INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

THE ANALYSIS OF LONGITUDINAL BINARY DATA

By

Joanna E. Mills

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
DALHOUSIE UNIVERSITY
HALIFAX, NOVA SCOTIA
AUGUST 2000

DALHOUSIE UNIVERSITY

FACULTY OF GRADUATE STUDIES

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "The Analysis of Longitudinal Binary Data"

by Joanna E. Mills

in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dated: July 18, 2000

External Examiner

Research Supervisor

Research Supervisor

Examining Committee

DALHOUSIE UNIVERSITY

Date: **August 2000**

Author: **Joanna E. Mills**

Title: **The Analysis of Longitudinal Binary Data**

Department: **Mathematics and Statistics**

Degree: **Ph.D.** Convocation: **October** Year: **2000**

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

_____

Signature of Author

To Scott Patrick Flemming

Contents

List of Tables

Acknowledgements

I wish to express my deepest gratitude to my supervisors Dr. Chris Field and Dr. Debbie Dupuis. Their guidance, encouragement, and patience will remain in my memory forever. I would also like to thank Dr. Jack Kalbfleisch for serving as an external examiner, as well as Dr. Ed Susko and Dr. Bruce Smith who were readers. Their questions and comments were invaluable.

My three years at Dalhousie has established many special friends: thank you to Nick Barrowman for being a great office mate, to Gretchen Smith for her reassurance and kindness, to David Hamilton for his sense of humour and interest, to Wade Blanchard and Kassiem Jacobs for their technical support, to Michele Millar for her confidence and Trish Moorhead for her encouragement.

To my parents and to Scott, I extend my sincere thanks. Their love and support, along with that of my friends and family, has made the completion of this thesis possible.

Financial support provided by the Natural Sciences and Engineering Research Council as well as Dalhousie University is gratefully acknowledged.

The Analysis of Longitudinal Binary Data

Joanna E. Mills
Doctor of Philosophy
Department of Mathematics and Statistics
Dalhousie University
2000

Abstract

Longitudinal data modelling is complicated by the necessity to deal appropriately with the correlation between observations made on the same individual. A thorough examination of popular approaches to longitudinal analysis establishes the essential features of an effective longitudinal model. Building upon an earlier non-robust version proposed by Heagerty [20], our robust marginally specified generalized linear mixed model (ROBMS-GLMM) is successful in exhibiting such features. This type of model is one of the first to allow both population-averaged and individual specific inference. As well, this type of model adopts the flexibility and interpretability of generalized linear models for introducing dependence, but builds regression structure for the marginal mean, allowing valid application with time-independent and time-dependent covariates. These new estimators are obtained as solutions of a robustified likelihood equation involving Huber's least favorable distribution and a collection of weights. Huber's least favorable distribution produces estimates which are resistant to deviations from the random effects distributional assumptions. Innovative weighting strategies enable the ROBMS-GLMM to perform well when faced with outlying observations both in the response and covariates. A simulation study allows us to investigate the sampling properties of the ROBMS-GLMM estimates. We illustrate the methodology with an analysis of a prospective longitudinal study of laryngoscopic endotracheal intubation, a skill which numerous health care professionals are expected to acquire. We also look at data collected on pregnancies and births in Nova Scotia with interest in the smoking habits of the expectant mothers. Psychiatric data concerning an anti-depression drug is also used for demonstrative purposes. The principal goal of our research is to achieve robust inference in longitudinal analyses. Robust model testing strategies and asymptotics properties of the ROBMS-GLMMs are also of interest. A concurrent goal is to investigate and potentially alleviate some of the difficulties with current model fitting software.

Chapter 1

Introduction

1.1 Motivation

This research was motivated by two main objectives:

- to work in an area of statistics which is both computationally challenging and of current interest,
- to contribute to the advancement of medical research.

Longitudinal binary data analysis seemed a natural candidate and has since proven to be an excellent choice.

1.2 Introduction

Longitudinal studies span a variety of disciplines including economics, pharmacology, sociology, biology and medicine. The defining characteristic of a longitudinal study is that individuals are measured *repeatedly* through time. Examples of such repeated measurements include daily stock prices for a selection of information technology companies; disease status of those involved in a clinical trial for a new cancer drug; behavior patterns for a group of recovering alcoholics; tree sizes over a growing season in an area with high ozone pollution; and the series of blood pressures of patients involved in a medical study of heart disease.

Effective modelling of longitudinal data is complicated by the necessity to deal appropriately with the correlation that exists between observations made on the same individual. In most longitudinal studies this correlation is a nuisance factor that must be taken into account in order to make valid statistical inferences. Diggle, Liang and Zeger [12] provide an excellent overview of methods for longitudinal data analysis, while more recent developments are reviewed by Heagerty and Zeger [23].

In essence there are three basic approaches to modelling longitudinal data: marginal, latent variable and response conditional models, with the first two being the most widely used and consequently our emphasis. Each of these approaches models the within-individual correlation differently and as a result achieves different objectives. Marginal models, for instance, separate the regression from the within-individual correlation which leads to descriptions of how the response average changes across various subsets of the population.

A thorough examination of popular approaches to longitudinal analysis establishes the essential features of an effective longitudinal model. Building upon an earlier non-robust model proposed by Heagerty [20], our robust marginally specified generalized linear mixed model (ROBMS-GLMM) is successful in exhibiting such features. This type of model is one of the first to allow both population-averaged and individual specific inference. As well, it adopts the flexibility and interpretability of generalized linear models for introducing dependence, but builds regression structure for the marginal mean, allowing valid application with time-independent and time-dependent covariates.

The ROBMS-GLMM estimators are obtained as solutions of a robustified likelihood equation involving Huber's least favorable distribution and a collection of weights. Huber's least favorable distribution [26] is introduced to make the ROBMS-GLMM less sensitive to violations on the distributional assumptions. Innovative weighting strategies enable the ROBMS-GLMM to perform well when faced with outlying observations both in the response and covariates. By changing the tuning constants defining the ROBMS-GLMM we can fit models with varying degrees of robustness. Close agreement of the estimates across a range of tuning constants suggests little, if any, contamination is present. Heagerty's non-robust model in fact becomes a special case of the ROBMS-GLMM when the tuning constants are set appropriately.

The ROBMS-GLMM is designed to simultaneously provide estimates of the parameters of interest and assess the fit of our data to the model. The robust enhancements it possesses

are paramount since these models are widely used in medical research where up to 10% of the observations may be contaminated [18]. In fact the development of robust procedures, like the ROBMS-GLMM, is one of the major developments in statistics this century.

Demonstrations and a simulation study allow us to investigate how our ROBMS-GLMM performs in practice. We see the ROBMS-GLMM succeed at identifying contamination when it exists while at the same time yielding reasonable parameter estimates based on the data that is well fit by the model. The ROBMS-GLMM performs similarly to its non-robust version when there is no contamination present and substantially better in the presence of contamination. The asymptotic properties of the ROBMS-GLMM estimators are derived in two different fashions by appealing to the results of White [48] as well as those of Yuan and Jennrich [49]. A simulation study demonstrates these properties.

One cannot estimate the parameters of a model robustly and then apply classical model selection procedures. We therefore recommend two approaches to robust model selection. The first is a robust version of the Akaike Information Criterion suggested by Ronchetti [44]. The second is based on obtaining robust deviances which can then be used for stepwise model selection as in the classical framework. With these additional tools one can achieve robust inference in longitudinal analyses.

The methodology is illustrated with analyses of longitudinal data currently of interest to medical researchers. We commence with an analysis of a prospective longitudinal study of laryngoscopic endotracheal intubation (LEI), a skill which numerous health care professionals are expected to acquire. Unfortunately, at present there is little information to indicate the amount of training required, or what signifies true competence in LEI. Hence our goal in working along side of the anesthesiologists directing the study, is to identify features of the process of LEI which are predictive of a successful LEI.

Longitudinal studies are also very popular in psychiatric clinical research. We breifly examine data involving 269 Nova Scotian adults taking an anti-depressant drug. Each patient makes regular visits to the psychiatrist for assessment, during each of which comprehensive information about health, drug use and psychiatric status are recorded. Interest is in the probability of remission from depression and how it varies with certain factors.

We also examine data extracted from a database which contains information on every women in Nova Scotia who has given birth since 1988. Interest is in examining the smoking behavior of women and whether certain outcomes influence smoking behavior during

subsequent pregnancies.

User-friendly *R* routines are provided for fitting ROBMS-GLMMs. These routines help conceal the numerical intensity of fitting such models. We address numerical issues associated with current model fitting software for the ROBMS-GLMM as well as its earlier non-robust version. Recommendations for future implementations are made in light of recent advances in statistical computing.

1.3 Outline

This thesis will be broken down in the following manner. In chapter 2 we introduce longitudinal data and present popular methods for its analysis. In chapter 3 we contrast these popular approaches using a real example and motivate the need for a new modelling approach. We select Heagerty's marginally specified generalized linear mixed model: a newly available alternative which is both flexible and easily interpreted. In chapter 4 we discuss our robust extension: the robust marginally specified generalized linear mixed model and in chapter 5 demonstrate its performance via a simulation study. Also included in this chapter is a section highlighting some important computational issues and recommendations. In chapter 6 we discuss the asymptotic properties of our model estimates and propose methodologies for robust model selection. Chapter 7 contains some interesting medical applications. Finally, chapter 8 summarizes our results and discusses some possible further research.

Chapter 2

Methods for the Analysis of Longitudinal Data

In this chapter we introduce longitudinal data and demonstrate the merits of performing longitudinal studies. We then give an overview of approaches used to model such data. This includes a detailed look at marginal, latent variable and response conditional models. A critical appraisal of these models follows in chapter 3.

2.1 What is Longitudinal Data?

In classical univariate statistics, one usually assumes that each of a number of subjects, or experimental units, gives rise to a single measurement on some relevant variable. This single measurement is termed the response. In multivariate statistics, the single measurement on each subject is replaced by a vector of measurements. For example, in a univariate medical study we might measure the weight of each subject, whereas in a multivariate study we might measure weight, blood pressure, cholesterol level and so on. In *longitudinal studies*, each subject gives rise to a matrix of measurements, these measurements now represent the same quantity(s) measured at a sequence of observation times. Thus, for example, we might measure a subject's blood pressure on each of five successive days. In one of the medical data sets we consider, the subjects are patients and the measurements on each subject consist of a collection of binary responses indicating whether or not the patient was in remission from depression at each of a series of visits to the psychiatrist for assessment.

Longitudinal data show up in many fields of empirical research. In sociology and economics longitudinal data arise frequently and are often referred to as panel data [12]. Longitudinal studies are an integral part of medical and biological science. They have been fundamental to the study of physical and cognitive development [16] and are important in the study of chronic diseases such as arthritis and diabetes.

Longitudinal data exhibits characteristics of multivariate, time series and survival data. Unlike multivariate data, longitudinal data generally exhibit a more highly structured pattern of interdependence. With multivariate data we again have a vector for each subject but this vector is comprised of a number of different measurements. As well, with longitudinal data we are measuring the *same* quantity at a sequence of observation times. Longitudinal data are also inherently unbalanced which is not usually the case with multivariate data. It is also worthwhile contrasting longitudinal data with that of time series. Longitudinal data consists of a large number of *short* time series, rather than a single, long time series. As well, in longitudinal studies the correlation structure is normally of secondary interest whereas in time series we often wish to determine whether the data exhibits an autoregressive or stationary structure, for example.

Finally we make comparisons with survival data. Such data is concerned with the time to a clinical outcome. Hence time is usually the principal response and generally there is only one observation per subject unlike with longitudinal data where we have repeated measurements on each subject.

2.1.1 Notation

To explore the ideas surrounding the approaches to longitudinal data analysis we commence with an introduction to the standard notation used. Consider the response Y and set of predictors X where n_i repeated measurements were taken on each subject i:

- The response is denoted $Y = \{Y_1, \cdots, Y_k\}$ where k represents the number of individuals. The response for each individual i is $Y_i = (Y_{i1}, \cdots, Y_{in_i})^T$ where Y_{ij} is the response at time j for individual i. In general n_i may not equal n_j for all $i \neq j$ and therefore Y is not a matrix. In longitudinal studies Y_i is considered to be the natural experimental unit as opposed to Y_{ij}.

- For each individual i, X_i represents an $n_i \times p$ matrix of p predictors. The elements of

X_i, X_{ijp}, represent the j^{th} observation of predictor p on individual i.

2.1.2 Merits of Longitudinal Studies

The defining characteristic of a longitudinal study is that subjects are measured repeatedly through time. Consequently each subject gives rise to a vector of measurements which represent the same physical quantity measured at a sequence of observation times.

The major advantage of a longitudinal study over a cross-sectional study (in which a single outcome is measured for each individual) was eloquently described by Diggle, Liang and Zeger [12]:

> Longitudinal studies can distinguish changes over time within subjects (ageing or *longitudinal* effects) from differences among subjects in their baseline levels (*cohort* effects).

This idea is best illustrated with an example. In Figure 2.1(a) some measure of computer literacy is plotted against age for a hypothetical cross sectional study of Nova Scotians between the ages of 10 and 40. The conclusion suggested by this plot is that computer literacy is poorer among adults. In Figure 2.1(b) we assume that the same data were obtained in a longitudinal study in which each person was measured twice. It is now clear that while younger people began at a higher literacy level, everyone improved with time.

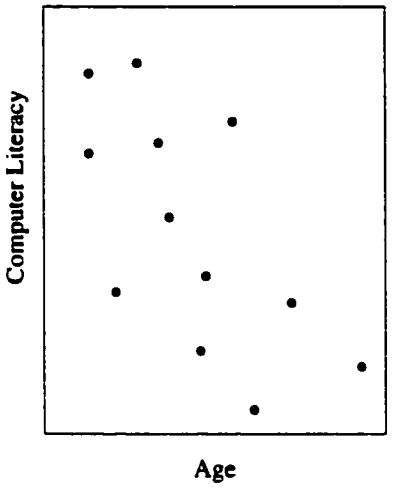
We can now generalize this example to the class of linear regression models. We consider the simple linear regression model without intercept. In a cross-sectional study ($n_i = 1$) we are restricted to the model

$$Y_{i1} = X_{i1}\beta_C + \varepsilon_{i1}, \qquad i = 1, \cdots, k \tag{2.1}$$

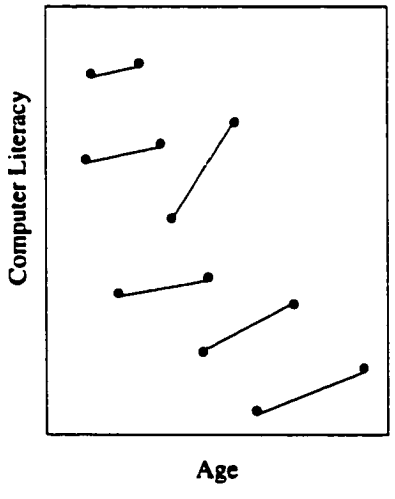
where β_C represents the difference in average Y across two sub-populations which differ by one unit in x. With repeated observations, the linear model can be extended to the form

$$Y_{ij} = X_{i1}\beta_C + (X_{ij} - X_{i1})\beta_L + \varepsilon_{ij}, \qquad j = 1, \cdots, n_i, \qquad i = 1, \cdots, k \tag{2.2}$$

Note that when $j = 1$, (2.2) reduces to (2.1) so β_C has the same cross-sectional interpretation. However, we can now also estimate β_L. The interpretation is made clear by subtracting



(a)



(b)

Figure 2.1: Computer Literacy

(2.1) from (2.2) to obtain

$$(Y_{ij} - Y_{i1}) = (X_{ij} - X_{i1})\beta_L + \varepsilon_{ij} - \varepsilon_{i1}.$$

That is, β_L represents the expected change in Y over time per unit change in X for a given subject.

To estimate how individuals change with time from a cross-sectional study, we must assume $\beta_C = \beta_L$. With a longitudinal study, this strong assumption is unnecessary since both can be estimated.

Even when $\beta_C = \beta_L$, longitudinal studies tend to be more powerful than cross-sectional studies. The basis of inference about β_C is a comparison of individuals with a particular value of X to others with a different value. In contrast, the parameter β_L is estimated by comparing a person's response at two times, assuming X changes with time. In a longitudinal study, each person can be thought of as serving as his or her own control. For most outcomes, there is considerable variability across individuals due to the influence of unmeasured characteristics such as genetic make-up, environmental exposures, personal habits, and so on. These tend to persist over time. Their influence is cancelled in the estimation of β_L; they obscure the estimation of β_C .

Another merit of the longitudinal study is its ability to distinguish the degree of variation in Y across time for one person from the variation in Y among people. This partitioning of the variation in Y is important for the following reason. With repeated values, we can borrow strength across time for the person of interest as well as across people. If there is little variability among people, one person's estimate can rely on data for others as in the cross-sectional case. However, if the variation across people is large, we might prefer to use only data for the individual. Given longitudinal data, we can acknowledge the naturally occurring differences among subjects when estimating a person's current value or predicting a future one.

2.1.3 Approaches to Analysis

Repeated observations on individuals enable direct study of change. Longitudinal data require special statistical methods because the repeated observations on a subject tend to be

correlated (however the subjects themselves are independent of one another). This within-subject correlation presents additional opportunities and challenges for analysis and must be taken into account to draw valid scientific inferences. Ignoring correlation when it exists results in two problems: inefficient estimates of regression parameters, and, more importantly, inconsistent estimates of precision [52].Consequently, we have two objectives for statistical models of longitudinal data:

1. to adopt the conventional regression tools, which relate the response variables to the explanatory variables; and

2. to account for the within-subject correlation.

In their cardinal book on longitudinal data [12], Diggle *et al.* discuss three distinct strategies available for analyzing longitudinal data: marginal, random effects and transition models. More recently [23] the latter two approaches have been renamed latent variable and response conditional models, respectively. All three are designed for the analysis of discrete and continuous longitudinal data using extensions of generalized linear models (GLMs). Consequently, we now first review GLMs and then look at each of the approaches in detail in subsequent subsections.

2.1.4 Review of Generalized Linear Models

In biomedical applications, it is common to confront both discrete and continuous outcome measures. Regression models for independent responses, discrete and continuous, have been unified under the class of generalized linear models (GLM). This has facilitated data analysis by providing a common set of methods regardless of the type of response.

Consider the cross-sectional situation with response Y_i and $1 \times p$ vector of explanatory variables $X_i, i = 1 \cdots, k$. The objective is to describe the dependence of the mean response $\mu_i = E(Y_i)$ on the covariates. A GLM is a member of the exponential family with likelihood function of the form

$$f(y_i) = \exp\{(y_i\theta_i + a(\theta_i))/\phi + c(y_i, \phi)\}$$

where the mean μ_i equals the first derivative of the function $a(\theta_i)$ and the variance of Y_i is proportional to the second derivative of $a(\theta_i)$. The mean μ_i is related to the explanatory

variables by $g(\mu_i) = X_i\beta$ where g is the *link* function. The variance is related to the mean by $var(Y_i) = v_i = v(\mu_i)\phi$ where v is called the variance function.

The GLM family includes linear, logistic, log-linear and some parametric survival regression models as special cases. In each GLM the regression coefficients β are estimated by solving the estimating equation

$$S(\beta) = \sum_{i=1}^{K} \frac{\partial \mu_i'}{\partial \beta} v_i^{-1}(Y_i - \mu_i(\beta)) = 0. \tag{2.3}$$

The solution β can be obtained by iteratively weighted least squares and is a consistent estimate as long as $g(\mu_i) = X_i\beta$ whether or not the variance function is correctly specified.

Wedderburn [47] first pointed out that the estimating equation (2.3) provides a consistent estimate of β with a variety of link and variance functions whether or not they correspond to a member of the exponential family of distributions. The name *quasi-likelihood estimate* was coined for the solution of equation (2.3) in the more general case since its integral does not necessarily constitute a proper likelihood function.

2.2 Marginal Models

The *marginal modeling* approach builds separate regressions for first, second and higher moments of the joint distribution, $[Y_i|X_i]$. The marginal expectation is linked to covariates using a generalized linear model. Additional moments are then specified for the second moment, and possibly for the higher moments. For example, Dale [11] parameterized the joint distribution of two binary variables in terms of their marginal means and their odds ratio. Others, including Heagerty and Zeger [22], extended this idea to vectors of ordinal responses specifying separate regression models for: the marginal means; pairwise odds ratios; and higher order contrasts among log-odds ratios to complete the likelihood function.

In these marginal models, the mean (or first moment) regression parameters represent the change in expected response, per unit change in a given predictor without conditioning on the other responses or any latent variables (random effects). Correlation among elements of Y_i given X_i, even if reasonably attributed to shared unobservable latent variables, are accounted for by a separate association regression.

Specifically, we can formulate a marginal model by assuming:

1. the marginal expectation of Y_{it}, $E(Y_{it}) = \mu_{it}$ is related to X_{it} by

$$g(\mu_{it}) = X_{it}\beta,$$

where g is a known link function such as the logit function for binary responses where $logit(\mu_{it}) = \log(\mu_{it}/(1-\mu_{it}))$;

2. the marginal variance is a function of the marginal mean, that is,

$$var(Y_{it}) = \nu(\mu_{it})/\phi,$$

where ν is a known function and ϕ is the over-dispersion parameter which accounts for the variation of Y_{it} not explained by $\nu(\mu_{it})$;

3. the covariance between Y_{is} and Y_{it}, $s < t = 1, \cdots, n_i$ is a function of the marginal means and additional parameter α, that is,

$$cov(Y_{is}, Y_{it}) = c(\mu_{is}, \mu_{it}; \alpha),$$

where c is a known function.

There are several advantages to the marginal approach. First, the interpretation of regression coefficients in the mean model does not depend on the dimension of Y_i as it does in response conditional models. Hence unbalanced data can be easily accommodated. Second, the interpretation of mean parameters, β, is invariant with respect to specification of the association or higher order models. This property does not hold for response conditional or latent variable models. Marginal models are often referred to as *population-averaged* models because they describe how the response average changes across various subsets of the population. These subsets are defined by the covariate values.

Note that only the first two moments of the joint distribution of Y_i are specified by assumptions 1-3. In general, marginal models separate the parameterization of the mean and higher order moments, so it is possible to estimate mean parameters without specifying the complete joint distribution of Y_i. Liang and Zeger [33] introduced *generalized estimating*

equations (GEE), a technique for fitting marginal models, requiring only the first two moments of the joint distribution of Y_i. This approach is now very popular and is discussed in the next subsection.

