ASSOCIATION ANALYSIS OF DISEASE STATUS WITH A CANDIDATE GENE USING GENERALIZED LINEAR MIXED MODEL

by

Salehin Khan Chowdhury

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of

the requirements for the degree of

MASTER OF SCIENCE

at

School of Mathematics and Statistics

Ottawa-Carleton Institute for Mathematics and Statistics

CARLETON UNIVERSITY

Ottawa, Ontario

September 05, 2008

Your file Votre référence
ISBN: 978-0-494-44119-0
Our file Notre référence
ISBN: 978-0-494-44119-0

Abstract

In this thesis, we explore techniques for analyzing genetic data. We are interested in an association analysis of disease status with a candidate gene using generalized linear mixed models (GLMMs). We have considered each of the families in the population as an independent cluster with intra cluster correlation. A binary logistic random intercept model has been used to simulate the genetic data. The random intercept is considered to be the random effect of the individual families. The maximum likelihood (ML) method has been used to estimate the fixed effects and variance components in GLMMs. We study the finite-sample properties of the ML estimators based on a series of simulations. We also investigate the impact of model misspecification on the estimates of the model parameters. For interval estimation of the regression parameters and the variance components we use both ML and bootstrap confidence intervals. For comparison, we use two strategies from a hierarchical nonparametric bootstrap approach. One strategy (Strategy 1) samples family units, preserving the structure and correlation within each family. The second strategy (Strategy 2) also samples family units but then randomly samples offspring with replacement in each family. Specifically, we evaluate the coverage probability of 95% confidence intervals, mean lengths of the confidence intervals and the relative biases of the estimators of both regression coefficients and variance components in the case of correctly specified and misspecified random effects.

Acknowledgments

I would like to thank my supervisor Dr. Sanjoy Sinha for his tireless efforts and patience in making this thesis become a reality. I gratefully acknowledge the partial financial support for this research provided from my supervisor's research grant. I also thank the School of Mathematics and Statistics for providing me the opportunity to pursue graduate study in Statistics. It is certainly a privilege to thank Valerie Daley, the Graduate Administrator of the school, for her valuable suggestions and for my study at Carleton. Many thanks to my aunt Nina Ahmed, uncle Faruq Hossain and my lovely cousin Ishti Hossain for their valuable support.

To
My Aunt Nina Ahmed,
Uncle Faruq Hossain
and
Lovely Cousin Ishti Hossain

Table of Contents

List of Tables

Abbreviations

CI	Confidence interval
GEE	Generalized estimating equation
GLMM	Generalized linear mixed model
GLM	Generalized linear model
LM	Linear model
LMM	Linear mixed model
ML	Maximum likelihood
MSE	Mean squared error
NR	Newton-Raphson

Chapter 1

Introduction

Studies concerning genetic association aim to test whether single-locus alleles or genotype frequencies (or more generally, multilocus haplotype frequencies) are different between 2 groups (usually diseased subjects and healthy controls). Genetic association studies are based on the principle that genotypes are measured "directly", i.e. by sequencing the actual genetic code. If certain genetic variations are found to be significantly more frequent in people with the disease compared to people without disease, the variations are said to be "associated" with the disease. The associated genetic variations can serve as powerful pointers to the region of the human genome where the disease-causing problem resides. Association became mainly a topic in population genetics.

As long ago as 1921, Buchanan and Higley [6] demonstrated a relation between pernicious anaemia and the ABO blood groups. Since that time other investiga-

tors have brought to light the association of cancer of the stomach with group A (Aird and Bentall, 1953 [1]), peptic ulceration with group O (Aird et al, 1954 [2]), diabetes mellitus with group A (McConnell et al., 1956 [17]) and other associations where the evidence was less convincing (Roberts, 1959 [25]). Marjory et al, (1965) [16] analyzed the association between ABO blood group and skin disease using χ^2-test.

Once new genetic associations are identified, researchers can use the information to develop better strategies to detect, treat and prevent the disease. Such studies are particularly useful in finding genetic variations that contribute to common, complex diseases, such as asthma, cancer, diabetes, heart disease and mental illnesses. However, the associated variants themselves may not directly cause the disease. They may just be "tagging along" with the actual causal variants. For this reason, researchers often need to take additional steps, such as sequencing DNA base pairs in that particular region of the genome, to identify the exact genetic change involved in the disease.

With the rapid accumulation of gene and genome sequences, genetics has entered a new age, one which depends heavily on statistical methods. Generalized Linear Models (GLM) are often used to evaluate the relationships between a disease trait and covariates, such as one or more candidate genes or an environmental exposure. Recently, attention has turned to study designs that mandate the inclusion of family members in addition to an individual.

Standard statistical methods that test for marginal relationships of risk factors with disease outcome and treat individual family members as independent are not valid because they ignore the correlation among family members. For example, in many cases the variation within a family will likely be smaller than the variation between families, because related family members share genetic characteristics and environmental influences (J. Shin et al. 2007)[28]. Standard models for analysis assume independent observations, which is unlikely to be true for family data, and the usual standard errors for the regression parameter estimates may be too large or too small, depending on the distribution of the covariates within and between families (Bull et al. 2001)[8]. Applying Generalized Linear Mixed Model (GLMM) is a vivid way to overcome this problem as it takes into account the intra cluster correlations.

A familial correlation structure that reflects genetic dependence would have high correlation between siblings, lower correlation for parent-offspring pairs, and virtually no correlation between parents. A more complex correlation structure may involve both genetic and environmental sharing among family members. Although it is possible to specify these types of general correlation structures (Bull et al., 1995 [7]), evaluations with varying family sizes have been limited. Sherman and le Cessie (1997) [27] used an exchangeable correlation structure to generate binary data in their simulations. Feng et al. (1996) [12] used a more general correlation structure to simulate their data, but examined a quantitative trait. Both of these

previous studies looked at clusters of fixed sizes.

The common complex diseases such as asthma are an important focus of genetic research, and studies based on large numbers of simple pedigrees ascertained from population-based sampling frames are becoming commonplace. Many of the genetic and environmental factors causing these diseases are unknown and there is often a strong residual covariance between relatives even after all known determinants are taken into account. This must be modeled correctly whether scientific interest is focused on fixed effects, as in an association analysis, or on the covariances themselves. Analysis is straightforward for multivariate Normal phenotypes, but difficulties arise with other types of trait. Generalized linear mixed models (GLMMs) offer a potentially unifying approach to analysis for many classes of phenotype including multivariate Normal traits, binary traits, and censored survival times. Burton et al. (1999) [9] discussed Bayesian inference using Gibbs Sampling (a generic Gibbs sampler; BUGS) in GLMMs for multivariate Normal and binary phenotypes in nuclear families. They investigated a suitable model structure for Normal phenotypes and showed how the model extends to binary traits. They discussed parameter interpretation and statistical inference and showed how to circumvent a number of important theoretical and practical problems that they encountered. Using simulated data they showed that model parameters seem consistent and appear unbiased in smaller data sets. They also illustrated their methods using data from an ongoing cohort study.

Scurrah et al. (2000) [26] used BUGS (Bayesian inference using Gibbs sampling: a readily available, generic Gibbs sampler) to fit GLMMs for right censored survival times in nuclear and extended families. They treated the random effects associated with a genetic component of variance in a GLMM as an adjusted phenotype and used as input to a conventional model-based or model-free linkage analysis. This provides a simple way to conduct a linkage analysis for a trait reflected in a right-censored survival time while comprehensively adjusting for observed confounders at the level of the individual and latent environmental effects shared across families.

Pawitan et al. (2000) [23] described various genetic models that can be analyzed using an extended family structure. They used a generalized linear mixed model (GLMM) to deal with the family structure and likelihood-based methodology for parameter inference. The method is completely general, accommodating arbitrary family structures and incomplete data. They illustrated the methodology in great detail using the Swedish birth registry data on pre-eclampsia, a hypertensive condition induced by pregnancy. The statistical challenges include the specification of sensible models that contain a relatively large number of variance components compared to standard mixed models. They used a generalized linear mixed model (GLMM) with a probit link and four variance components, due to maternal additive polygenic effects that may influence the intra-uterine environment , foetal additive polygenic effects, environmental effects shared between siblings, and envi-

ronmental effects common to the pedigree consisting of a pair of siblings and their respective offspring.

J. Shin et al. (2007) [28] compared two statistical approaches for analyzing correlated binary data from randomly ascertained nuclear families. They used the generalized estimating equations approach (GEE) to adjust for familial correlation. The relationship between covariates and the response was modelled, and the correlations among family members were treated as nuisance parameters. For comparison, they proposed two strategies from a hierarchical nonparametric bootstrap approach. One strategy (S1) samples family units, preserving the structure and correlation within each family. A second and novel strategy (S2) also samples family units but then randomly samples offspring with replacement in each family. They applied the methods to data from a study of cardiovascular disease, and followed up with a simulation study in which family data were generated from an underlying multifactorial genetic model. Although the bootstrap approach was more computationally demanding, it outperformed the GEE in terms of confidence interval coverage probabilities for all sample sizes considered.

1.1 Statement of the Problem

In this thesis, our goal is to explore a disease model that includes correlated familial data. We are interested in an association analysis of disease status with

a candidate gene using Generalized Linear Mixed Model (GLMM). We have considered each of the families in the population as an independent cluster with intra cluster correlation. We generated familial data considering binary logistic random intercept model that has candidate genotype of the individuals as a covariate. Here the random intercept is the random effect of the individual families. To compare the maximum likelihood estimates of the simulated data, we used two strategies from the hierarchical nonparametric bootstrap approach. One strategy (Strategy 1) samples family units, preserving the structure and correlation within each family. The second strategy (Strategy 2) also samples family units but then randomly samples offspring with replacement in each family. We ran a series of simulations to investigate the coverage probability of 95% confidence intervals, mean length of the confidence intervals and the empirical relative biases for the both estimation of regression coefficients and variance components in the case of correctly specified and misspecified random effects.

1.2 Organization of Thesis

This thesis is organized as follows. In Chapter 2, we introduce generalized linear models (GLMs) and review some commonly used methods for inference in GLM's. We also briefly review linear mixed models (LMMs) and generalized linear mixed models (GLMMs) for analyzing clustered correlated data. Chapter 3 presents the

preliminaries of the statistical genetic analysis. Here we also review the nonparametric bootstrap and the hierarchical bootstrap methods for inference in mixed models.

In chapter 4, we illustrate the computational issues using a simple binary mixed model for genetic data. Chapter 5 discuses the results from a simulation study, which was carried out to investigate the performance of the bootstrap strategies for inference in GLMMs for genetic data. Chapter 6 concludes the thesis with some discussion and direction for further research.

Chapter 2

Generalized, Linear and Mixed Models

In this chapter, we introduce the generalized linear models (GLMs) and review some commonly used methods for inference in the GLM's. We also briefly review the linear mixed models (LMMs). Finally, we describe the generalized linear mixed models (GLMMs) as an extension of the GLM and LMM for analyzing clustered correlated data.

2.1 Generalized Linear Models (GLM)

The introduction of the general linear regression models for normal data was a great achievement in statistical inference in the early 20th century. In this pre-

computer age, there was much effort devoted to experimental designs that would, not only statistically efficient, but be easily analyzed by hand computation. Fisher and Yates were two pioneers of this work, both working at the Rothamsted agricultural research station. With the advent of computers the effort of computation was removed and statistical research moved on to the extension of these methods to a wider class of statistical problem, in particular non-normal data. Much of the impetus come from biostatistics in which it was common that observations are *binary*, that is there were only two possible outcomes e.g. dead or alive. Numerical procedures were devised for each situation in isolation and computer programs for model fitting were specific to that situation. A breakthrough came with the work of Nelder and Wedderburn (also working at Rothamsted) who devised a wide class of models that could be fitted by a common numerical algorithm, the *Generalized Linear Model* (1972) [21].

Hypothesis tests applied to the generalized linear model do not require normality of the response variable, nor do they require homogeneity of variances. Hence generalized linear models can be used when response variables follow distributions other than the normal distribution, and when variances are not constant. For example, count data would be appropriately analyzed using a Poisson random variable within the context of the generalized linear model.

2.1.1 The Components of a Generalized Linear Model

Generalized linear models are an extension of classical linear models. The classical linear model can be summarized as follows:

The components of $\mathbf{y}$ are independent normal variables with constant variance σ^2 and response mean vector

$$E(\mathbf{y}) = \boldsymbol{\mu}, \tag{2.1}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)'$ with $\mu_i = E(y_i) = \sum_{j=1}^{p} x_{ij}\beta_j$.

The above model has the following three specific components:

i. The *"random component"*: The random components of $\mathbf{y}$ are independently normally distributed with $E(\mathbf{y}) = \boldsymbol{\mu}$ and constant variance σ^2,

ii. The *"systematic component"*: covariates $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_p$ produce a "linear predictor" $\boldsymbol{\eta}$ given by $\boldsymbol{\eta} = \sum_{j=1}^{p} \mathrm{x}_j\beta_j$, and

iii. The *"link"* between random and systematic components, $\boldsymbol{\mu} = \boldsymbol{\eta}$. Now, if we write $\eta_i = g(\mu_i)$, then g is called the *"link function"*.

In this formulation, classical linear models have a normal distribution in component **i** and identity function for the link in component **iii**. Generalized linear models allow two extensions; first, the distribution in component **i** comes from an exponential family other than the normal, and second, the link function in component **ii** may become any monotonic differentiable function.

2.1.2 Likelihood Function for GLMs

We assume that each component of y has a distribution in the exponential family, taking the form:

$$f(y,\theta,\phi) = exp\left\{\left(\frac{y\theta - b(\theta)}{a(\phi)}\right) + c(y,\phi)\right\}, \tag{2.2}$$

where ϕ is the scale parameter and θ is called the natural location for some functions a, b and c. For members of the exponential family, if ϕ is known, this is an exponential family model with conditional parameter θ.

We write,

$$l(\theta,\phi;y) = \log f(y;\theta,\phi) = \left\{\left(\frac{y\theta - b(\theta)}{a(\phi)}\right) + c(y,\phi)\right\} \tag{2.3}$$

for the loglikelihood function considered as a function of θ and ϕ, y being given. The mean and variance can be easily derived from the well-known relations

$$E\left(\frac{\delta l}{\delta\theta}\right) = 0, \tag{2.4}$$

and

$$E\left(\frac{\delta^2 l}{\delta\theta^2}\right) + E\left(\frac{\delta l}{\delta\theta}\right)^2 = 0. \tag{2.5}$$

From (2.3) we have

$$\frac{\delta l}{\delta\theta} = \left\{\frac{y - b'(\theta)}{a(\phi)}\right\}, \tag{2.6}$$

and

$$\frac{\delta^2 l}{\delta\theta^2} = \frac{-b''(\theta)}{a(\phi)}. \tag{2.7}$$

From (2.4) and (2.6), we have

$$E\left(\frac{\delta l}{\delta \theta}\right) = \left\{\frac{\mu - b'(\theta)}{a(\phi)}\right\},$$

so that

$$E(y) = \mu = b'(\theta). \tag{2.8}$$

Similarly, from (2.5) and (2.7),

$$0 \equiv -\frac{b''(\theta)}{a(\phi)} + \frac{v(y)}{a^2(\phi)} \quad ,$$

so that

$$var(y) = b''(\theta)a(\phi). \tag{2.9}$$

Thus the variance of y is the product of two function; one, $b''(\theta)$, depends on the canonical parameter (and hence on the mean) only and will be called the variance function, while the other is independent of θ and depends only on ϕ.

2.1.3 Link Functions

A link function relates the linear predictor $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ to the mean response μ_i:

$$E[y_i] = \mu_i,$$

$$g(\mu_i) = \eta_i = \mathbf{x}_i'\boldsymbol{\beta}. \tag{2.10}$$

Each distribution in the exponential family has a link function for which there exists a sufficient statistic. A canonical link occurs when $\theta = \eta$, where θ is the canonical parameter in the exponential family.

In classical linear models, the mean and the linear predictor are identical, and the identity link is plausible in that both η and μ can take any value on the real line. However when we are dealing with counts and the distribution is Poisson, we must have $\mu > 0$, so that the identity link is less attractive, in part because η may be negative while μ not be. Models for counts based on independence in cross-classified data lead naturally to multiplicative effects, and this is expressed by the log link, $\eta = \log \mu$, with its inverse $\mu = e^{\eta}$. Now additive effects contributing to η become multiplicative effects contributing to μ, and μ is necessarily positive.

For the binomial distribution, we have $0 < \mu < 1$ and a link should satisfy the condition that it maps the interval $(0, 1)$ on to the whole real line. Now we can consider three principal functions, namely:

i. logit:

$$\eta = \log\left(\frac{\mu}{1-\mu}\right). \tag{2.11}$$

ii. probit:

$$\eta = \Phi^{-1}(\mu), \tag{2.12}$$

where $\Phi(.)$ is the normal cumulative distribution function.

iii. complementary log-log:

$$\eta = \log\{-\log(1-\mu)\}. \tag{2.13}$$

2.1.4 Fitting GLMs Using the ML Method

The log-likelihood for a single observation, in canonical form, is given by,

$$l_i = \frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi). \tag{2.14}$$

Now, differentiating (2.14) with respect to β_j,

$$\begin{aligned}
\frac{\delta l_i}{\delta \beta_j} &= \frac{\delta l_i}{\delta \theta_i}\frac{\delta \theta_i}{\delta \mu_i}\frac{\delta \mu_i}{\delta \eta_i}\frac{\delta \eta_i}{\delta \beta_j} \\
&= \frac{y_i - b'(\theta_i)}{a_i(\phi)}\frac{1}{b''(\theta_i)}\frac{\delta \mu_i}{\delta \eta_i}x_{ji} \\
&= (y_i - \mu_i)\frac{1}{a_i(\theta)b''(\theta_i)}\frac{\delta \mu_i}{\delta \eta_i}x_{ji} \\
&= (y_i - \mu_i)\frac{1}{v(y_i)}\left(\frac{\delta \mu_i}{\delta \eta_i}\right)^2\frac{\delta \eta_i}{\delta \mu_i}x_{ji}.
\end{aligned} \tag{2.15}$$

For a set of n observations $y_1, y_2, \ldots, y_n$, (2.15) can be written in matrix form as

$$(y - \mu).w.d.x_j \quad ,$$

where $w = \frac{1}{v(y_i)}\left(\frac{\delta \mu_i}{\delta \eta_i}\right)^2$ is called weight and the dispersion parameter $d = \frac{\delta \eta_i}{\delta \mu_i}$.

