERRDISTMU MULINK;
global PLINK OFFSET;
clear paramtrs;
[toler,maxits,illctol]=paramtrs
BIG=1.0e10; SMALL=1.0e-10; converge=0;
[quadwts,quadvals]=getgaussherm(mquad);
beta=betainit gamma=gammainit sigma=sigmainit
```

```
dimbeta=size(beta,1); dimgamma=size(gamma,1); dimsigma=1;
dimparm=dimbeta+dimgamma+dimsigma;
its=0; fail=0;
bk=0;
while(its < maxits)
    its=its+1
    betaold=beta; gammaold=gamma; sigmaold=sigma;
    [beta,gamma,bb1, bb2,converge] = emnr(Y,beta,gamma,sigma,mquad,
    quadvals, quadwts);
    bb3=bb2
    [yy] = gerdata(beta,gamma,sigma);
    [beta,gamma,bb1,bb2,converge] = emnr(yy,beta,gamma,sigma,mquad,
    quadvals, quadwts);
    bias = bb2-bb1;
    bk=(1-1/its)*bk+1/its*bias;
    sigma = sqrt(bb3-bk)
    maxchange=max(abs(sigma-sigmaold))
    if (maxchange < toler*10)
        converge=1
        its=maxits+1;
    end;
end;
[beta,gamma,bb1, bb2,converge] = emnr(Y,beta,gamma,sigma,mquad,
    quadvals, quadwts);
nbeta=beta; ngamma=gamma; nsigma=sigma
```

/**** Program name : EMNR.M ******/**

/* Purpose : ML estimation */

```
function [beta,gamma,bb1, bb2,converge] = emnr(y,beta,gamma,sigma,mquad,
    quadvals, quadwts);
global ID ID2 N N2 GMAT BMAT BINN ERRDISTP ERRDISTMU MULINK
global PLINK ;
clear paramtrs;
[toler,maxits,illctol]=paramtrs
BIG=1.0e10; SMALL=1.0e-10; converge=0;
dimbeta=size(beta,1); dimgamma=size(gamma,1);
dimparm=dimbeta+dimgamma;
its2=0; fail=0;
b1vecini=zeros(N2,1); b2vecini=zeros(N2,1);
while(its2 < maxits)
    its2=its2+1
    betaold=beta; gammaold=gamma; sigmaold=sigma;
    blp=BMAT*beta; glp=GMAT*gamma; p=getmixp(glp);
    kp=kron(p, ones(1,mquad));
    b1vec=b1vecini; b2vec=b2vecini;
    [hatb1,neghessdiagb1,hatb2,neghessdiagb2]=optimb(b1vec,b2vec,
        y,beta,gamma,sigma,blp,p,toler);
    b1vecini=hatb1;
    for l=1:mquad;
        b1vecstar(:,l)=hatb1 + sqrt(2)*quadvals(l)./sqrt(neghessdiagb1);
        b2vecstar(:,l)=hatb2 + sqrt(2)*quadvals(l)./sqrt(neghessdiagb2);
    end;
    nb1vecstar=zeros(N, mquad); nb2vecstar=zeros(N, mquad);
    count1=0; count2=0;
    for i=1:N2
```

```
    ti=size(ID(ID==i),1);
    count1=count2+1; count2=count2+ti;
    nb1vecstar(count1:count2,:)=kron(b1vecstar(i,:),ones(ti,1));
    nb2vecstar(count1:count2,:)=kron(b2vecstar(i,:),ones(ti,1));
end;
ky=kron(y, ones(1,mquad));
kbmat=kron(BMAT, ones(mquad,1));
kgmat=kron(GMAT, ones(mquad,1));
kbinn=kron(BINN, ones(1,mquad));
u=(y==0);
kforpi1=zeros(N, mquad); kforbin1=zeros(N, mquad);
kforpi2=zeros(N, mquad); kforbin2=zeros(N, mquad);
for l=1:mquad
    kforpi1(:,l)=getdkappa([BMAT,nb1vecstar(:,l)]*[beta;1]);
    kforbin1(:,l)=getfcond(y,getmu([BMAT,nb1vecstar(:,l)]*[beta;1],BINN),BINN);
    kforpi2(:,l)=getdkappa([BMAT,nb2vecstar(:,l)]*[beta;1]);
    kforbin2(:,l)=getfcond(y,getmu([BMAT,nb2vecstar(:,l)]*[beta;1],BINN),BINN);
end;
fofycond1=zeros(N2,mquad); fofycond2=zeros(N2,mquad);
for i=1:N2
    term1=(kp(ID==i)+(1-kp(ID==i))
        .*(1-kforpi1(ID==i))**(kbinn(ID==i)))**(ku(ID==i));
    term2=((1-kp(ID==i)).*kforbin1(ID==i))
        **(1-ku(ID==i));
    fofycond1(i,:)=prod(term1.*term2);
    term1=(kp(ID==i)+(1-kp(ID==i)).*
        (1-kforpi2(ID==i))**(kbinn(ID==i)))**(ku(ID==i));
```