2.2.1 Generalized Estimating Equations

Generalized estimating equations (GEE) were developed by Zeger and Liang [52] by building on the quasi-likelihood theory. They enable us to fit marginal models.

Quasi-likelihood functions, are a useful tool when we only know the form of the relationship between mean and variance. Quasi-likelihood is a method of making inference even if there is insufficient information to construct a likelihood.

Suppose that $Y_1, \cdots, Y_k$ are k independent observations such that $E[Y_i] = \mu_i$ and $Var[Y_i] = V(\mu_i)$. In other words, the variance is assumed to be proportional to some function of the mean. We can now present the quasi-likelihood function $Q(Y_i, \mu_i)$ for a generic observation Y_i having mean μ_i and variance $V(\mu_i)$. The function $Q(Y_i, \mu_i)$ is defined by

$$\frac{\partial}{\partial \mu} Q(Y_i, \mu_i) = \frac{Y_i - \mu_i}{V(\mu_i)},$$

or equivalently

$$Q(Y_i, \mu_i) = \int_{Y_i}^{\mu_i} \frac{Y_i - t}{V(t)} dt.$$

The global quasi-likelihood for a whole sample is then defined to be the sum of the individual contributions, that is,

$$Q(Y, \mu) = \sum_{i=1}^{k} Q(Y_i, \mu_i),$$

where $\mu = (\mu_1, \cdots, \mu_k)$ and $Y = (Y_1, \cdots, Y_k)$.

If we suppose that μ is expressed as a function of β, then $Q(Y, \mu)$ has the following properties:

1. $E[\frac{\partial}{\partial \mu} Q(Y, \mu)] = 0$,
2. $E[\frac{\partial}{\partial \beta_i} Q(Y, \mu)] = 0$,
3. $E[\frac{\partial}{\partial \mu} Q(Y, \mu) \frac{\partial}{\partial \mu} Q^T(Y, \mu)] = -E[\frac{\partial^2}{\partial \mu \partial \mu^T} Q(Y, \mu)] = 0$, and

4. $E[\frac{\partial}{\partial\beta_i}Q(Y,\mu)\frac{\partial}{\partial\beta_j}Q^T(Y,\mu)] = -E[\frac{\partial^2}{\partial\beta_i\partial\beta_j}Q(Y,\mu)]$.

For the GLM the mean μ_i can be expressed as a function of $X_i\beta$ via the link function, that is $\mu_i = g^{-1}(X_i\beta)$. We may then use the quasi-likelihood (in place of the likelihood function) for estimation and inference.

Consider the longitudinal observations (Y_{ij}, X_{ij}) for times $j = 1, \cdots, n_i$ and subjects $i = 1, \cdots, k$. Here Y_{ij} is the outcome variable and X_{ij} is a $1 \times p$ vector of covariates. Quasi-likelihood was originally applied to the regression context where $n_i = 1$ for all i. So, if we drop the subscript j the quasi-likelihood estimator is the solution of the score-like equation system (which we initially saw in Section 2.1.4, see equation (2.3))

$$S_m(\beta) = \sum_{i=1}^{k} \frac{\partial \mu_i'}{\partial \beta_m} v_i^{-1}(Y_i - \mu_i) = 0, \qquad m = 1, \cdots, p. \tag{2.4}$$

GEE are a modification of the estimating equations for quasi-likelihood (2.4) which allow the estimation of regression parameters when dealing with longitudinal data.

To apply the quasi-likelihood approach to the analysis of longitudinal data, the mean and covariance of the vector of responses, Y_i, for the ith subject must be considered. Zeger and Liang extend the quasi-likelihood theory by letting $R_i(\alpha)$ be an $n_i \times n_i$ symmetric matrix which fulfills the requirement of being a correlation matrix, and α be an $s \times 1$ vector which fully characterizes $R_i(\alpha)$. They refer to $R_i(\alpha)$ as a *working* correlation matrix because they do not expect it to be correctly specified and desire estimators that are consistent and have consistent variance estimates even when $R_i(\alpha)$ is incorrect. Note that the observation times and correlation matrix can differ from subject to subject. $R_i(\alpha)$ however, is assumed to be fully specified by the unknown parameter, α, which is the same for all subjects. Then following the quasi-likelihood approach, the working covariance matrix for Y_i is given by

$$V_i = a(\phi)A_i^{1/2}R_i(\alpha)A_i^{1/2}/\phi$$

where A_i is an $n_i \times n_i$ diagonal matrix with $g(\mu_{ij})$ as the jth diagonal element and ϕ is some scale parameter. If $R_i(\alpha)$ is the true correlation matrix of Y_i, then V_i will be the true covariance matrix of Y_i.

Zeger and Liang's [52] extension of equations (2.4) to the longitudinal data case is

given by

$$U(\beta,\alpha) = \sum_{i=1}^{k} (\frac{\partial \mu_i}{\partial \beta})' V_i^{-1} (Y_i - \mu_i) = 0 \tag{2.5}$$

The estimating equations (2.5) now depend on α as well as β, but can be re-expressed as a function of β alone by first replacing α by a $k^{1/2}$-consistent estimator, $\hat{\alpha}(Y,\beta,\phi)$ and then replacing ϕ in $\hat{\alpha}$ by a $k^{1/2}$-consistent estimator, $\hat{\phi}(Y,\beta)$. Consequently, for any given $R(\alpha)$, the estimate $\hat{\beta}_R$, of β is defined as the solution of equation (2.5) with the appropriate estimates involved.

To solve the GEE for $\hat{\beta}$ we iteratively solve for the regression coefficients and the correlation and scale parameters, α and ϕ. Given an estimate of $R_i(\alpha)$ and of ϕ, an updated estimate of β is calculated by iteratively re-weighted least squares as described by McCullagh and Nelder [34]. Given an estimate of β one calculates Pearson residuals, $e_{ij} = \frac{Y_{ij} - \hat{\mu}_{ij}}{\sqrt{v(\hat{\mu}_{ij})}}$, which are used to consistently estimate α and ϕ. These two steps are iterated until convergence.

There are several choices for the working correlation matrix, R_i. The simplest is to assume $R_i = I_{n_i}$, the $n_i \times n_i$ identity matrix, i.e. that repeated observations are uncorrelated. The GEEs then reduce to the score equations (2.4) used with independent observations. A second extreme case is applicable when observations times are the same for all subjects so that $R_i(\alpha) = R(\alpha)$ and $n_i = n$. One can then let $R(\alpha)$ be fully unspecified and estimate the $n(n-1)/2$ correlations. The following are the structures of the working correlation matrix supported by most software packages. Let $R_{t,s}$ denote the t,s element of R_i.

- **fixed**

$$R_i(\alpha) = R_0$$

  where R_0 is a user-specified correlation matrix.

- **Exchangeable**

$$R_{t,s} = \begin{cases} 1 & t = s, \\ \alpha & \text{otherwise.} \end{cases}$$

- **Unstructured**

$$R_{t,s} = \begin{cases} 1 & t = s, \\ \alpha_{ts} & \text{otherwise, } \alpha_{ts} = \alpha_{st}. \end{cases}$$

- **Auto-regressive**:

$$R_{t,s} = \begin{cases} 1 & t = s, \\ \alpha^{|t-s|} & \text{otherwise.} \end{cases}$$

- ***m*-dependent**:

$$R_{t,s} = \begin{cases} 1 & t = s, \\ \alpha_{|t-s|} & \text{otherwise.} \end{cases}$$

2.3 Latent Variable Models

Both latent variable and response conditional models (to be treated in the next subsection) are different from marginal models in that they attempt to address both the regression objective and the within-subject correlation simultaneously. In other words, parameters for the dependence on X and for correlation are introduced on the same scale in a common equation.

One way to model the joint distribution, $[Y_i, X_i]$, is to postulate the existence of unobserved latent variables, b_i, which are shared by, and hence introduce correlation among the elements of Y_i. The observed data likelihood is constructed by integrating over the latent variable distribution:

$$P_\theta[Y_i|X_i] = \int P_\theta[Y_i|X_i, b_i] f_\theta(b_i|X_i) db_i.$$

There are two common assumptions to simplify the model:

- conditional independence among responses such that

$$P_\theta[Y_i|X_i, b_i] = \prod_j P_\theta[Y_{ij}|X_i, b_i]$$

- homogeneous latent variable distribution

$$f_\theta(b_i|X_i) = f_\theta(b_i).$$

The first assumption forms the basis for structuring the correlation among responses within individuals. The second assumption is a strong one and recently Heagerty and Zeger [23] have considered a less stringent regression structure for the random effects variance components.

The most common partition of θ is into β and α where

$$P_\theta[Y_i|X_i] = \int \{\prod_j P_\beta[Y_{ij}|X_i, b_i]\} f_\alpha(b_i|X_i) db_i.$$

Here β are canonical regression parameters in a GLM for the conditional expectation of the response:

$$E(Y_{ij}|X_i, b_i) = \mu_{ij}$$

$$g(\mu_{ij}) = X_{ij}\beta + b_{ij}.$$

Assumptions about b_{ij} commonly used in practice include: mixed models where $b_{ij} = Z_{ij}u_{ij}$ for Z_{ij} a subset of X_{ij} and u_{ij} is a $q \times 1$ vector of random effects; and serial or spatial models where b_{ij} represents an autocorrelated stochastic process. The parameter α identifies the specific distribution of b_i from within the chosen parametric family.

Such models are often referred to as *multilevel* models and are popular in the empirical sciences for a number of reasons. First, it is often reasonable to postulate that shared, unobservable, variables influence the response, thereby making observations on one individual correlated with one another. Second, the multilevel regression parameter has a desirable causal interpretation as the change in (possibly transformed) expected response per unit change in X_j, holding the other observed variables and unobserved latent factors fixed. Third, these models make possible the estimation of individual-specific regression coefficients, for example intercepts, that use information from observations on a specific individual but which also borrow information from other individuals. Such estimates are often superior to competitors which rely only on an individual's own data [23].

In the conditional mean parameterization, the regression contrasts, β, measure the

change in transformed mean per unit change in a covariate, controlling for all other variables including the latent variables b_{ij}. Since the latent variable assumptions determine what values of b_{ij} are equivalent, these assumptions also determine the interpretation of the parameter β. If the b_{ij} are constant over time (which is the *random intercepts* assumption) the parameters β have a subject-specific interpretation [50]. However one can view the random intercepts assumption as a special case of a more general serially dependent stochastic process model with $cov(b_{ij}, b_{ik}) = \sigma^2\rho^{|j-k|}$, where for the random intercepts model, $\rho = 1$ [23]. One can then relax the random intercepts model to allow dependence to decay as the time separation increases and then β no longer measures the change in an individual's log-odds since controlling for the individual no longer ensures that the latent variables b_{ij} and b_{ik} are equal. A simple change in the latent variable assumptions can make β both subject and time specific.

As an example of a simple latent variable model, consider a study of changes in weight during pregnancy. It may be reasonable to assume that the relationship between weight and the gestational week is linear for each expectant mother, but with linear regression parameters that vary among mothers. In this formulation, the probability distribution for the multiple measurements is the same for each mother, but the parameters of that distribution vary over mothers. These parameters are the *random effects* or latent variables.

At present, the latent variable models that are most commonly used in practice assume random intercepts. These models are most useful when the objective is to make inference about individuals rather than the population average. Sometimes they are referred to as *cluster-specific* methods.

2.4 Response Conditional Models

There are two main classes of models for multivariate data that can naturally be viewed as models for the expected value of one response conditional on subsets of the other responses from the same individual. These are (i) transition models and (ii) log-linear models and are both effective for modeling associations but do not admit simple models for the marginal means.

2.4.1 Transition Models

Transition models focus on the conditional expectation of Y_{ij} given past outcomes, $Y_{ij-1},\cdots,Y_{i1}$. Here the data analyst specifies a regression model for the conditional expectation, $E(Y_{ij}|Y_{ij-1},\cdots,Y_{i1},X_{ij})$, as an explicit function of X_{ij} and of the past outcomes.

Transition models assume

1. the conditional expectation of Y_{it}, $\mu_{it}^c = E(Y_{it}|Y_{it-1},\cdots,Y_{i1})$, depends on X_{it} and past responses as follows:

$$g(\mu_{it}^c) = X_{it}\beta^* + \sum_{j=1}^{\nu} \gamma_j^* f_j(Y_{it-1},\cdots,Y_{i1}), \tag{2.6}$$

   where $\{f_j\}, j = 1,\cdots,\nu$ are known functions.

2. the conditional variance of Y_{it} given the past is a function of μ_{it}^c; that is,

$$var(Y_{it}|Y_{it-1},\cdots,Y_{i1}) = v(\mu_{it}^c)\phi,$$

   where v is a known function.

It is possible to construct a multivariate model by decomposing the joint distribution into a sequence of predictive distributions:

$$P_{\theta_i}(Y_{i1},Y_{i2},\cdots,Y_{in}) = P_{\theta_i}(Y_{i1}) \prod_{j=2}^{n} P_{\theta_i}(Y_{ij}|Y_{ik} : k < j).$$

It is straightforward to allow the parameters to depend on covariates, X_i, but difficult to obtain simple expressions for $E(Y_{ij}|X_i)$ since sums over the joint distribution of times $1,2,\cdots,j$ are required. Azzalini [4] has been successful in marginalizing some simple transition models for discrete responses. We will see in section 3.3 that the development of the marginalized latent variable model parallels the approach taken by Azzalini[23] as well by Fitzmaurice and Laird [14].

2.4.2 Log-linear Models

Log-linear models [5] have been widely used for the analysis of cross-classified discrete observations. In a log-linear model, the natural univariate regressions are for the conditional expectations $logit\ E(Y_{ij}|Y_{ik} : k \neq j)$. Unfortunately, the models result in complicated functions for the marginal expectations since these are obtained as sums over the response variable joint distribution, yielding mixtures of exponential functions of the parameters. Therefore, although log-linear models are well suited for describing multivariate dependencies or for modeling joint and conditional distributions, they do not facilitate generalized linear regression modeling of the marginal means.

There have been a number of attempts to marginalize log-linear models so as to permit likelihood based regression estimation of the marginal means. Fitzmaurice and Laird [14] were successful in doing so for balanced data and Lang and Agresti [32], for small or moderate cluster sizes. Unfortunately such restrictions limit the use of these models.

2.5 Computational Aspects

SAS, Stata, SUDAAN and S-Plus are the four packages most commonly used for fitting marginal models to longitudinal data. On the whole, GEEs were found to be well-supported by all of these software packages and straightforward to use [25]. Since these methods are not likelihood based they may afford robustness to misspecification of the multivariate dependence structure, yet do not allow for the use of likelihood based procedures such as profile likelihood functions and likelihood ratio tests. In many situations likelihood based methods may be preferred.

Latent variable models are more difficult to fit because evaluation of the likelihood requires numerical integration in most problems. Serious computational difficulties have resulted in a wide range of methods being investigated. One approach to avoid the numerical integrals is to approximate the integrands (see equation (2.3)) with simple expansions whose integrals have closed forms. This was the approach proposed from a Bayesian perspective by Stiratelli *et al.* [46] and recently reviewed by Breslow and Clayton [7]. These approximate techniques give effective estimates of the fixed effects but are somewhat biased for estimating random effects and the random effects variance matrix, especially when the variance is large. A Bayesian approach for fitting random effects GLMs using Gibbs

sampling has been proposed by Zeger and Karim [51]. Other methods include maximum penalized quasi-likelihood and marginal quasi-likelihood via EM or quasi-Newton (Fisher scoring) algorithms, and Markov Chain Monte Carlo (MCMC) methods.

When fitting latent variable models many suggest that when the random effects are only of one-dimension, numerical integration can be implemented fairly easily. Our experience suggests that this is not the case. Even with one-dimensional random effects, algorithms are slow, and careful approximation is required. These computational issues will be elaborated on in Chapter 5.

We have now given an overview of longitudinal data and introduced the popular modeling approaches. In the next chapter we contrast these approaches via a medical application. It becomes clear that a new model is required, one which combines desirable properties from each of these distinct approaches.

Chapter 3

A Critical Appraisal of Longitudinal Data Analysis Methods

In chapter 2 we outlined three approaches to longitudinal data analysis. Heagerty [20] states that marginal and latent variable models are the two major regression approaches for the analysis of longitudinal binary data. Our literature review supports this statement and suggests that it is also true for longitudinal data in general. We therefore discuss all three approaches while placing less emphasis on response conditional models. While there are no substantial reasons to suggest that response conditional models are inferior, in the context of biomedical research they are less commonly used. Possible explanations may include the lack of readily available routines in standard statistical software packages as well as the difficulty in arriving at simple models for the marginal means.

In this chapter we expose limitations of both marginal and latent variable models using a medical application. We look in detail at the interpretation of coefficients. We then present Heagerty's marginally specified generalized linear mixed model [20], a newly available alternative.

3.1 Contrasting Approaches

In order to effectively contrast the commonly used approaches to longitudinal data analysis we introduce a longitudinal data set which typifies problems arising in biomedical research.

Dr. John LeBlanc and colleagues in the Department of Psychiatry at Dalhousie University conducted a study of Nova Scotian adults using Zoloft, an anti-depressant drug. Interest was in determining what factors influence the probability of remission from depression. Each patient involved in the study made regular visits over a six month period to the psychiatrist for assessment. Explanatory variables (covariates) include dose level (*dose*), and a quantitative depression score (*score*) which are time-dependent. That is, $dose_{it}$ equals 1 if patient i is taking the high dose of Zoloft at visit t and 0 if they are taking the low dose. $score_{it}$ is the Hamilton-D depression index score for patient i at the tth visit. The Hamilton-D depression index score is on a 29 point scale: the higher the score the more severe the depression. The response is binary, that is, Y_{ij} equals 1 if patient i is in remission from depression at visit j and 0 otherwise.

3.1.1 Marginal Model

We are interested in whether the probability of remission is related to the level of dose of Zoloft prescribed. In a marginal model, we might assume

1. $logit(\mu_{it}) = \beta_0 + \beta_1 dose_{it} + \beta_2 score_{it}$, where $\mu_{it} = E(Y_{it}) = Pr(Y_{it} = 1)$ and
2. $var(Y_{it}) = \mu_{it}(1 - \mu_{it})$, and
3. $corr(Y_{is}, Y_{it}) = \alpha_{|s-t|}, s < t$.

That is, the correlation coefficient between two repeated observations from the same subject depends only on the time between the two visits. The covariance between Y_{is} and Y_{it} is fully specified by assumptions 2 and 3 above.

- Remark 1. In this example, e^{β_1} has the interpretation as approximately the ratio of odds of remittance for two populations of patients, those taking the high dose of Zoloft and those taking the low dose at a given visit. In other words

$$e^{\beta_1} \cong \frac{\text{prevalence of remittance among patients taking the high dose}}{\text{prevalence of remittance among patients taking the low dose}}.$$

  The marginal model regression coefficients have what we call *population-averaged* interpretations because they contrast odds of remission from depression in the two populations.

- Remark 2. Note that the effect of the depression score, β_2, can be estimated cross-sectionally if there is heterogeneity in the depression score distribution of the patients sampled at one time. However, this cross-sectional depression score effect may be subject to bias if so-called cohort effects are present in the study population. For instance, suppose that remittance from depression is more likely for those with a lower depression score at baseline. This bias can be detected and corrected with longitudinal data by expanding the model to include depression score at baseline and change in depression score from baseline as covariates. The coefficient for score at baseline is sensitive to cohort effects; that for change in score is not. It is of interest to examine if, and in what way, these two depression score coefficients are different.

- Remark 3. Because β describes the effect of covariates on the marginal expectation of the $Y's$, it has the same interpretation regardless of the number of repeated observations, n_i, which may vary among subjects.

- Remark 4. The third assumption is needed to account for within-subject correlation. However, the magnitude of this correlation, indexed by α, does not alter the interpretations of β.

3.1.2 Response Conditional Model

In order to fit a response conditional model to the Zoloft data, we might assume

1. $logit\ Pr(Y_{it} = 1|Y_{it-1}, \cdots, Y_{i1}) = \beta_0^* + \beta_1^* dose_{it} + \beta_2^* score_{it} + \gamma^* Y_{it-1}$,
2. $var(Y_{it}|Y_{it-1}) = \mu_{it}^c(1 - \mu_{it}^c)$.

In this special case, $v = 1$ (see Section 2.4.1), so that Y_{it} is assumed to depend upon the past responses only through the immediately preceding response.

- Remark 1. The table that follows gives the transition probability of remission from depression.

		Y_{it}	
		0	1
Y_{it-1}	0	$\frac{1}{1+e^{x_{it}\beta^*}}$	$\frac{e^{x_{it}\beta^*}}{1+e^{x_{it}\beta^*}}$
	1	$\frac{1}{1+e^{x_{it}\beta^*+\gamma^*}}$	$\frac{e^{x_{it}\beta^*+\gamma^*}}{1+e^{x_{it}\beta^*+\gamma^*}}$

Suppose the depression duration is similar to the sampling interval. Then because of the adjustment for Y_{it-1} in assumption 1, $e^{\beta_1^*}$ is approximately the ratio of incidences from two groups, one taking the high dose of Zoloft and one not. In other words,

$$e^{\beta_1^*} \cong \frac{\text{Incidence of remittance among patients taking the high dose}}{\text{Incidence of remittance among patients taking the low dose}}.$$

- Remark 2. Examining assumption 1 closely as well as equation (2.6), we note that the interpretation of β^* changes with the length of the dependencies, that is, by including additional terms such as $\gamma_2^* Y_{it-2}$. This is to be contrasted with Remark 4 of the previous subsection concerning the parameters in marginal models.

- Remark 3. Transition models can be fit with standard software by treating $f_1, \cdots, f_v$ in equation (2.6) along with x, as the set of regressors.

3.1.3 Latent Variable Model

In order to propose a latent variable model to describe the Zoloft data, we might assume a random intercept model as follows:

1. $logit\ Pr(Y_{it} = 1|b_i) = \beta_0^{**} + b_i + \beta_1^{**} dose_{it} + \beta_2^{**} score_{it}$.
2. $Var(Y_{it}|b_i) = E(Y_{it}|b_i)\{1 - E(Y_{it}|b_i)\}$.
3. $b_i \sim N(0, \theta)$.

- Remark 1. $e^{\beta_1^{**}}$ has interpretation as the odds ratio of remission from depression for a patient who is taking the high dose versus the same patient if he/she is taking the low dose. In other words, it describes the change in the odds of remission for a patient

whose dose level changes. Note the odds ratio is assumed in the model to be the same across subjects.