By Taylors series expansion, we can write

$$0 \equiv \frac{\delta l}{\delta \beta} \cong \frac{\delta l}{\delta \beta}\bigg|_{\beta_0} + \frac{\delta^2 l}{\delta \beta \delta \beta'}\bigg|_{\beta_0}(\beta - \beta_0), \tag{2.16}$$

where $\frac{\delta^2 l}{\delta \beta \delta \beta'}\Big|_{\beta_0} = I_0(\beta_0)$ is called Fisher observed information. In matrix notation, (2.15) can be written as

$$\frac{\delta l}{\delta \beta} = \mathbf{X}\mathbf{W}_0\mathbf{D}_0(\mathbf{y} - \boldsymbol{\mu}_0), \tag{2.17}$$

where $\mathbf{W}_0 = \text{diag}\left\{\frac{1}{v(y_i)}\left(\frac{\delta\mu_{0_i}}{\delta\eta_{0_i}}\right)^2\right\}$, $\mathbf{D}_0 = \text{diag}\left\{\frac{\delta\eta_{0_i}}{\delta\mu_{0_i}}\right\}$, $\boldsymbol{\mu}_0 = g^{-1}(\boldsymbol{\eta}_0)$ and $\boldsymbol{\eta}_0 = \mathbf{X}\boldsymbol{\beta}_0$.

Fisher's scoring technique replaces $I_0(\beta_0)$ in (2.16) by $I_E(\beta_0)$, where

$$I_E(\beta_0) = -E\left(\frac{\delta^2 l}{\delta\beta\delta\beta'}\right)\Big|_{\beta_0}.$$

From (2.15),

$$\frac{\delta l}{\delta\beta_j} = \sum w(y-\mu)\frac{\delta\eta}{\delta\mu}x_j. \tag{2.18}$$

Now, differentiating (2.18) with respect to β_k, we get

$$\frac{\delta^2 l}{\delta\beta_j\delta\beta_k} = \sum\left[(y-\mu)\frac{\delta}{\delta\beta_k}\left(w\frac{\delta\eta}{\delta\beta_k}x_j\right) + \left(w\frac{\delta\eta}{\delta\mu}x_j\right)\frac{\delta}{\delta\beta_k}(y-\mu)\right],$$

so that

$$E\left(\frac{\delta^2 l}{\delta\beta\delta\beta'}\right) = -\sum w\frac{\delta\eta}{\delta\mu}x_j\frac{\delta\mu}{\delta\beta_k}. \tag{2.19}$$

In matrix notation,

$$I_E(\boldsymbol{\beta}_0) = -E\left(\frac{\delta^2 l}{\delta\boldsymbol{\beta}\delta\boldsymbol{\beta}'}\right)\Big|_{\boldsymbol{\beta}_0} = \mathbf{X}'\mathbf{W}_0\mathbf{X}. \tag{2.20}$$

Hence from (2.16), we find

$$\frac{\delta l}{\delta\boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}_0} \approx I_E(\boldsymbol{\beta}_0)(\boldsymbol{\beta}-\boldsymbol{\beta}_0).$$

Now using NR and Fisher scoring, we can write

$$\begin{aligned}
\boldsymbol{\beta} &\cong \boldsymbol{\beta}_0 + [I_E(\boldsymbol{\beta}_0)]^{-1}\left(\frac{\delta l}{\delta\boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}_0}\right) \\
&= \boldsymbol{\beta}_0 + (\mathbf{X}'\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_0\mathbf{D}_0(\mathbf{y}-\boldsymbol{\mu}_0) \\
&= (\mathbf{X}'\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_0\left\{\mathbf{X}\boldsymbol{\beta}_0 + \mathbf{D}_0(\mathbf{y}-\boldsymbol{\mu}_0)\right\} \\
&= (\mathbf{X}'\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_0\mathbf{Z}_0,
\end{aligned} \tag{2.21}$$

where $\mathbf{z} = \mathbf{x}\beta_0 + \mathbf{D}_0(\mathbf{y} - \boldsymbol{\mu})$ is called a pseudo vector.

In particular,

$$\begin{aligned} \mathbf{z}_{i0} &= \mathbf{x}_i\boldsymbol{\beta}_0 + \left.\frac{\delta\eta_i}{\delta\mu_i}\right|_{\mu_{i0}} (y_i - \mu_{i0}) \\ &= \eta_i + \left.\frac{\delta\eta_i}{\delta\mu_i}\right|_{\mu_{i0}} (y_i - \mu_{i0}) \\ &= g(\mu_{i0}) + \left.\frac{\delta\eta_i}{\delta\mu_i}\right|_{\mu_{i0}} (y_i - \mu_{i0}), \end{aligned} \tag{2.22}$$

and

$$v(\mathbf{z}_{i0}) = \eta_i + \left(\left.\frac{\delta\eta_i}{\delta\mu_i}\right|_{\mu_{i0}}\right)^2 v(y_i). \tag{2.23}$$

Thus we can use the weighted regression of $\mathbf{z}_0$ on $\mathbf{x}$ with weights w_0. This method is referred to as the Iteratively Reweighted Least Square (IRWLS) method.

2.1.5 Test of Hypothesis

a. Likelihood Ratio Tests

Likelihood ratio tests follow the usual prescription of comparing the maximized values of the log likelihood both under H_0 and not restricted to H_0. If the difference is large (i.e., the unrestricted model fit is much better), then H_0 is rejected.

When there are multiple parameters we will often be interested in hypothesis concerning only a subset of the parameters. Accordingly, let the parameter vector $\boldsymbol{\theta}$ be partitioned into two components $\boldsymbol{\theta}' = (\boldsymbol{\theta}_1', \boldsymbol{\theta}_2')$ and suppose interest focuses on $\boldsymbol{\theta}_1$ while $\boldsymbol{\theta}_2$ is left unspecified. $\boldsymbol{\theta}_2$ is often called a nuisance parameter. Either or

both of θ_1 and θ_2 could be vector valued.

Suppose our hypothesis is of the form $H_0 : \theta_1 = \theta_{10}$, where θ_{10} is a specified value of θ_1, and let $\hat{\theta}_{20}$ be the MLE of θ_2 under the restriction that $\theta_1 = \theta_{10}$. The likelihood ratio test statistic is given by

$$-2\log\Lambda = -2\left[l(\theta'_{10}, \hat{\theta}_{20}) - l(\hat{\theta}_1, \hat{\theta}_2)\right], \tag{2.24}$$

where $\hat{\theta}' = (\hat{\theta}'_1, \hat{\theta}'_1)$ and the large-sample critical region of the test is to reject H_0 in favor of the alternative when

$$-2\log\Lambda > \chi^2_{v,1-\alpha}, \tag{2.25}$$

where v is the dimension of θ_1.

b. Wald Tests

As alternative method of testing is to use the large-sample normality of the ML estimator in order to form a test. From standard results,

$$\sqrt{n}(\hat{\theta} - \theta) \sim \mathcal{AN}[0, \mathbf{I}^{-1}(\theta)], \tag{2.26}$$

where $\mathbf{I}(\theta)$ is the Fisher information for $\hat{\theta}$. Again, if we write $\theta' = (\theta'_1, \theta'_2)$, and write conformably

$$\mathbf{I}(\theta) = \begin{bmatrix} \mathbf{I}_{11} & \mathbf{I}_{12} \\ \mathbf{I}_{21} & \mathbf{I}_{22} \end{bmatrix}, \tag{2.27}$$

then standard matrix algebra for partitioned matrices and multivariate calculations show that the asymptotic variance of $\hat{\theta}_1$ is given by

$$var(\hat{\theta}_1) = \left(\mathbf{I}_{11} - \mathbf{I}_{12}\mathbf{I}_{22}^{-1}\mathbf{I}_{21}\right)^{-1}. \tag{2.28}$$

To test $H_0 : \theta_1 = \theta_{10}$, we can form the Wald statistic

$$W = (\hat{\theta}_1 - \theta_{10})' \left[var(\hat{\theta}_1)\right]^{-1} (\hat{\theta}_1 - \theta_{10}), \tag{2.29}$$

which under H_0 has the same large-sample χ^2 distribution as the LRT with degrees of freedom equal to the dimension of θ_1. More explicitly we would reject the H_0 : $\theta_1 = \theta_{10}$ if

$$W > \chi^2_{v,1-\alpha}. \tag{2.30}$$

c. Confidence Intervals

Either the LRT or Wald test can be used to construct large-sample confidence interfvals for θ_1. For the LRT we include in the confidence set all values θ_1 such that

$$-2\left[l(\theta_1', \hat{\theta}_{2,1}) - l(\hat{\theta}_1, \hat{\theta}_2)\right] \leq \chi^2_{v,1-\alpha}. \tag{2.31}$$

In (2.31), $\hat{\theta}_{2,1}$ represents the MLE of θ_2 for each value of θ_1 checked for inclusion in the set.

For the Wald test we include the confidence set of all values of θ_1 such that

$$(\hat{\theta}_1 - \theta_1)' \left[var_\infty(\hat{\theta}_1)\right]^{-1} (\hat{\theta}_1 - \theta_1) \leq \chi^2_{v,1-\alpha}. \tag{2.32}$$

The computational burden of the likelihood-based confidence interval is thus larger than that for the Wald-based interval. However, the small and moderate-sized sample performance of the LRT-based confidence region has generally been found to be better.

2.1.6 Example: Binary Logistic Model

Consider a binary model,

$$y_i \sim indep.\ Bernoulli(p_i), \quad i = 1, \ldots, n; \tag{2.33}$$

$$E[y_i] = p_i, \tag{2.34}$$

$$logit(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta x_i. \tag{2.35}$$

Here the data y are binary, $logit(p_i)$ is the link function (since it links together the mean of y_i and the linear form of predictors), x_i is the ith observed covariate, α and β are the parameters in the linear predictor. Note that, β is the log odds ratio.

a. Likelihood

Since the y_i's are independent and Bernoulli, the likelihood function is easy to evaluate:

$$\begin{aligned} L &= \prod_i^n p_i^{y_i} (1-p_i)^{(1-y_i)} \\ &= \prod_i^n \left\{\frac{p_i}{(1-p_i)}\right\}^{y_i} (1-p_i). \end{aligned} \tag{2.36}$$

Using

$$\frac{p_i}{(1-p_i)} = \left(1 + e^{\alpha+\beta x_i}\right),$$

and

$$1 - p_i = \left(1 + e^{\alpha+\beta x_i}\right)^{-1},$$

the likelihood can be expressed as

$$L = \prod_i^n e^{y_i(\alpha+\beta x_i)} \left(1 + e^{\alpha+\beta x_i}\right)^{-1}, \tag{2.37}$$

and the log-likelihood as

$$l = \log L = \sum_i^n y_i(\alpha + \beta x_i) - \log(1 + e^{\alpha+\beta x_i}). \tag{2.38}$$

b. ML Estimation

Differentiating (2.38) with respect to α and β gives

$$\begin{aligned} \frac{\delta l}{\delta \alpha} &= \sum_{i=1}^n \left(y_i - \frac{e^{\alpha+\beta x_i}}{1 + e^{\alpha+\beta x_i}} \right) \\ &= \sum_{i=1}^n \left(y_i - \frac{1}{1 + e^{-(\alpha+\beta x_i)}} \right). \end{aligned} \tag{2.39}$$

Using $p_i = \frac{1}{1+e^{-(\alpha+\beta x_i)}}$, we have

$$\frac{\delta l}{\delta \alpha} = \sum_{i=1}^n (y_i - p_i), \tag{2.40}$$

$$\begin{aligned} \frac{\delta l}{\delta \beta} &= \sum_{i=1}^n \left(x_i y_i - \frac{x_i e^{\alpha+\beta x_i}}{1 + e^{\alpha+\beta x_i}} \right) \\ &= \sum_{i=1}^n x_i(y_i - p_i). \end{aligned} \tag{2.41}$$

The second derivatives of l take a convenient form:

$$\begin{aligned}\frac{\delta^2 l}{\delta\alpha^2} &= -\sum_{i=1}^{n}\frac{e^{-(\alpha+\beta x_i)}}{\left(1+e^{-(\alpha+\beta x_i)}\right)^2} \\ &= -\sum_{i=1}^{n} p_i(1-p_i), \end{aligned} \tag{2.42}$$

$$\begin{aligned}\frac{\delta^2 l}{\delta\alpha\delta\beta} &= -\sum_{i=1}^{n}\frac{x_i e^{-(\alpha+\beta x_i)}}{\left(1+e^{-(\alpha+\beta x_i)}\right)^2} \\ &= -\sum_{i=1}^{n} x_i p_i(1-p_i), \end{aligned} \tag{2.43}$$

and

$$\frac{\delta^2 l}{\delta\beta^2} = -\sum_{i=1}^{n} x_i^2 p_i(1-p_i) \tag{2.44}$$

with $\mathbf{W} = \begin{pmatrix} p_1(1-p_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & p_n(1-p_n) \end{pmatrix}$ and $\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$. Then by (2.20) we can compactly write the information matrix as

$$-E\begin{bmatrix} l_{\alpha\alpha} & l_{\alpha\beta} \\ l_{\beta\alpha} & l_{\beta\beta} \end{bmatrix} = \mathbf{X}'\mathbf{W}\mathbf{X}. \tag{2.45}$$

This yields a convenient computing algorithm to find $\hat{\alpha}$ and $\hat{\beta}$. Following (2.21), the algorithm proceeds as follows, with m denoting the iteration number and superscripts indicating sequential values of the parameters:

1. Obtain starting values $\alpha^{(0)}$ and $\beta^{(0)}$. Set $m = 0$.

2. Calculate

$$\begin{pmatrix} \alpha^{(m+1)} \\ \beta^{(m+1)} \end{pmatrix} = \begin{pmatrix} \alpha^{(m)} \\ \beta^{(m)} \end{pmatrix} + (\mathbf{X}'\mathbf{W}^{(m)}\mathbf{X})^{-1}\mathbf{X}'[\mathbf{y} - \mathbf{p}^{(m)}]. \tag{2.46}$$

3. Check for convergence of $\begin{pmatrix} \alpha^{(m+1)} \\ \beta^{(m+1)} \end{pmatrix}$. If it has converged, stop; otherwise set $m = m + 1$ and return to step 2.

In this algorithm $\mathbf{p}^{(m)}$ is the notation we used for the vector

$$\begin{pmatrix} 1/(1 + e^{-(\alpha^{(m)} + \beta^{(m)} x_1)}) \\ \vdots \\ 1/(1 + e^{-(\alpha^{(m)} + \beta^{(m)} x_n)}) \end{pmatrix} \text{ and } \mathbf{W}^{(m)} = \text{diag}\left\{p_i^{(m)}(1 - p_i^{(m)})\right\}.$$

2.2 Linear Mixed Model (LMM)

The starting point for an LM is $E[\mathbf{y}] = \mathbf{X}\beta$ with β being fixed effects; for an LMM we still use $\mathbf{X}\beta$ for effects but add to it $\mathbf{Zu}$ where $\mathbf{Z}$, like $\mathbf{X}$, is a known (model) matrix and $\mathbf{u}$ is the vector of random effects that occur in the vector $\mathbf{y}$. Although the elements of $\mathbf{u}$ are random variables, it is convenient to specify the model conditional on their unobservable but realized values. The conditional mean of $\mathbf{y}$ is defined as

$$E[\mathbf{y}|\mathbf{u}] = \mathbf{X}\beta + \mathbf{Zu}. \tag{2.47}$$

It is assumed that

$$\mathbf{u} \sim (\mathbf{0}, \mathbf{D}), \quad \text{that is} \quad E[\mathbf{u}] = 0 \quad \text{and} \quad var(\mathbf{u}) = \mathbf{D}. \tag{2.48}$$

For specifying $var(\mathbf{y})$, we have $var(\mathbf{u}) = \mathbf{D}$ from (2.48) and assume

$$var(\mathbf{y}|\mathbf{u}) = \mathbf{R}. \tag{2.49}$$

Then the marginal mean and variance of $\mathbf{y}$ can be obtained as:

$$E(\mathbf{y}) = \mathbf{X}\beta \qquad \text{and} \qquad var(\mathbf{y}) = \mathbf{ZDZ}' + \mathbf{R}. \tag{2.50}$$

It is clear that the fixed effects enter only the mean whereas the random effects variance components enter only the variance of $\mathbf{y}$. The regression parameters β can be estimated using the best linear unbiased estimation (BLUE) and the variance components can be estimated by the ML or REML method.

2.3 Generalized Linear Mixed Model (GLMM)

Generalized linear mixed models (GLMMs) are a natural extension of both linear mixed models and generalized linear models. They are useful for the accommodation of non-normally distributed responses and specification of a possibly nonlinear link between the mean response and the predictors, and for calculating overdispersion and correlation by incorporating random effects. GLMMs are particularly useful for structuring multiple sources of variation, including components associated with measured factors, such as covariates and variation attributed to unmeasured factors or random effects.

GLMMs are widely used in the analysis of clustered data, including longitudinal data or repeated measurements (Breslow and Clayton, 1993)[5]. These mixed models are useful for accommodating the overdispersion often observed among nonnormally distributed responses and for modeling the dependence among responses inherent in longitudinal or repeated measures data by incorporating random effects (Stiratelli, Laird, andWare 1984 [32]; Zeger, Liang, and Albert 1988 [37]). A full maximum likelihood (ML) analysis based on the joint marginal likelihood of the responses can be used for estimating both fixed- and random-effects parameters in GLMMs, which requires numerical integration techniques for calculating the log-likelihood, score equations, and information matrix. However, its use in GLMMs is limited to relatively simple models, and it is found intractable for more complicated problems involving irreducibly high-dimensional integrals. To avoid such computational problems, a number of Bayesian approaches have been suggested that generate repeated samples from the posterior distributions of the random effects using Gibbs sampling techniques (Besag, York, and Mollie 1991 [4]; Zeger and Karim 1991 [36]).