```
    term2=((1-kp(ID==i)).*kforbin2(ID==i))**(1-ku(ID==i));
    fofycond2(i,:)=prod(term1.*term2);
end;
W=zeros(N2,mquad); Wdenom=zeros(N2,mquad);
for l=1:mquad
    W(:,l)= sqrt(2)*quadwts(l)*exp(quadvals(l)*quadvals(l))
        *fofycond1(:,l).*normpdf(b1vecstar(:,l),0, sigma)./sqrt(neghessdiagb1);
    Wdenom(:,l)= sqrt(2)*quadwts(l)*exp(quadvals(l)*quadvals(l))
        *fofycond2(:,l).*normpdf(b2vecstar(:,l), 0, sigma)./sqrt(neghessdiagb2);
end;
loglik=sum(log(sum(Wdenom,2)))
for l=1:mquad;
    W(:,l)=W(:,l)./sum(Wdenom,2);
    W(W(:,l)==Inf,l)=zeros(sum(W(:,l)==Inf),1);
end;
weight=zeros(N, mquad);
count1=0; count2=0;
for i=1:N2
    ti=size(ID(ID==i),1);
    count1=count2+1; count2=count2+ti;
    weight(count1:count2,:)=kron(W(i,:),ones(ti,1));
end;
if (its2==1)
    bb1=sum(1./neghessdiagb2 + hatb2.*hatb2)/N2
end;
wts=stackrows(weight);
cc=kp+(1-kp).*(1-kforpi1)**kbinn;
```

```
term1=ku.*(1-kp).*(-kbinn).*kforpi1.*(1-kforpi1)**kbinn./cc;
term2=(1-ku).*(ky-kbinn.*kforpi1);
nrscoreb=(stackrows(term1).*wts)'*kbmat + (stackrows(term2).*wts)'*kbmat;
term1=ku.*kp.*(1-kp).*(1-(1-kforpi1)**kbinn)./cc;
term2=(1-ku).*(-kp);
nrscoreg=(stackrows(term1).*wts)'*kgmat + (stackrows(term2).*wts)'*kgmat;
nrscore=[nrscoreb,nrscoreg]';
nrhess=zeros(dimparm, dimparm);
termb1=ku.*(-kbinn).*(1-kp).*(kforpi1.*(1-kforpi1)**(kbinn+1)-
    kbinn.*kforpi1**2.*(1-kforpi1)**kbinn)./cc;
aa=kbinn.*(1-kp).*kforpi1.*(1-kforpi1)**kbinn;
termb2=ku.*aa.*aa./(cc.*cc);
termb3=(1-ku).*(-kbinn).*kforpi1.*(1-kforpi1);
term=stackrows(termb1-termb2+termb3).*wts;
term=sparse(diag(term));
nrhess(1:dimbeta,1:dimbeta)=kbmat'*term*kbmat;
termg1=ku.*(1-(1-kforpi1)**kbinn).*kp.*(1-kp).*(1-2.*kp)./cc;
aa=kp.*(1-kp).*(1-(1-kforpi1)**kbinn);
termg2=ku.*aa.*aa./(cc.*cc);
termg3=(1-ku).*kp.*(1-kp);
term=stackrows(termg1-termg2-termg3).*wts;
term=sparse(diag(term));
nrhess((dimbeta+1):dimparm,(dimbeta+1):dimparm)= kgmat'*term*kgmat;
termbg1=ku.*kbinn.*kforpi1.*(1-kforpi1)**kbinn.*kp.*(1-kp)./cc;
termbg2=ku.*(1-kp).*kbinn.*kforpi1.*(1-kforpi1)
    **kbinn.*kp.*(1-kp).*(1-(1-kforpi1)**kbinn)./(cc.*cc);
term=stackrows(termbg1+termbg2).*wts;
```

```
        term=sparse(diag(term));
        nrhess(1:dimbeta,(dimbeta+1):dimparm)= kbmat'*term*kgmat;
        nrhess((dimbeta+1):dimparm,1:dimbeta)=(nrhess(1:dimbeta,(dimbeta+1):dimparm))';
        para=[beta',gamma']';
        update=inv(nrhess)*nrscore;
        para=para-update;
        beta=para(1:dimbeta); gamma=para((dimbeta+1):dimparm)
        maxchangefix=max(abs([beta-betaold;gamma-gammaold]))
        if (maxchangefix¡toler)
            converge=1
            its2=maxits+1;
        end;
        if (converge==1)
            b2vec=b2vecini; b1vec=b1vecini;
            [hatb1,neghessdiagb1,hatb2,neghessdiagb2]=
                optimb(b1vec,b2vec,y,beta,gamma,sigma,blp,p,toler);
            b2vecini=hatb2;
            b1vecini=hatb1;
            bb2=sum(1./neghessdiagb2 + hatb2.*hatb2)/N2
        end;
end;
```