- Remark 2. The value of β^{**} depends on the assumed distribution F for the b_i's. For discrete longitudinal data with small n_i, it is difficult to check the validity of F. It is therefore good practice, but difficult, to examine how sensitive the results are to the specified distribution for the b_i's.

Table 3.1 summarizes the interpretation of the regression coefficients β, β^* and β^{**} using the variable dose to illustrate. Given different interpretations for coefficients in each model, their magnitudes are expected to be different as well.

Table 3.1: Summary of regression coefficient interpretations.

Model	Parameter	Interpretation
Marginal	$\exp(\beta_1)$	ratio of population prevalences
Response Conditional	$\exp(\beta_1^*)$	ratio of population incidences
Latent Variable	$\exp(\beta_1^{**})$	odds ratio for individual patient

The relationship between β and β^* can be established in special cases. If we ignore the depression score variable for simplicity, e^{β_1} is the odds ratios relating the frequency of remission from depression to dose level. On the other hand, $e^{\beta_1^*}$, is the odds ratio for remission from depression and dose level controlling for the remission status at the previous visit. If a positive correlation exists between Y_{t-1} and Y_t as we expect, and there is a positive effect of dose level on the probability of remission, it can be shown that [5]

$$|\beta_1^*| \leq |\beta_1|.$$

The latent variable model discussed here is most useful when the objective is to make inference about individuals rather than the population average. In the Zoloft Study, the latent variable model would permit inference about the probability of a particular patient being in remission from depression. The regression coefficients, β^{**}, represent the effects of the explanatory variables on an *individual*'s chance of remission. This is in contrast to the marginal model coefficients which describe the effect of explanatory variables on the

population average. In the next section we look in greater detail at the interpretation of latent variable and marginal model coefficients.

3.2 Interpretation of Regression Coefficients

A latent variable model results in the probability distribution of Y_{ij} being modeled as a function of the covariates X_{ij}, and parameters b_i specific to the ith subject. With a marginal model, the marginal or population-averaged expectation of Y_{ij} is modeled as a function of the covariates.

To compare these two approaches in more detail, we will begin by considering a hypothetical linear model. We can formulate the two regression approaches to have coefficients with the same interpretation. To illustrate, consider the simple linear regression model:

$$Y_{ij} = \beta_0 + X_{ij}\beta_1 + \varepsilon_{ij}$$

where X_{ij} is the age, in years of individual i at visit j, Y_{ij} is the response at age X_{ij} and ε_{ij} is a mean-zero deviation. The residuals, $\varepsilon_{i1}, \cdots, \varepsilon_{in_i}$, for individual i will likely be correlated with one another. The marginal modeling approach is to assume:

1. $E(Y_{ij}) = \beta_0 + X_{ij}\beta_1$
2. $Corr(\varepsilon_{ij}, \varepsilon_{ik}) = \rho(X_{ij}, X_{ik}, \alpha)$.

Assumption 1 is that the average response for all individuals in the population at any age X is $\beta_0 + X\beta_1$. The parameter β_1 is therefore the change per year in the population-average response. Assumption (2) specifies the nature of the correlation. Clearly, in the marginal approach we separate the modeling of the regression and the correlation; either can be changed without necessarily changing the other.

A linear latent variable model can be written

$$Y_{ij} = \beta_0^* + b_i + X_{ij}\beta_1^* + \varepsilon_{ij}^*$$

where the ε_{ij}^* are independent $N(0, \sigma_e^2)$ variates and the b_i are independent $N(0, \sigma_A^2)$ variates. In this model, the regression coefficients also have a marginal interpretation since $E(Y_{ij}) =$

$\beta_0^* + X_{ij}\beta_1^*$. This is because the average of the rates of change in response for individuals is the same as the change in the population-average response across time in a linear model.

We have just shown for the linear model that regression coefficients can have a marginal interpretation for both the marginal and latent variable approaches. This fact may help explain why there has been so much confusion and difficulty understanding why in some cases the regression coefficients have different interpretations. For instance, with non-linear link functions, such as the logit, this is not the case. It is the non-linear link that causes the problem as will be shown by what follows. We consider the logistic regression model which is suitable for longitudinal binary data.

For demonstrative purposes we return to the Zoloft data but for the sake of simplicity consider only the dose level covariate. That is, we let X_{ij} equal 1 if patient i is taking the high dose of Zoloft at visit j and 0 if he/she is taking the low dose. Given that $logit^{-1}(x) = \exp(x)/\{1+\exp(x)\}$, the latent variable logistic model

$$P(Y_{ij} = 1|b_i) = logit^{-1}(\beta_0^* + b_i + X_{ij}\beta_1^*) \tag{3.1}$$

then states that each patient has their own baseline chance of remission $\exp(\beta_0^* + b_i)/\{1 + \exp(\beta_0^* + b_i)\}$ and that a patient's odds of remission are multiplied by $\exp(\beta_1^*)$ if they switch to the high dose of Zoloft. Hence, the odds ratio for dose level is the same for every patient. However the corresponding change in odds of remission differs depending on the baseline rate.

The population rate of remission is given by

$$P(Y_{ij} = 1) = \int logit^{-1}(\beta_0^* + b_i + X_{ij}\beta_1^*)dF(b_i)$$

where $F(.)$ is the Gaussian distribution function since we are assuming b_i are independent $N(0, \sigma_A^2)$.

In the marginal model, we ignore the differences among individuals and model the population-average, $P(Y_{ij} = 1)$ rather than $P(Y_{ij} = 1|b_i)$, by assuming that

$$P(Y_{ij} = 1) = logit^{-1}(\beta_0 + X_{ij}\beta_1). \tag{3.2}$$

Here, the remission rate in the sub-group taking the low dose is $\exp(\beta_0)/\{1+\exp(\beta_0)\}$.

This establishes the important point: the marginal and latent variable model parameters differ in the logistic model. The former describes the ratio of population odds; the latter describes the ratio of an individual's odds. Integrating the individual odds ratio (3.1) over the random effects will also not give the same result as the population odds ratio (3.2) from the marginal model. Note that the marginal parameter values will always be smaller in absolute value than their latent variable analogues. Neuhaus et al. [39] show that if $Var(b_i) > 0$, then the elements of the marginal (β) and random effects (β^*) regression vectors satisfy

1. $|\beta_k| \leq |\beta_k^*|, k = 1, \cdots, p$;
2. equality holds if and only if $\beta_k^* = 0$;
3. the discrepancy between β_k and β_k^* increases with $Var(b_i)$.

Marginal models and latent variable models fitted to longitudinal binary data, and models with non-linear link functions in general, yield parameters with different interpretations. It turns out that additional complications also arise depending upon the type of covariates to which the parameters correspond.

There are basically two types of covariates; those which are *time-independent* and those which are *time-dependent*. A time-independent covariate might be the *sex* of a subject involved in a longitudinal study. This covariate will not change within a subject, i.e. remains constant as we repeatedly measure the subject through time. On the other hand, a time-dependent covariate might be say the *depression score* of a subject. This covariate could change from visit to visit depending on how he/she feels at a particular visit. Hence with time-dependent covariates we see a change within observations made on the same subject.

Recently, Neuhaus and Kalbfleisch [38] went further to divide time-dependent covariates into those which were *designed* versus *non-designed*. A designed time-dependent covariate varies with identical distribution across the measurements on each individual. For example, consider a longitudinal study of infants which commences at birth, and monitors them on the first day of every month for a two year period. *Age* (in months) here is a designed time-dependent covariate varying in the identical manner for each infant involved in the study. On the other hand, a non-designed time-dependent covariate is more general. Now, suppose that for a similar longitudinal study of infants, *Age* is recorded at each

visit (scheduled or otherwise) to their family doctor. *Age* now represents a *non-designed* time-dependent covariate, assuming a different value for each observation on each infant, with the covariate pattern and individual mean of the covariate, varying between infants. Such a covariate then actually has both a time-dependent and time-independent component. For example, we can distinguish between (i) an overall effect of age on response, as measured by the association of the mean age with the response, and (ii) the effects of deviations from the average age on the series of responses on an individual. Often authors do not distinguish these effects in latent variable models and so implicitly assume that they are the same, with the result that we obtain estimates without any substantive interest. Consequently, Neuhaus and Kalbfleisch recommend analysts of longitudinal data examine whether the between- and within-individual components of covariate variability exhibit common effects on response [38].

If we imagine fitting a latent variable model with random intercepts then the parameters are measuring a contrast in covariates when the random effects are controlled for, that is, held fixed. For time-dependent covariates this is straightforward, since for a particular patient there will be empirical evidence of this contrast in the data. However, with time-independent covariates we never see a change in the covariate level so in a sense the parameter estimate obtained almost invites an unjustified causal statement. That is, there is no empirical verification of this statement available from the data. For example, consider a longitudinal study where the *sex* of the individuals is one covariate of interest. There is an unobserved random effect associated with each individual and by holding this random effect fixed we restrict our attention only on observations made on one particular individual. Clearly, we will observe no change in the sex covariate for these observations and yet the latent variable model will generate an individual-specific coefficient for sex. Such a coefficient consequently has no reasonable interpretation.

With marginal models, time-independent covariates generate parameters with straightforward interpretation. These parameters simply describe how the response average changes across various subsets of the study population where subsets are defined by covariate values. Unfortunately problems arise with time-dependent covariates. Marginal models don't use within subject comparisons so consequently may substantially underestimate the regression coefficients.

In summary, marginal models effectively handle time-independent covariates whereas

latent variable models are often best suited to time-dependent covariates. However, neither can deal effectively with both types of covariates. A new modeling strategy which can handle both time-dependent and time-independent covariates is sought. Since latent variable models have long been praised for their ability to provided individual-level estimates, it would be nice to have a model which could provide population-averaged and well as individual-level estimates. In the next section we present an innovative modeling strategy, recently proposed by Heagerty [20], that overcomes both these limitations.

3.3 The Marginally Specified Generalized Linear Mixed Model

Latent variable models are now often referred to as *generalized linear mixed models* (GLMMs), and as pointed out earlier, are individual-specific methods since inference is about individuals rather than the population average. As well, for covariates that do not vary within individuals, the interpretation of individual-specific coefficients can be difficult or misleading since they measure a contrast in covariates when the random effects are held equal.

Heagerty's *marginally specified generalized linear mixed model* (MS-GLMM)[20], originally designed for longitudinal binary data, presents an alternative parameterization for the latent variable model in which the marginal mean, rather than the conditional mean given random effects, is regressed on covariates.

Basically the MS-GLMM separates the mean model from the correlation model by first assuming a regression model for the marginal mean, and then by assuming that correlation among response variables is attributable to unobserved latent variables. Specifically it adopts two models: a model for the distribution of Y_i conditional on a vector of unobserved latent variables, or random effects, $b_i = vec(b_{ij})$; and a model for the population distribution of random effects. Covariates X_i can be accommodated by conditioning on them at each level of the hierarchical model: $[Y_i|b_i, X_i]$; and $[b_i|X_i]$. With the assumption that the observations $(Y_{i1}, Y_{i2}, \cdots, Y_{in_i})$ are conditionally independent given b_i, the likelihood is obtained as:

$$L_i(\theta) = pr_\theta(Y_i|X_i) = \int_{b_i} \{\prod_{j=1}^{n_i} pr_\theta(Y_{ij}|b_i, X_i)\} f_\theta(b_i|X_i) db_i \tag{3.3}$$

where θ is a parameter vector that characterizes the distribution of $[Y_i|X_i]$, or equivalently characterizes the distributions of both $[Y_i|b_i, X_i]$ and $[b_i|X_i]$.

The parameter vector θ is typically partitioned into a generalized linear model regression coefficient, β, that models averages of Y_{ij} as a function of covariates, and a vector of variance components, α, that characterize the distribution of random effects b_i. GLMMs adopt a generalized linear model for the conditional mean, $\mu_{ij}^b = E(Y_{ij}|b_i, X_i)$. Alternatively MS-GLMMs adopt a generalized linear model for the marginal mean $\mu_{ij} = E(Y_{ij}|X_i)$, while still modeling μ_{ij}^b.

A marginal regression model describes variation in the mean, $\mu_{ij} = E(Y_{ij}|X_i)$, as a function of measured covariates:

$$g(\mu_{ij}) = X_{ij}\beta.$$

To model μ_{ij}^b, we note that

$$E_b(\mu_{ij}^b) = \mu_{ij}.$$

This correspondence between the marginal and conditional means can also be expressed via the convolution equation (also referred to as the integral equation):

$$\mu_{ij} = \int_{b_{ij}} (\mu_{ij}^b) f_\alpha(b_{ij}|X_i) db_{ij}$$

where upon taking $h(x) = g^{-1}(x)$ as the inverse link function, $g(\mu_{ij}) = X_{ij}\beta$, and $g(\mu_{ij}^b) = \Delta_{ij} + b_{ij}$ we arrive at

$$h(X_{ij}\beta) = \int_{b_{ij}} h(\Delta_{ij} + b_{ij}) f_\alpha(b_{ij}|X_i) db_{ij} \tag{3.4}$$

which can be solved for Δ_{ij}.

In this approach, the mixed model is used primarily as a parsimonious mechanism for modeling the multivariate dependence structure. The assumption of an underlying mixed model is given by the conditional mean model $g(\mu_{ij}^b) = \Delta_{ij} + b_{ij}$, where the parameter Δ_{ij} is determined from (β, α) as the solution to the convolution equation.

A model which adopts the flexibility and interpretability of latent variable models for introducing dependence, but builds regression structure for the marginal mean, allowing

valid application with time-dependent and time-independent covariates, is now in place. However we should not be satisfied since models of this sort are often used in medical research where up to 10% of the observations may be contaminated [19]. We will see in Chapter 5 that such contamination can greatly affect the parameter estimates. There is a need for a robust model, one which will still achieve the aforementioned objectives but also afford us robustness.

In the next chapter we introduce the theory of robustness, discuss some robust efforts that have been proposed for longitudinal data, and then present our robust marginally specified generalized linear mixed model (ROBMS-GLMM).

Chapter 4

Extension to Robust Estimation

In a wide range of research areas a great deal of progress has been made in finding procedures which are robust. In fact, the robustification of many statistical procedures is one of the major developments in statistics this century. Our goal is to show the usefulness and necessity of robust techniques for modelling longitudinal data.

We begin with an overview of robust estimation and then introduce some robust procedures currently available for the analysis of longitudinal data. We then present our robust marginally specified generalized linear mixed model (ROBMS-GLMM), which is an effective method for dealing with longitudinal data in a robust fashion.

4.1 The Theory of Robustness

> A tacit hope in ignoring deviations from ideal models was that they would not matter; that statistical procedures which were optimal under the strict model would still be approximately optimal under the approximate model. Unfortunately, it turned out that this hope was often drastically wrong; even mild deviations often have much larger effects than were anticipated by most statisticians (J. Tukey, [45]).

Estimation is an art, as well as a science, whose goal is to infer information about an unknown quantity based on available data. Estimators are selected to perform well under assumed underlying conditions. However, since these conditions are never known exactly,

we should choose estimators which are robust, that is, perform well under a variety of underlying conditions. In so doing we obtain estimators which are reasonably accurate and efficient in the presence of outliers and other departures from the assumptions.

Outliers and other deviations from underlying model assumptions are unfortunately a fact of life. Common examples are recording and definition errors. Increased accessibility to high speed computing and data storage has facilitated the collection and analysis of larger data sets. Unfortunately little effort has been spent ensuring data quality. Therefore, often with larger data sets come more errors and consequently an even greater need for robust procedures.

4.1.1 Influence Function

A useful tool for investigating the robustness properties of a statistical procedure is the influence function. Roughly speaking, it describes the asymptotic bias caused on the estimator by a small amount of contamination in the underlying distribution. Therefore, a desirable local stability property is a bounded influence function.

The idea of influence functions was formulated by Hampel [18] and developed later by Hampel *et al.* [17]. Let F_n represent the empirical distribution of the data putting mass $1/n$ on each observation. As estimators of θ we consider statistics $T_n = T_n(y_1, \cdots, y_n)$ which can be represented, at least asymptotically, as functionals of the empirical distribution, i.e. $T_n(y_1, \cdots, y_n) = T(F_n)$. The robustness of the estimator T is assessed by means of the influence function (IF), which is defined at the model F_θ by

$$IF(y; T, F_\theta) = \lim_{\varepsilon \to 0} \left[\frac{T((1-\varepsilon)F_\theta + \varepsilon\Delta_y) - T(F_\theta)}{\varepsilon} \right]. \tag{4.1}$$

Δ_y is a probability measure which puts all its mass in y. Notice that $(1-\varepsilon)F_\theta + \varepsilon\Delta_y$ yields *good* observations from F_θ with high probability 1-ε and *bad* observations from point y with small probability ε. The IF describes the effect of this small contamination ($\varepsilon\Delta_y$) at the point y on the estimate (standardized by the mass of the contamination). In fact, the linear approximation $\varepsilon IF(y; T, F_\theta)$ measures the asymptotic bias of the estimator caused by the contamination.

One of the great strengths of influence functions rests with their ability to expose the

limitations of many commonly used estimators in the presence of contamination. Unboundedness of the IF permits small deviations from the model distribution to cause large changes in the estimator. Such a characteristic is most detrimental as it can lead to badly biased results.

Alternatively, estimators exhibiting the desirable robustness property of a bounded IF are more attractive. Such estimators are called *B-robust* (Bias-robust). By bounding the IF, we can ensure that small deviations from the model distribution do not cause large changes in the estimates.

4.1.2 Robust Estimators

Good robust estimators should be efficient as well as insensitive to small and large deviations in the data. As is often the case in real life, these goals cannot normally be reached simultaneously. Instead we must often compromise by sacrificing some degree of efficiency to increase robustness or vice versa.

A great many robust procedures involve weighting functions. Such functions assign weights, between 0 and 1, to each data point. A weight near 0 indicates that the data point is not well fitted by the model. Hence the weight function gives us precisely the information needed to determine whether our model is consistent with the data. Our options are to base inferences on the points which are well fitted by the model and downweight the observation(s) which are not, or try to develop alternative models in which all the weights are close to one. In identifying such potential outliers we may obtain very useful information regarding the data.

4.2 Robust Models for Longitudinal Data

We now examine some robust models which have been proposed for various sorts of longitudinal data. Given that both marginal and latent variable models build upon generalized linear models, we discuss these first.

4.2.1 Robust Estimation for Generalized Linear Models

As seen in Section 2.1.4, generalized linear models allow one to model the relationship between the predictors and a function of the mean of the response for both continuous and discrete response variables. The response variables Y_i, for $i, \cdots, n$ are supposed to come from a distribution belonging to the exponential family, such that $E[Y_i] = \mu_i$ and $V[Y_i] = V(\mu_i)$ for $i = 1, \cdots, n$ and

$$g(\mu_i) = X_i\beta,$$

$i = 1, \cdots, n$, where $\beta \subset \Re^p$ is the vector of parameters, $X_i \subset \Re$, and $g(.)$ is the link function.

The non-robustness of the maximum likelihood estimation for β has been studied extensively in the literature [31], [36]. The quasi-likelihood estimator for β shares the same non-robustness properties. As a solution, Cantoni and Ronchetti [8] recently proposed a natural robust generalization based on the idea of quasi-likelihood functions. This generalization leads to a class of estimators that are (1) easy to deal with, and (2) admit handy inference for the whole class of generalized linear models.

Their new estimators are a solution of a set of estimating equations which involve two sets of weighting functions. The first is introduced to control deviations in the y-space while the second downweights leverage points in the x-space.

These robust estimators are said [8] to be a valuable complement to classical techniques and are more reliable than their predecessors in the presence of outlying points and other deviations from the assumed model.

4.2.2 Resistant Generalized Estimating Equations

Preisser and Qaqish [41] considered the robustification of marginal models. As we learned in chapter 3, generalized estimating equations are often used to fit marginal models. Pressier and Qaqish demonstrated that parameter estimates from GEE may be highly influenced by a small subset of the data and consequently introduced resistant generalized estimating equations (REGEE). REGEE is an alternative estimation procedure to GEE, which automatically downweights influential observations or clusters. A *cluster* in their discussion would be analogous to an individual for the types of data we discuss.

When there are no unusual observations present, then all observations receive equal weight and REGEE reduces to GEE. Their approach is a multivariate generalization of

Carroll and Pederson [9], who provide robust estimates in the logistic regression model that are of the Mallows class. Estimates of the Mallows class are obtained by downweighting large leverage values. Alternatively, Schweppe estimates are obtained by downweighting according to residuals. They consider two approaches for the Mallows and Schweppe classes, which they call *observation* downweighting and *cluster* downweighting. The former downweights each observation separately, whereas the latter method assigns equal weight to all observations in a cluster based on some aggregate measure of the influence of the entire cluster.

Mallows cluster downweighting is achieved by assigning the weights equal to some function of the trace of the hat matrix. In the Schweppe class, clusters may be downweighted according to a summary measure of fit of the observations in the cluster based on residuals.

Mallows observation downweights are based on some measure of covariate distance or an observation's leverage. Carrol and Pederson [9] define non-iterative Mallows weights as a function of the distance of a covariate vector from a robust estimate of its central value. In contrast, leverage weights based on the hat matrix assign the smallest weights to observations with potentially large influence on the overall fit.

REGEE provides an alternative approach to data that is commonly analyzed with the GEE procedure. Statistical inference based on the GEE procedure is often sensitive to the influence of unusual data values. In contrast, the REGEE procedure automatically downweights individual observations or clusters with large influence. It is therefore a very useful procedure which allows us to fit marginal models robustly. It can also be used in parallel with classical procedures, especially to aid with data screening and preliminary analyses.

4.2.3 Robust Latent Variable Models

Latent variable models have been shown to be sensitive to deviations from the model distribution, to outlying observations, and to model misspecifications. A number of approaches to the robustness of latent variable models have been proposed.

In 1993 Huggins [27] proposed a robust inference procedure consisting of a robustified version of the log-likelihood for multivariate normal data. The method was proposed for the analysis of repeated measures in the presence of outliers. His concern was that likelihood

techniques in this setting are least squares estimators and may be expected to be sensitive to outliers and other departures from multivariate normality. For this reason he robustified the log-likelihood for multivariate normal data by using Tukey's bisquare function. The resulting estimators were shown to be consistent and asymptotically normal. His estimates were calculated using optimization routines, which required only the input of a function, such as -log-likelihood, to be minimized. All derivatives and matrices of second derivatives were then calculated numerically.

Richardson and Welsh [43] reviewed and extended robust estimation in the mixed linear model. Robustification was achieved by applying Huber-type ψ functions to the residuals in the estimating equations.