McCulloch (1994) [18] investigated a Monte Carlo EM (MCEM) approach for analyzing models with complicated fixed- and random-effects structure but is limited to binary data with probit link. McCulloch (1997) [19] developed a Monte Carlo Newton-Raphson (MCNR) algorithm for approximating the ML estimates in GLMMs. The MCNR estimates were compared to the exact ML likelihood estimates

for simple models, and it was found that MCNR inherits the properties of the exact ML estimates. Recently, Sutradhar and Sinha (2002) [33] developed a pseudo-likelihood approach for estimating the variance components of a binary mixed model for longitudinal data. This approach is based on an assumption that the variance components of the random effects are small in magnitude.

Although these likelihood algorithms are useful for fitting the GLMMs efficiently under strict model assumptions, they can be highly in influenced by the presence of unusual data points. Sinha (2004) [29] developed a technique for finding robust maximum likelihood (RML) estimates of the model parameters in GLMMs, which appears to be useful in downweighting the influential data points when estimating the parameters. The asymptotic properties of the robust estimators are investigated under some regularity conditions. To avoid the computational problems involving highdimensional integrals, he proposed a robust Monte Carlo Newton-Raphson (RMCNR) algorithm for fitting GLMMs. Sinha (2006) [30], developed a robust quasi-likelihood method, which appears to be useful for downweighting any influential data points when estimating the model parameters. He illustrated the computational issues of the method in an example. He used simulations to study the behavior of the robust estimates when data are contaminated with outliers, and he compared these estimates to those obtained by the ordinary quasi-likelihood method.

The EM algorithm is often used for finding the maximum likelihood estimates

in generalized linear models with incomplete data. Sinha (2008)[31] presented a robust method in the framework of the maximum likelihood estimation for fitting generalized linear models when nonignorable covariates are missing. His robust approach is useful for downweighting any influential observations when estimating the model parameters. To avoid computational problems involving irreducibly high-dimensional integrals, he adopted a Metropolis-Hastings algorithm based on a Markov chain sampling method. He carried out simulations to investigate the behavior of the robust estimates in the presence of outliers and missing covariates; furthermore, he compared these estimates to the classical maximum likelihood estimates. Finally, he illustrated his approach using data on the occurrence of delirium in patients operated on for abdominal aortic aneurysm.

However, the sensitivity of the estimated regression coefficients to random effects assumptions in mixed models has only been explored to a limited degree. Neuhaus et al. (1992) [22] investigate the effects of mixture distribution misspecification when fitting mixed-effects logistic models. They concluded that although regression estimates are asymptotically biased, the magnitude of bias is typically small. Liang and Hanfelt (1994) [15] considered likelihood inference using the beta binomial model and showed that ignoring the variation in the within cluster correlation as a function of cavariates can result in severe bias for regression estimates. Ten Have et al. (1999) [35] studied binary data with multiple levels of clustering and showed that fitting an incomplete multilevel structure can lead to

bias in the estimates. Tan et al. (1999) [34] showed that using generalized estimating equations for an underlying latent variable model does not lead to biased estimation or inference.

Heagerty and Kurland (2001)[13] investigated the impact of model violations on the estimate of a regression coefficient in a GLMM. They evaluated the asymptotic relative bias that results from incorrect assumptions regarding the random effects. They found, a marginally specified regression structure that is estimated using maximum likelihood is much less susceptible to bias resulting from random effects model misspecification.

Feng et al. (1996) [12] and Sherman and le Cessie (1997) [27] used a nonparametric bootstrap approach on independent clusters and observed that, in general, when the robust estimate of the standard error was used GEE provided comparable coverage to the bootstrap. Feng et al. (1996) [12] found that the bootstrap estimates were more efficient when the sample size was small. Overall, these authors concluded that the bootstrap approach is superior in small and large samples compared to other methods used for correlated data, including the GEE and mixed linear models.

2.3.1 Structure of the Generalized Linear Mixed Model

The use of random factors is not restricted to linear mixed models. For many reasons, we may want to incorporate random factors into nonlinear models. We

may wish to build a model that accommodates correlated data, or to consider the levels of a factor from a selected population in order to make inference about the population. A basic linear model has mean $E[\mathbf{y}] = \mathbf{X}\boldsymbol{\beta}$. We incorporate random effects by enlarging the model as $E[\mathbf{y}|\mathbf{u}] = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}$. If we write a combined model matrix $\mathbf{X}^* = [\mathbf{X}\mathbf{Z}]$ and an enlarged "parameter" vector $\boldsymbol{\beta}^* = [\boldsymbol{\beta}', \mathbf{u}']$, we obtain $E[\mathbf{y}] = \mathbf{X}^*\boldsymbol{\beta}^*$.

This suggests a straightforward extension of the generalized linear models. Append the random effects in the form $\mathbf{Z}\mathbf{u}$ to the linear predictor $\mathbf{X}\boldsymbol{\beta}$. This will achieve the two main goals of incorporating correlation and allowing broader inference. In this section, we define the generalized linear mixed model (GLMM), explore the consequences of adding random factors and discuss a variety of inferential methods.

Conditional distribution of y

To specify the model, we start with the conditional distribution of $\mathbf{y}$ given $\mathbf{u}$. As in (2.2) and (2.3), the response vector $\mathbf{y}$ is typically, but not necessarily, assumed to consist of conditionally independent elements, each with a distribution with density from the exponential family:

$$y_i|\mathbf{u} \sim indep. \quad f_{y_i|u}(y_i|\mathbf{u}),$$

$$f_{y_i|\mathbf{u}}(y_i|\mathbf{u}) = exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i;\phi)\right\}. \tag{2.51}$$

From (2.8), we know the conditional mean of y_i is related to θ_i in (2.51). It is a transformation of this mean that we wish to model as a linear model in both fixed and random factors:

$$E(y_i|\mathbf{u}) = b'(\theta_i) = \mu_i,$$

$$g(\mu_i) = \theta_i = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i'\mathbf{u}. \tag{2.52}$$

As in section 2.2, $g(.)$ is a known function, called the link function, $\mathbf{x}_i'$ is the i-th row of the design matrix $\mathbf{X}$ for the fixed effects, and $\boldsymbol{\beta}$ is the fixed effects parameter vector. To that specification we have added $\mathbf{z}_i'$, which is the i-th row of the design matrix for the random effects, and $\mathbf{u}$, the random effect vector of dimension q. Note that we are using μ_i here to denote the conditional mean of y_i given $\mathbf{u}$, not the unconditional mean. To complete the specification, we assign a distribution to the random effects:

$$\mathbf{u} \sim f_{\mathbf{U}}(\mathbf{u}). \tag{2.53}$$

In light of the fact that the conditional distribution of $\mathbf{y}$ given $\mathbf{u}$ is just a notational extension of the generalized linear model, μ_i represents the conditional mean of y_i.

2.3.2 ML Estimation

The likelihood function for GLMMs can be obtained as

$$\begin{aligned} L &= \int f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) f_{\mathbf{u}}(\mathbf{u}) d\mathbf{u} \\ &= \int \left[\prod_{i=1}^{n} f_{\mathbf{y}_i|\mathbf{u}}(\mathbf{y}_i|\mathbf{u}) \right] f_{\mathbf{u}}(\mathbf{u}) d\mathbf{u}, \end{aligned} \tag{2.54}$$

where the integration is over the q-dimensional distribution of $\mathbf{u}$.

Example: The Poisson Mixed Model

Consider the Poisson mixed model with normally distributed random effects:

$$y_{ij} \sim indep.Poisson(\mu_{ij}), \quad i = 1, 2, \ldots, m; \; j = 1, 2, \ldots, n_i;$$

$$\log(\mu_{ij}) = \mathbf{x}'_{ij}\boldsymbol{\beta} + u_i,$$

$$u_i \sim i.i.d. \quad \mathcal{N}(0, \sigma_u^2). \tag{2.55}$$

This uses a log-link function and a normal distribution for the random cluster effects. For model (2.55), the log-likelihood is

$$
\begin{aligned}
l &= \log\left[\prod_{i=1}^{m}\int_{-\infty}^{+\infty} f_{\mathbf{y}_i|u_i}(\mathbf{y}_i|u_i) f_{\mathbf{u}}(u_i) du_i\right] \\
&= \log\left[\prod_{i=1}^{m}\int_{-\infty}^{+\infty}\left[\prod_{j=1}^{n_i} f_{y_{ij}|u_i}(y_{ij}|u_i)\right] f_{\mathbf{u}}(u_i) du_i\right] \\
&= \log\left[\prod_{i=1}^{m}\int_{-\infty}^{+\infty}\left[\prod_{j=1}^{n_i}\frac{e^{\mu_{ij}}\mu_{ij}^{y_{ij}}}{y_{ij}!}\right] f_{\mathbf{u}}(u_i) du_i\right] \\
&= \sum_{i=1}^{m}\log\left[\int_{-\infty}^{+\infty}\frac{e^{\left(-\sum_{j=1}^{n_i}\mu_{ij}+\sum_{j=1}n_i y_{ij}\ \log\mu_{ij}\right)}}{\prod_{j=1}^{n_i}(y_{ij}!)} f_{\mathbf{u}}(u_i) du_i\right] \\
&= \sum_{i=1}^{m}\log\left[\frac{1}{\prod_{j=1}^{n_i}(y_{ij}!)}\int_{-\infty}^{+\infty} e^{-\sum_{j=1}^{n_i}\mu_{ij}+\sum_{j=1}^{n_i} y_{ij}(\mathbf{x}_{ij}\boldsymbol{\beta}+u_i)} f_{\mathbf{u}}(u_i) du_i\right] \\
&= \sum_{i=1}^{m}\left[-\sum_{j=1}^{n_i}\log(y_{ij}!)+\sum_{j=i}^{n} y_{ij}\mathbf{x}'_{ij}\boldsymbol{\beta}\right]+\sum_{i=1}^{m}\log\int_{-\infty}^{+\infty}\left[e^{-\sum_j\mu'_{ij}+\sum_j y_{ij}u_i}\right] f_{\mathbf{u}}(u_i) du_i \\
&= \mathbf{y}'\mathbf{X}\boldsymbol{\beta}-\sum_{ij}\log(y_{ij}!)+\sum_{i=1}^{m}\log\int_{-\infty}^{+\infty}\left[e^{-\sum_j\mu'_{ij}+u_i y_{i.}}\right] f_{\mathbf{u}}(u_i) du_i. \qquad (2.56)
\end{aligned}
$$

Unfortunately, (2.56) cannot be simplified further or evaluated in closed form and hence maximizing values cannot be expressed in closed form either.

In the simplest case, numerical integration for calculating the likelihood is very straightforward and hence numerical maximization of the likelihood is not too difficult. For example, for (2.55) as seen in (2.56), the log likelihood is the sum of independent contributions from each cluster, each of which involves just a single-dimensional integral. This integral can be evaluated accurately using standard quadrature techniques.

Fixed Effect Parameters

Even though the likelihood equations are numerically difficult, we can write them in simpler form. From (2.54), we get,

$$l = \log \int f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) f_{\mathbf{u}}(\mathbf{u}) d\mathbf{u} = \log f_{\mathbf{y}}(\mathbf{y}), \tag{2.57}$$

so that

$$\begin{aligned} \frac{dl}{d\boldsymbol{\beta}} &= \frac{\delta}{\delta \boldsymbol{\beta}} \int f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) f_{\mathbf{u}}(\mathbf{u}) d\mathbf{u} \frac{1}{f_{\mathbf{y}}(\mathbf{y})} \\ &= \int \left[\frac{\delta}{\delta \boldsymbol{\beta}} f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) \right] f_{\mathbf{u}}(\mathbf{u}) d\mathbf{u} \frac{1}{f_{\mathbf{y}}(\mathbf{y})} \end{aligned} \tag{2.58}$$

since $f_{\mathbf{u}}(\mathbf{u})$ does not involve $\boldsymbol{\beta}$. Noting that

$$\begin{aligned} \frac{\delta}{\delta \boldsymbol{\beta}} f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) &= \left(\frac{1}{f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})} \frac{\delta f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})}{\delta \boldsymbol{\beta}} \right) f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) \\ &= \frac{\delta \log f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})}{\delta \boldsymbol{\beta}} f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}), \end{aligned} \tag{2.59}$$

we can write (2.58) as

$$\begin{aligned} \frac{dl}{d\boldsymbol{\beta}} &= \int \frac{\delta \log f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})}{\delta \boldsymbol{\beta}} f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u}) f_U(\mathbf{u}) d\mathbf{u} \frac{1}{f_{\mathbf{y}}(\mathbf{y})} \\ &= \int \frac{\delta \log f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})}{\delta \boldsymbol{\beta}} f_{\mathbf{u}|\mathbf{y}}(\mathbf{u}|\mathbf{y}) d\mathbf{u}. \end{aligned} \tag{2.60}$$

Using (2.17) which gives the derivative of the log-likelihood for a GLM, (2.60) gives

$$\begin{aligned} \frac{dl}{d\boldsymbol{\beta}} &= \int \mathbf{X}' \mathbf{W}^* (\mathbf{y} - \boldsymbol{\mu}) f_{\mathbf{u}|\mathbf{y}}(\mathbf{u}|\mathbf{y}) d\mathbf{u} \\ &= \mathbf{X}' \mathbf{y} E\left[\mathbf{W}^* | \mathbf{y}\right] - \mathbf{X}' E\left[\mathbf{W}^* \boldsymbol{\mu} | \mathbf{y}\right] \end{aligned} \tag{2.61}$$

where $\mathbf{W}^* = \left\{ {}_d \left[a_i(\phi) v(\mu_i) g_{\mu}(\mu_i) \right]^{-1} \right\}$.

The likelihood equation for β is therefore,

$$\mathbf{X}'\mathbf{y}E\left[\mathbf{W}^*|\mathbf{y}\right] = \mathbf{X}'E\left[\mathbf{W}^*\boldsymbol{\mu}|\mathbf{y}\right]. \tag{2.62}$$

Random Effect Parameters

A result similar to (2.60) can be derived for the ML equations for the parameters in the distribution of $f_{\mathbf{u}}(\mathbf{u})$. Let φ denote those parameters, so that

$$\begin{aligned} \frac{dl}{d\varphi} &= \int \frac{\delta \log f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})}{\delta\varphi} f_{\mathbf{u}|\mathbf{y}}(\mathbf{u}|\mathbf{y})d\mathbf{u} \\ &= E\left[\frac{\delta \log f_{\mathbf{y}|\mathbf{u}}(\mathbf{y}|\mathbf{u})}{\delta\varphi}\middle|\mathbf{y}\right]. \end{aligned} \tag{2.63}$$

2.4 Computing ML Estimates

2.4.1 Gauss-Hermite Quadrature

Generalized linear mixed models pose special challenges beyond linear mixed models because of the high dimensional integration required to evaluate (hence maximize) the likelihood. We start by considering a GLMM with a single, normally distributed random effect. Let y_{ij} be the j-th observation corresponding to the i-th level of the random effect so that

$$y_{ij}|\mathbf{u} \sim indep. f_{Y_{ij}|\mathbf{U}}(y_{ij}|\mathbf{u}),$$

$$f_{Y_{ij}|\mathbf{u}}(y_{ij}|\mathbf{u}) = exp\left\{\frac{y_{ij}\theta_{ij} - b(\theta_{ij})}{a_i(\phi)} + c(y_{ij},\phi)\right\},$$

$$E\left[y_{ij}|\mathbf{u}\right] = \mu_{ij}, \tag{2.64}$$

$$g(\mu_{ij}) = \mathbf{x}'_{ij}\boldsymbol{\beta} + u_i,$$

with

$$u_i \sim i.i.d.\mathcal{N}(0, \sigma^2_{\mathbf{u}}). \tag{2.65}$$

The likelihood for the model is

$$\begin{aligned} L &= \int \prod_{ij} f_{y_{ij}|u_i}(y_{ij}|u_i) f_{u_i}(u_i) du_i \\ &= \prod_i \int_{-\infty}^{+\infty} e^{\sum_j \frac{y_{ij}\theta_{ij} - b(\theta_{ij})}{a_i(\phi)} + \sum_j c(y_{ij},\phi)} \frac{e^{-u_i^2/(2\sigma^2_{\mathbf{u}})}}{\sqrt{2\pi\sigma^2_{\mathbf{u}}}} du_i \\ &= \prod_i \int_{-\infty}^{+\infty} h_i(u_i) \frac{e^{-u_i^2/(2\sigma^2_{\mathbf{u}})}}{\sqrt{2\pi\sigma^2_{\mathbf{u}}}} du_i, \end{aligned} \tag{2.66}$$

where $h_i(u_i) = e^{\sum_j \frac{y_{ij}\theta_{ij} - b(\theta_{ij})}{a_i(\phi)} + \sum_j c(y_{ij},\phi)}$ and θ_{ij} is a function of u_i.

It can be seen that the likelihood is the product of one-dimensional integrals of the form

$$\int_{-\infty}^{+\infty} h(u) \frac{e^{-u^2/(2\sigma^2_{\mathbf{u}})}}{\sqrt{2\pi\sigma^2_{\mathbf{u}}}} du$$

which, upon a change of variables of $u = \sqrt{2\sigma_u}v$, can be written as

$$\int_{-\infty}^{+\infty} h(\sqrt{2\sigma_u}v) \frac{e^{-v^2}}{\sqrt{\pi}} dv \equiv \int_{-\infty}^{+\infty} h^*(v) e^{-v^2} dv, \tag{2.67}$$

where $h^* = h(\sqrt{2\sigma_u}v)/\sqrt{\pi}$.