Krishnakumar and Ronchetti [29] extended the work of Richardson and Welsh to a general context of econometric models with a full covariance matrix of errors. They limited the influence of any single observation on the estimators of both the coefficients and the covariance parameters thus yielding estimators associated with the bulk of the data when contamination is present.

4.3 The Robust Marginally Specified Generalized Linear Mixed Model

We now introduce our robust model: the robust marginally-specified generalized linear mixed model (ROBMS-GLMM). The ROBMS-GLMM is designed to simultaneously provide estimates of the parameters and assess the fit of our data to the model. It includes weighting strategies to handle outlying observations both in the response and covariates. Huber's least favorable distribution is also introduced to make the ROBMS-GLMM less sensitive to violations on the distributional assumptions. By changing the tuning constants defining the ROBMS-GLMM we can fit models with varying degrees of robustness, close agreement of the estimates across a range of tuning constants usually suggests little, if any, contamination is present. The MS-GLMM in fact becomes a special case of the ROBMS-GLMM, obtained by setting tuning constants appropriately.

The subsections that follow detail the various robust features of our ROBMS-GLMM.

4.3.1 Random Effects Distributional Assumptions

We first consider how to protect against violations of the random effects (latent variable) distributional assumptions. Commonly the random effects $[b_i|X_i]$ are assumed to follow a unit spherical multivariate Gaussian distribution with b_i being a linear transformation of a possible lower dimensional random effect, $b_i = C_i z_i$, for an $n_i \times q$ matrix C_i, and $z_i \in R^q$ spherically normal, $z_i \sim \phi_q(z_i)$. To achieve our goal of robustness we use Huber's Least favorable distribution for scale (ϕ_q^H) in place of the Gaussian distribution where $\phi_q^H = \prod_{i=1}^q \phi^H(z_{ik})$ with

$$\phi^H(z) = \begin{cases} \frac{1-\varepsilon}{\sqrt{2\pi}} \exp[-\frac{z^2}{2}] & \text{if} |z| \leq k, \\ \frac{1-\varepsilon}{\sqrt{2\pi}} \exp[-\frac{k^2}{2}] \left[\frac{k}{|z|}\right]^{k^2} & \text{otherwise.} \end{cases} \tag{4.2}$$

Huber's Least favorable distribution for scale is normal in the middle, behaves like a t-distribution with k degrees of freedom in the tails, and consequently downweights the influence of extremes. It is not downweighting in the traditional sense (as described in Section 4.1.2) but rather does so by allowing longer tails which is a usual type of deviation. This makes outliers less *unlikely* under the model so that their effect on the parameter estimates is reduced. Choices for ε, k pairs are given in Huber [26]. A reasonably efficient choice is $\varepsilon = .005$ and $k = 2.46$. Note that Huber's least favorable density appears in both the likelihood and the convolution equation. Our mixed model is now comprised of the marginal regression model, $g(\mu_{ij}) = X_{ij}\beta$, the conditional assumption $g(\mu_{ij}^b) = \Delta_{ij} + b_{ij}$ and the distributional assumptions $\phi_q^H(z_i)$. For this model the parameters are $\theta = (\beta, \alpha)$ and the observed data likelihood for subject i can be written

$$L_i^R(\theta) = pr_\theta(Y_i|X_i) = \int \{\prod_{j=1}^{n_i} pr_\theta(Y_{ij}|b_i, X_i)\}\phi_q^H(z_i)dz_i \tag{4.3}$$

Heagerty and Kurland [21] helped establish the need for our ROBMS-GLMM by examining the performance of the MS-GLMM for scenarios in which the marginal mean regression models were correctly specified; but where the distributional assumptions for the random effects were wrongly specified. They found that maximum likelihood estimates of β may contain modest bias if the dependence (random effects) assumptions are grossly

violated. Final recommendations suggested that careful attention should be given to the random effects model assumptions when using generalized linear mixed models for regression inference with longitudinal data. Our ROBMS-GLMM automates this task making $\hat{\beta}$ less sensitive to violations of the random effects distributional assumptions.

If we set $k = \infty$ and $\varepsilon = 0$, ϕ_q^H becomes the unit spherical multivariate Gaussian density and likelihood (4.3) corresponds exactly to Heagerty's MS-GLMM thereby yielding the classical maximum likelihood estimates.

One of the most popular MS-GLMMs for longitudinal binary data is the marginally specified logistic-normal model with random intercepts. It assumes $b_{ij} = b_{i0} \sim N(0, \sigma^2)$ allowing us to rewrite $b_{ij} = \sigma z$ where $z \sim \phi$. With Y_{ij}'s that take on values 0 or 1,

$$g(\mu_{ij}) = logit\ P(Y_{ij} = 1 | X_{ij}) = X_{ij}\beta$$

and

$$g(\mu_{ij}^b) = logit\ P(Y_{ij} = 1 | b_i, X_{ij}) = \Delta_{ij} + \sigma z.$$

In this case the ROBMS-GLMM yields a robustified likelihood for individual i of the form:

$$\begin{aligned} L_i^R(\beta, \alpha) &= \int \prod_{j=1}^{n_i} h(\Delta_{ij} + \sigma z)^{y_{ij}} \{1 - h(\Delta_{ij} + \sigma z)\}^{1 - y_{ij}} \phi^H(z) dz \\ &= \int \exp\left[\sum_{j=1}^{n_i} \{ y_{ij}(\Delta_{ij} + \sigma z) + \log[1 - h(\Delta_{ij} + \sigma z)] \} \right] \phi^H(z) dz \end{aligned} \tag{4.4}$$

where $h = g^{-1} = logit^{-1}$. The corresponding convolution equation (3.4) is

$$h(X_{ij}\beta) = \int h(\Delta_{ij} + \sigma z) \phi^H(z) dz. \tag{4.5}$$

We are particularly interested in modelling longitudinal binary data and consequently we now restrict our attention to model (4.4), referring to it as the *robust marginally specified logistic-normal model*, while still recognizing that our conclusions can be generalized back to model (4.3).

4.3.2 Outlying Observations in the Covariates

Having now robustified against violations of the random effects distributional assumptions there are still ways to further improve the model. Specifically, we are concerned with the estimation of β and how it may be impacted by deviations from the model assumptions.

In section 4.1.1 we learned that estimators are deemed *non-robust* if they possess IF which are unbounded. M-estimators have IF proportional to their score functions. Consequently, proving that an M-estimator's score function is unbounded allows one to proceed directly to the conclusion that the estimator is *non-robust*.

Heagerty's MS-GLMM estimators are in fact maximum likelihood estimators which are a special form of M-estimator. We therefore examine score functions to evaluate their robustness, or lack thereof. Looking specifically at the marginally-specified logistic-normal model, the score function for β_k takes the form:

$$\frac{1}{L_i}\int \exp\left[\sum_{j=1}^{n_i}\{y_{ij}(\Delta_{ij}+\sigma z)+\log[1-h(\Delta_{ij}+\sigma z)]\}\right]\left[\sum_{j=1}^{n_i}\{y_{ij}-h(\Delta_{ij}+\sigma z)\}\frac{\partial\Delta_{ij}}{\partial\beta_k}\right]\phi(z)dz.$$

The above score function, and consequently also the IF, are unbounded. This is most easily seen by recognizing that the partial derivative $\frac{\partial\Delta_{ij}}{\partial\beta_k}$ can be written:

$$\frac{\partial\Delta_{ij}}{\partial\beta_k}=\frac{\partial\Delta_{ij}}{\partial\eta_{ij}}.\frac{\partial\eta_{ij}}{\partial\beta_k}.$$

In the above expression $\eta_{ij}=X_{ij}\beta$, $\frac{\partial\Delta_{ij}}{\partial\eta_{ij}}=\mu_{ij}(1-\mu_{ij})/A_{ij}$ where $A_{ij}=\int h(\Delta_{ij}+\sigma z)\{1-h(\Delta_{ij}+\sigma z)\}\phi(z)dz$, see Heagerty [20] for further details. Now given that $\eta_{ij}=X_{ij}\beta$, we have $\frac{\partial\eta_{ij}}{\partial\beta_k}=X_{ijk}$. Therefore $\frac{\partial\Delta_{ij}}{\partial\beta_k}$ will include a term involving only the covariate X_{ijk}. Hence, by making this covariate arbitrarily large we can unbound the score function. We conclude that these estimators of β are non-robust and hence a small amount of contamination can severely distort the parameter estimates (as will be demonstrated numerically later). Our approach to obtaining more robust estimators is to further robustify the likelihood equations such that they are not so sensitive to outlying observations in the covariates.

Outlying observations in the covariates are often referred to as observations with high leverage. To reduce the impact of high leverage we introduce weights w_{ij} into the likelihood

producing:

$$L_i^R(\beta,\alpha) = \int \prod_{j=1}^{n_i} [h(\Delta_{ij}+\sigma z)^{y_{ij}}\{1-h(\Delta_{ij}+\sigma z)\}^{1-y_{ij}}]^{w_{ij}} \phi^H(z)dz.$$

A simple and effective choice for w_{ij} is to base it on some function of the hat matrix while noting that more sophisticated choices are also available. Specifically, we take $w_{ij} = \sqrt{1-h_{jj}}\sqrt{1-H_{ii}}$ where h_{jj} are the diagonal elements of the hat matrix $h_i = X_i(X_i^T X_i)^{-1}X_i^T$ and H_{ii} are the diagonal elements of the hat matrix $H = X(X^T X)^{-1}X^T$. h_i contains only the time-dependent covariates while H contains the time-independent covariates. When no time-independent covariates are present we require only the first term in the definition of w_{ij}.

An observation with a small weight will contribute less information to the likelihood equation than one with a large weight. We design our robust approach such that those observations (or individuals) who are *outlying* obtain small weights. In the maximum likelihood approach this is not the case since all weights are one and hence an observation with outlying covariates could significantly influence the maximum likelihood estimator.

The Convolution Equation

The convolution equation (4.5) is solved for Δ_{ij} using a quadrature program to approximate the integral and a root finder algorithm to obtain Δ_{ij}. The quadrature program was recommended by Dr. Pat Keast, a numerical analyst in the Department of Mathematics and Statistics, Dalhousie University. This program uses formulae originally due to Patterson but later modified by Krogh and Snyder [30]. Patterson [40] derived a family of quadrature formulae in which additional abscissae are added so as to maximize precision. Krogh and Snyder reduced the number of coefficients necessary to represent Patterson's quadrature formulae as well as the amount of storage necessary for function values. We have chosen a root finder algorithm given by Forsythe *et al.* [15]. This algorithm makes use of the bisection procedure combined with linear or quadratic inverse interpolation. At every step the algorithm selects one of two new approximations, the first being obtained by the bisection procedure and the second resulting from the interpolation.

Outlying observations in the covariates can render the convolution equation (4.5) unsolvable (as well as the convolution equation (3.4) for the MS-GLMM). This happens

because such extreme observations force the left hand side of the convolution equation, $h(X_{ij}\beta)$, towards either 0 or 1. Our experience suggests that when $h(X_{ij}\beta)$ is within approximately 10^{-5} of either 0 or 1, then no solution for Δ_{ij} in

$$h(\eta_{ij}) = \int h(\Delta_{ij} + \sigma z)\phi^H(z)dz$$

can be found. Our robust procedure automatically checks the value of $h(\eta_{ij})$ so as to identify instances where there will be no solution for Δ_{ij}, and then sets the corresponding leverage weight w_{ij} equal to 0. Unfortunately, the MS-GLMM does not protect against the convolution equation becoming unsolvable and consequently cannot be fit when such outlying observations are present. One could put similar flags in with the MS-GLMM, and then ignore the observations which do not yield a solution for Δ_{ij}. However this was not done in the Heagerty implementation.

4.3.3 Outlying Observations in the Response

When considering longitudinal binary data there is also another type of outlying observation which may arise: outlying observations in the response. In the case of the robust marginally specified logistic-normal model, an outlying observation in the response will be one for which the differences $|y_{ij} - P(Y_{ij} = 1|X_{ij}, b_i)|$ and $|y_{ij} - P(Y_{ij} = 1|X_{ij})|$ are very close to one. Given that the random effects are unobserved we can only examine the latter difference. Fortunately, in cases where the first difference is large, Δ_{ij} is necessarily large, and hence the second difference must also be large. This is a direct result of that fact that $P(Y_{ij} = 1|X_{ij})$ and $P(Y_{ij} = 1|X_{ij}, b_i)$ are linked by the convolution equation. For such outlying observations the data and the model are in fairly strong disagreement suggesting that the contribution to the likelihood should possibly be *re-weighted.* Specifically we propose to *downweight* the likelihood contribution $P(Y_{ij} = 1|X_{ij}, b_i)^{y_{ij}}$ and *upweight* that of $P(Y_{ij} = 0|X_{ij}, b_i)^{1-y_{ij}}$, or vice versa. Re-weighting the likelihood in this way results in a robustified likelihood of the form:

$$L_i^R(\beta, \alpha) = \int \prod_{j=1}^{n_i} [h(\Delta_{ij} + \sigma z)^{y_{ij}-\lambda_{ij}} \{1 - h(\Delta_{ij} + \sigma z)\}^{1-y_{ij}+\lambda_{ij}}]^{w_{ij}} \phi^H(z)dz \qquad (4.6)$$

where the λ_{ij} determine the degree of re-weighting. Care must be taken to ensure that we do not re-weight too drastically such that we begin fitting the outliers instead of the bulk of the data. We define a function λ which takes as an argument the Pearson residual r_{ij} for the particular observation and produces a value λ_{ij}. That is, $\lambda_{ij} = \lambda(r_{ij})$ where

$$\lambda(r_{ij}) = \begin{cases} 0 & \text{if} |r_{ij}| \leq c, \\ sign(r_{ij})(1 - \frac{c}{|r_{ij}|}) & \text{otherwise} \end{cases}$$

and c is an appropriately chosen constant. It is common with robust procedures to choose values for tuning constants like this one such that one achieves 95% efficiency when there is no contamination present. Let us examine how this function behaves with $c = 1.7$. Taking $y_{ij} = 1$ and $P(y_{ij} = 1|b_i, X_{ij}) = .2$, then $r_{ij} = 2.0$ making $\lambda_{ij} = .15$. This decreases the weight of the first contribution to the likelihood (by $\lambda_{ij} = .15$) and increases the weight of the second as is sensible here. On the contrary when $y_{ij} = 0$ and $P(y_{ij} = 0) = .8$, then $\lambda_{ij} = -.15$ and the reverse weighting occurs. An appropriate choice for c will depend on various properties of the data set, including both size and the amount of within-individual correlation. c will also affect the efficiency of the estimates of the model.

Since we are dealing with repeated observations on individuals there are two weighting schemes available. *Observation* weighting assigns a weight to each observation (Y_{ij}) separately, whereas individual weighting assigns the same weight to all observations on an individual (Y_i) based on some aggregate measure of their influence. These weighting schemes were first considered by Preisser and Qaqish [41] in the context of marginal models, as previously discussed in subsection 4.2.2 .

For individual weighting we set $\lambda_{ij} = \lambda_i$ where λ_i is some summary measure of the deviations for the entire individual. For example, we can define $\lambda_i = \sum_{i=j}^{n} \lambda_{ij}$ and use λ_i in place of λ_{ij} in Equation (4.6). Notice that this is a non-robust mean which will consequently allow one outlier to significantly impact the λ_i. If there is indeed correlation between the individual observations then one outlying observation can influence all those that follow it. This choice of λ_i will behave in a similar manner and re-weight all observations on the particular individual. Individual weighting is most applicable in situations where we believe the correlation within an individual is fairly large. This is more frequently the case when we have only a few observations per individual.

Alternatively we can restrict our attention to the observation level which amounts to keeping λ_{ij} as defined earlier. Either approach to re-weighting could result in a loss of efficiency for the resulting estimators as is generally the case with robust statistics. In section 6.1.1 we discuss this issue in further detail.

Up to now we have focused on re-weighting the likelihood contributions. However, instead of re-weighting we can weight the entire contribution to the likelihood for a particular observation. Again, these weights are defined based on the Pearson residuals. For an appropriately chosen constant c we have the weight $v_{ij} = v(r_{ij})$ where

$$v(r_{ij}) = \begin{cases} 1 & \text{if} |r_{ij}| \leq c, \\ \frac{c}{|r_{ij}|} & \text{otherwise.} \end{cases}$$

Notice that the first type of weighting we propose actually shifts mass through $y_{ij} \pm \lambda_{ij}$ by downweighting one term in the likelihood contribution and upweighting the other. The second weighting approach is multiplicative in that we downweight the entire contribution to the log-likelihood from 1 to v_{ij}. We choose to use only one or the other of the weighting approaches, their performances are compared in section 5.1. In terms of implementation, the second approach is more straightforward since it simply involves multiplying each term in the log-likelihood by v_{ij}. The first approach is more complicated in that it requires that the λ_{ij} be inserted into each likelihood term twice (and with different sign).

We have now arrived at the final ROBMS-GLMM, defined by the robustified likelihood

$$L_i^R(\beta, \alpha) = \int \prod_{j=1}^{n_i} [h(\Delta_{ij} + \sigma z)^{y_{ij} - \lambda_{ij}} \{1 - h(\Delta_{ij} + \sigma z)\}^{1 - y_{ij} + \lambda_{ij}}]^{w_{ij} v_{ij}} \phi^H(z) dz \qquad (4.7)$$

along with the convolution equation:

$$h(\eta_{ij}) = \int h(\Delta_{ij} + \sigma z) \phi^H(z) dz.$$

4.3.4 The Infiuence Function

In section 4.1.1 we saw that it is very desirable to have estimators with bounded IF. Hence we now investigate the ROBMS-GLMM estimators to see if they in fact exhibit this feature.

The IF for the ROBMS-GLMM is proportional to its score function which takes the

form

$$\frac{1}{L_i^R}\int u \exp\left[\sum_{j=1}^{n_i} w_{ij}v_{ij}\{(y_{ij}-\lambda_{ij})(\Delta_{ij}+\sigma z)+\log[1-h(\Delta_{ij}+\sigma z)]\}\right]\phi^H(z)dz \qquad (4.8)$$

where for $\frac{\partial l_i^R}{\partial \sigma}$

$$u=\sum_{j=1}^{n_i} w_{ij}v_{ij}\{y_{ij}-\lambda_{ij}-h(\Delta_{ij}+\sigma z)\}\frac{\partial \Delta_{ij}}{\partial \beta_k}, \qquad (4.9)$$

and similarly for $\frac{\partial l_i^R}{\partial \beta}$

$$\begin{aligned} u=\sum_{j=1}^{n_i}\Big[& w_{ij}v_{ij}\{[y_{ij}-\lambda_{ij}-h(\Delta_{ij}+\sigma z)]\frac{\partial \Delta_{ij}}{\partial \beta_k}-(\Delta_{ij}+\sigma z)\frac{\partial \lambda_{ij}}{\partial \beta_k}\}+ \\ & w_{ij}\frac{\partial v_{ij}}{\partial \beta_k}\{(y_{ij}-\lambda_{ij})(\Delta_{ij}+\sigma z)+\log(1-h(\Delta_{ij}+\sigma z)\}\Big]. \end{aligned} \qquad (4.10)$$

In order to unbound the score function (4.8) the function of z must be non-integrable. For this to happen our integral over z must involve a term which is either undefined (i.e. division by zero) or grows at too rapid a rate. We need not be concerned with the y_{ij} since they are always 0 or 1. However we must examine what happens to the score function when the covariate X_{ijk} becomes extreme.

The covariate X_{ijk} enters (4.8)-(4.10) through Δ_{ij} and well as through the residual r_{ij}. We begin by considering Δ_{ij} which is a function of X_{ijk}, obtained by solving the convolution equation. We must determine whether Δ_{ij}, or any function of Δ_{ij}, is unbounded as this would lead to the score function (4.8) being unbounded.

In subsection 4.3.2, we saw that making X_{ijk} arbitrarily large renders the convolution equation unsolvable and hence we robustify against such extreme X_{ijk} by setting the corresponding leverage weight w_{ij} equal to 0. With problematic observations eliminated by setting $w_{ij}=0$, Δ_{ij} is never unbounded.

Next we consider the partial derivative $\frac{\partial \Delta_{ij}}{\partial \beta_k}$. We saw in section 4.3.2 that $\frac{\partial \Delta_{ij}}{\partial \beta_k}$ can be written:

$$\frac{\partial \Delta_{ij}}{\partial \beta_k}=\frac{\partial \Delta_{ij}}{\partial \eta_{ij}}\cdot\frac{\partial \eta_{ij}}{\partial \beta_k},$$

or equivalently as,

$$\frac{\partial \Delta_{ij}}{\partial \beta_k} = \frac{\mu_{ij}(1-\mu_{ij})X_{ijk}}{\int \frac{e^{\Delta_{ij}+\sigma z}}{(1+e^{\Delta_{ij}+\sigma z})^2}\phi^H(z)dz}.$$

Clearly then, by making X_{ijk} arbitrarily large we can unbound $\frac{\partial \Delta_{ij}}{\partial \beta_k}$. However examination of the expressions (4.9-4.10) reveals that $\frac{\partial \Delta_{ij}}{\partial \beta_k}$ only appears in conjunction with w_{ij}. Consequently, if we take X_{ijk} very large it will render the convolution equation unsolvable in which case w_{ij} becomes 0 and hence desirably, any terms including $w_{ij}\frac{\partial \Delta_{ij}}{\partial \beta_k}$ are bounded.

We now consider terms involving the residual r_{ij}, specifically, $\frac{\partial \lambda_{ij}}{\partial \beta_k}$ and $\frac{\partial v_{ij}}{\partial \beta_k}$. Both of these terms are well-defined but unbounded with X_{ijk} extreme. Fortunately, they too always occur with w_{ij} and hence are again protected by the fact that when X_{ijk} is made arbitrarily large, the convolution equation will be rendered unsolvable such that the corresponding w_{ij} is set equal to 0 by our robust procedure.

We are now able to state that the score function (4.8) is bounded and consequently the ROBMS-GLMM are robust.

4.3.5 Robust Starting Values

Our ROBMS-GLMM estimates are calculated using iterative optimization routines, which require only the input of the negative of the log of the robustified likelihood. All derivatives and matrices of second derivatives are then calculated numerically. The optimization routines require reasonable starting values for good performance. By naively assuming that observations on individuals are independent, we can obtain an initial fit to the data using a generalized linear model. This was the approach taken by Heagerty. In section 4.2.1 we noted that Cantoni and Ronchetti have recently developed a procedure for fitting such models robustly. Therefore we use their estimates as starting values, computation of which involves a Fisher scoring procedure. S-Plus routines are available from Cantoni. Use of these robust starting values results in faster convergence to the solution which is desirable.