Numerical integration over an unbounded range can be difficult. However, for integrals of smooth functions $h^*(.)$ multiplied by the function e^{-v^2}, the method of

Gauss-Hermite-quadrature is available. This approximates the integral in (2.67) as a weighted sum

$$\int_{-\infty}^{+\infty} h^*(v)e^{-v^2}dv = \sum_{k=1}^{d} h^*(x_k)w_k, \tag{2.68}$$

where the weights w_k and the evaluation points x_k are designed to provide an accurate approximation in the case where $h^*(.)$ is a polynomial. Abramowitz and Stegun (1964), calculated x_k and w_k in the following forms

$$x_k = i\text{-th zero of} H_n(x),$$

$$w_k = \frac{2^{n-1}n!\sqrt{\pi}}{n^2\left[H_{n-1}(x_k)\right]^2}, \tag{2.69}$$

where $H_n(x)$ is the Hermite polynomial of degree n. By using quadrature of a high-enough degree, accurate approximation can be calculated to integrals of functions that are similar to those of any high-degree polynomial.

2.4.2 Newton-Raphson Method

Recall the GLMM in the general form:

$$y_i|\mathbf{u} \sim indep. f_{Y_i|\mathbf{u}}(y_i|\mathbf{u}),$$

$$f_{Y_i|\mathbf{u}}(y_i|\mathbf{u}) = exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i,\phi)\right\},$$

$$E\left[y_i|\mathbf{u}\right] = \mu_i, \tag{2.70}$$

$$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}'\mathbf{u},$$

$$\mathbf{u} \sim f_{\mathbf{u}}(\mathbf{u}|\mathbf{D}),$$

where $\mathbf{D}$ represents the parameters governing the distribution of $\mathbf{u}$.

Whenever the marginal density of $\mathbf{y}$ is of the form (2.70) with separate parameters for $f_{\mathbf{Y}|\mathbf{u}}$ and $f_{\mathbf{u}}$, then the ML estimations for $\boldsymbol{\theta} = (\boldsymbol{\beta}', \phi)$ and $\mathbf{D}$ take the following form:

$$E\left[\frac{\delta f_{Y|\mathbf{u}}(\mathbf{y}|\mathbf{u},\boldsymbol{\theta})}{\delta\boldsymbol{\theta}}\middle|\mathbf{y}\right] = 0, \tag{2.71}$$

$$E\left[\frac{\delta f_{\mathbf{u}}(\mathbf{u}|\mathbf{D})}{\delta\mathbf{D}}\middle|\mathbf{y}\right] = 0. \tag{2.72}$$

Equation (2.72) involves only the distribution of $\mathbf{u}$ and is often fairly easy to solve, e.g. when the distribution is normal. On the other hand, (2.71) is amenable to Newton-Raphson or scoring approach.

Expanding $\frac{\delta f_{Y|\mathbf{u}}(\mathbf{y}|\mathbf{u},\boldsymbol{\theta})}{\delta\boldsymbol{\beta}}$ as a function of $\boldsymbol{\beta}$ around a value $\boldsymbol{\theta}_0$ gives

$$\frac{\delta f_{Y|\mathbf{u}}(\mathbf{y}|\mathbf{u},\boldsymbol{\theta})}{\delta\boldsymbol{\beta}} = \frac{\delta f_{Y|\mathbf{u}}(\mathbf{y}|\mathbf{u},\boldsymbol{\theta})}{\delta\boldsymbol{\beta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} + \frac{\delta^2 f_{Y|\mathbf{u}}(\mathbf{y}|\mathbf{u},\boldsymbol{\theta})}{\delta\boldsymbol{\beta}\delta\boldsymbol{\beta}'}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}. \tag{2.73}$$

Specializing this to our model, and dropping the term with a conditional expected value of zero, the formula for a scoring-type algorithm becomes

$$\frac{\delta f_{Y|\mathbf{u}}(\mathbf{y}|\mathbf{u},\boldsymbol{\theta})}{\delta\boldsymbol{\beta}} \cong \frac{1}{\phi}\mathbf{X}'\mathbf{W}\Delta(\mathbf{y}-\mathbf{u}) - \frac{1}{\phi}\mathbf{X}'\mathbf{W}\mathbf{X}(\boldsymbol{\beta}-\boldsymbol{\beta}_0), \tag{2.74}$$

where $\mathbf{W}^* = \left\{{}_d\left[v(\mu_i)g^2_{\boldsymbol{\mu}}(\mu_i)\right]^{-1}\right\}$ and $\Delta = \{{}_c g_{\boldsymbol{\mu}}(\mu_i)\}$. It is clear that $\mathbf{W}$, Δ and $\mu = E[\mathbf{y}|\mathbf{u}]$ are all functions of $\mathbf{u}$ and that all parameters are evaluated as $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Using this approximation in (2.4.8) leads to an iterative equation of the form

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} + \left(\mathbf{X}'E[\mathbf{W}|\mathbf{y}]\,\mathbf{X}\right)^{-1}\mathbf{X}'E[\mathbf{W}\Delta(\mathbf{y}-\boldsymbol{\mu})|\mathbf{y}]. \tag{2.75}$$

All terms on the right side are evaluated at $\beta^{(m)}$. This analog of scoring would proceed by iteratively solving (2.72), (2.75) and an equation for ϕ.

Chapter 3

Genetic Analysis and Bootstrap Methods

This chapter presents the preliminaries of the statistical genetic analysis. We also review the nonparametric bootstrap estimation procedure and the hierarchical bootstrap in this chapter.

3.1 Methods for Genetic Analysis

3.1.1 Genetics Background

The classical genetic definitions of interest to us predate the modern molecular era. *Genes* occur at different site, or *loci*, along a *chromosome*. Each locus can be occupied by several variant genes called *alleles*. Most human cells contain 46

chromosomes. Two of these are *sex* chromosomes — two paired X's for female, and an X and Y for a male. The remaining 22 homologous pairs of chromosomes are *autosomes*. One member of each chromosome pair is maternally derived via egg; the other member is paternally derived by sperm. Except for the sex chromosomes, it follows that there are genes at every locus. If the two alleles are identical, then the person is a *homozygote*; otherwise, he is a *heterozygote*. Typically, one denotes a genotype by two allele symbols separated by a slash /. Genotypes may not be observable. By definition, what is observable is a person's *phenotype*.

A simple example will serve to illustrate these definitions. The ABO locus resides on the long run of chromosome 9 at band q34. This locus determines detectable *antigens* on the surface of red blood cells. There are three alleles, A,B and O, which determine an A antigen, a B antigen, and the absence of either antigen, respectively. Phenotypes are recorded by reacting antibodies for A and B against a blood sample. The four observable phenotypes are A (antigen A alone detected), B (B along detected), AB (antigens A and B both detected) and O (neither antigen A nor B detected). These correspond to the genotype sets given in Table 3.1.

Table 3.1: *Phenotypes at the ABO locus*

Phenotype	**Genotype**
A	$A/A, A/O$
B	$B/B, B/O$
AB	A/B
O	O/O

Note that phenotype A results from either the homozygous genotype A/A or the heterozygous genotype A/O; similarly, phenotype B results from either B/B or B/O. Alleles A and B both mask the presence of the O allele and are said to be *dominant* to it. Alternatively, O is *recessive* to A and B. Relative to one another, alleles A and B are *codominant*.

The six genotypes listed above at the ABO locus are unordered in the sense that maternal and paternal contributions are not distinguished. In some cases it is helpful to deal with *ordered* genotypes. We can adopt the convention that the maternal allele is listed to the left of the slash and the paternal allele is listed to the right. With three alleles, the ABO locus has nine distinct ordered genotypes.

The **Hardy-Weinberg Law** of population genetics permits calculation of genotype frequencies form allele frequencies. In the ABO example above, if the frequency of the A allele is p_A and the frequency of the B allele is p_B, then a random individual will have phenotype AB with frequency $2p_Ap_B$. The factor of 2 in this frequency reflects the two equally likely ordered genotypes A/B and B/A. In essence, Hardy-Weinberg equilibrium corresponds to the random union of two *gametes*, one gamete being an egg and other being a sperm. A union of two gametes incidentally is called a *zygote*.

In gene mapping studies, several genetic loci on the same chromosome are phenotyped. When these loci are simultaneously followed in a human *pedigree*, the phenomenon of *recombination* can often be observed. This reshuffling of genetic

material manifest itself when a parent transmits to a child a chromosome that differs from both of the corresponding homologous parental chromosomes. A gamete's sequence of alleles along a chromosome constitutes a *haplotype*. The alleles appearing in a haplotype are said to be in phase. Two such haplotypes together determine a *multilocus* genotype.

3.1.2 Hardy-Weinberg Equilibrium

Let us now consider a formal mathematical model for the establishment of Hardy-Weinberg equilibrium. This model relies on the seven following explicit assumptions:

a. Infinite population size,

b. Discrete generations,

c. Random mating,

d. No selection,

e. No migration,

f. No mutation, and

g. Equal initial genotype frequencies in the two sexes.

Suppose for the sake of simplicity that there are two alleles A_1 and A_2 at some autosomal locus in this population and that all genotypes are unordered. Consider

Table 3.2: *Mating outcomes for Hardy-Weinberg Equilibrium*

Mating Type	Nature of Offspring	Frequency
$A_1/A_1 \times A_1/A_1$	A_1/A_1	u^2
$A_1/A_1 \times A_1/A_2$	$\frac{1}{2}A_1/A_1 + \frac{1}{2}A_2/A_2$	$2uv$
$A_1/A_1 \times A_2/A_2$	A_1/A_2	$2uw$
$A_1/A_2 \times A_1/A_2$	$\frac{1}{4}A_1/A_1 + \frac{1}{2}A_1/A_2 + \frac{1}{4}A_2/A_2$	v^2
$A_1/A_2 \times A_2/A_2$	$\frac{1}{2}A_1/A_2 + \frac{1}{2}A_2/A_2$	$2vw$
$A_2/A_2 \times A_2/A_2$	A_2/A_2	w^2

the result of crossing the genotype A_1/A_1 with A_2/A_2. The first genotype produces only A_1 gametes and the second genotype yields gametes A_1 and A_2 in equal proportion. For the cross under consideration, gametes produced by the genotype A_1/A_1 are equally likely to combine with either gamete type issuing from the genotype A_1/A_2. Thus, for the cross $A_1/A_1 \times A_1/A_2$, the frequency of offspring is $\frac{1}{2}A_1/A_1$ and $\frac{1}{2}A_1/A_2$. Similarly the cross $A_1/A_1 \times A_2/A_2$ yields only A_1/A_2 offspring. The cross $A_1/A_2 \times A_1/A_2$ produced offspring in the ratio $\frac{1}{4}A_1/A_1$, $\frac{1}{2}A_1/A_2$ and $\frac{1}{4}A_2/A_2$. These proportions of outcomes for the various possible crosses are known as *segregation ratios*.

Suppose the initial proportions of the genotypes are u for A_1/A_1, v for A_1/A_2 and w for A_2/A_2. Under the stated assumptions, the next generation will be as shown in Table 3.2. The entries in the Table 3.2 yield for the three genotypes A_1/A_1, A_1/A_2 and A_2/A_2 the new frequencies:

$$u^2 + uv + \frac{1}{4}v^2 = \left(u + \frac{1}{2}v\right)^2,$$

$$uv + 2uw + \frac{1}{2}v^2 + vw = 2\left(u + \frac{1}{2}v\right)\left(u + \frac{1}{2}w\right),$$
$$\frac{1}{4}v^2 + vw + w^2 = \left(\frac{1}{2}v + w^2\right)^2, \tag{3.1}$$

respectively. If we define the frequencies of the two alleles A_1 and A_2 as $p_1 = u + \frac{v}{2}$ and $p_2 = \frac{v}{2} + w$, then A_1/A_2 with frequency p_1^2, A_1/A_2 with frequency $2p_1p_2$, and A_2/A_2 with frequency p_2^2. After a second round of random mating, the genotypes A_1/A_1, A_1/A_2 and A_2/A_2 are

$$\left(p_1^2 + \frac{1}{2}2p_1p_2\right)^2 = [p_1(p_1 + p_2)]^2 = p_1^2,$$
$$2\left(p_1^2 + \frac{1}{2}2p_1p_2\right)\left(\frac{1}{2}2p_1p_2 + p_2^2\right) = 2p_1(p_1 + p_2)p_2(p_1 + p_2) = 2p_1p_2,$$
$$\left(\frac{1}{2}2p_1p_2 + p_2^2\right)^2 = 2p_1(p_1 + p_2)p_2(p_1 + p_2) = [p_2(p_1 + p_2)]^2 = p_2^2. \tag{3.2}$$

Thus, after a single round of random mating, genotype frequencies stabilize at the Hardy-Weinberg proportions.

We may deduce the same result by considering the gamete population. A_1 gametes have frequency p_1 and A_2 gametes frequency p_2. Since random union of gametes is equivalent to random mating, A_1/A_1 is present in the next generation with frequency p_1^2, A_1/A_2 with frequency $2p_1p_2$ and A_2/A_2 with frequency p_2^2. In the gamete pool from the new generation, A_1 again occurs with frequency $p_1^2 + p_1p_2 = p_1(p_1 + p_2) = p_1$ and A_2 with frequency p_2. In other words, stability is attained in a single generation. This random union of gametes argument generalizes easily to more than two alleles.

3.1.3 Gamete Transmission Probability

Let $\text{Tran}(G_k|G_i, G_j)$ denote the probability that a mother i with genotype G_i and a father j with genotype G_j produce a child k with genotype G_k. For ordered genotypes, the child's genotype G_k can be visualized as an ordered pair of gametes (U_k, V_k), U_k being maternal in origin and V_k being paternal in origin. If all participating loci reside on the same chromosome, then U_k and V_k are haplotypes. Because any two parents create gametes independently, the transmission probability

$$\text{Tran}\,(G_k|G_i, G_j) = \text{Tran}\,(U_k|G_i)\,\text{Tran}\,(V_k|G_j) \tag{3.3}$$

factors into two *gamete transmission probabilities*. Unordered genotypes do not obey this factorization rule.

Specification of gamete transmission probabilities is straightforward for single-locus models. For a single autosomal locus, $\text{Tran}(H|G)$ is either 1, $\frac{1}{2}$, or 0, depending on whether the single allele H is identical in state to both, one, or neither of the two alleles of the parental genotype G, respectively.

3.2 Bootstrap Methods

3.2.1 Nonparametric Bootstrap Estimates

Bootstrap methods depend on the notion of a bootstrap sample. Let $\hat{F}$ be the empirical distribution, putting probability $1/n$ on each of the observed values x_i,

$i = 1, 2, \ldots, n$. A bootstrap sample is defined to be random sample of size n drawn from $\hat{F}$, say $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$,

$$\hat{F} \rightarrow (x_1^*, x_2^*, \ldots, x_n^*). \tag{3.4}$$

The star notation indicates that $\mathbf{x}^*$ is not the actual data set $\mathbf{x}$, but rather a randomized, or resampled version of $\mathbf{x}$.

There is another way to say (3.4): the bootstrap data points $(x_1^*, x_2^*, \ldots, x_n^*)$ are a random sample of size n drawn with replacement from the population of n objects $(x_1, x_2, \ldots, x_n)$. Thus we might have $x_1^* = x_7, x_2^* = x_3, x_4^* = x_{22}, \ldots, x_n^* = x_7$. The bootstrap data set $(x_1^*, x_2^*, \ldots, x_n^*)$ consists of members of the original data set $(x_1, x_2, \ldots, x_n)$, some appearing zero times, some appearing once, some appearing twice, etc.

Corresponding to a bootstrap data set $\mathbf{x}^*$ is a bootstrap replication of $\hat{\theta}$,

$$\hat{\theta}^* = s(\mathbf{x}^*). \tag{3.5}$$

The quantity $s(\mathbf{x}^*)$ is the result of applying the same function $s(.)$ to $\mathbf{x}^*$ as was applied to $\mathbf{x}$. For example if $s(\mathbf{x})$ is the sample mean $\bar{x}$ then $s(\mathbf{x}^*)$ is the mean of the bootstrap data set, $\bar{x}^* = \sum_{i=1}^{n} \frac{x_i^*}{n}$.

The bootstrap estimate of $se_F(\hat{\theta})$, the standard error of a statistic $\hat{\theta}$, is a plug-in estimate that uses the empirical distribution function $\hat{F}$ in place of the unknown distribution F. Specifically, the bootstrap estimate of $se_{\hat{F}}(\hat{\theta})$ is defined by

$$se_{\hat{F}}(\hat{\theta}^*). \tag{3.6}$$

In other words, the bootstrap estimate of $se_F(\hat{\theta})$ is the standard error of $\hat{\theta}$ for data sets of size n randomly sampled from F.

It is easy to implement bootstrap sampling on the computer. A random number device selects integers $i_1, i_2, \ldots, i_n$, each of which equals any value between 1 and n with probability $1/n$. The bootstrap sample consists of the corresponding members of $\mathbf{x}$.

$$x_1^* = x_{i_1}, x_2^* = x_{i_2}, \ldots, x_n^* = x_{i_n}. \tag{3.7}$$

The bootstrap algorithm works by drawing many independent bootstrap samples, evaluating the corresponding bootstrap replications and estimating the standard error of $\hat{\theta}$ by the empirical standard deviation of the replications. The result is called the bootstrap estimate of the standard error, denoted by $\hat{se}_B$, where B is the number of bootstrap samples used.

These ideal bootstrap estimates and its approximations are called nonparametric bootsrap estimates because they are based on $\hat{F}$, the nonparametric estimate of the population F.

3.2.2 Hierarchical Bootstrap

In some studies the variation in response may be hierarchical or multilevel. Depending upon the nature of the parameter being estimated, it may be important to take careful account of the two (or more) sources of variation when setting up a resampling scheme. In principle, there should be no difficulty with parametric

resampling: having fitted the model parameters, resample data will be generated according to a completely defined model. Nonparametric resampling is not straightforward: certainly it will not make sense to use simple nonparametric resampling, which treats all observations as independent.