In the next chapter we examine the performance of the ROBMS-GLMM via a simulation study.

Chapter 5

The Performance of the ROBMS-GLMM

We now investigate how ROBMS-GLMMs perform in practice. This chapter includes a demonstration for illustrative purposes as well as a simulation study designed to investigate the sampling properties of the estimates. This chapter concludes with a discussion of some important computational issues concerning the fitting of these models.

5.1 Demonstration

We begin by considering a simulated data set which allows us to compare MS-GLMM and ROBMS-GLMM estimates to *true* parameter values. At the same time we judge how well we are able to identify contamination when it exists. We look specifically at a logistic-normal formulation, that is,

$$logit(\mu_{ij}) = \beta_0 + X_{ij1}\beta_1 + X_{ij2}\beta_2,$$

and

$$logit(\mu_{ij}^b) = \Delta_{ij} + b_i$$

with $\beta^T = (-2, 3, 1)$ and $[b_i|X_i]$ following a standard normal distribution such that $\alpha = ln(\sigma) = 0$. The covariate X_1 was generated from a uniform distribution on [0,1] and X_2 from a Bernoulli distribution with probability of success $p = 1/2$. Hence we have two non-designed time-dependent covariates, one continuous and one discrete. The response is binary, coded by 0 or 1. *Leverage* contamination was introduced by changing 5% of the X_1 covariates. Their decimal was moved one place to the left. This contamination mimics data recording errors commonly occurring in practice. Prediction contamination was introduced by changing 5% of the response (from 0 to 1, 1 to 0 respectively). This contamination is again reflective of what occurs in practice, when a 0 is recorded instead of a 1, and vice versa.

The simulation involved 40 individuals with 20 observations per individual. Initially we fitted both the ROBMS-GLMM and MS-GLMM to the non-contaminated data. The results, which are shown in Table 5.1, are in fairly close agreement, with the ROBMS-GLMM estimates giving up a small amount of efficiency for robustness. Standard errors are computed from the inverse of the hessian matrix, which implies that those for the ROBMS-GLMM are computed under the assumption that the robustified likelihood is the *true* likelihood. This is discussed in further detail in Chapter 6. Note that the ROBMS-GLMM was fitted twice in order to utilize both prediction weighting strategies as discussed in section 4.3.3. The first *re-weighted* likelihood contributions where appropriate via the λ_{ij} (recall that a small value of $|\lambda_{ij}|$ results in fairly little re-weighting), and the second *downweighted* likelihood terms where appropriate using the v_{ij} (recall that a small value of v_{ij} implies a large amount of downweighting). Results were similar for both. All of the prediction re-weights (λ_{ij}) were smaller in absolute value than 0.4, and both the prediction downweights (v_{ij}) and leverage downweights (w_{ij}) all larger than 0.6. Hence fairly little weighting was performed suggesting that most of the data was well fit by the model as would be expected with no contamination.

Table 5.1: Parameter estimates (standard errors) with non-contaminated data, $*k = 2.46$, $c = 1.70$.

		Model		
Variable	**True Value**	**MS-GLMM**	**ROBMS-GLMM***	
			Re-Weight	*Downweight*
β_0	-2.00	-1.63(.22)	-1.83(.26)	-1.74(.25)
β_1	3.00	2.89(.28)	3.24(.33)	3.10(.35)
β_2	1.00	0.80(.14)	0.91(.16)	0.86(.13)
α	0.00	0.04(.16)	-0.07(.17)	-0.03(.21)

Table 5.2 presents the estimates obtained by fitting both the ROBMS-GLMM and MS-GLMM in the presence of leverage contamination. Such contamination renders the convolution equation unsolvable so that one cannot fit the MS-GLMM. The ROBMS-GLMM yields sensible parameter estimates while at the same time correctly downweighting over 95% of the contamination. The contamination and associated weights are shown in Table 5.3. Weights above 0.9 have fairly little impact so this is the contamination that we consider *missed* by our model.

Table 5.2: Parameter estimates (standard errors) with 5% leverage contamination, $*k = 2.46, c = 1.70$.

		Model		
Variable	**True Value**	**MS-GLMM**	**ROBMS-GLMM***	
			Re-Weight	*Downweight*
β_0	-2.00	—	-1.80(.25)	-1.72(.25)
β_1	3.00	—	3.20(.34)	3.08(.32)
β_2	1.00	—	0.82(.14)	0.78(.15)
α	0.00	—	-.10(.18)	-.06(.18)

Table 5.3: 5% Leverage contamination and associated weights.

Contaminant		Weight
Ind.	*Obs.*	w_{ij}
0	4	0.0
	18	0.0
1	11	0.0
2	10	.79
4	16	.91
6	3	.54
9	13	.86
10	0	.95
11	2	.88
	18	0.0
13	19	0.0
14	9	0.0
	17	.94
16	19	0.0
17	1	0.0
18	9	0.0
	19	.94
19	17	0.0
20	8	0.0
	10	.88
21	14	0.0
22	0	0.0
23	0	0.0
25	13	.94
	19	0.0
26	1	0.0
27	2	0.0
29	5	0.0
	10	0.0
30	12	0.0
31	1	0.0
	9	0.0
	13	0.0
32	6	0.0
34	7	0.0
	9	0.0
	12	.95
35	0	0.0
38	17	0.0

Table 5.4 presents the results obtained in the presence of prediction contamination. Again the ROBMS-GLMM outperforms the MS-GLMM by producing parameters estimates closer to the true values. Prediction contamination is much more difficult to deal with than leverage contamination. Multiple contamination on one individual increases the probability of masking. This makes it unreasonable to try to identify all of the contamination precisely. Rather, we are interested in the overall effect on the parameter estimates and the amount of weighting occurring. One would hope that the amount of weighting that occurs is in line with the amount of contamination. This is indeed the case as approximately

5% of the observations are being re-weighted (or downweighted).

Although α is a nuisance parameter, care must be taken when interpreting its estimates. In replacing the Gaussian distribution with Huber's Least Favorable distribution, α acquires a different interpretation. Since the variance of Huber's Least Favorable distribution is larger, we expect α to be smaller to ensure that the function is still a density. Results are as expected.

Table 5.4: Parameter estimates (standard errors) with 5% prediction contamination, $*k = 2.46, c = 1.70$.

		Model		
Variable	**True Value**	**MS-GLMM**	**ROBMS-GLMM***	
			Re-Weight	*Downweight*
β_0	-2.00	-1.46(.21)	-1.58(.21)	-1.53(.19)
β_1	3.00	2.58(.26)	2.81(.29)	2.72(.26)
β_2	1.00	0.72(.15)	0.80(.16)	0.77(.15)
α	-0.00	-0.12(.17)	-0.22(.18)	-0.20(.18)

5.2 Simulation Study

To further evaluate the performance of our robust procedure, we carried out a simulation study that compared ROBMS-GLMMs to MS-GLMMs. For one simulation run, we proceeded as follows. A design matrix was generated and values for the parameters were given. Random effects were generated according to a random effects distribution, and vectors y_{ij} were then generated as conditionally independent Bernoulli random variables with means $h(\Delta_{ij} + \sigma z_i)$. Note that Δ_{ij} must be recovered as the solution of the convolution equation.

We consider models that contain both a time-dependent (*within-individual*) and time-independent (*individual-level*) covariate. The multinomial time-independent covariate, $X_{ij1} = x_{i1}$ takes on integer values between 0 and 5 (rescaled) with equal probability. The continuous uniform non-designed time-dependent covariate X_{ij2} varies between 0 and 1. We generate data involving 60 individuals with 10 observations on each.The model was assumed to be

$$logit(\mu_{ij}) = \beta_0 + X_{ij1}\beta_1 + X_{ij2}\beta_2$$

with $\beta^T = (-2, 3, 1)$ and $[b_i|X_i]$ following a standard normal distribution such that $\alpha = ln(\sigma) = 0$.

5.2.1 Background

Neuhaus, Hauck, and Kalbfleisch [37] have studied generalized linear mixed models when the mean parameters are conditionally specified (CS-GLMM). Note that such models are often referred to simply as GLMMs or latent variable models. They investigated the impact of assuming the random effects are distributed as F when in truth they are G. They looked specifically at logistic-normal models and found that the maximum likelihood estimate (MLE) of a mean parameter has bias of less than 20% when random effects are non-normally distributed. These results suggest that likelihood based mean estimates using a CS-GLMM may be moderately insensitive to distributional assumptions.

Heagerty [20] performed simulation studies to investigate the potential bias due to maximum likelihood estimation using MS-GLMM. He looked specifically at logistic-normal models with random intercepts, when σ is a function of subject-level covariates or when

random effects are autocorrelated. Bias may result, but it was found to be slight (relative bias $\leq 15\%$) in the scenarios considered.

Heagerty and Kurland [21] explored the bias that arises due to incorrectly assuming that the variance of the random effects is independent of the covariates. In particular, they considered a model where the random effects variance differs according to a binary covariate. They assessed the impact of this model misspecification on both marginally specified and conditionally specified mean parameters. They compared MS-GLMMs and CS-GLMMs and found that the MLEs for MS-GLMMs are much less sensitive to variance component specification than for CS-GLMMs. Their conclusion was that MLEs for MS-GLMMs may be biased due to variance component misspecification but the magnitude of the bias is generally small.

The aforementioned work influenced our decision to focus on robustness of the random effects distribution to outliers, rather than to misspecified distributions, given that such misspecification appears to have little impact.

Both the MS-GLMM and ROBMS-GLMM allow for a multi-dimensional random effects distribution. This makes a wide variety of models available, including those which involve serial correlation of the random effects. However with the increased complexity of the random effects distribution comes heavy computational requirements. Investigations are therefore suggested as a direction of future research.

5.2.2 Design

In the study we consider three factors: random effects distribution, design matrix, and prediction. We choose two random effects distributions labeled as follows: *no mix* is standard normal; and *mix* is 5% wild (i.e., 95% from a standard normal distribution and 5% from a normal with $\mu = 0$ and $\sigma = 9$).

We used two design matrices. The first as discussed in Section 5.2 and labeled *no lev*; the second, labeled *lev*, is similar except that it contains 5% high leverage points. This leverage was introduced by changing covariates for a randomly chosen 5% of the individuals. These covariates had their decimal place shifted one place to the left so as to mimic data recording errors.

Prediction contamination was introduced by changing 5% of the responses y_{ij} (from 0 to 1, 1 to 0 respectively). We label the non-contaminated design *no pred* whereas that with

prediction contamination as *pred*.

Table 5.5 gives the results. For each situation, we carried out 50 simulation runs. The entries include the parameter estimates and their associated standard errors for both the ROBMS-GLMM and the MS-GLMM. Those marked with an $\times$ could not be fitted due to the contamination. The ROBMS-GLMM is utilizing the prediction downweighting strategy as this was found to be slightly more stable. We would expect results to be similar when using re-weighting as suggested by the demonstration.

The standard errors in Table 5.5 are based on performing 50 simulations (i.e. they are the standard errors of the 50 corresponding $\hat{\beta}_k$). The standard errors of the means obtained by simulation (Monte Carlo) are therefore those reported divided by $\sqrt{50}$. If we were to compare the $\hat{\beta}_1$ estimates for the case where there is only prediction contamination we would find that the two means -2.80 and 3.03 are only just significantly different (using a two-sided t_α-test with $\alpha = .05$). One can see that even when there is no contamination present the standard errors of the ROBMS-GLMM estimates are slightly smaller than those for the MS-GLMM estimates. We believe this is due to the increased stability of the ROBMS-GLMM algorithm.

Table 5.5: Parameter estimates (standard errors) based on 50 simulations.

	no mix			
	no pred		**pred**	
	MS	*ROBMS*	*MS*	*ROBMS*
	no lev			
β_0	-2.06(.36)	-2.12(.27)	-1.90(.31)	-2.04(.24)
β_1	3.02(.49)	3.13(.37)	2.80(.42)	3.03(.69)
β_2	1.06(.28)	1.05(.21)	0.96(.21)	0.96(.24)
α	-.07(.20)	-.06(.14)	-.03(.17)	-.14(.36)
	lev			
β_0	×	-1.94(.29)	×	-2.04(.31)
β_1	×	2.94(.36)	×	3.08(.46)
β_2	×	.98(.26)	×	1.02(.27)
α	×	-.00(.15)	×	-.04(.22)
	mix			
	no pred		**pred**	
	MS	*ROBMS*	*MS*	*ROBMS*
	no lev			
β_0	-1.92(.48)	-1.95(.37)	-1.91(.28)	-1.94(.28)
β_1	2.89(.68)	2.90(.50)	2.90(.40)	2.92(.70)
β_2	0.93(.25)	1.02(.29)	.85(.30)	.93(.31)
α	0.12(.14)	.08(.15)	.14(.17)	.11(.17)
	lev			
β_0	×	-1.99(.25)	×	-1.92(.25)
β_1	×	3.04(.41)	×	2.89(.40)
β_2	×	.98(.25)	×	.95(.29)
α	×	.11(.18)	×	.18(.14)

Note: × indicates model could not be fitted.

5.2.3 Analysis of Results

While examining the results in Table 5.5, it is useful to consider the following insightful comments of David Cox [10] in discussing the role of statistical models:

> "It is important to distinguish the parts of the model that define aspects of subject matter interest, the primary aspects, and the secondary aspects that indicate efficient methods of estimation and assessment of precisions."
>
> "Especially in empirical models, it is desirable that parameters (e.g., contrasts, regression coefficients and the like) have an interpretation largely independent of secondary features of the models used."

The primary objective of both MS-GLMMs and ROBMS-GLMMs is to make inference regarding the mean response as a function of the covariates. We are therefore most interested in the estimates of β_k, $k = 1, \cdots, p$ rather than α, which here is a secondary feature of the model. Consequently the comments that follow are based largely on the estimation of β.

We discuss first the results with no contamination (no mix, no flip, no lev) and we see that, as expected, both models perform similarly, with the MS-GLMM estimates being slightly closer to the true parameter values. However, as soon as we go to the contaminated normal situation (mix), the result is different, with the ROBMS-GLMM outperforming the MS-GLMM by producing estimates which are somewhat closer to the true values. This result demonstrates that the ROBMS-GLMM has good robustness characteristics with a contaminated normal, whereas the MS-GLMM does not perform as well.

We next consider the results under leverage contamination. The ROBMS-GLMM gives reasonable results unlike the MS-GLMM which cannot be fitted to the data as it has no mechanism in place for dealing with cases where the convolution equation is rendered unsolvable by the contamination. Our robust procedure identifies such observations and downweights their contribution to the likelihood.

In the presence of prediction contamination the ROBMS-GLMM once again yields estimates slightly closer to the true values than the MS-GLMM. The results are not as startling as those with leverage contamination but still support the use of our ROBMS-GLMM.

Finally, the results also illustrate that the ROBMS-GLMM performs well when faced with a variety of types of contamination. We see it perform better than the MS-GLMM

when faced with both prediction contamination and a contaminated normal distribution for the random effects. As well, when faced with all three types of contamination (mix, flip, lev) the ROBMS-GLMM still performs well unlike the MS-GLMM which once again cannot be fitted to the data.

It is important to remember that the ROBMS-GLMM provides additional information that is not shown in Table 5.5. This insightful information is provided by the weights and tells us about the fit of the model to the data. These weights are invaluable for assessing the performance of the models and provide another reason for fitting ROBMS-GLMMs.

In summary, our robust procedure performs similarly to the MS-GLMM when there is no contamination present but does better in the presence of contamination. The behavior that we have observed is quite typical of robust estimates compared to maximum likelihood.

5.3 Computational Issues

Many researchers have commented on the serious computational difficulties in fitting GLMMs and for this reason a wide range of methods have been proposed. We look first at efforts that have been made at effectively fitting CS-GLMMs. We then consider fitting MS-GLMMs and finally discuss our experiences when fitting ROBMS-GLMMs. We conclude this section with some recommendations.

5.3.1 CS-GLMMs

There are serious limitations to methods for fitting CS-GLMMs because of the need for numerical integration of dimension q to evaluate the likelihood. The computational burden has limited data analysis in several ways. First, investigators have largely restricted their attention to random intercepts models (q=1) to avoid higher dimensional numerical integration. Second, specialized software is required and is typically optimized for a particular random effects distribution (e.g. the normal). Such limitations have recently led to the development of several methods using analytical approximations to the likelihood.

Methods involving analytical approximations to the likelihood include maximum penalized quasi-likelihood and marginal quasi-likelihood [7] via EM or quasi-Newton (Fisher scoring) algorithms, and Markov Chain Monte Carlo (MCMC) methods. McCulloch [35] demonstrates that calculating ML estimates for CS-GLMMs is feasible using either a Monte Carlo EM algorithm or a Monte Carlo Newton-Raphson algorithm.

Zeger and Karim [51] illustrate the use of the Gibbs sampler for estimating parameters in CS-GLMMs. They focus on the logistic-normal case because it is the most common example and yet still poses numerical difficulties. In models with only a random intercept, likelihood evaluation by numerical integration is a competitive alternative to Gibbs sampling from a computational viewpoint. However, the strength of the Gibbs sampling approach is its extendibility to multivariate and non-normal random effects.

Booth and Hobert [6] discuss using the EM algorithm for finding maximum likelihood estimates in the CS-GLMM setting. Because the E-step of the algorithm involves an integral which cannot be evaluated analytically, they consider computer-intensive alternatives. They propose two different implementations of the Monte Carlo EM algorithm in which simulation methods are used to evaluate the intractable integral at the E-step. The first

method uses rejection sampling whereas the second method uses an importance sampling approximation. Their Monte Carlo EM algorithm is not restricted to models with normally distributed random effects.

5.3.2 MS-GLMMs

When Heagerty first proposed the MS-GLMM [20] he investigated both likelihood and estimating equation approaches to parameter estimation. For each of the estimation methods, there is the additional complexity associated with Δ_{ij}, which must be computed as a function of the marginal mean parameters β and the random effects covariance parameters α. This is achieved through numerical solution of the convolution equation (3.4). To numerically evaluate the convolution equation Heagerty uses 20-point Gauss-Hermite quadrature [1] which is a specialized method tailored to the logistic-normal integral. Unfortunately with $\sigma > 2$ Heagerty reports that this method produces an error in the approximation to the integral which is no longer acceptable. Our experience supports this statement. We view this as a limitation to Heagerty's implementation of the MS-GLMM.

An estimating-equations approach requires the specification of a model for the marginal mean and a *working model* for the marginal covariance $\text{cov}(Y_i)$. It is possible to use estimating equations [33] and a *working* logistic-normal model because we are able to calculate the induced covariance structure. Using paired estimating equations [42], Heagerty [20] obtained estimates of the mean regression parameter and the logistic-normal variance components and used these model estimates to construct covariance matrices for iterative estimation of β. Use of this approach requires that $\text{cov}(Y_{ij}, Y_{ik}|X_i)$ be computed via numerical integration for which Heagerty again uses Gauss-Hermite quadrature. The primary advantage of an estimating equations approach is that inference regarding regression parameters β is robust to incorrect specification of the within-subject dependence model. However, although estimates of β obtained as the solution of the estimating equation have been shown to be consistent under quite general conditions, efficient estimation still requires that the within-subject dependence model be correctly specified.

In many situations likelihood based methods may be preferred. One advantage of likelihood-based methods is that they can handle data that may be missing at random. To obtain valid inference using estimating equations requires the data be missing *completely*

at random. This ubiquitous phenomenon of missing data is an important issue with longitudinal data for which the natural consequence is individuals with a varying number of observations. We have seen that the observed data likelihood for subject i is a mixture over the random effects distribution and is given by

$$L_i(\beta, \alpha) = \int \prod_{j=1}^{n_i} P(Y_{ij} = y_{ij} | b_i, X_i) dF_{b_i}.$$

Since L_i cannot be evaluated analytically, numerical methods are required to compute the q-dimensional integral. Heagerty suggests using Gauss-Hermite quadrature for low-dimensional random effects models. However, as q (the dimension of the random effects increases), the computational burden for quadrature methods grows exponentially, and alternative approaches are desirable.

5.3.3 ROBMS-GLMMs

A great deal of time and effort has been spent in arriving at a precise method for obtaining ROBMS-GLMM estimates. We focused on maximizing the robustified likelihood, rather that working with estimating equations. This likelihood approach was usually taken by Heagerty perhaps because the gains of estimating-equations in term of robustness to a misspecified covariance are questionable, and both methods still require numerical integration. As well, likelihood based methods are usually preferred.

Maximization of the robustified likelihood for ROBMS-GLMMs as compared with the likelihood for MS-GLMMs, is additionally complicated by a number of factors. First, we are no longer assuming normally distributed random effects and are consequently unable to use Gauss-Hermite integration or take advantage of its many simplifications. Second, calculation of the leverage weights requires the inversion of the design matrices which is also computationally demanding. Third, prediction downweights or re-weights must be calculated at every step. Fourth, we must set tuning constants appropriately so as to achieve the desired amount of robustness. In summary, maximizing the robustified likelihood requires:

- appropriate selection of tuning constants,
- solving the convolution equation (requires both a root finder algorithm and numerical integration, see section 4.3.2),

- calculation of weights (requires inversion of the design matrices),
- numerical integration of the robustified likelihood (requires three separate numerical integrations for each individual),
- numerical computation of the hessian and the scores.

`nlm` in R [28] was the nonlinear minimizer chosen to take us on an efficient path towards the solution.

Let us consider one of the simulated data sets discussed in Section 5.2 and the effort required of the nonlinear minimizer in its search towards the maximum likelihood estimate. To compute the robustified likelihood at one location requires approximately 2,950,000 integrand evaluations. Each integrand evaluation takes approximately $5e^{-6}$ seconds. The fact that we must compute so many of them makes obtaining a solution very expensive and extremely nontrivial. Fitting ROBMS-GLMMs (as well as MS-GLMMs) is therefore computationally intensive and requires some approximation (introduced by numerical integration etc.). This is certainly motivation for considering other approaches to fitting ROBMS-GLMMs.

Our routines are readily available and listed in Appendix A. To reduce the time required to fit the ROBMS-GLMMs we provide a collection of *C* programs for computation of the likelihood. This *C* code must be dynamically loaded into R and can then be called via some user friendly R routines.