The most basic problem involving hierarchical variation can be formulated as follows. For each of a groups we obtain b responses y_{ij} such that

$$y_{ij} = x_i + z_{ij}, \qquad i = 1, \ldots, a; \quad j = 1, ..b; \tag{3.8}$$

where the x_is are randomly sampled from F_x and independently the z_{ij}s are randomly sampled from F_z, with $E(Z) = 0$ to force uniqueness of the model. Thus there is homogeneity of variation in Z between groups, and the structure is additive. The feature of this model that complicates resampling is the correlation between observations within a group,

$$var(Y_{ij}) = \sigma_x^2 + \sigma_z^2, \qquad cov(Y_{ij}, Y_{ik}) = \sigma_x^2. \tag{3.9}$$

For data having this nested structure, one might be interested in parameters of F_x and F_z, or some combination of both.

There are two simple strategies, for both of which the first stage is to randomly sample groups with replacement. At the second stage we randomly sample within the groups selected at the first stage, either without replacement (Strategy 1) or with replacement (Strategy 2). Consider selecting $y_{i1}^*, \ldots, y_{ib}^*$. At first stage we select random integer I^* from $\{1, 2, \ldots, a\}$. At the second stage, we select ran-

dom integers $j_1^*, \ldots, j_b^*$ from $\{1, 2, \ldots, b\}$, either without replacement (Strategy 1) or with replacement (Strategy 2): the sampling without replacement is equivalent to keeping the I^*th group intact.

Chapter 4

Binary Logistic Mixed Model for Genetic Data

In this chapter, we discuss the generalized linear mixed model that we consider and the estimation procedure, including the misspecified models.

4.1 The Disease Model

We consider the binary mixed model:

$$y_{ij}|a_i \sim Bernoulli(p_{ij})\,; \quad i = 1, 2, \ldots, m; j = 1, 2, \ldots, n_i \quad , \tag{4.1}$$

$$logit(p_{ij}) = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_o + \beta_1 G_{ij} + a_i \quad , \tag{4.2}$$

$$a_i \sim \mathcal{N}(0, \sigma_a^2) \quad , \tag{4.3}$$

where

- $y_{ij} = 0$ or 1, implies not-affected or affected respectively with the disease of interest for individual j in family i,
- p_{ij} = probability of being affected with the disease of interest for individual j in family i,
- G_{ij} = candidate genotype indicator of individual j in family i,
- β_1 = log odds ratio for the effect of the candidate gene G, and
- a_i is the random effect of family i.

4.1.1 The likelihood function

The likelihood function for the model is

$$\begin{aligned} L &= \prod_{i=1}^{m} \int_{-\infty}^{\infty} \prod_{j=1}^{n_i} p_{ij}^{y_{ij}} (1-p_{ij})^{1-y_{ij}} \frac{e^{-\frac{1}{2\sigma_a^2} a_i^2}}{\sqrt{2\pi\sigma_a^2}} \, da_i \\ &= \prod_{i=1}^{m} \int_{-\infty}^{\infty} \prod_{j=1}^{n_i} e^{(\beta_o+\beta_1 G_{ij}+a_i)y_{ij}} \left(1+e^{\beta_o+\beta_1 G_{ij}+a_i}\right)^{-1} \frac{e^{-\frac{1}{2\sigma_a^2} a_i^2}}{\sqrt{2\pi\sigma_a^2}} \, da_i \\ &= e^{\beta_o \sum_{i=1}^{m}\sum_{j=1}^{n_i} y_{ij} + \beta_1 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij} y_{ij}} \\ &\quad \times \prod_{i=1}^{m} \int_{-\infty}^{\infty} e^{a_i \sum_{j=1}^{n_i} y_{ij}} \frac{e^{-\frac{1}{2\sigma_a^2} a_i^2}}{\sqrt{2\pi\sigma_a^2}} \prod_{j=1}^{n_i} \left(1+e^{\beta_o+\beta_1 G_{ij}+a_i}\right)^{-1} da_i. \end{aligned} \tag{4.4}$$

The log-likelihood can be simplified as follows

$$l = logL = \beta_o \sum_{i=1}^{m}\sum_{j=1}^{n_i} y_{ij} + \beta_1 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij}y_{ij} + \sum_{i=1}^{m} \log \int_{-\infty}^{\infty} e^{a_i \sum_{j=1}^{n_i} y_{ij}} \frac{e^{-\frac{1}{2\sigma_a^2}a_i^2}}{\sqrt{2\pi\sigma_a^2}} \prod_{j=1}^{n_i} \left(1 + e^{\beta_o+\beta_1 G_{ij}+a_i}\right)^{-1} da_i. \tag{4.5}$$

Unfortunately (4.5) cannot be simplified further or evaluated in closed form and hence maximizing values cannot be expressed in closed form either.

4.1.2 Newton-Rapson Iterative Equation

We apply Newton-Raphson algorithm to estimate the maximum likelihood estimates $\hat{\boldsymbol{\theta}} = \left(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}_a\right)$. The maximum likelihood estimate of the fixed effect parameters can be obtained by solving the following iterative equation

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} + \left(\mathbf{X}' E_a\left[\mathbf{W}|\mathbf{y}\right]\mathbf{X}\right)^{-1}\mathbf{X}'(\mathbf{y} - E_a\left[\mathbf{P}|\mathbf{y}\right]). \tag{4.6}$$

Here,

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_o \\ \\ \beta_1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & G_{11} \\ \vdots & \vdots \\ 1 & G_{mn_m} \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} p_{11} \\ \vdots \\ p_{mn_m} \end{pmatrix}$$

$$\mathbf{y} = \begin{pmatrix} y_{11} \\ \vdots \\ y_{mn_m} \end{pmatrix}, \quad \mathbf{W} = \begin{pmatrix} p_{11}(1-p_{11}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & p_{mn_m}(1-p_{mn_m}) \end{pmatrix}.$$

4.1.3 Conditional Distribution of the Random Term

In order to maximize the likelihood, we need to derive the conditional distribution of the random intercept a_i given Y. From (4.1) and (4.3) we have

$$f_{y|a}(y_{ij}) = p_{ij}^{y_{ij}}(1-p_{ij})^{1-y_{ij}}\ ; \quad y_{ij}=0,1\ ; \quad 0<p_{ij}<1 \tag{4.7}$$

$$f_a(a_i) = \frac{e^{-\frac{1}{2\sigma_a^2}a_i^2}}{\sqrt{2\pi\sigma_a^2}}\ ; \quad -\infty<a_i<\infty,\ ; \quad \sigma_a^2>0. \tag{4.8}$$

We can find the conditional distribution by,

$$f_{a|Y}(a_i) = \frac{\prod_{j=1}^{n_i} f_{Y|a}(y_{ij}) f_a(a_i)}{\int_{-\infty}^{\infty}\prod_{j=1}^{n_i} f_{Y|a}(y_{ij}) f_a(a_i)\, da_i}. \tag{4.9}$$

Using (4.7) and (4.8) we can write,

$$\begin{aligned}
&\prod_{j=1}^{n_i} f_{y|a}(y_{ij}) f_a(a_i) \\
&= \prod_{j=1}^{n_i} p_{ij}^{y_{ij}}(1-p_{ij})^{1-y_{ij}} f_a(a_i) \\
&= \prod_{j=1}^{n_i}\left(\frac{e^{\beta_o+\beta_1 G_{ij}+a_i}}{1+e^{\beta_o+\beta_1 G_{ij}+a_i}}\right)^{y_{ij}}\left(\frac{e^1}{1+e^{\beta_o+\beta_1 G_{ij}+a_i}}\right)^{1-y_{ij}} f_{a|y}(a_i) \\
&= \prod_{j=1}^{n_i} e^{(\beta_o+\beta_1 G_{ij}+a_i)y_{ij}}\left(1+e^{\beta_o+\beta_1 G_{ij}+a_i}\right)^{-1} f_{a|Y}(a_i) \\
&= \prod_{j=1}^{n_i} e^{\beta_o+\beta_1 G_{ij}} e^{a_i y_{ij}}\left(1+e^{\beta_o+\beta_1 G_{ij}+a_i}\right)^{-1}\frac{e^{-\frac{1}{2\sigma_a^2}a_i^2}}{\sqrt{2\pi\sigma_a^2}} \\
&= \frac{e^{n_i\beta_o+\beta_1\sum_{j=1}^{n_i}G_{ij}}}{\sqrt{2\pi\sigma_a^2}} e^{a_i\left(\sum_{j=1}^{n_i}y_{ij}-\frac{a_i}{2\sigma_a^2}\right)}\prod_{j=1}^{n_i}\left(1+e^{\beta_o+\beta_1 G_{ij}+a_i}\right)^{-1}.
\end{aligned} \tag{4.10}$$

From (4.9) and (4.10) finally we get

$$f_{a|Y}(a_i) = \frac{e^{a_i\left(\sum_{j=1}^{n_i} y_{ij} - \frac{a_i}{2\sigma_a^2}\right)} \prod_{j=1}^{n_i} \left(1 + e^{\beta_o + \beta_1 G_{ij} + a_i}\right)^{-1}}{\int_{-\infty}^{\infty} e^{a_i\left(\sum_{j=1}^{n_i} y_{ij} - \frac{a_i}{2\sigma_a^2}\right)} \prod_{j=1}^{n_i} \left(1 + e^{\beta_o + \beta_1 G_{ij} + a_i}\right)^{-1} da_i}. \tag{4.11}$$

4.1.4 Conditional Expectation of p_{ij}

The conditional expectation of the of p_{ij} w. r. to the conditional distribution of a can be obtained by numerically solving the following formula

$$\begin{aligned} & E_a(p_{ij}|y) \\ = & \int_{-\infty}^{\infty} p_{ij}\, f_{a|y}(a_i) da_i \\ = & \int_{-\infty}^{\infty} \frac{\left(\frac{e^{\beta_o + \beta_1 G_{ij} + a_i}}{1 + e^{\beta_o + \beta_1 G_{ij} + a_i}}\right) e^{a_i\left(\sum_{j=1}^{n_i} y_{ij} - \frac{a_i}{2\sigma_a^2}\right)} \prod_{j=1}^{n_i} \left(1 + e^{\beta_o + \beta_1 G_{ij} + a_i}\right)^{-1}}{\int_{-\infty}^{\infty} e^{a_i\left(\sum_{j=1}^{n_i} y_{ij} - \frac{a_i}{2\sigma_a^2}\right)} \prod_{j=1}^{n_i} \left(1 + e^{\beta_o + \beta_1 G_{ij} + a_i}\right)^{-1} da_i} da_i \end{aligned} \tag{4.12}$$

Conditional expectation of $p_{ij}(1 - p_{ij})$ can be found in the similar way.

4.1.5 ML Estimate of the Variance Component

The log-likelihood function of the random term is,

$$\begin{aligned} l_a &= \log\left(\Pi_{i=1}^{m} f_a(a_i)\right) \\ &= \frac{m}{2}\log(2\pi) - \frac{m}{2}\log\sigma_a^2 - \sum_{i=1}^{m} \frac{a_i^2}{2\sigma_a^2}. \end{aligned} \tag{4.13}$$

We differentiate the log likelihood with respect to σ_a^2,

$$\begin{aligned}\frac{\delta l_a}{\delta \sigma_a^2} &= 0 - \frac{1}{2}\frac{m}{\sigma_a^2} - \sum_{i=1}^{m} \frac{a_i^2}{2}\left(-\frac{1}{\sigma_a^4}\right) \\ &= \sum_{i=1}^{m} \frac{a_i^2}{2\sigma_a^4} - \frac{m}{2\sigma_a^2}. \end{aligned} \tag{4.14}$$

The Maximum likelihood estimate of the random effect parameter σ_a^2 can be found explicitly by solving,

$$E_a\left[\frac{\delta l_a}{\delta \sigma_a^2}\middle| y\right] = E_a\left[\left(\sum_{i=1}^{m} \frac{a_i^2}{2\sigma_a^4} - \frac{m}{2\sigma_a^2}\right)\middle| y\right] = 0. \tag{4.15}$$

Replacing σ_a^2 with $\hat{\sigma}_a^2$ in (4.15) we get,

$$\begin{aligned} &\frac{1}{2\hat{\sigma}_a^4}\sum_{i=1}^{m} E\left[a_i^2\middle| y\right] - \frac{m}{2\hat{\sigma}_a^2} = 0 \\ &\Rightarrow \frac{1}{\hat{\sigma}_a^2}\sum_{i=1}^{m} E\left[a_i^2\middle| y\right] = m \\ &\Rightarrow \hat{\sigma}_a^2 = \frac{\sum_{i=1}^{m} E\left[a_i^2\middle| y\right]}{m}. \end{aligned} \tag{4.16}$$

4.2 Misspecified Model

Maximum likelihood (ML) is a commonly used method for analyzing generalized linear mixed models. The ML estimates are the most efficient under the assumption that the models are correctly specified. However, for an incorrectly specified regression model, the ML method generally provides biased estimates.

In this thesis, we evaluate $(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}_a^2)$ when the random effect assumptions are incorrect. In particular, we consider separately the following two situations:

4.2.1 Non-Gaussian Random Effect

We consider a gamma random intercept, such that

$$a_{ij} = a_i = \sigma_a(u_i - \lambda)/\sqrt{\lambda}, \tag{4.17}$$

where $u_i \sim gamma(\lambda, 1)$ and $var(u_i) = E(u_i) = \lambda$.

It is easy to show that

$$E(a_i) = \frac{\sigma_a}{\sqrt{\lambda}} E(u_i - \lambda) = 0, \tag{4.18}$$

and

$$\begin{aligned} var(a_i) &= \frac{\sigma_a^2}{\lambda} var(u_i - \lambda) \\ &= \frac{\sigma_a^2}{\lambda} var(u_i) = \sigma_a^2. \end{aligned} \tag{4.19}$$

Here it is clear that the random intercept a_i has mean 0 and variance σ_a^2, but follows a non-normal distribution.

4.2.2 Random Intercepts and Random Slopes are Mutually Independent

Recall the regression model

$$logit(p_{ij}) = \log\left(\frac{p_{ij}}{1 - p_{ij}}\right) = \beta_o + \beta_1 G_{ij} + a_{ij}, \tag{4.20}$$

where a_{ij} assumed to have the form

$$a_{ij} = a_{i,0} + a_{i,1} G_{ij} \tag{4.21}$$

with $a_{i,0} \sim \mathcal{N}(0, \sigma_0^2)$ and $a_{i,1} \sim \mathcal{N}(0, \sigma_1^2)$. Here $a_{i,0}$ and $a_{i,1}$ are mutually independent. In this case,

$$var(a_{ij}) = \sigma_0^2 + \sigma_1^2 G_{ij}^2. \tag{4.22}$$

So for a non-zero value of G_{ij}, $var(a_{ij})$ depends on the within-cluster covariate.

Chapter 5

Simulation Study

5.1 Introduction

In this chapter, we discuss how the family data is generated. The modified hierarchical bootstrap methods that we apply and the statistics that we use to compare and evaluate the estimates are discussed here as well. Finally, we present and investigate the simulation results.

5.2 Simulating Familial Data

For the simulation study, we use a disease model to randomly generate familial data that includes a candidate gene(G) and a random intercept that exhibit the effects of unmeasured polygenes and unmeasured environmental factors.

The genetic component in the model consisted of a bi-allelic candidate gene. The genotypes comprised alleles A and a for the candidate gene (G) and are coded categorically based on additive inheritance: $AA = 2$, $Aa = 1$, $aa = 0$. The parental genotypes were simulated with predefined population allele frequencies, which is $p_A = 0.20, p_a = 0.80$. According to the *Hardy-Weinberg Law* of population genetics we get,

$$P(AA) = p_A^2 = 0.04 = P(G = 2),$$

$$P(Aa) = 2p_A p_a = 0.32 = P(G = 1),$$

$$P(aa) = p_a^2 = 0.64 = P(G = 0). \tag{5.1}$$

We considered families of size $(3, 4, 5, 6, 7, 8)$ and their probabilities were preassigned $(.20, .40, .20, .10, .05, .05)$ respectively. The genotypes (G) for the offsprings are then assigned following Mendelian transmission probabilities assuming random parental mating types. For example, if a parent pair has genotype $(aa, aa) = (0, 0)$, then the possible genotype of a random offspring is $aa = 0$ with probability 1 i.e. $P(aa|aa, aa) = P(0|0, 0) = 1$. Again, if a parent pair has genotype $(Aa, Aa) = (0, 1)$, then the probability of the genotypes of a random offspring is,

$$P(aa|Aa, Aa) = (0|1, 1) = .25,$$

$$P(Aa|Aa, Aa) = (1|1, 1) = .5,$$

$$P(AA|Aa, Aa) = (2|1, 1) = .25, \tag{5.2}$$

The probability of the genotypes of a random offspring for all possible parent pairs are determined in the similar way.

The disease probability p_{ij} is determined using the model (4.2). Finally, disease status y_{ij} is randomly determined from the disease probability p_{ij}, using Bernoulli distribution. Each simulated family and their data are generated independently.

5.3 Extension of Hierarchical Bootstrap

Following Davison & Hinkley (1997)[10], we adopt an extension of the hierarchical bootstrap resampling scheme that is proposed by Shin et al. (2007)[28]:

Strategy 1: Stage 1: Randomly sample families with replacement, keep parents.

Stage 2: For those families selected in stage 1, randomly sample offspring within each family *without replacement*.

Strategy 2: Stage 1: Randomly sample families with replacement, keep parents.

Stage 2: For those families selected in stage 1, randomly sample offspring within each family *with replacement*.

In our study, all bootstrap estimates are were obtained from $B = 1000$ bootstrap samples that were generated from their corresponding simulated samples. We found the ML estimates of the regression coefficients and the variance components for all bootstrap samples.

5.4 Comparative Study

We used a few statistics for comparison and evaluation purposes. They are defined below.

$100 \times (1-\alpha)\%$ CI for the ML estimates: If $\hat{\theta}$ is the ML estimate of θ then the $100 \times (1-\alpha)\%$ CI for θ is

$$\left(\hat{\theta} \pm \mathcal{Z}_{\alpha/2} \times \sqrt{Var(\hat{\theta})}\right). \tag{5.3}$$

$100 \times (1-\alpha)\%$ Bootstrap CI: We use nonparametric bootstrap percentile confidence intervals that was suggested by Efron and Tibshirani (1993) [11]. We generate B bootstrap samples from each sample considered. Then calculate the bootstrap version of the statistic of interest $\hat{\theta}^*$ for every bootstrap sample and get $\left(\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*\right)$.Finally, the $[100 \times \alpha/2]$th and $[100 \times (1-\alpha/2)]$th percentiles of the empirical distribution formed the limits for the $100 \times (1-\alpha)\%$ bootstrap percentile confidence interval.