5.3.4 Recommendations

Our experience suggests that the future of statistical computation for longitudinal data will include the heavy use of iterative simulation methods. We believe that recent advances in computer hardware combined with the development of new statistical methodology, such as the ROBMS-GLMM, will create a bright future for generalized linear mixed models, particularly in the context of longitudinal data.

At present, the methods available for fitting both MS-GLMMs and ROBMS-GLMMs are extremely numerically intensive and rely heavily on numerical integration. Fortunately we anticipate that many, if not most, of the methods involving analytical approximations to

the likelihood suggested for CS-GLMMs can be applied to both MS-GLMMs and ROBMS-GLMMs. We intend to investigate application of the work of Booth and Hobert [6] to fitting ROBMS-GLMMs.

Recent procedures which avoid the need for numerical integration by using importance or Gibbs sampling techniques are attractive. Potential drawbacks include the intensive computations and some uncertainty as to when the sampling process has achieved equilibrium. Regardless, they are certainly worthy of investigation.

Chapter 6

Inference and Model Selection

We begin this chapter with a discussion of the asymptotic properties of ROBMS-GLMM estimators, including the examination of some simulation results. Some philosophical issues pertaining to robust estimation are also addressed. In section 6.2 we consider model selection since choosing a model is often an important goal of a statistical analysis. We recommend robust methods of model selection and testing procedures to be used when fitting ROBMS-GLMMs.

6.1 Asymptotics

Neuhaus, Hauck and Kalbfleisch [37] followed the approach of Akaike [2] and White [48] to investigate the effect of mixture distribution misspecification on estimates of model parameters for CS-GLMMs. Heagerty and Kurland [21] applied the results of White [48] to investigate the effects of misspecification of the mixture distribution for MS-GLMMs. We now apply the results of White [48] in order to investigate the asymptotic properties of ROBMS-GLMMs.

6.1.1 ROBMS-GLMM Estimates

For simplicity, we assume that $n_i = n$ for all i, so that all individuals have the same number of observations and that the covariate vectors X_i are independent and identically distributed, for $i = 1, \cdots, k$. We denote the response vector by $Y = (Y_1, \cdots, Y_k)$ and the covariate matrix

by $X = (X_1, \cdots, X_k)$. Without loss of generality we can assume that $E(X) = 0$. We investigate the asymptotic properties of ROBMS-GLMM estimates as $k \to \infty$ with n fixed. This is the standard approach to investigating the asymptotic properties of models for longitudinal data [21]. We also assume that the weights w_{ij} depend only on the covariates for individual i, that is X_i. This will be the case for models involving only time-dependent covariates, the LEI data being one example (analysis of which is presented in section 7.1). In section 6.1.2 we propose an alternative approach to investigating the asymptotic properties of ROBMS-GLMM estimate for the more realistic scenario which allows weights to depend on the entire covariate matrix X. Both time-dependent and time-independent covariates can then be accommodated.

Following the approach of White, we take our robustified likelihood to represent a *misspecified* likelihood and the likelihood corresponding to Heagerty's MS-GLMM to be that of our model distribution. This implies that the random effects are assumed to be normally distributed. Such an approach is consistent with much of robustness theory where often the model distribution is assumed to correspond to that model which would best describe the data were there no contamination present.

We denote by $\theta^* = (\beta^*, \alpha^*)$ the value which minimizes the Kullback-Leibler divergence between the true and misspecified models. That is, θ^* minimizes

$$E_X E_{Y|X} \log\{P_G(Y = y|\theta, X)/P_F(Y = y|\theta^*, X)\}, \tag{6.1}$$

where F denotes the misspecified model, G denotes the true model and the expectation is taken with respect to the true model. Akaike and White show that the *maximum likelihood* estimator, $\hat{\theta}^*$, under the misspecified model converges to the value θ^*. In the above expression (6.1),

$$P_G(Y = y|\theta, X) = \int \prod_j (p_j)^{y_j} (q_i)^{1-y_j} \phi(z) dz,$$

and

$$P_F(Y = y|\theta^*, X) = \int \prod_j [(p_j)^{y_j - \lambda_j} (q_i)^{1-y_j+\lambda_j}]^{w_j v_j} \phi^H(z) dz,$$

where $\alpha = \log(\sigma)$, $\alpha^* = e^{\sigma^*}$, $logit\ p_j = \Delta_{ij} + \sigma z$, $logit\ p_j^* = \Delta_{ij}^* + \sigma z$, $q_j = 1 - p_j$, $q_j^* = 1 - p_j^*$, Δ_{ij} is a function of θ and Δ_{ij}^* is a function of θ^*. Dependence on i has been suppressed.

Assumptions

In order to get asymptotic normality (via the results of White [48]) we require the following assumptions:

A1 The independent random $1 \times n$ vectors Y_i, $i = 1, \cdots, k$, have common joint distribution function $G(y, \theta)$ on Ω, a measurable Euclidean space, with density $g = dG/dv = P_G(Y = y|\theta, X)k(X)$ where v is the appropriate counting measure. Here, and in what follows, $k(X)$ denotes the density of X.

A2 The family of distribution functions $F(y, \theta)$ has densities $f(y, \theta) = dF(y, \theta)/dv = P_F(Y = y|\theta, X)k(X)$ which are measurable in y for every θ in Θ, a compact subset of a p-dimensional Euclidean space, and continuous in θ for every y in Ω.

A3 (a) $E[\log P_G(Y = y|\theta, X)]$ exists and $|\log P_F(Y = y|\theta, X)| \leq m(y|x)$ for all θ in Θ, where m is integrable with respect to G; (b) The Kullback-Leibler divergence (see equation (6.1)) has a unique minimum at θ^* in Θ.

A4 $\partial \log P_F(Y = y|\theta, X)/\partial\theta_i$, $i = 1, \cdots, p$, are measurable functions of $y|x$ for each θ in Θ and continuously differentiable functions of θ for each $y|x$ in Ω.

A5 $|\partial^2 \log P_F(Y = y|\theta, X)/\partial\theta_i\partial\theta_j|$ and $|\partial \log P_F(Y = y|\theta, X)/\partial\theta_i \cdot \partial \log P_F(Y = y|\theta, X)/\partial\theta_j|$, $i, j = 1, \cdots, p$, are dominated by functions integrable with respect to G for all $y|x$ in Ω and θ in Θ.

A6 (a) θ^* is interior to Θ; (b) $B(\theta^*)$ is nonsingular where $B(\theta^*) = \{E(\partial \log P_F(Y = y|\theta^*, X)/\partial\theta_i \cdot \partial \log P_F(Y = y|\theta^*, X)/\partial\theta_j)\}$; (c) θ^* is a regular point of $A(\theta)$ where $A(\theta) = \{E(\partial^2 \log P_F(Y = y|\theta, X)/\partial\theta_i\partial\theta_j)\}$. A *regular point* of the matrix $A(\theta)$ is a value of θ such that $A(\theta)$ has constant rank in some open neighborhood of θ.

A7 $|\partial[\partial P_F(Y = y|\theta, X)/\partial\theta_i \cdot P_F(Y = y|\theta, X)]/\partial\theta_j|$, $i, j = 1, \cdots, p$, are dominated by functions integrable with respect to v for all θ in Θ, and the minimal support of $P_F(Y = y|\theta, X)$ does not depend on θ.

Discussion of Assumptions

A1 and A2 are satisfied by letting G correspond to the distribution specified in the MS-GLMM and F correspond to those specified in the ROBMS-GLMM.

A3 requires that $E[\log P_G(Y = y|\theta, X)]$ exists. We can be certain this expectation exists as it is simply a finite summation. That is,

$$E[\log P_G(Y = y|\theta, X)] = \sum_y \log P_G(Y = y|\theta, X) * P_G(Y = y|\theta, X).$$

Condition A3(a) also requires that $|\log P_F(Y = y|\theta, X)|$ be bounded above by some function $m(y|x)$ not involving θ. Since $P_F(Y = y|\theta, X)$ must be between 0 and 1 it follows that

$$0 \leq |\log P_F(Y = y|\theta, X)| < \infty$$

with $|\log P_F(Y = y|\theta, X)| \to \infty$ as $P_F(Y = y|\theta, X) \to 0$. Hence if we can make $P_F(Y = y|\theta, X)$ arbitrarily small then $|\log P_F(Y = y|\theta, X)|$ will be unbounded.

Can we make $P_F(Y = y|\theta, X)$ arbitrarily small? Let us suppose that we can. Then, since

$$P_F(Y = y|\theta, X) = \int \prod_j [(p_j)^{y_j - \lambda_j} (q_j)^{1 - y_j + \lambda_j}]^{w_j v_j} \phi^H(z),$$

$P_F(Y = y|\theta, X)$ will be arbitrarily small only when at least one term in the right hand side of the above expression is arbitrarily small. In order for one of these terms to be arbitrarily small we require that the corresponding p_j (or q_j) be arbitrarily small. Requiring p_j be arbitrarily small means that $\frac{e^{\Delta_j + \sigma z}}{1 + e^{\Delta_j + \sigma z}}$ must be arbitrarily small since

$$p_j = \frac{e^{\Delta_j + \sigma z}}{1 + e^{\Delta_j + \sigma z}}.$$

Now since p_j is also involved in the convolution equation this implies that the right hand side of the convolution equation (4.5):

$$h(\eta_j) = \int p_j \phi^H(z) dz$$

must be arbitrarily small. However our robust procedure ensures that when $\int p_j \phi^H(z) dz < \delta$

($\delta > 0$) then $w_j = 0$ and hence

$$[(p_j)^{y_j-\lambda_j}(q_j)^{1-y_j+\lambda_j}]^{w_j v_j} = 1.$$

(A similar argument can be made for the case in which q_j is arbitrarily small). Therefore we cannot make one term in the expression for $P_F(Y = y|\theta, X)$ arbitrarily small and consequently $P_F(Y = y|\theta, X)$ cannot be made arbitrarily small. Hence by contradiction $|\log P_F(Y = y|\theta, X)|$ must be be bounded above. Condition A3(a) is therefore satisfied. Note that there is a problem with Heagerty's MS-GLMM since his implementation allows $\int p_j \phi^H(z)$ to be made arbitrarily small and hence violates assumption A3(a).

Classical estimation theory is based on assuming a unique maximum likelihood estimate [18]. Following this approach, we assume both the true and misspecified likelihoods have unique maxima. Assumption A3(b) then follows.

A4 requires that the first two derivatives with respect to θ in Θ exist. By locally smoothing functions λ and v in the definition of the ROBMS-GLMM we satisfy this assumption.

A5 requires that the derivatives are appropriately dominated by functions integrable with respect to G. Since integrating with respect to G in this case is simply performing a finite summation, this assumption is verified.

For A6(a) we must assume θ^* is interior to Θ. A6(b) and A6(c) as well as A7 follow from the fact that the expectations involved in these expressions are again simply finite summations.

With A1 - A7 satisfied, we can now state four important theorems based on the results of White [48].

Results

We define the robust log-likelihood of the sample as

$$L_k(Y = y|\theta, X) \equiv \frac{1}{k}\sum_{t=1}^{k} \log P_F(Y_t = y_t|\theta, X),$$

and our robust maximum likelihood estimator (ROBMLE) as the parameter vector $\hat{\theta}_k^*$ which solves the problem

$$\max_{\theta \in \Theta} L_k(Y|X, \theta).$$

THEOREM 1 (Existence): Given assumptions A1 and A2, for all k there exists a measurable ROBMLE, $\hat{\theta}_k$. (see Theorem 2.1 of [48])

Having established the existence of the ROBMLE, we examine its properties. It is well known that when F is in fact the model distribution, the MLE is consistent for θ_0 under suitable regularity conditions.

THEOREM 2 (Consistency): Given Assumptions A1-A3, $\hat{\theta}_k \to \theta^*$ as $k \to \infty$ for almost every sequence (Y_t); i.e., $\hat{\theta}_k \xrightarrow{a.s.} \theta^*$. (see Theorem 2.2 of [48])

In other words, the ROBMLE is generally a strongly consistent estimator for θ^*, the parameter vector which minimizes the Kullback-Leibler divergence.

If the probability model agrees with the model distribution $\hat{\theta}_k$ is consistent for the *true* parameter vector θ_0. Stated differently, we have, under F, $\hat{\theta}_k \to \theta_0$, whereas under G, $\hat{\theta}_k \to \theta^*$. These results raise some important philosophical questions pertaining to what we should consider our model distribution to be. A detailed discussion follows in section 6.1.3.

We define the matrices

$$A_k(\theta) = \{k^{-1} \sum_{t=1}^{k} \partial^2 \log P_F(Y_t = y_t|\theta, X)/\partial\theta_i \partial\theta_j\},$$

$$B_k(\theta) = \{k^{-1} \sum_{t=1}^{k} \partial \log P_F(Y_t = y_t|\theta, X)/\partial\theta_i \cdot \partial \log P_F(Y_t = y_t|\theta, X)/\partial\theta_j\},$$

$$C_k(\theta) = A_k(\theta)^{-1} B_k(\theta) A_k(\theta)^{-1},$$

and

$$C(\theta) = A(\theta)^{-1} B(\theta) A(\theta)^{-1}.$$

THEOREM 3 (Asymptotic Normality): Given Assumptions A1-A6

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \overset{A}{\sim} N(0, C(\theta^*)).$$

(see Theorem 3.2 of [48])

Moreover, $C_k(\hat{\theta}_k) \xrightarrow{a.s.} C(\theta^*)$, element by element.

If we further assume that the model is correctly specified, then with A7, we have the familiar equality in maximum likelihood theory which ensures the equivalence of the Hessian (left-hand side) and outer product (right-hand side) form for the information matrix.

> THEOREM 4 (Information Matrix Equivalence): Given Assumptions A1-A7, if $P_G(Y = y|\theta_0, X) = P_F(Y = y|\theta_0, X)$ for θ_0 is Θ, then $\theta^* = \theta_0$ and $A(\theta_0) = -B(\theta_0)$, so that $C(\theta_0) = -A(\theta_0)^{-1} = B(\theta_0)^{-1}$, where $-A(\theta_0)$ is Fisher's information matrix.(see Theorem 3.3 of [48])

Application

In summary, we have shown that if our model distribution is assumed to correspond to the MS-GLMM then the ROBMLE are asymptotically normal with variance as given in Theorem 3. However, if our model distribution is assumed to correspond to the ROBMS-GLMM then the ROBMLE are consistent for θ_0 and asymptotically normal with variance as given in Theorem 4.

Table 6.1 compares the standards errors of the ROBMS-GLMM estimates for 50 simulated data sets containing 5% leverage contamination. As in section 5.2, each data set involves 60 individuals, with 10 observations per individual. However, to avoid repetition, the leverage contamination is now restricted to only the X_{ij1} unlike in section 5.2 where both X_{ij1} and X_{ij2} were contaminated. The standard errors *sd0* are computed based on the simulations. The standard errors *sd1* are computed based on assuming the ROBMS-GLMM is the model distribution whereas the standard errors *sd2* are computed assuming the MS-GLMM is the model distribution. Both *sd1* and *sd2* are based on matrices $A_k(\theta)$ and $B_k(\theta)$ which are calculated at the estimated parameter values. One can see that as expected the standard errors (*sd1*) based on assuming the true model is the ROBMS-GLMM are slightly smaller. The fact that there is not a great deal of difference gives us additional confidence in reporting either form of standard error depending on what we wish to assume about the *true* model.

Table 6.1: Standard errors based on 50 simulations.

	Estimate	*sd0*	*sd1*	*sd2*
β_0	-2.05	.48	.44(.08)	.58(.24)
β_1	3.14	.62	.63(.14)	.87(.32)
β_2	1.01	.30	.28(.06)	.29(.08)
α	0.01	.19	.17(.04)	.18(.06)

With no contamination present and normally distributed random effects, the ROBMS-GLMM estimates were found to be approximately 85% to 95% as efficient as the MS-GLMM estimates as c varied from 2 to 3, respectively.

6.1.2 Extension

The asymptotic development presented in the previous subsection requires an assumption that cannot always be satisfied. The assumption is that the w_{ij}, required in the definition of the ROBMS-GLMM, depend only on the covariates specific to the ith individual. With only time-dependent covariates present in our model, this is always the case since we define our weights w_{ij} as being $\sqrt{1-h_{jj}}$ where h_{jj} are the diagonal elements of the hat matrix $h_i = X_i(X_i^T X_i)^{-1}X_i$. However when there are time-independent covariates present we instead define the w_{ij} based on the entire hat matrix and hence do not satisfy the assumption. Without this assumption, we no longer have independent and identically distributed observations and consequently cannot apply the results of White [48].

In the presence of both time-dependent and time-independent covariates we therefore propose an alternative asymptotic development based on the work of Yuan and Jennrich [49]. We believe this development will lead to results similar to those given in the previous subsection.

By assuming that the covariate matrix $X = (X_1, X_2, \cdots, X_N)$ is fixed we have observations Y_i which are are independent but not identically distributed. Yuan and Jennrich [49] give very general conditions under which the existence, strong consistency, and asymptotic normality of estimators can be obtained. They state that their results have application to maximum likelihood as well as robust estimation.

In order to apply the results of Yuan and Jennrich [49] we must be able to write down our robust estimators as the solution of a set of estimating equations $G_k(\theta) = 0$ where $G_k(\theta) = \sum_{i=1}^{k} g_i(\theta)$ and interest is in estimating a value θ. In what follows θ_0 represents the true parameter value. Yuan and Jennrich [49] state that the following conditions lead to the existence, consistency and asymptotic normality of a sequence $\hat{\theta}_k$ of roots of $G_k(\theta) = 0$.

1. For each i, $g_i(\theta_0)$ has mean zero and covariance V_i, and $\overline{V} = 1/k\sum_{i=1}^{k} V_i \rightarrow V$ which is positive definite.

2. For the V_i in Assumption 4, there are positive numbers B and δ such that for all i, $E|g_i^T(\theta_0)(I+V_i)^{-1}g_i(\theta_0)|^{1+\delta} \leq B$.

3. With probability one $g_i(\theta)$ is twice continuously differentiable on Θ.

4. For each $\theta \in \Theta$, $E\dot{G}_k(\theta) \rightarrow \dot{G}(\theta)$ with $A = \dot{G}(\theta_0)$ being nonsingular, and with probability one $\dot{G}_k(\theta) \rightarrow \dot{G}(\theta)$. $\dot{G}$ denotes the matrix of derivatives taken with respect to θ.

5. For each i, $||\ddot{g}_i(\theta)|| \leq T_i$ and $P(sup_{i\geq 1}T_i < \infty) = 1$ where $\ddot{g}$ denotes the matrix of second derivatives taken with respect to θ.

If the $G_k(\theta)$ satisfy Assumptions 1 through 5, then the estimating equations have roots which are consistent and asymptotically normal such that

$$\sqrt{k}(\hat{\theta}_k - \theta_0) \rightarrow N(0, \Omega)$$

where $\Omega = A^{-1}VA'^{-1}$ with $A = \dot{G}(\theta_0)$.

The distinguishing feature of Yuan and Jennrich's approach [49] is the requirement that the $g_i(\theta_0)$ have mean zero (as stated in Assumption 1). In order to ensure this condition is met we introduce a Fisher consistency correction term $a(\theta)$ and take

$$g_i(\theta) = \frac{\partial}{\partial\theta}\log L_i^R(\theta) - a(\theta),$$

with $L_i^R(\theta)$ as defined in (4.7) and

$$a(\theta) = \sum_{k=1}^{2^{n_i}} \int \log L_k^R(\theta(F)) \prod_{j=1}^{n_i} \frac{(e^{\Delta_{jk}+\sigma z_k})^{y_{kj}}}{1+e^{\Delta_{jk}+\sigma z_k}} \phi^H(z_k)dz_k,$$

where F is the assumed model distribution (as discussed in section 6.1.3). Although non-trivial, one can compute the Fisher consistency correction term. We believe this result will then lead to estimators; similar in form to our ROBMS-GLMM estimators, which are both consistent and asymptotically normal. Unfortunately, the computation of these new estimators requires solving a set of simultaneous equations (the estimating equations) rather than the use of iterative maximization procedures. We expect that finding the solution will be even more computationally challenging that our current approach.

6.1.3 What is the *True* Model?

In classical estimation theory we are often in search of a *true* model, with *true* constants which describe it. However, as Hampel [18] suggests, there are relatively few cases when a simple *true model* appears to hold, even for huge data sets. Often this is due to the fact that we only have a limited sample and much of classical statistics is based on assuming that the sample size can be increased arbitrarily. As well, even if one believes in a *true* constant, its value is tied to our practical experience only through measurements with their errors. Therefore it seems reasonable to admit that there will generally be some amount of distortion associated with what we often refer to as the *true model.*

In acknowledging that there is an unknown though limited distortion of the true model, we accept that there must then be an unknown bias which is bounded but in general non-vanishing with increasing sample size, as is shown in Huber [26]. Then with this *unavoidable bias*, what can be estimated? Hampel [18] believes that this question seems to be one of the most difficult philosophical questions in the background robust statistics.

For a moment we will ignore the problems of existence of a *true model.* In doing so, we begin with a data set and are instead concerned with the totality of all model distributions *compatible* with the data. Given our data and the set of all probability distributions which might have generated them, how do we select a model distribution to describe the data? Keeping in mind that our model is intended to serve as a summary of the data it would be nice to choose one which is *simple.* However, generally speaking, the more data, the more complicated the model and consequently we may consider entertaining a simple model even if it is only approximate. Experience shows that simplified models which do not explain all idiosyncrasies of the data, are often much better than more complicated ones [18]. In theories of robust statistics we consider keeping the same simple model, although

explicitly only as an approximation. This does not destroy the beauty of classical parametric models but rather keeps them and fortifies them against mishap. The price is that any model distribution and hence any parameter in some neighborhood of the actual distribution may be chosen to describe the data. However as we have shown, this limitation is only one of several existing already for other reasons.

The idea is to choose an estimator which estimates the correct quantity if the model were exactly true, and which changes as little as possible in neighborhoods of the model. We assume our actual underlying distribution has been generated from a model distribution with *ideal* but unknown parameter by some unknown but bounded distortion; the distortion causes a bounded but unknown and unavoidable bias.

In light of some of the concerns that have been raised above, we prefer to speak of *assumed model distributions* as opposed to a *true model*. This seems to alleviate the difficult problem associated with trying to specify a single true model, especially since one will rarely exist. We instead investigate results obtained with different model distributions, and as demonstrated by Table 6.1, insightful conclusions can still be drawn.