Coverage Probability of a CI: To find the coverage probability of a CI, we repeat the estimation procedure K times, such that we get K CIs. Then we determine the number of times (say K') the true value of the parameter lies in the K CIs. Finally, we obtain the coverage probability by

$$\frac{K'}{K}. \tag{5.4}$$

Mean Length of a CI: To find the mean length of a CI, we repeat the estimation procedure K times for K simulated samples, so that we get K CIs. For every single confidence intervals we get a lower limit $\hat{\theta}_{L_i}$ and an upper limit $\hat{\theta}_{U_i}$, $i = 1, \ldots, K$. Then we determine the mean length by

$$\sum_{1}^{K} \frac{\hat{\theta}_{U_i} - \hat{\theta}_{L_i}}{K}. \tag{5.5}$$

Empirical Relative Bias of the ML Estimates: To find the empirical relative bias of the ML Estimator of θ, we repeat the estimation procedure K times for K simulated samples, so that we get K estimates $(\hat{\theta}_1, \ldots, \hat{\theta}_K)$ of θ. Then we determine the empirical relative bias of the estimator of θ as

$$100 \times \frac{\theta^* - \theta_0}{\theta_0},$$

where $\theta^* = \frac{\sum_1^K \hat{\theta}_i}{K}$, $i = 1, \ldots, K$ and θ_0 is the true value of θ.

5.5 Results and Findings

We ran a series of simulations to evaluate the coverage probability of 95% confidence intervals for both regression parameters and variance components, mean length of these confidence intervals , and the empirical relative biases of the ML estimators, which were obtained under both correctly specified and misspecified random effects distributions. Specifically, $K = 1000$ replicates of the data for three

different numbers of families (50, 100 and 200) were generated to compare the ML and bootstrap methods. For all cases, we considered $\beta_0 = -1$ and $\beta_1 = 1.5$.

The statistical package R [24] was used for generating the data, bootstrapping and fitting GLMMs to the data by the ML method. The R function "glmmML" fits generalized linear mixed models with random intercepts by maximum likelihood and numerical integration via Gauss-Hermite quadrature.

5.5.1 Gaussian Random Effects

Initially, we considered a perfectly specified model i.e. the random term $a_i \sim \mathcal{N}(0, \sigma_a^2)$. We explore a range of values for the standard deviation σ_a of the random effect a_i. We compared the coverage probabilities and the mean lengths of the confidence intervals for β and σ_a obtained from ML and bootstrap methods. We also explore the finite-sample relative biases and mean squared errors of the ML estimates under this correctly specified random effects distribution.

Table 5.1 presents the empirical coverage probabilities and mean lengths of 95% confidence intervals for $(\beta_0, \beta_1, \sigma_a)$ under Gaussian random effects. It is clear from the table that the ML method provides coverage probabilities that are generally close to the nominal 95% level. The two bootstrap strategies also provide CI's which are close to the nominal 95% level. But the mean lengths of the CI's from the bootstrap methods are longer as compared to the ML confidence intervals.

We also observe from Table 5.1 that ML estimates provides better coverage probabilities for large sample sizes. But the bootstrap strategies, specially Strategy 2, generally produces better coverage probabilities for small number of families. In all three methods, it is clear form the table that, the mean lengths of the 95% confidence interval decrease with the increase in sample size but increase with the true value of σ_a.

Table 5.2 presents empirical relative bias of the ML estimators under perfectly specified Gaussian random effects. Here we observe that the relative bias of the ML estimates are less than 5% for all the cases considered. It is clear from the table that the biases are small for larger number of families and they are higher as the true values of σ_a rise. We observe negative bias for σ_a and the biases are positive when the true value of $\sigma_a = 2.0$.

Table 5.3 presents the mean squared error of the ML estimators under perfectly specified Gaussian random effects. We observe from this table that the MSEs of ML estimates are generally small for larger number of families. When the true value of σ_a is large, we observe large MSE of the ML estimators. This results support the pattern that is found in the relative bias of the ML estimates.

Table 5.1: *Empirical Coverage Probabilities and Mean Lengths of 95% CI's for* $(\beta_0, \beta_1, \sigma_a)$ *under perfectly specified Gaussian random effects for GLMM's.*

Number of Families	σ_a		ML Method		Bootstrap Method			
					Strategy1		Strategy2	
			Coverage	Mean Length	Coverage	Mean Length	Coverage	Mean Length
50	1.0	β_0	0.956	1.086	0.942	1.068	0.952	1.311
		β_1	0.914	1.249	0.939	1.438	0.944	1.814
		σ_a	0.919	1.006	0.893	1.212	0.955	1.340
	1.5	β_0	0.973	1.441	0.954	1.348	0.957	1.640
		β_1	0.941	1.541	0.948	1.662	0.961	2.091
		σ_a	0.909	1.228	0.907	1.401	0.964	1.621
	2.0	β_0	0.981	1.913	0.953	1.738	0.949	2.182
		β_1	0.966	1.899	0.944	1.904	0.956	2.466
		σ_a	0.929	1.731	0.933	1.736	0.954	2.161
100	1.0	β_0	0.953	0.768	0.942	0.738	0.940	0.895
		β_1	0.911	0.880	0.949	0.986	0.954	1.228
		σ_a	0.900	0.668	0.920	0.821	0.891	0.880
	1.5	β_0	0.967	1.014	0.940	0.924	0.943	1.100
		β_1	0.935	1.080	0.948	1.130	0.957	1.397
		σ_a	0.911	0.858	0.903	0.936	0.929	1.076
	2.0	β_0	0.982	1.329	0.950	1.146	0.953	1.352
		β_1	0.958	1.314	0.938	1.272	0.932	1.567
		σ_a	0.936	1.184	0.917	1.146	0.934	1.323
200	1.0	β_0	0.947	0.539	0.939	0.515	0.926	0.621
		β_1	0.894	0.619	0.940	0.691	0.916	0.854
		σ_a	0.889	0.464	0.898	0.551	0.752	0.606
	1.5	β_0	0.961	0.710	0.934	0.643	0.921	0.760
		β_1	0.943	0.756	0.949	0.786	0.939	0.964
		σ_a	0.915	0.598	0.897	0.646	0.882	0.739
	2.0	β_0	0.981	0.926	0.954	0.789	0.941	0.924
		β_1	0.968	0.914	0.960	0.876	0.943	1.071
		σ_a	0.938	0.819	0.884	0.787	0.919	0.902

Table 5.2: *Empirical Relative Bias,* $100 \times (\theta^* - \theta_0)/\theta_0$, *of the* **ML** *estimators under perfectly specified Gaussian random effects for GLMM's.*

		Number of families		
		50	**100**	**200**
$\sigma_a = 1.0$	β_0	1.751	0.386	−0.255
	β_1	2.108	−0.133	0.359
	σ_a	−4.529	−0.883	−0.752
$\sigma_a = 1.5$	β_0	0.723	0.828	0.477
	β_1	0.558	0.990	0.824
	σ_a	−0.779	−0.040	−0.344
$\sigma_a = 2.0$	β_0	1.416	1.641	0.601
	β_1	3.426	2.636	0.462
	σ_a	1.785	1.106	0.167

Table 5.3: *Empirical Mean Squared Error of the* **ML** *estimators under perfectly specified Gaussian random effects for GLMM's.*

		Number of families		
		50	**100**	**200**
$\sigma_a = 1.0$	β_0	0.0733	0.0358	0.0191
	β_1	0.1311	0.0611	0.0332
	σ_a	0.1017	0.0433	0.0219
$\sigma_a = 1.5$	β_0	0.1054	0.0597	0.0292
	β_1	0.1688	0.0820	0.0393
	σ_a	0.1249	0.0639	0.0296
$\sigma_a = 2.0$	β_0	0.1658	0.0779	0.0372
	β_1	0.2157	0.1083	0.0477
	σ_a	0.2029	0.0981	0.0492

5.5.2 Non-Gaussian Random Effects

In this case, we consider a random intercept $a_{ij} = a_i = \sigma_a(u_i - \lambda)/\sqrt{\lambda}$ with $u_i \sim ind.\ Gamma(\lambda, 1)$. We explore a range of values for the shape parameter λ and the scale parameter σ_a.

Table 5.4 and 5.5 present the empirical coverage probabilities and mean lengths of 95% confidence intervals for $(\beta_0, \beta_1, \sigma_a)$ under misspecified Gaussian random effects for GLMM's, where the random effects have a gamma distribution with shape parameter λ. We choose two values of λ, 1.0 and 2.0 for the simulation study. It is clear from these tables that no method seems to perform significantly better than the other. We observe that for all three methods the coverage probabilities decrease as the sample sizes increase. Strategy 2 provides better coverage probabilities for σ_a than the other two methods for any number of families but with longer mean lengths. ML method and Strategy 1 provide coverage probabilities for β_1, which is more close to the nominal level of 95% than that of β_1 and σ_a. However, mean lengths of the 95% confidence interval are lower for large number of families and higher for larger values of σ_a and λ.

Table 5.6 presents empirical relative bias of the ML estimators under misspecified Gaussian random effects for GLMM's, where the random effects actually have a gamma distribution. Here we observe that the relative bias of the ML estimates is small for β_1 in most cases considered. However, the ML estimates of σ_a have a

moderate amount of bias (10% – 20%) and β_0 has larger bias (10% – 35%). In most cases the bias seems to be decreasing with increased sample sizes. The bias is larger in situations with higher skewness and substantial heterogeneity, for example, when true values of $\lambda = 1.0$ and $\sigma = 2.0$ or $\lambda = 2.0$ and $\sigma = 2.0$. These results support the conclusion of Heagerty and Kurland (2001)[13], the distributional misspecification of the random term may lead to seriously biased estimators when the mixing distribution is highly skewed and the between cluster heterogeneity is substantial.

Table 5.7 presents the mean squared error of the ML estimators under misspecified Gaussian random effects for GLMM's where the random effects actually have a gamma distribution. This table shows that the mean squared errors of ML estimate are generally small for larger number of families. When the true value of σ_a is large, we observe, large MSE of the ML estimates. We observe that the mean squared errors are the largest when $\lambda = 2.0$ and $\sigma = 2.0$. This results support the pattern that is found in the relative bias of the ML estimates under misspecified Gaussian random effects.

Table 5.4: *Empirical Coverage Probabilities and Mean Lengths of 95% CI's for* $(\beta_0, \beta_1, \sigma_a)$ *under misspecified Gaussian random effects for GLMM's where the random effects have a gamma distribution,* $a_{ij} = a_{i,0} = \sigma_a(u_i - \lambda)/\sqrt{\lambda}$ *for* $u_i \sim gamma(\lambda, 1)$. *Here* $\lambda = 1.0$.

Number of Families	σ_a		ML Method		Bootstrap Method			
					Strategy1		Strategy2	
			Coverage	Mean Length	Coverage	Mean Length	Coverage	Mean Length
50	1.0	β_0	0.952	1.043	0.928	1.020	0.913	1.272
		β_1	0.929	1.208	0.950	1.388	0.948	1.774
		σ_a	0.886	1.056	0.830	1.222	0.966	1.367
	1.5	β_0	0.928	1.308	0.887	1.244	0.859	1.553
		β_1	0.941	1.422	0.931	1.558	0.934	1.994
		σ_a	0.730	1.106	0.776	1.431	0.952	1.583
	2.0	β_0	0.933	1.609	0.832	1.488	0.808	1.897
		β_1	0.948	1.657	0.930	1.711	0.944	2.234
		σ_a	0.682	1.339	0.738	1.624	0.926	1.914
100	1.0	β_0	0.944	0.733	0.920	0.703	0.861	0.865
		β_1	0.930	0.847	0.945	0.953	0.921	1.204
		σ_a	0.794	0.685	0.805	0.883	0.952	0.895
	1.5	β_0	0.886	0.914	0.825	0.845	0.747	1.029
		β_1	0.954	0.996	0.940	1.052	0.914	1.319
		σ_a	0.686	0.753	0.718	0.938	0.960	1.031
	2.0	β_0	0.807	1.127	0.703	1.005	0.619	1.228
		β_1	0.958	1.157	0.936	1.152	0.923	1.444
		σ_a	0.596	0.930	0.607	1.071	0.913	1.217
200	1.0	β_0	0.904	0.520	0.874	0.494	0.748	0.606
		β_1	0.925	0.599	0.943	0.663	0.889	0.832
		σ_a	0.764	0.454	0.763	0.594	0.913	0.612
	1.5	β_0	0.791	0.643	0.723	0.588	0.580	0.711
		β_1	0.944	0.700	0.942	0.733	0.874	0.914
		σ_a	0.562	0.526	0.521	0.641	0.962	0.711
	2.0	β_0	0.645	0.786	0.514	0.694	0.376	0.837
		β_1	0.956	0.808	0.930	0.800	0.881	0.995
		σ_a	0.404	0.646	0.328	0.731	0.869	0.831

Table 5.5: *Empirical Coverage Probabilities and Mean Lengths of 95% CI's for* $(\beta_0, \beta_1, \sigma_a)$ *under misspecified Gaussian random effects for GLMM's where the random effects have a gamma distribution,* $a_{ij} = a_{i,0} = \sigma_a(u_i - \lambda)/\sqrt{\lambda}$ *for* $u_i \sim gamma(\lambda, 1)$. *Here* $\boldsymbol{\lambda = 2.0}$.

Number of Families	σ_a		ML Method		Bootstrap Method Strategy1		Bootstrap Method Strategy2	
			Coverage	Mean Length	Coverage	Mean Length	Coverage	Mean Length
50	1.0	β_0	0.958	1.055	0.927	1.030	0.912	1.281
		β_1	0.939	1.218	0.949	1.401	0.942	1.788
		σ_a	0.910	1.050	0.855	1.240	0.967	1.374
	1.5	β_0	0.940	1.293	0.862	1.228	0.828	1.525
		β_1	0.935	1.406	0.934	1.535	0.943	1.963
		σ_a	0.720	1.108	0.768	1.421	0.953	1.567
	2.0	β_0	0.916	1.634	0.822	1.522	0.782	1.948
		β_1	0.956	1.673	0.931	1.747	0.928	2.282
		σ_a	0.726	1.358	0.781	1.648	0.947	1.953
100	1.0	β_0	0.930	0.741	0.895	0.708	0.858	0.871
		β_1	0.929	0.853	0.949	0.952	0.931	1.200
		σ_a	0.814	0.663	0.828	0.885	0.952	0.896
	1.5	β_0	0.878	0.920	0.820	0.852	0.726	1.037
		β_1	0.927	1.000	0.935	1.067	0.915	1.335
		σ_a	0.673	0.755	0.700	0.943	0.958	1.039
	2.0	β_0	0.816	1.124	0.711	1.006	0.638	1.124
		β_1	0.968	1.154	0.948	1.159	0.925	1.449
		σ_a	0.588	0.925	0.604	1.067	0.915	0.925
200	1.0	β_0	0.915	0.522	0.893	0.497	0.771	0.608
		β_1	0.953	0.600	0.963	0.666	0.910	0.834
		σ_a	0.771	0.455	0.764	0.594	0.904	0.614
	1.5	β_0	0.772	0.643	0.712	0.588	0.544	0.712
		β_1	0.952	0.699	0.956	0.730	0.902	0.909
		σ_a	0.561	0.526	0.537	0.642	0.962	0.710
	2.0	β_0	0.635	0.791	0.524	0.698	0.370	0.841
		β_1	0.954	0.811	0.946	0.801	0.881	0.993
		σ_a	0.431	0.650	0.373	0.738	0.871	0.834

Table 5.6: *Empirical Relative Bias,* $100 \times (\theta^* - \theta_0)/\theta_0$*, of the* **ML** *estimators under misspecified Gaussian random effects for GLMM's where the random effects have a gamma distribution,* $a_{ij} = a_{i,0} = \sigma_a(u_i - \lambda)/\sqrt{\lambda}$ *for* $u_i \sim gamma(\lambda, 1)$.

			Number of families		
			50	**100**	**200**
$\lambda = 1.0$	$\sigma_a = 1.0$	β_0	19.753	10.029	9.644
		β_1	2.794	2.619	2.366
		σ_a	−15.001	−12.918	−10.484
	$\sigma_a = 1.5$	β_0	22.372	21.496	20.559
		β_1	4.926	4.347	3.960
		σ_a	−16.483	−15.672	−15.357
	$\sigma_a = 2.0$	β_0	33.192	34.662	32.899
		β_1	5.634	5.263	3.470
		σ_a	−18.324	−18.016	−18.529
$\lambda = 2.0$	$\sigma_a = 1.0$	β_0	11.228	10.503	9.777
		β_1	4.197	2.518	2.523
		σ_a	−13.214	−10.496	−9.876
	$\sigma_a = 1.5$	β_0	23.363	22.485	20.617
		β_1	4.806	5.228	2.974
		σ_a	−18.164	−15.096	−15.467
	$\sigma_a = 2.0$	β_0	34.142	34.865	33.779
		β_1	8.334	5.222	3.819
		σ_a	−16.356	−18.385	−18.123

Table 5.7: *Empirical Mean Squared Error of the* **ML** *estimators under misspecified Gaussian random effects for GLMM's where the random effects have a gamma distribution,* $a_{ij} = a_{i,0} = \sigma_a(u_i - \lambda)/\sqrt{\lambda}$ *for* $u_i \sim gamma(\lambda, 1)$.