6.2 Model Selection

Model selection is a key component in any statistical analysis. Typically the choice of the final model(s) is an iterative procedure based on subject matter knowledge and on formal selection criteria. Here we recommend robust methods of model selection and testing useful when fitting ROBMS-GLMMs. After all, one cannot estimate the parameters robustly and apply unmodified classical model selection procedures [44].

We consider two approaches to performing model selection. The first is based on the Akaike Information Criterion while the second mimics the approach taken by Cantoni and Ronchetti [8] in the context of generalized linear models.

6.2.1 A Robust Akaike Criterion

In 1997 Ronchetti [44] considered a general parametric model $\{P_\theta | \theta \in \Theta\}$ and discussed a robust version of the Akaike Information Criterion.

Suppose we have n independent and identically distributed observations $z_1, \cdots, z_n$ and

we denote by L_p the log-likelihood of the model with p parameters. Akaike's Criterion amounts to choosing the model that minimizes $-2L_p + 2p$. This procedure may be viewed as an extension of the likelihood principle and is based on a general information theoretic criterion, which is in turn based on the computation of the log-likelihood function at the maximum likelihood estimator for θ. However since it is well known that maximum likelihood estimators are non-robust for many important parametric models, Ronchetti instead considered general M-estimators.

A general M-estimator is defined as the minimum with respect to θ of the objective function $\sum_i \tau(z_i, \theta)$, for a given function τ, and satisfies the first order condition

$$\sum_i \psi(z_i, \theta) = 0, \tag{6.2}$$

where $\psi(z, \theta) = \partial\tau(z, \theta)\partial\theta$. If we choose $\tau(z, \theta) = -log\ p_\theta(z)$, where p_θ is the density of P_θ, the objective function equals minus the log-likelihood function, ψ is the score function, and the corresponding M-estimator is the maximum likelihood estimator.

In order to derive the Akaike Criterion based on a general M-estimator, Ronchetti [44] looks at such an estimator as a maximum likelihood estimator with respect to an underlying density $p_\theta(z)$ proportional to $exp(-\tau(z, \theta))$. This is of course only true when the function τ satisfies certain conditions but this does not affect the result which follows. The usual Akaike Criterion based on this density can then be written, yielding the following robust version

$$AICR(p; \alpha_p, \tau) = 2\sum_i \tau(z_i, \hat{\theta}) + \alpha_p, \tag{6.3}$$

where $\hat{\theta}$ is the general M-estimator defined by (6.2), $\alpha_p = tr(M^{-1}Q)$, $M = -E[\partial\psi/\partial\theta]$, and $Q = E[\psi\psi^T]$.

Application to the ROBMS-GLMM

It is straightforward to apply the robust Akaike Criterion (6.3) to the ROBMS-GLMM since we have k observations $y_1, \cdots, y_k$ with $\sum_i \tau(y_i, \theta)$ equal to $\sum_i \log L_i^R$ and L_i^R as defined in (4.7). In the next chapter we apply this robust model selection criterion as part of the statistical analysis of a real medical application.

6.2.2 Robust Testing

Cantoni and Ronchetti [8] proposed a natural class of robust estimators for generalized linear models based on the notion of quasi-likelihood. They then derived robust deviances that can be used for stepwise model selection as in the classical framework. The asymptotic distribution of tests based on robust deviances were also discussed. We suggest that a similar approach can be followed for ROBMS-GLMMs.

To evaluate the adequacy of a model, we define a robust goodness-of-fit measure which we call robust deviance, that is:

$$D_{L^R}(y,\mu^b) = -2L^R(y,\mu^b) = -2\sum_{i=1}^{k} L_i^R(y_i,\mu_i^b), \tag{6.4}$$

where μ_i^b contains the $\mu_{ij}^b = h(\Delta_{ij} + \sigma z)$ and L_i^R is as defined in (4.7) but now expressed alternatively as a function of the data and the parameters, that is,

$$L_i^R(y_i,\mu_i^b) = \int \prod_{j=1}^{n_i} [\{\mu_{ij}^b\}^{y_{ij}-\lambda_{ij}} \{1-\mu_{ij}^b\}^{1-y_{ij}+\lambda_{ij}}]^{w_{ij}v_{ij}} \phi^H(z)dz.$$

$D_{L^R}(y,\mu^b)$ describes the quality of a fit and we will use it to define a statistic for model selection. Let us consider the model M_p, with p parameters. Suppose that the corresponding set of parameters is $\theta = (\beta_1,\beta_2,\cdots,\beta_{p-1},\alpha)' = (\theta'_{(1)},\theta'_{(2)})$. We consider a nested model $M_{p-q} \in M_p$ with $(p-q)$ parameters, and testing whether the sub-model M_{p-q} holds.

We estimate the vector of parameters by minimizing $-\sum_i^k \log L_i^R$ for the complete model, and we obtain an estimator $\hat{\theta}$ of θ. Under the null hypothesis, the same procedure yields an estimator $\dot{\theta}$ of $(\theta_{(1)},0)$. We write $\hat{\mu}^b$ and $\dot{\mu}^b$ for the estimated linear predictors associated with the estimate $\hat{\theta}$ and $\dot{\theta}$ respectively. Then, we define a robust measure of discrepancy between the two nested models by

$$\Lambda_{L^R} = [D_{L^R}(y,\dot{\mu}^b) - D_{L^R}(y,\hat{\mu}^b)] \tag{6.5}$$

where $D_{L^R}(y,\mu^b)$ is defined by (6.4).

Cantoni and Ronchetti [8] go further to establish the asymptotic distribution of their test statistic, similar to ours as given by (6.5). By assuming the conditions for the existence,

consistency, and asymptotic normality of M-estimators, they show that the asymptotic distribution of their test statistic is a linear combination of χ^2 random variables with one degree of freedom. The reader is referred to their discussion for further details. At present we are at work on establishing the asymptotic distribution for the test statistic (6.5).

We now have a set of tools to undertake a complete robust analysis. We can obtain robust parameters estimates using our ROBMS-GLMM and can then use either the test statistic (6.3) or potentially (6.5) to make inference and model choice. Complete robust analyses of some important medical applications are presented in the next chapter.

Chapter 7

Applications

In this chapter we apply our robust procedure to data arising from two biomedical studies. We are able to answer some important questions posed by associated medical researchers as well as compare the performance of the ROBMS-GLMM with its earlier non-robust version. Robust model selection strategies are also demonstrated. First, we consider data collected on laryngoscopic endotracheal intubation (LEI) with the goal of identifying the parts of the process of LEI which are most predictive of successful completion. We also look at data collected on Nova Scotian women who have had more than one pregnancy. Interest is in their smoking habits as related to pregnancy outcomes.

7.1 Evaluation of Proficiency of Laryngoscopic Intubation

Laryngoscopic endotracheal intubation (LEI) is a potentially life saving procedure in which numerous health care professionals are trained and expected to be competent. Unfortunately, there is little information to indicate the amount of training required, or what signifies true competence in LEI. As it is, training programs for personnel such as paramedics and respiratory therapists are arbitrary and potentially inadequate. This is a concern given the critical importance of good airway management.

Data from a prospective longitudinal study of novice endotracheal intubators directed by Dr. Orlando Hung of the Department of Anesthesia, Dalhousie University is examined. The goal is to identify features of the process of LEI which are predictive of a successful

LEI. These features are entered as covariates into the proposed model and we are interested in how the performance of a successful LEI changes across various subsets of the population.

In this section we apply the MS-GLMM and ROBMS-GLMM to the LEI data. Traditional teaching of LEI has focused on certain key aspects of successful intubation including proper positioning, timely performance and proper lifting of the laryngoscope blade during LEI. Accordingly we sought to clarify the issue of quality and competence on these grounds by studying specific features of the LEI thought to be important.

A total of 438 LEI were analyzed during this longitudinal study. We let the response Y_{ij} equal 1 if trainee i performs a complete LEI in less than 30 seconds on trial j, and 0 otherwise. We restrict our attention to judging trainees based on: whether they inserted the scope properly (PROPLGSP); whether they performed the lift successfully (PROPLIFT); and if there was unsolicited intervention by the attending anesthesiologist (HELP). Each of these criteria were included as covariates in our model. Later, in subsection 7.1.1 we demonstrate how our robust model selection procedure actually led us to arrive at this model. 19 trainees performed anywhere from 18 to 33 trials. A covariate TRIALCAT was defined; TRIALCAT=1 for trials 1-5, TRIALCAT=2 for trials 6 thru 10, and so forth.

The results of fitting both the ROBMS-GLMM and MS-GLMM are shown in Table 7.1. Recall that k and c are the tuning constants required for Huber's least favorable distribution and the response weighting strategies, respectively. We assumed that the underlying model involved normally distributed random effects. Corresponding standard errors could then be computed from the inverse of the Hessian matrix for the MS-GLMM, and using the sandwich estimator (Theorem 3)for the ROBMS-GLMM. Observation weighting was performed since there were a fairly large number of observations per individual and relatively small correlation between them ($\hat{\sigma} \approx .61$). The parameter estimates are quite similar and as expected the corresponding standard errors are slightly larger for the ROBMS-GLMM. Table 7.2 summarizes some additional information given by the ROBMS-GLMM. 2% of the observations were actually re-weighted with $c = 3.0$ suggesting that a small but significant amount of the data was not well fitted by the model. As well, the 20th observation on trainee 12 was assigned a leverage weight of .61 suggesting that it should be re-examined. Upon doing so we noticed that this trainee performed particularly poorly (PROPLGSP=0, PROPLIFT=0) on this trial, even though the trainee had a reasonable amount of practice

(19 previous trials) and previous success. Hence, the relatively small weight appears reasonable. With $c = 2.0$ approximately 4% of the data were re-weighted, and more extremely.

Table 7.1: Parameter estimates (standard errors) for the LEI data, $*k = 2.46$.

	Model				
Variable	**MS-GLMM**	**ROBMS-GLMM***			
		Re-Weight		Downweight	
		c=2.0	*c=3.0*	*c=2.0*	*c=3.0*
INTERCEPT	-4.26(.49)	-5.45(.66)	-4.62(.56)	-5.34(.64)	-4.61(.55)
TRIALCAT	.30(.09)	.36(.11)	.32(.10)	.36(.11)	.32(.10)
PROPLGSP	.94(.31)	1.13(.38)	1.02(.35)	1.11(.38)	1.02(.35)
PROPLIFT	1.82(.33)	2.60(.48)	2.00(.38)	2.50(.46)	1.99(.38)
HELP	.78(.26)	.85(.30)	.87(.29)	.85(.30)	.87(.29)
α	-.49(.49)	-.74(.48)	-.66(.43)	-.69(.46)	-.65(.42)

Table 7.2: Weights assigned to the LEI data by the ROBMS-GLMM.

Observation					
Ind.	*Obs.*	λ_{ij}		v_{ij}	
		c=2.0	*c=3.0*	*c=2.0*	*c=3.0*
0	10	.77	.52	.24	.49
1	4	.72	.42	.30	.59
	9	.31	0	.72	1
	14	.65	.26	.38	.76
5	0	.58	.10	.45	.91
	6	.50	0	.54	1
	8	.50	0	.54	1
	19	.12	0	.93	1
8	16	.52	.05	.51	.96
	28	-.14	0	.89	1
10	13	.17	0	.86	1
11	5	.49	0	.54	1
12	2	.84	.65	.17	.35
13	11	.65	.26	.37	.76
14	5	.31	0	.72	1
17	3	.11	0	.92	1
	11	.60	.19	.42	.82
18	17	.58	.12	.45	.89
	21	.12	0	.93	1

To demonstrate how the maximum likelihood estimates for the MS-GLMM may be highly influenced by modest contamination in the data we examine its performance in the

presence of both influential and outlying observations. First let us suppose that some trial categories were confused. That is, the trial number was used instead of trial category for each of 20 randomly chosen trials. This would amount to approximately 5% contamination. The leverage weights of the ROBMS-GLMM should adjust for this sort of contamination. Resulting estimates are shown in Table 7.3 and demonstrate that 5% contamination can drastically change MS-GLMM estimates, especially for the contaminated covariate. The corresponding ROBMS-GLMM estimates are also provided and are less impacted by the contaminated data. Note especially the coefficient for TRIALCAT. Given the large discrepancy between the ROBMS-GLMM and MS-GLMM, one would suspect that contamination was present and thoroughly examine the weights. Notice that the ROBMS-GLMM estimates are closer to that obtained with the original, non-contaminated data. In general the ROBMS-GLMM is more reliable and also provides additional information concerning the fit of the model.

Table 7.4 presents the contamination that was introduced and the corresponding weights. A number of interesting features are apparent. Our robust procedure does best at identifying contamination when there is only a single contaminant per individual. It also does better at identifying contamination when it occurs later in an individual's series of observations. This is a direct result of the type of contamination we have introduced. Later observations will have larger trial numbers and consequently higher leverage that those occuring earlier in the series for a particular individual. The more contaminants per individual the more difficulty the procedure has. It appears that if more than approximately 10% of an individual's observations are contaminated, this contamination can no longer be identified.

Table 7.3: Parameter estimates (standard errors) for LEI data with 5% leverage contamination introduced, $*k = 2.46$.

	Model				
Variable	**MS-GLMM**	**ROBMS-GLMM***			
		Re-Weight		Downweight	
		c=2.0	*c=3.0*	*c=2.0*	*c=3.0*
INTERCEPT	-3.44(.38)	-5.06(.64)	-4.10(.52)	-4.88(.65)	-4.08(.49)
TRIALCAT	.00(.01)	.12(.05)	0.10(.05)	.11(.05)	.10(.05)
PROPLGSP	.99(.31)	1.21(.39)	1.11(.36)	1.19(.38)	1.11(.36)
PROPLIFT	1.92(.32)	2.96(.52)	2.14(.38)	2.81(.50)	2.13(.38)
HELP	.79(.26)	.78(.30)	.81(.29)	.77(.30)	.81(.26)
α	-.62(.37)	-.98(.63)	-.82(.50)	-.91(.57)	-.81(.49)

Table 7.4: Weights on the leverage contamination for the LEI data.

Real Contamination		**Weight**
Ind.	*Obs.*	*Leverage*
0	16	.28
1	5	-
2	22	.27
6	16	-
	20	.56
8	10	-
	23	-
	29	-
9	13	.48
10	15	.46
11	17	.27
12	14	-
	19	.60
14	17	.33
15	18	-
	24	.58
16	3	-
	20	.29
18	4	-
	20	.24

Now we consider another type of contamination. We randomly choose one individual: Individual 14. This is equivalent to choosing 5% of the individuals in the data set. We then introduce 5% contamination by examining all 22 of Individual 14's observations and flipping the first half of the responses. This amounts to flipping observations 1 thru 11.

This contamination seems reasonable when compared with real situations, where a 0 is commonly recorded instead of a 1 and vice versa. The results of fitting both the MS-GLMM and ROBMS-GLMM are shown in Table 7.5. One can see that the ROBMS-GLMM estimates are much more in line with the original estimates than those of the MS-GLMM. Table 7.6 illustrates that much of the contamination is identified by our procedure. It is immediately clear that Individual 14 is outlying and should be re-examined.

Table 7.5: Parameter estimates (standard errors) for the LEI data with 5% prediction contamination introduced, $^*k = 2.46$.

		Model			
Variable	**MS-GLMM**	**ROBMS-GLMM***			
		Re-Weight		Downweight	
		c=2.0	*c=3.0*	*c=2.0*	*c=3.0*
INTERCEPT	-3.61(.45)	-4.78(.58)	-3.96(.50)	-4.60(.56)	-3.93(.50)
TRIALCAT	.23(.09)	.29(.10)	.25(.10)	.28(.10)	.25(.09)
PROPLGSP	.97(.30)	1.17(.37)	1.05(.33)	1.15(.36)	1.05(.33)
PROPLIFT	1.43(.29)	2.12(.40)	1.58(.33)	2.00(.39)	1.56(.33)
HELP	.70(.25)	.84(.29)	.80(.27)	.82(.29)	.80(.27)
α	-.23(.26)	-.62(.41)	-.42(.33)	-.55(.38)	-.40(.32)

Table 7.6: Weights on the prediction contamination for the LEI data.

Contamination		Weights			
Ind.	*Obs.*	*Reweight*		*Downweight*	
		c=2.0	*c=3.0*	*c=2.0*	*c=3.0*
0	10	.71	.40	.31	.61
1	4	.62	.21	.42	.81
	9	.29	-	.74	-
	14	.57	.10	.47	.91
5	0	.42	-	.63	-
	6	.33	-	.73	-
	8	.33	-	.73	-
8	15	.05	-	.98	-
	16	.41	-	.64	-
9	19	.05	-	.98	-
10	13	.18	-	.85	-
11	5	.33	-	.73	-
12	2	.79	.53	.24	.48
13	11	.57	.10	.47	.91
14	0	.62	.21	.42	.81
	1	.39	-	.64	-
	4	.42	-	.63	-
	6	.75	.47	.27	.54
	8	.75	.47	.27	.54
	9	.56	.10	.48	.91
	18	.05	-	.98	-
17	3	.07	-	.97	-
	11	.49	-	56	-
18	17	.50	-	.54	-

In general, it appears that the choice of $c = 2.00$ is superior for the LEI data. This is substantiated by the fact that there is fairly close agreement with the parameter estimates obtained with the uncontaminated data and in the presence of both types of contamination. As well, the choice of $c = 2.00$ results in a modest amount of weighting while at the same time identifying the majority of the contamination.

7.1.1 Robust Model Selection

Deletion of covariates from a model can in fact improve a model as well as reduce collinearity. In many problems, we may seek a relatively small set of covariates that have nearly the same information as the full set; further analysis can then concentrate on this subset of predictors, and possibly simplify results. Both cost and time savings are also motivation for performing model selection. The large computational requirements for fitting ROBMS-GLMMs (as well as MS-GLMMs) make the possibility of performing model selection very desirable, provided we can do it robustly.

In the original LEI data there were actually two more covariates thought to be of relevance to the performance of a successful intubation. The first was NECKFLEX, an indicator variable which was set equal to 1 if the neck flexion was performed properly. The second, EXTOA, also an indicator variable, was set equal to 1 when there was proper extension of the Atlanto-Occipital joint.

The robust Akaike Criterion (AICR), discussed in section 6.2.1, provides us with a straightforward mechanism for performing model selection. Basically we choose the model that minimizes this criterion. Table 7.1.1 presents the models that were under consideration along with their corresponding AICR values. One immediately sees that the model involving only TRIALCAT, PROPLGSP, PROPLIFT and HELP is the *best* model based on the robust Akaike Information Criterion. This justifies its use in earlier sections and more importantly suggests that NECKFLEX and EXTOA are not important covariates and can therefore be removed for the sake of simplicity and computational efficiency. It is of interest to note that model selection procedures performed on the corresponding MS-GLMMs yield the same conclusion concerning the removal of EXTOA and NECKFLEX.

Table 7.7: Models with smallest AICR for the LEI Data.

p	AICR	Predictors in the Model					
4	303.98	TRIALCAT	PROPLGSP	PROPLIFT	HELP		
5	323.72	TRIALCAT	PROPLGSP	PROPLIFT	HELP	EXTOA	
5	307.81	TRIALCAT	PROPLGSP	PROPLIFT	HELP	NECKFLEX	
6	320.32	TRIALCAT	PROPLGSP	PROPLIFT	HELP	NECKFLEX	EXTOA

7.2 Smoking Habits During Pregnancies

Dr. Linda Dodds of the IWK-Grace Hospital, Halifax, Nova Scotia, works closely with the Nova Scotia Atlee Perinatal Database (NSAPD), a unique repository of comprehensive maternal and newborn information. It contains demographic variables, procedures, interventions, maternal and newborn diagnoses, and morbidity and mortality information for pregnancies and births occurring in Nova Scotia hospitals since 1988. The information is collected from patient care records by health records personnel and is entered into the database soon after collection [3]. The research objectives of Dr. Dodds include attempting to learn more about the smoking habits of women who are childbearing. Interesting questions that have been posed include:

- Does it take a bad pregnancy outcome to convince women to stop smoking for subsequent pregnancies?
- Are women who smoke during pregnancy more likely than nonsmoking women to have a low birth weight baby?

For each pregnancy in the NSAPD there are a vast number of covariates recorded, including the amount smoked (average number of cigarettes/day), mother's age, baby's birth weight, etc. The database in its entirety includes approximately 24,000 women. If we limit it to only those who smoked during their first pregnancy, the database is reduced by about 70%. Note that a woman who smokes before she is pregnant but stops as soon as she knows she is pregnant is categorized as a non-smoker since research suggests that the effects of smoking prior to pregnancy are negligible [13].

For the purposes of our analysis we restricted our attention to only those women who had two or more pregnancies and smoked during the first. We then chose a random sample of 300 of these women. Our final data set consisted of 669 pregnancies distributed as follows:

Number of pregnancies (n_i)	2	3	4	5	6
Women	224	54	12	1	1

In Nova Scotia it has also been found that the smoking rate is highest among women under 20 years of age and decreases with increasing age [13]. These findings suggest that

mother's age may be an important predictor of smoking behavior. Studies have also shown that both ceasing to smoke at some point during pregnancy and reducing smoking during pregnancy can be beneficial to the baby [24]. The predominant effect of nicotine (inhaled during smoking) is to decrease blood supply to the fetus. Medical research has established that the risks to the baby of smoking during pregnancy include low birth weight, SIDS, respiratory diseases, and morbidity (illness) [13]. Of these risks, low birth weight seems to be the one most commonly experienced. Discussions with Dr. Dodds suggested that it would be useful to include in our model a covariate for low birth weight as well as another covariate summarizing the remaining aforementioned problems. Hence we define a covariate PROB and set it equal to 1 if any of the problems were experienced in a previous pregnancy and equal to 0 otherwise.

To complete our model formulation we let the response Y_{ij} equal 1 if woman i smokes during pregnancy j, and 0 otherwise. This is consistent with a previous paper by Dodds [13] in which smoking is analyzed as a dichotomous variable. We include the covariate PROB as well as the age (standardized) of the mother (AGE); and the birth weight (kg) of the baby (BIRTHWEIGHT).

The results of fitting both the ROBMS-GLMM and MS-GLMM with and without the covariate PROB suggested that this covariate was not of significance to the model. This statement is supported by the fact that the corresponding AIC and AICR values decreased with its exclusion. As well, the parameter estimates and associated standard errors indicated that the coefficient for PROB was not significantly different from zero. From a medical perspective the conclusion that the covariate PROB is not an important predictor of smoking behavior suggests that poor outcomes during a previous pregnancy do not seem to influence a womans smoking behavior during subsequent pregnancies. Although unfortunate, this does seem reasonable.