			Number of families		
			50	**100**	**200**
$\lambda = 1.0$	$\sigma_a = 1.0$	β_0	0.0771	0.0415	0.0255
		β_1	0.1203	0.0601	0.0311
		σ_a	0.1455	0.0753	0.0341
	$\sigma_a = 1.5$	β_0	0.1471	0.0896	0.0642
		β_1	0.1597	0.0739	0.0380
		σ_a	0.1991	0.1138	0.0821
	$\sigma_a = 2.0$	β_0	0.2390	0.1863	0.1401
		β_1	0.1923	0.0926	0.0480
		σ_a	0.3080	0.2086	0.1753
$\lambda = 2.0$	$\sigma_a = 1.0$	β_0	0.0753	0.0458	0.0241
		β_1	0.1229	0.0589	0.0267
		σ_a	0.1353	0.0629	0.0357
	$\sigma_a = 1.5$	β_0	0.1468	0.0981	0.0656
		β_1	0.1557	0.0878	0.0355
		σ_a	0.2136	0.1137	0.0808
	$\sigma_a = 2.0$	β_0	0.2654	0.1848	0.1449
		β_1	0.2016	0.0879	0.0458
		σ_a	0.2713	0.2096	0.1708

5.5.3 Random Intercepts and Random Slopes are Mutually Independent

Here we assumed that the random intercept a_{ij} have the form $a_{ij} = a_{i,0} + a_{i,1}G_{ij}$ with $a_{i,0} \sim \mathcal{N}(0, \sigma_0^2)$ and $a_{i,1} \sim \mathcal{N}(0, \sigma_1^2)$. Here $a_{i,0}$ and $a_{i,1}$ are mutually independent and for a non-zero value of G_{ij}, $var(a_{ij})$ depends on the within-cluster covariate. Hence we do not know the true value of $var(a_{ij})$. Here we investigate β_0 and β_1 for a range of values of σ_0 and σ_1.

Table 5.8 and 5.9 present the empirical coverage probabilities and mean lengths of 95% confidence intervals for $(\beta_0, \beta_1, \sigma_a)$ under misspecified Gaussian random effects for GLMM's when there are mutually independent random intercepts and random slopes for $\sigma_1 = 0.5$ and $\sigma_1 = 1.0$ respectively. It is very interesting to notice that both bootstrap strategies provide better coverage probabilities than that of the ML method for all the sample sizes. Although these bootstrap CI's generally have longer average lengths than the ML confidence intervals, but Strategy 1 produces average length of the intervals which are very close to that of ML method. Thus it is evident from the simulation that, when random intercepts and random slopes are mutually independent, intervals from bootstrap strategies perform better in terms of producing coverage probabilities as compared to the standard ML approach.

We observe from Table 5.8 and 5.9 that for any number of families, the coverage probabilities of ML estimates for β_1 is not close to the nominal level of 95%

as it is close for β_0. Coverage probabilities of 95% confidence interval from the ML method and Strategy 1 increase for larger sample sizes but this pattern is opposite for Strategy 2. In all three methods, the mean lengths of the 95% confidence intervals decrease with the increase in sample size, but the lengths increase with increased values of σ_0 and σ_1.

Table 5.10 presents empirical relative bias of the ML estimators under misspecified Gaussian random effects for GLMM's when there are mutually independent random intercepts. Here we observe that the bias of the estimators of β_0 is very low (less than 3.5%) for all the situations considered. It is clear from the table that β_1 is always underestimated and the biases of the estimates of β_1 increase with the increase in number of families but they always remain less then 7%. We find that the biases get higher as the true values of σ_1 rise. However, when $\sigma_1 = 0.5$, the biases are decreasing as the the values of σ_0 rise but the opposite is observed when $\sigma_1 = 1.0$, i.e. the bias has a positive relationship with the value of σ_0 when $\sigma_1 = 1.0$.

Table 5.11 presents mean squared errors of the ML estimators under misspecified Gaussian random effects for when there are mutually independent random intercepts. This table shows that the mean squared errors of ML estimate are generally small for larger number of families. When the true values of both σ_0 and σ_1 are large, we observe large MSEs for the ML estimates.

Table 5.8: *Empirical Coverage Probabilities and Mean Lengths of 95% CI's for* $(\beta_0, \beta_1, \sigma_a)$ *under misspecified Gaussian random effects for GLMM's, when there are mutually independent random intercepts and random slopes:* $a_{ij} = a_{i,0} + a_{i,1} CG_j$ *with* $a_{i,0} \sim \mathcal{N}(0, \sigma_0^2)$ *and* $a_{i,1} \sim \mathcal{N}(0, \sigma_1^2)$. *Here* $\boldsymbol{\sigma_1 = 0.5}$.

Number of Families	σ_0		**ML Method**		**Bootstrap Method**			
					Strategy1		**Strategy2**	
			Coverage	Mean Length	Coverage	Mean Length	Coverage	Mean Length
50	0.5	β_0	0.944	0.862	0.930	0.858	0.955	1.087
		β_1	0.863	1.049	0.946	1.294	0.953	1.640
	1.0	β_0	0.951	1.102	0.942	1.074	0.962	1.320
		β_1	0.891	1.261	0.938	1.490	0.951	1.863
100	0.5	β_0	0.949	0.608	0.948	0.600	0.947	0.756
		β_1	0.867	0.737	0.940	0.896	0.948	1.128
	1.0	β_0	0.953	0.780	0.933	0.746	0.927	0.905
		β_1	0.883	0.889	0.929	1.023	0.943	1.266
200	0.5	β_0	0.961	0.427	0.954	0.421	0.916	0.525
		β_1	0.834	0.519	0.948	0.627	0.948	0.785
	1.0	β_0	0.955	0.546	0.947	0.518	0.926	0.626
		β_1	0.880	0.622	0.934	0.711	0.939	0.872

Table 5.9: *Empirical Coverage Probabilities and Mean Lengths of 95% CI's for* $(\beta_0, \beta_1, \sigma_a)$ *under misspecified Gaussian random effects for GLMM's, when there are mutually independent random intercepts and random slopes:* $a_{ij} = a_{i,0} + a_{i,1}CG_j$ *with* $a_{i,0} \sim \mathcal{N}(0, \sigma_0^2)$ *and* $a_{i,1} \sim \mathcal{N}(0, \sigma_1^2)$. *Here* $\boldsymbol{\sigma_1 = 1.0}$.

Number of Families	σ_0		ML Method		Bootstrap Method			
					Strategy1		Strategy2	
			Coverage	Mean Length	Coverage	Mean Length	Coverage	Mean Length
50	0.5	β_0	0.951	0.907	0.942	0.893	0.960	1.128
		β_1	0.806	1.092	0.930	1.410	0.967	1.749
	1.0	β_0	0.954	1.150	0.937	1.112	0.948	1.371
		β_1	0.859	1.292	0.929	1.580	0.964	1.961
100	0.5	β_0	0.960	0.645	0.949	0.624	0.929	0.782
		β_1	0.794	0.773	0.911	0.986	0.955	1.211
	1.0	β_0	0.959	0.805	0.938	0.760	0.933	0.921
		β_1	0.857	0.903	0.945	1.089	0.975	1.324
200	0.5	β_0	0.958	0.454	0.947	0.437	0.905	0.545
		β_1	0.718	0.543	0.892	0.689	0.963	0.839
	1.0	β_0	0.963	0.569	0.948	0.534	0.911	0.642
		β_1	0.813	0.637	0.913	0.758	0.964	0.916

Table 5.10: *Empirical Relative Bias,* $100 \times (\theta^* - \theta_0)/\theta_0$, *of the* **ML** *estimators under misspecified Gaussian random effects for GLMM's, when there are mutually independent random intercepts and random slopes:* $a_{ij} = a_{i,0} + a_{i,1}CG_j$ *with* $a_{i,0} \sim \mathcal{N}(0, \sigma_0^2)$ *and* $a_{i,1} \sim \mathcal{N}(0, \sigma_1^2)$.

			Number of families		
			50	**100**	**200**
$\sigma_1 = 0.5$	$\sigma_0 = 0.5$	β_0	0.730	1.256	0.297
		β_1	−1.331	−1.897	−2.638
	$\sigma_0 = 1.0$	β_0	0.012	3.057	2.170
		β_1	−0.777	−6.230	−7.199
$\sigma_1 = 1.0$	$\sigma_0 = 0.5$	β_0	1.581	1.506	0.603
		β_1	−6.927	−1.142	−2.162
	$\sigma_0 = 1.0$	β_0	1.879	1.747	1.654
		β_1	−6.076	−5.740	−6.544

Table 5.11: *Empirical Mean Squared Error of the* **ML** *estimators under misspecified Gaussian random effects for GLMM's, when there are mutually independent random intercepts and random slopes:* $a_{ij} = a_{i,0} + a_{i,1}CG_j$ *with* $a_{i,0} \sim \mathcal{N}(0, \sigma_0^2)$ *and* $a_{i,1} \sim \mathcal{N}(0, \sigma_1^2)$.

			Number of families		
			50	**100**	**200**
$\sigma_1 = 0.5$	$\sigma_0 = 0.5$	β_0	0.0476	0.0224	0.0111
		β_1	0.1095	0.0522	0.0277
	$\sigma_0 = 1.0$	β_0	0.0707	0.0252	0.0133
		β_1	0.1455	0.0747	0.0424
$\sigma_1 = 1.0$	$\sigma_0 = 0.5$	β_0	0.0506	0.0395	0.0171
		β_1	0.1348	0.0729	0.0365
	$\sigma_0 = 1.0$	β_0	0.0800	0.0380	0.0184
		β_1	0.1657	0.0816	0.0467

Chapter 6

Conclusion

The purpose of the thesis was to study the finite-sample properties of the ML estimators and to compare the ML confidence intervals (based on the asymptotic normality property of the ML estimators) with the bootstrap confidence intervals. It appears that the bootstrap strategies generally provide better coverage probabilities when the sample size is relatively small. Moreover, when the random effects distribution is misspecified, the bootstrap strategy 2 appears to be more robust as compared to the ML and the bootstrap strategy 1 when finding the coverage probability of the confidence intervals for σ_a. However, the ML confidence intervals for (β_0, β_1) appear to be better in most cases. Specifically, when the Gaussian random effects assumption is not violated, for larger sample sizes (No. of families ≥ 100) ML method performed better. But for small sample sizes (No. of families ≤ 50), the modified bootstrap strategy (Strategy 2) seems to be a better option.

For the case of Non-Gaussian random effects, no method seems to perform significantly better than the other. Though the bias for the ML estimates appeared to be small for the cluster level covariate (β_1) in most of the cases, we conclude that the distributional misspecification of the random term may lead to seriously biased estimators when the mixing distribution is highly skewed and the between cluster heterogeneity is substantial, $\sigma_a \geq 2.0$.

It is evident from the simulation that, when random intercepts and random slopes are mutually independent, the confidence intervals from bootstrap strategies are better in terms of coverage probabilities, compared to the standard ML approach. In particular, Strategy 1 produces good coverage probabilities and also produces average length of the intervals that are very close to that of ML method.

Overall, in an association analysis using GLMM, we recommend the bootstrap confidence intervals when the number of families is small. A drawback of the bootstrap method is in its demand for more computational time relative to the ML. Therefore, it appears that the better performance of the bootstrap strategies comes with modest additional cost in computation.

6.1 Future Research

In this study, we considered only one covariate which is the genotype of the family members. There might be some known environmental factors in a study.

Future research is warranted to evaluate the impact of environmental effect and the interaction between the genotype and the environmental effect in addition to the candidate genotype.

We used nonparametric bootstrap percentile confidence intervals as suggested by Efron and Tibshirani (1993) [11]. Further research can also be performed to evaluate the Bias-corrected and accelerated (BC_a) nonparametric bootstrap percentile confidence intervals that were recommended by them.

Appendix A

R Codes Used for Simulation Study

A.1 R Codes When Random Intercepts are Gaussian

```
#------------------------ Data-function --------------------------#

 library(stats)
 library(boot)
 library(glmmML)

data.glmm = function(p=c(.64,.32,.04),fno=100, beta_0= -1,
                        beta_1= .5, sigma_a=.5)
{
                  # ====CG====
# p=probability of (aa, Aa, AA), fno= Number of families in sample

# Taking two independent sample of size fno w.r.t "p"
Pgenotype=cbind(sample(c(0,1,2), fno, replace = TRUE,prob=p),
          sample(c(0,1,2), 100, replace = TRUE,prob=p))
f1=Pgenotype[,1] # Parent 1
f2=Pgenotype[,2] # Parent 2
  # Generating fno random family size
fsize=sample(c(3,4,5,6,7,8), fno, replace = TRUE,
```

```
              prob=c(.20,.40,.20,.10,.05,.05))
  ## Family (Cluster) ID
fid=rep(1:fno,fsize)
  # Sample size
N=sum(fsize)
  # Creating independent possible offsprings for each parent pairs
fmid=NULL
CG=NULL
A=NULL
for (i in 1:fno) {
fmid=c(fmid, c(1:fsize[i]))
if (f1[i]==0) {
if (f2[i]==0) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(1,0,0))
}
if (f2[i]==1) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(.5,.5,0))
}
if (f2[i]==2) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(0,1,0))
}}
if (f1[i]==1) {
if (f2[i]==0) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(.5,.5,0))
}
if (f2[i]==1) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(.25,.5,.25))
}
if (f2[i]==2) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(0,.5,.5))
}}
if (f1[i]==2) {
if (f2[i]==0) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(0,1,0))
}
if (f2[i]==1) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(0,0,.5))
}
if (f2[i]==2) {
cgen = sample(c(0,1,2), fsize[i]-2,replace = TRUE,prob=c(0,0,1))
}}
```

```
CG=rbind(CG,as.matrix(Pgenotype[i,]),as.matrix(cgen))
  # Simulating random intercept
aij= rnorm(1, mean=0, sd=sigma_a)
ai= rep(aij,fsize[i])
A= rbind(A,as.matrix(ai))
}
  # Generating P from the Logit model
lgt=c(rep(beta_0,N))+((beta_1)*CG)+A
P=inv.logit(lgt)

  # Generating Y using binomial distribution w.r.t. P
Y=rbinom(N, 1, P)

data0=cbind(fid,fmid,CG,Y)
colnames(data0)=c("fid","fmid","CG","Y")
data.frame(data0)
datalist=NULL
c1=1
c2=0
for (i in 1:fno)
{
  c2=c2 + fsize[i]
  datalist[[i]]=data0[c1:c2,]
  c1=c2+1
}
dat=list(datalist=datalist,data=data0,fid=fid,fsize=fsize,N=N)
}

#---------------------- ML Estimation --------------------------#
glmm.ML=function (alpha =.1,dat.glmm=dat.glmm)
{
glmm=glmmML(dat.glmm$data[,4] ~ dat.glmm$data[,3],
            cluster = dat.glmm$data[,1],method ="ghq")
glmm.CI=rbind(c(glmm$coefficients[1]-
                qnorm(1-(alpha/2))*glmm$coef.sd[1],
glmm$coefficients[1]+ qnorm(1-(alpha/2))*glmm$coef.sd[1]),
c(glmm$coefficients[2]- qnorm(1-(alpha/2))*glmm$coef.sd[2],
glmm$coefficients[2]+ qnorm(1-(alpha/2))*glmm$coef.sd[1]),
c(glmm$sigma - qnorm(1-(alpha/2))*glmm$sigma.sd,
glmm$sigma + qnorm(1-(alpha/2))*glmm$sigma.sd))
dimnames(glmm.CI)=list(c("Beta_o","Beta_1","sigma_a"),
```

```
                        c("Lower","Upper"))
glmm.CI
glmm.table=c(glmm$coefficients[1],glmm.CI[1,1],glmm.CI[1,2],
abs(glmm.CI[1,2]-glmm.CI[1,1]),
glmm$coefficients[2],glmm.CI[2,1],glmm.CI[2,2],
abs(glmm.CI[2,2]-glmm.CI[2,1]),
glmm$sigma,glmm.CI[3,1],glmm.CI[3,2],
abs(glmm.CI[3,2]-glmm.CI[3,1]))
names(glmm.table)=c("beta_0","lower_(beta_0)",
"Upper_(beta_0)","Length_(beta_0)",
"beta_1","Lower_(beta_1)","Upper_(beta_1)","Length_(beta_1)",
"sigma_a","lower_(sigma_a)","Upper_(sigma_a)","Length_(sigma_a)")
glmm.table
glmm.out=list(CI=glmm.CI,table=glmm.table)
}

#-------------------- Bootstrap Sample Generation ----------------#
boot.strategy <- function(dat=dat.glmm)
{
# --------------  Strategy 1 -----------------#
fid <- dat$fid
unique.fid <- unique(fid)
k <- length(unique.fid)
   fid.samp <- sample(unique.fid, size=k, replace=T)
   boot.dat1 <- NULL
for (i in 1:k)
{
boot.dat10 <- dat$data[dat$data[,"fid"]==fid.samp[i], ]
boot.dat1 <- rbind(boot.dat1, boot.dat10)

}
   fid.boot <- rep(c(1:k), table(fid)[fid.samp])
  boot.dat1 <- cbind(fid.boot, boot.dat1)

# --------------  Strategy 2 -----------------#
boot.dat2 <- NULL
for (i in 1:k)
{
boot.dat20 <- boot.dat1[boot.dat1[,"fid.boot"]==i, ]
n <- nrow(boot.dat20)
if(n <= 3)
```

```
fmid.boot <- boot.dat20[,"fmid"]
else
{
element <- c(3:n)
n1 <- length(element)

fmid.boot0 <- sample(element, size=n1, replace=T)
fmid.boot <- c(1, 2, fmid.boot0)
}
boot.new <- NULL
for (j in 1:n)
{
index <- fmid.boot[j]
boot.dat200 <- boot.dat20[boot.dat20[,"fmid"]==index, ]
boot.new <- rbind(boot.new, boot.dat200)

}
boot.dat2 <- rbind(boot.dat2, boot.new)
}
list(boot.dat1=boot.dat1, boot.dat2=boot.dat2)
}

#---------------------- Bootstrap Estimation ----------------------#
boot.glmm= function (B=1000,alpha=.05,dat= dat.glmm)
{
boot.s1.coef=NULL
boot.s2.coef=NULL
for (b in 1:B)
{
data1 <- boot.strategy(dat=dat)
d1 <- data1$boot.dat1
d2 <- data1$boot.dat2
boot.glmm.s1=glmmML(Y ~ CG, cluster=fid.boot, data=as.data.frame(d1))
boot.s1.coef=rbind(boot.s1.coef,c(boot.glmm.s1$coefficients,
                    boot.glmm.s1$sigma))
boot.glmm.s2=glmmML(Y ~ CG, cluster=fid.boot, data=as.data.frame(d2))

boot.s2.coef=rbind(boot.s2.coef,
              c(boot.glmm.s2$coefficients,boot.glmm.s2$sigma))
}
boot.s1.CI=rbind(c(sort(boot.s1.coef[,1],
```

```
decreasing = FALSE)[(alpha/2)*B],
sort(boot.s1.coef[,1],decreasing = FALSE)[(1-(alpha/2))*B]),
c(sort(boot.s1.coef[,2],decreasing = FALSE)[(alpha/2)*B],
sort(boot.s1.coef[,2],decreasing = FALSE)[(1-(alpha/2))*B]),
c(sort(boot.s1.coef[,3],decreasing = FALSE)[(alpha/2)*B],
sort(boot.s1.coef[,3],decreasing = FALSE)[(1-(alpha/2))*B]))
dimnames(boot.s1.CI)=list(c("Beta_o","Beta_1","sigma_a"),
                          c("Lower","Upper"))

boot.s2.CI=rbind(c(sort(boot.s2.coef[,1],
decreasing = FALSE)[(alpha/2)*B],
sort(boot.s2.coef[,1],decreasing = FALSE)[(1-(alpha/2))*B]),
c(sort(boot.s2.coef[,2],decreasing = FALSE)[(alpha/2)*B],
sort(boot.s2.coef[,2],decreasing = FALSE)[(1-(alpha/2))*B]),
c(sort(boot.s2.coef[,3],decreasing = FALSE)[(alpha/2)*B],
sort(boot.s2.coef[,3],decreasing = FALSE)[(1-(alpha/2))*B]))
dimnames(boot.s2.CI)=list(c("Beta_o","Beta_1","sigma_a"),
                          c("Lower","Upper"))

boot.s1.table=c(boot.s1.CI[1,1],boot.s1.CI[1,2],
                abs(boot.s1.CI[1,2]-boot.s1.CI[1,1]),
boot.s1.CI[2,1],boot.s1.CI[2,2], abs(boot.s1.CI[2,2]-boot.s1.CI[2,1]),
boot.s1.CI[3,1],boot.s1.CI[3,2], abs(boot.s1.CI[3,2]-boot.s1.CI[3,1]))
names(boot.s1.table)=c("lower_(beta_0)",
"Upper_(beta_0)","Length_(beta_0)",
"Lower_(beta_1)","Upper_(beta_1)","Length_(beta_1)",
"lower_(sigma_a)","Upper_(sigma_a)","Length_(sigma_a)")

boot.s2.table=c(boot.s2.CI[1,1],boot.s2.CI[1,2],
abs(boot.s2.CI[1,2]-boot.s2.CI[1,1]),
boot.s2.CI[2,1],boot.s2.CI[2,2], abs(boot.s2.CI[2,2]-boot.s2.CI[2,1]),
boot.s2.CI[3,1],boot.s2.CI[3,2], abs(boot.s2.CI[3,2]-boot.s2.CI[3,1]))
names(boot.s2.table)=c("lower_(beta_0)",
"Upper_(beta_0)","Length_(beta_0)",
"Lower_(beta_1)","Upper_(beta_1)","Length_(beta_1)",
"lower_(sigma_a)","Upper_(sigma_a)","Length_(sigma_a)")

boot.s1.mean= apply(boot.s1.coef,2,mean)
names(boot.s1.mean)=c("Beta_o","Beta_1","sigma_a")
boot.s2.mean= apply(boot.s2.coef,2,mean)
names(boot.s2.mean)=c("Beta_o","Beta_1","sigma_a")
```

```
boot.out=list(s1=boot.s1.CI,s2=boot.s2.CI,boot.s1.mean= boot.s1.mean,
        boot.s2.mean=boot.s2.mean,table.s1=boot.s1.table,
        table.s2=boot.s2.table)
}

#---------------------- Confidence Intervals ----------------------#
glmm.CI = function(p=c(.64,.32,.04),fno=100, beta_0= -1, beta_1= .5,
                        sigma_a=.5, B=100,alpha=.10)
{
  dat.glmm=data.glmm(p=p,fno=fno,beta_0= beta_0,
  beta_1= beta_1, sigma_a= sigma_a)
  glmm.out= glmm.ML(alpha=alpha,dat.glmm=dat.glmm)

  boot.out= boot.glmm (B=B,alpha=alpha,dat=dat.glmm)

  CI= list(glmm.CI=glmm.out$CI, boot.s1.CI=boot.out$s1,
      boot.s2.CI=boot.out$s2,
      N=dat.glmm$N,boot.s1.mean=boot.out$boot.s1.mean,
      boot.s2.mean=boot.out$boot.s2.mean,
      glmm.table=glmm.out$table, boot.s1.table=boot.out$table.s1,
      boot.s2.table=boot.out$table.s2)
}

#---------------------------- Coverage ----------------------------#
glmm.out= function(p=c(.64,.32,.04),fno=100, beta_0= -1, beta_1= .5,
                        sigma_a=.5,B=100,alpha=.10,iter=10)
{
  c11=0
  c12=0
  c13=0
  c21=0
  c22=0
  c23=0
  c31=0
  c32=0
  c33=0

  glmm.table=NULL
  boot.s1.mean=NULL
  boot.s2.mean=NULL
```

```
boot.s1.table=NULL
boot.s2.table=NULL
N= NULL

for (r in 1:iter)
{
  CI=glmm.CI (p=p,fno=fno, beta_0= beta_0, beta_1= beta_1,
              sigma_a=sigma_a,B=B,alpha=alpha)
  if (CI$glmm.CI[1,1]<=beta_0)
  {
    if (CI$glmm.CI[1,2]>=beta_0)
    {
      c11=c11 + 1
    }
  }
  if (CI$glmm.CI[2,1]<=beta_1)
  {
    if (CI$glmm.CI[2,2]>=beta_1)
    {
      c12=c12 + 1
    }
  }
  if (CI$glmm.CI[3,1]<=sigma_a)
  {
    if (CI$glmm.CI[3,2]>=sigma_a)
    {
      c13=c13 + 1
    }
  }
  if (CI$boot.s1.CI[1,1]<=beta_0)
  {
    if (CI$boot.s1.CI[1,2]>=beta_0)
    {
      c21=c21 + 1
    }
  }
  if (CI$boot.s1.CI[2,1]<=beta_1)
  {
    if (CI$boot.s1.CI[2,2]>=beta_1)
    {
      c22=c22 + 1
```

```
    }
  }
  if (CI$boot.s1.CI[3,1]<=sigma_a)
  {
    if (CI$boot.s1.CI[3,2]>=sigma_a)
    {
      c23=c23 + 1
    }
  }
  if (CI$boot.s2.CI[1,1]<=beta_0)
  {
    if (CI$boot.s2.CI[1,2]>=beta_0)
    {
      c31=c31 + 1
    }
  }
  if (CI$boot.s2.CI[2,1]<=beta_1)
  {
    if (CI$boot.s2.CI[2,2]>=beta_1)
    {
      c32=c32 + 1
    }
  }
  if (CI$boot.s2.CI[3,1]<=sigma_a)
  {
    if (CI$boot.s2.CI[3,2]>=sigma_a)
    {
      c33=c33 + 1
    }
  }
  glmm.table=rbind(glmm.table,CI$glmm.table)
  boot.s1.mean=rbind(boot.s1.mean,CI$boot.s1.mean)
  boot.s2.mean=rbind(boot.s2.mean,CI$boot.s2.mean)
  boot.s1.table=rbind(boot.s1.table,CI$boot.s1.table)
  boot.s2.table=rbind(boot.s2.table,CI$boot.s2.table)
  N=rbind(N,CI$N)
}

parameter=c(beta_0,beta_1,sigma_a)
names(parameter)= c("Beta_0","Beta_1","sigma_a")
coverage = rbind(c(c11/iter,c21/iter,c31/iter),
```

```
                c(c12/iter,c22/iter,c32/iter),
                c(c13/iter,c23/iter,c33/iter))
dimnames(coverage)=list(c("Beta_0","Beta_1","sigma_a"),
                        c("ML","Strategy1","Strategy2"))

mean.length=rbind(c(mean(glmm.table[,4]),mean(boot.s1.table[,3]),
            mean(boot.s2.table[,3])),c(mean(glmm.table[,8]),
            mean(boot.s1.table[,6]),mean(boot.s2.table[,6])),
            c(mean(glmm.table[,12]),mean(boot.s1.table[,9]),
            mean(boot.s2.table[,9])))
dimnames(mean.length)=list(c("Beta_0","Beta_1","sigma_a"),
                           c("ML","Strategy1","Strategy2"))

bias=cbind(c(mean(glmm.table[,1])- beta_0,
        mean(glmm.table[,5])- beta_1,
        mean(glmm.table[,9])- sigma_a),
        c(mean(boot.s1.mean[,1])- beta_0,
        mean(boot.s1.mean[,2])- beta_1,
        mean(boot.s1.mean[,3])- sigma_a),
        c(mean(boot.s2.mean[,1])- beta_0,
        mean(boot.s2.mean[,2])- beta_1,
        mean(boot.s2.mean[,3])- sigma_a))
dimnames(bias)=list(c("Beta_0","Beta_1","sigma_a"),
                    c("ML","Strategy1","Strategy2"))

MSE.ML=c(mean((glmm.table[,1]- beta_0)^2),
mean((glmm.table[,5]- beta_1)^2),mean((glmm.table[,9]- sigma_a)^2))
names(MSE.ML)= c("Beta_0","Beta_1","sigma_a")
avsample=mean(N)

output=list(true.parameter.value=parameter, No.of.families=fno,
            average.sample.size=avsample,
coverage=coverage,mean.length=mean.length,bias=bias,MSE.ML=MSE.ML,
glmm.table=glmm.table,boot.s1.table=boot.s1.table,
            boot.s2.table=boot.s2.table)

cat("\n True parameter values: \n")
print(parameter)
cat("\n No of families: \n")
print(fno)
cat("\n Average sample size: \n")
```

```
  print(avsample)
  cat("\n No. of simulation: \n")
  print(iter)
  cat("\n The coverage probabilities: \n")
  print(coverage)
  cat("\n The mean lengths of the CIs: \n")
  print(mean.length)
  cat("\n Bias of the estimtes: \n")
  print(bias)
  cat("\n MSE of the ML estimtes: \n")
  print(MSE.ML)

}

output=glmm.out(p=c(.64,.32,.04),fno=200,beta_0= -1,
                beta_1= 1.5, sigma_a=2,B=1000,alpha=.05,iter=1000)
```

Bibliography

[1] Aird,I.; Bentall, H. H. and Roberts, J. A. Fraser, "Relationship Between Cancer of Stomach and the ABO Blood Groups", *British Medical Journal,* **1** (1953), 799–801.

[2] Aird,I.; Bentall, H. H.; Mehigan, J. A. and Roberts, J. A. Fraser, "The Blood Groups in Relation to Peptic Ulceration and Carcinoma of Colon, Rectum, Breast, and Bronchus", *British Medical Journal,* **2** (1954), 315–321.

[3] Baral, Janardhan, "The effect of misspecification when fitting generalized linear mixed models", Master's thesis, *Carleton University*, Ottawa, Canada, 2006.

[4] Besag, J.; York, J., and Mollie, A., "Bayesian Image Restoration, With Two Applications in Spatial Statistics", *Annals of the Institute of Statistical Mathematics,* **43** (1991), 1–59.

[5] Breslow, N.E. and Clayton, D.G., "Approximate Inference in Generalized Lin-

ear Mixed Models", *Journal of the American Statistical Association,* **88** (1993), 9–25.

[6] Buchanan, J.A. and Higley, E.T., "The Relationship of Blood Groups to Disease", *British Journal of Experimental Pathology,* **2** (1921), 247.

[7] Bull, S. B.; Chapman, N. H.; Greenwood, C. M. T. and Darlington, G. A., "Evaluation of genetic and environmental effects using GEE and APM methods", *Genetic Epidemiology,* **12** (1995), 729–734.

[8] Bull, S.B.; Darlington; G.A.; Greenwood, C.M.T. and Shin, J., "Design Considerations for Association Studies of Candidate Genes in Families", *Genetic Epidemiology,* **20** (2001), 149–174.

[9] Burton, Paul R.; Tiller, Katrina J.; Gurrin, Lyle C.; Cookson, William O.C.M.; Musk, A. William and Palmer, Lyle J., "Genetic Variance Components Analysis for Binary Phenotypes Using Generalized Linear Mixed Models (GLMMs) and Gibbs Sampling", *Genetic Epidemiology,* **17** (1999), 118–140.

[10] Davison, A.C. and Hinkley, D.V., *Bootstrap methods and their application,* Cambridge University Press Inc., New York, 1997.

[11] Efron, B. and Tibshirani, R. J., *An introduction to the bootstrap,* Chapman and Hall, New York, 1993.

[12] Feng, Z., McLerran, D. and Grizzle, J., "A comparison of statistical methods for clustered data analysis with Gaussian error", *Statistics in Medicine,* **15** (1996), 1793–1806.

[13] Heagerty, Patrick J. and Kurland, Brenda F., "Misspwcified Maximum Likelihood Estimates and Generalized Linear Mixed Models", *Biometrika,* **88** (2001), 973–985.

[14] Kenneth Lange, *Mathemetical and statistical methods for genetic analysis,* Springer, New York.

[15] Liang, K.Y. and Hanfelt, J., "On the use of Quasi-Likelihood Method in Teratological Experiments", *Biometrics,* **50** (1994), 872–880.

[16] Marjory P.; MacSWEEN, M.B. and Una A. Syme, M.A., "ABO Blood Group and Skin Diseases", *British Journal of Dermatology,* **77** (1965), 30–34.

[17] McConnell, R. B.; Pyke, D. A. and Roberts, J. A. Fraser, "Blood Groups in Diabetes Mellitus", *British Medical Journal,* **1** (1956), 772–776.

[18] McCulloch, C. E., "Maximum Likelihood Variance Components Estimation for Binary Data", *Journal of the American Statistical Association,* **89** (1994), 330–335.

[19] ———, "Maximum Likelihood Algorithms for Generalized Linear Mixed Models", *Journal of the American Statistical Association,* **92** (1997), 162–170.

[20] McCulloch, C.H. and Searle, S.R., *Generalized, linear and mixed models*, John Wiley and Sons, Inc., 605 Third Avenue, New York.

[21] Nelder, J.A and Wedderburn, R.W.M, "Generalized Linear Models", *Journal of the Royal Statistical Society,* **Series A (General), 135(3)** (1972), 370–384.

[22] Neuhaus, J.M., Hauck, W.W. and Kalbfleisch, J.D., "The Effects of Mixture Distribution Misspecification when Fitting Mixed-Effect Logistic Models", *Biometrica,* **79** (1992), 755–762.

[23] Pawitan, Y.; Reilly, M.; Nilsson, E.; Cnattingius; S. and Lichtenstein, P., "Estimation of genetic and environmental factors for binary traits using family data", *Statistics in Medicine,* **23** (2004), 449–465.

[24] R Development Core Team, "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2007, ISBN 3-900051-07-0.

[25] Roberts, J. A. Fraser, "Some associations between blood groups and disease", *British Medical Bull.,* **15** (1959), 129–133.

[26] Scurrah, Katrina J.; Palmer, Lyle J. and Burton1, Paul R., "Variance Components Analysis for Pedigree-Based Censored Survival Data Using Generalized Linear Mixed Models (GLMMs) and Gibbs Sampling in BUGS", *Genetic Epidemiology,* **19** (2000), 127–148.

[27] Sherman, M. and le Cessie, S., "A comparison between bootstrap methods and generalized estimating equations for correlated outcomes in generalized linear models", *Communications in Statistics - Simulation and Computation,* **26** (1997), 901–925.

[28] Shin, J.; Darlington, G.A.; Cotton, C.; Corey, M. and Bull, S.B., "Confidence Intervals for Candidate Gene Effects and Environmental Factors in Population-based Association Studies of Families", *Annals of Human Genetics,* **71** (2007), 421–432.

[29] Sinha, S. K., "Robust Analysis of Generalized Linear Mixed Models", *Journal of the American Statistical Association* **99** (2004), 451–460.

[30] ———, "Robust Inference in Generalized Linear Models for Longitudinal Data", *The Canadian Journal of Statistics,* **34(2)** (2006), 261–278.

[31] ———, "Robust Methods for Generalized Linear Models with Nonignorable Missing Covariates", *The Canadian Journal of Statistics,* **36(2)** (2008), 277–299.

[32] Stiratelli, R.; Laird, N. and Ware, J., "RandomEffects Models for Serial Observations With Binary Responses", *Biometrics,* **40** (1984), 961–971.

[33] Sutradhar, B. C. and Sinha, S. K., "On Pseudo-Likelihood Inference in the Bi-

nary Longitudinal Mixed Model", *Communications in Statistics, Part A–Theory and Methods,* **31** (2002), 397–417.

[34] Tan, M.; Qu, Y. and Rao, J.S., "Robustness of the Latent Variable Model for Correlated Binary Data", *Biometrics,* **55** (1999), 258–263.

[35] Ten Have, T.R.; Kunselman, A.R. and Tran, L., "A Comparison of Mixed Effects Logistic Regression Models for Binary Response Data with Two Nested Level of Clustering", *Statistics in Medicine,* **18** (1999), 947–960.

[36] Zeger, S. L. and Karim, M. R., "Generalized Linear Models With Random Effects; A Gibbs Sampling Approach", *Journal of the American Statistical Association,* **86** (1991), 79–86.

[37] Zeger, S. L.; Liang, K. Y. and Albert, P. S., "Models for Longitudinal Data: A Generalized Estimating Equation Approach", *Biometrics,* **44** (1988), 1049–1060.