We now discuss the results of fitting the ROBMS-GLMM and MS-GLMM with the covariates AGE and BIRTHWEIGHT. A number of preliminary fits suggested that $c = 3.0$ and $k = 2.46$ were appropriate values for the tuning constants. Individual leverage weighting was chosen since there are few observations per individual and we performed prediction downweighting (v_{ij}).

Table 7.8: Parameter estimates (standard errors) for the Pregnancy data, $*k = 2.46, c = 3.0$.

	Model	
Variable	**MS-GLMM**	**ROBMS-GLMM***
INTERCEPT	4.85(1.02)	6.49(.36)
AGE	-.48(.16)	-.59(.22)
BIRTHWEIGHT	-.65(.29)	-1.03(.06)
α	-.05(.52)	-.94(.97)

We begin by considering the weights assigned by the ROBMS-GLMM. All of the leverage weights (w_{ij}) were larger than .95. However, upon examining the prediction downweights (v_{ij}) we found that there were 9 women for which the prediction downweights on their second (and final) pregnancy were close to or less than .5 as shown in Table 7.9. Re-examination of the data for these women revealed that in all 9 cases these women had given up smoking after their first pregnancy (although the outcome of this first pregnancy was fine). As well, the birth weight for their second pregnancy was in fact lower than that of their first. The smaller the weight in Table 7.9, the lower the birth weight of the second pregnancy. Having low birth weights associated with smoking cessation is not consistent with medical findings [13] which have shown that reducing smoking increases birth weight. Such behavior is also not consistent with the majority of the data which is well fitted by the ROBMS-GLMM, and consequently explains why such observations are assigned small weights. Our ROBMS-GLMM has been successful in identifying this aberrant behavior.

Table 7.9: Weights assigned to the Pregnancy data by the ROBMS-GLMM.

Observation		
Woman	*Pregnancy*	v_{ij}
135	2	.47
158	2	.47
164	2	.44
169	2	.22
210	2	.38
220	2	.56
238	2	.52
245	2	.48
288	2	.51

The present robust analysis is an example of a situation where the outlying data may well be the most interesting part of the data set. Perhaps some women who give up smoking after their first child inadvertently pursue some other behavior associated with a decrease in birth rate. Clearly these women are worthy of further investigation.

We now look in detail at the MS-GLMM and ROBMS-GLMM parameter estimates (standard errors were computed from the inverse of the Hessian matrix) shown in Table 7.8. Their moderate differences are no doubt a result of the modest amount of contamination (3% of the women) that has been detected and properly corrected for by the ROBMS-GLMM. Both models yield estimated marginal log odds for AGE and BIRTHWEIGHT that make it reasonable to conclude that the probability of smoking is higher among younger women, and those who have low birth weight pregnancies. These conclusions are consistent with the findings of many other medical researchers [24], [13]. It is also worthwhile comparing the estimates of α. The MS-GLMM yields $\hat{\sigma} = \exp(\hat{\alpha}) = \exp(-.05) = .95$ which can be interpreted as the variance between women. Notice that the ROBMS-GLMM, having downweighted women not consistent with the model, yields $\hat{\sigma} = \exp(-.94) = .39$. That is, the ROBMS-GLMM suggests that the women well fitted by the model are not as different as the MS-GLMM would originally have us believe.

Both applications discussed in this chapter demonstrate the additional and very insightful information that is gained from fitting ROBMS-GLMMs. As part of a complete analysis we therefore recommend fitting both ROBMS-GLMMs and MS-GLMMs. When they are in agreement we have additional confidence in our results, when is disagreement further

investigation is essential in order to hope to draw valid conclusions.

Chapter 8

Conclusions

In this final chapter we provide a summary of the results presented in this thesis and then outline some possible directions for future research.

8.1 Summary

In this thesis we have derived, demonstrated and recommended the ROBMS-GLMM as a powerful method for modelling longitudinal data. This model is useful in a variety of disciplines, biology and medicine being just two of many examples. Attention was focussed on biomedical applications where longitudinal *binary* data arise naturally and make the ROBMS-GLMM achieve a number of very useful objectives.

There have been a number of attempts at modelling longitudinal data, all of which are complicated by the necessity to deal appropriately with the correlation that exists between observations made on the same individual. Previous methods have been successful in providing either population-averaged *or* individual-specific inference but have been unable to provide both simultaneously. At the same time there is the additional requirement to simultaneously handle time-dependent and time-independent covariates. As we have shown, the ROBMS-GLMM can meet both of these objectives.

In many biomedical applications, the primary objective of the analysis is to make inference regarding the mean response as a function of covariates. Implicitly, this means that even if random effects are a necessary part of the model, they are usually regarded as a nuisance factor. For instance, in the LEI study we include a random effect to account for the

variance *between* trainees. However, we are not overly concerned with what this variance actually is, but rather include it to ensure we are making valid inferences about the other parameters such as PROPLIFT, for example. Often the motivation for including random effects in models like the ROBMS-GLMM is simply to account for correlation among an individual's observations. The ROBMS-GLMM allows us to include necessary random effects while at the same time yielding the desired population-averaged interpretations of the parameters of interest. Even with a fairly simple random effects assumption involving a small number of parameters, the ROBMS-GLMM provides a rich class of models for the association among observations from the same individual. This is not the case with many likelihood based methods.

In many real life problems a significant portion of the data may be contaminated, up to 10% in biomedical studies, for example. Such contamination is a fact of life and in being able to deal with it the ROBMS-GLMM is more durable than than its predecessors. Not only does it possess the ability to check that likelihood estimation has not been unduly influenced by a few extreme observations but it is also able to identify and analyze the *good* and applicable portion of the data. That is, if outliers are present, or model assumptions are violated, the ROBMS-GLMM estimates characteristics associated with the bulk of the data, consequently reducing the influence of aberrant observations. It can deal with outlying observations in both the response and covariates and at the same time is less sensitive to violations of the random effects distributional assumptions. Robust inference procedures like the ROBMS-GLMM are now an essential part of the statistician's tool box, particularly for longitudinal data where model assumptions may be difficult to check directly.

The ROBMS-GLMM functions as an invaluable aid in data screening and preliminary analyses. It performs similarly to the MS-GLMM when there is no contamination but does substantially better when there is contamination present. When the two models are in agreement we can place additional confidence in our results. When the models disagree more investigation is required. Perhaps the greatest strength of the ROBMS-GLMM is its ability to facilitate such investigation by providing us with additional information concerning the fit of the data to the model. This information, provided in the form of easily interpretable weights, was previously unavailable with similar but non-robust procedures. By way of example, in the smoking habits application, the bulk of the data supported the generally accepted conclusion that smoking decreases birth weight. However, the data that

were identified by the ROBMS-GLMM as *contamination* are in fact the most interesting. These are women who had stopped smoking after their first pregnancy and yet had subsequent babies of significantly lower birth weight. It may be possible to reach some important conclusions from the data for these women with further study.

The derivation of the asymptotic properties of the ROBMS-GLMM along with the robust model selection strategies, provides us with the tools necessary to achieve robust inference for longitudinal data. User friendly *R*-routines have been designed to make these tools readily available to statisticians and analysts alike.

Fitting both ROBMS-GLMMs and MS-GLMMs is difficult. Both require iterative solutions which are dependent on the selection of reasonable starting values. While considerable effort was expended to obtain good starting values, accurate approximations and effective maximization strategies, these methods remain computationally intensive (although this is masked somewhat by the *R*-routines). We were prevented from implementing our ROBMS-GLMM in *Splus* due to its well known problems with memory allocation. Although high speed computing is now easily accessible more work needs to be done to make the fitting of models like these more computationally efficient. After all, robust procedures will almost certainly play an increasingly important role in the analysis of longitudinal data.

8.2 Future Directions

Currently, the limiting factor in the use of the ROBMS-GLMM is its computational inefficiency. After all, the availability of high speed computing notwithstanding, high speed statistical analysis is still not available, at least not in the context of longitudinal data analysis. Although realizable optimization procedures make it possible to analyze data which was not possible even a few years ago, there is still much work to be done. Specifically, the amount of approximation should be lessened and more efficient paths to the solution should be found. Preliminary investigations suggest that new implementations of the EM algorithm proposed by Booth and Hobert [6] may be very useful. These methods use random sampling to construct Monte Carlo approximations at the E-step and can be considerably more efficient than those based on Markov chain Monte Carlo algorithms.

With new model fitting strategies in place it will be possible to more easily implement ROBMS-GLMMs which include multiple random effects. As well, we can then fine tune

the algorithm to handle other types of longitudinal data.

Appendix A

Program Listings

A.1 rob-like.c

```
/* ------------------------------------------------------------ */
/* rob_like.c (to be used with the nlm optimizer in R           */
/* Computes the Robustified Log Likelihood - J.Mills 16/12/99 */
/* ------------------------------------------------------------ */
/* This function takes as arguments the following:              */
/* S_id = The vector detailing which observations               */
/*        correspond to the various individuals                 */
/* S_y = The vector of responses (0s and 1s)                    */
/* S_x = The X matrix of covariates                             */
/* S_beta = The starting value for the vector beta              */
/*          (returns the maximized likelihood estimates)        */
/* S_epsilon = The epsilon to be used for Huber's Least         */
/*             favorable density.                               */
/*             (epsilon = 0 yields the MLE)                     */
/* S_lw = The leverage weight flag, 0 for weights all 1,        */
/*        1 for weights calculated based on the hat matrix.     */
/* S_pr = The prediction weight flag, 0 for no weighting,       */
/*        1 for reweighting, 2 for downweighting                */
/* S_eta =                                                      */
/* S_ints = The vector which includes:                          */
/*                      p=dim(beta)                             */
/*                      q=dim(alpha)                            */
/*                      nobs=total number of observations       */
/*                      nclust= number of individuals           */
/* S_logL = Returns the value of the maximized log likelihood */
/* S_indlike = Returns the individual log likelihoods values  */
/* S_flag =                                                     */
/* S_c = Tuning constant c for prediction weighting             */
/* S_print = The printing flag, 0 for no printing, 1 for        */
/*           printing.                                          */
/* ------------------------------------------------------------ */

#include "chanmatstruct.h"
#include "chanmatfuns.h"

/* Common Block Declarations */
struct {
```

```
    MATRIX **Y, **N, **X, **Z, **Eta, *gamma, *E, **Mu;
    MATRIX **Muinit;
    MATRIX **LWGHT, **PWGHT, **Lbda, **Hat;
    int c, jind, kind, ppar, qpar;
    double  HLF_k, HLF_eps, cflag;
} com1;

#define glob com1
#define abs(i) (i)<0 ? -(i) : (i)

extern double      matmaxabs(), antilogit(), logit(), get_HLF_k();
extern double      likelihood_();
extern double      likelihlf_();
extern double      get_lambda();
extern double      get_pweight();
extern double      HLF_f();
extern void        convol();
extern int         quad_();

extern MATRIX  *matantilogit(), *matxdiagasvec(), *luinv();

void rob_like( S_id, S_y, S_x, S_beta, S_betainit,S_z, S_alpha,
               S_epsilon, S_lw, S_pr, S_eta, S_ints, S_logL, S_indlike,
               S_flag, S_c, S_print )

  double  *S_id, *S_y, *S_x, *S_beta, *S_betainit, *S_z, *S_alpha;
  double  *S_epsilon, *S_eta, *S_logL, *S_indlike, *S_c;
  int     *S_ints, *S_flag, *S_print, *S_lw, *S_pr;
{

  double  *lbd;
  double  *ubd;
  double  error=(float)1e-6;
  double  a,b,cutl,cutu;
  double  loglike, like_i;
  double  answer, results[8];
  double  w_i;
  MATRIX  *idin, *yin, *xin, *zin, *etain;
  MATRIX  *beta, *alpha, *betainit, *indlike;
  MATRIX  *mu, *eta, *hatin, *semi, *ILWGHT;
  MATRIX  *HATi, *SEMIi, *redx;
  MATRIX  *lp, *lpinit;
  MATRIX  *indx;

  double  sumweight, mweight;
  double  *z, *w;
  double  HLF_eps, HLF_k, lambdai;
  double  y_i;
  double  eta_i, gi, ps, pm, p_q_s, h_i;
  double  wind_i, weight_i, wov_i, x_c;
  double  mu_i;

  int     i, j, k, nobs, nclust, p, q, one, count, s, Ni;
  int     c, lwflag, prflag, printflag;
  int     flagger;
  int     npots, m, checker;

  FILE    *wfile, wfile2;

  /* ----------------------- */
```

```
/* Parameter Initialization */
/* ------------------------ */
one = 1;

p = *(S_ints+0);
glob.ppar=p;

q = *(S_ints+1);
glob.qpar=q;

nobs = *(S_ints+2);

nclust = *(S_ints+3);

glob.HLF_eps = *S_epsilon;

lwflag=*S_lw;

prflag=*S_pr;

printflag=*S_print;

glob.cflag=*S_c;

/* ------------------------------------------------- */
/* Choose between ROBMS-GLMM and MS-GLMM (non-robust) */
/* ------------------------------------------------- */
if(glob.HLF_eps==0 ) { /* NON-ROBUST */ glob.HLF_k=1000.; }
else {  glob.HLF_k = get_HLF_k(glob.HLF_eps); }

/* ---------------------------- */
/* Transfering information for S */
/* ---------------------------- */
from_S( S_beta, &p, &one, beta );
make_permanent( beta );

from_S( S_betainit, &p, &one, betainit );
make_permanent( betainit );

from_S( S_alpha, &q, &one, alpha );
make_permanent( alpha );

from_S(S_indlike, &nclust, &one, indlike );

from_S( S_id, &nobs, &one, idin );
from_S( S_y, &nobs, &one, yin );
from_S( S_x, &nobs, &p, xin );
make_permanent(xin);

from_S( S_z, &nobs, &q, zin );
from_S( S_eta, &nobs, &one, etain );

glob.Y = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
glob.X = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
glob.Z = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
glob.Eta = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
glob.Mu = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
glob.Muinit = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix));
glob.Lbda = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
glob.PWGHT = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
```

```
glob.LWGHT = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );

make_permanent( idin );
split( yin, idin, glob.Y );
split( xin, idin, glob.X );
split( zin, idin, glob.Z );
split( etain, idin, glob.Eta );

destroy_matrix( idin );
destroy_matrix( yin );
destroy_matrix( xin );
destroy_matrix( zin );
destroy_matrix(etain);

for( c=0; c<nclust; c++ ){
  make_permanent( glob.Y[c] );
  make_permanent( glob.X[c] );
  make_permanent( glob.Z[c] );
  make_permanent( glob.Eta[c] );
  glob.Mu[c] = create_matrix( glob.Y[c]->nrows, 1, EPHEMERAL );
  glob.Muinit[c] = create_matrix( glob.Y[c]->nrows, 1, EPHEMERAL );
  glob.Lbda[c] = create_matrix( glob.Y[c]->nrows, 1, EPHEMERAL );
  glob.PWGHT[c] = create_matrix( glob.Y[c]->nrows, 1, EPHEMERAL);
  glob.LWGHT[c] = create_matrix( glob.Y[c]->nrows, 1, EPHEMERAL);
}

indx= create_matrix(nclust,3,EPHEMERAL);

/*Individual weighting preliminaries*/
for(c=0; c<nclust; c++){
  x_c= MEL(glob.X[c],1,1);
  MEL(indx,c,1)=x_c;
  x_c= MEL(glob.X[c],1,2);
  MEL(indx,c,2)=x_c;
  MEL(indx,c,0)=(float)1.;
}

make_permanent(indx);
semi=luinv(matmult(transp(indx),indx));
hatin = matmult( indx, matmult(semi,transp(indx)));
destroy_matrix(semi);
destroy_matrix(indx);


/*-------------------------------*/
/* Preliminaries                 */
/*-------------------------------*/
    sumweight=0.; /* sums the prediction downweights*/


for( glob.c=0; glob.c<nclust; glob.c++ ){/* loop thru individuals*/
    Ni = glob.Y[glob.c]->nrows;
    lp = matmult( glob.X[glob.c], beta );
    lpinit = matmult( glob.X[glob.c], betainit);
   /* Looping through observations on individual c */
    h_i=MEL(hatin,glob.c,glob.c);
      for( i=0; i<Ni; i++ ){

        /*Individual Leverage weights*/
         MEL(glob.LWGHT[glob.c],i,0) =(float)1.;
```

```
          if(lwflag==1){
            if (h_i>=0 && h_i<=1) { /* makes sure weights well-defined */
          MEL(glob.LWGHT[glob.c],i,0) = sqrt(1-h_i);}
             }


          MEL( glob.Mu[glob.c], i, 0 ) = antilogit( MEL(lp,i,0) );


         MEL( glob.Muinit[glob.c], i, 0 ) = antilogit( MEL(lpinit,i,0) );
         /*If any of these are 0 or 1, can't solve convolution!!! */
         /* Basing them on betaint, stop algorithm driving to 0*/
         if(glob.HLF_eps!=0){
          if(MEL(glob.Muinit[glob.c],i,0)>=(float)1.) {
             MEL(glob.LWGHT[glob.c],i,0) = (float)0.; }
        if(MEL(glob.Muinit[glob.c],i,0)<=(float)0.) {
           MEL(glob.LWGHT[glob.c],i,0) = (float)0.;}
         }
          if(prflag==0){
            MEL(glob.PWGHT[glob.c],i,0)=1.;}

/* Calculate fixed prediction weights */
          if(prflag==1){/*Reweighting*/
          y_i = MEL( glob.Y[glob.c], i, 0 );
          mu_i = MEL( glob.Muinit[glob.c], i, 0 );
          MEL(glob.PWGHT[glob.c],i,0)=1.;
          MEL( glob.Lbda[glob.c], i, 0 ) = get_lambda(glob.c,i,y_i,mu_i);}

          if(prflag==2){/*Downweighting*/
          y_i = MEL( glob.Y[glob.c], i, 0 );
          mu_i = MEL( glob.Muinit[glob.c], i, 0 );
          MEL(glob.PWGHT[glob.c],i,0 )=get_pweight(glob.c,i,y_i,mu_i);
          sumweight+=MEL(glob.PWGHT[glob.c],i,0);}

       } /* end loop thru observations */
}/* end loop thru individuals */

/*Rescale prediction weights and print*/
mweight=sumweight/nobs;
if(printflag==1){wfile=fopen("PWGHTS.out","a");}

for( glob.c=0; glob.c<nclust; glob.c++ ){
     Ni = glob.Y[glob.c]->nrows;
     for( i=0; i<Ni; i++ ){
if(prflag==2){
  MEL(glob.PWGHT[glob.c],i,0) = MEL(glob.PWGHT[glob.c],i,0)/mweight;
  if(printflag==1){
   fprintf( wfile, "ind.= %i, obs.= %i, downweight= %f \n", glob.c, i,
            MEL(glob.PWGHT[glob.c],i,0));}}

  if(prflag==1){
    if(printflag==1){
  fprintf( wfile, "ind.=%i, obs=%i, reweight = %f\n",
           glob.c, i, MEL(glob.Lbda[glob.c],i,0));}}
     }
destroy_matrix(glob.Muinit[glob.c]);
}
if(printflag==1) {fclose(wfile);}
```

```
 /* ---------------------------- */
 /* Begin Likelihood Maximization */
 /* ---------------------------- */

loglike = 0.0;
for( glob.c=0; glob.c<nclust; glob.c++ ){/* loop thru individuals*/
     Ni = glob.Y[glob.c]->nrows;

     /*Observation leverage weight initializations */
     ILWGHT=create_matrix(Ni,1,EPHEMERAL );
     if(lwflag==1){
     HATi=create_matrix(Ni,Ni,EPHEMERAL);

     /* With no time-independent don't need these */

     /*SEMIi=create_matrix(p-1,p-1,EPHEMERAL);
     redx=create_matrix(Ni,p-1,EPHEMERAL);
     redx=extract_cols(glob.X[glob.c],2,p);
     make_permanent(redx);*/

     SEMIi = luinv(matmult(transp(glob.X[glob.c]),glob.X[glob.c]));
     HATi = matmult(glob.X[glob.c],matmult(SEMIi,transp(glob.X[glob.c])));
     /*destroy_matrix(redx);*/
     destroy_matrix(SEMIi);
     }

     glob.gamma = matexp(matmult(glob.Z[glob.c],alpha));
     make_permanent( glob.gamma );

     /* score and hessian for individual i*/
     like_i = 0.0;

     /* Calculation of observation leverage weights */
     for(i=0; i<Ni; i++){
        MEL(ILWGHT,i,0)=(float)1.;
        /*if(lwflag==1){
               h_i = MEL(HATi,i,i);
            if (h_i>=0 && h_i<=1) { /* makes sure weights well-defined */
        /*MEL(ILWGHT,i,0) = sqrt(1-h_i);}
          }*/
     }
       if(lwflag==1){ destroy_matrix(HATi);}

       /* Looping through observations on individual c */
       for( i=0; i<Ni; i++ ){

         eta_i = MEL( glob.Eta[glob.c], i, 0 );

         gi = MEL( glob.gamma, i, 0 );

         mu_i = MEL( glob.Mu[glob.c], i, 0 );

         /* Multiply overall and individual weights*/
         /* For LEI data we have no time-independent covariates */
         wind_i= MEL(ILWGHT,i,0);
         wov_i = MEL(glob.LWGHT[glob.c],i,0);
         weight_i = wind_i*wov_i;
         /*if(weight_i<=.4){*/
         MEL(glob.LWGHT[glob.c],i,0)= weight_i; /*}
         else{
```

```
   /*if(weight_i<=.8){*/
   /*MEL(glob.LWGHT[glob.c],i,0)=(float)1.;/*}
   else{MEL(glob.LWGHT[glob.c],i,0)=(float)1.;}
   /*store final value for if statement to follow*/
     /* MEL(glob.LWGHT[glob.c],i,0)= (float)1.;}*/

   flagger=0;

   /* Solve convolution equation */

   if(weight_i!=(float)0.){
     convol( &eta_i, mu_i, gi, glob.HLF_k, glob.HLF_eps, &flagger );}
   else{
     if(glob.HLF_eps!=0){/*robust*/
           flagger=1;}
     else{/*non-robust*/
       convol( &eta_i, mu_i, gi, glob.HLF_k, glob.HLF_eps, &flagger );}}

   if( flagger==0 ){
     MEL( glob.Eta[glob.c], i, 0 ) = eta_i;
           }else{
    if(glob.HLF_eps!=0){
              MEL(glob.LWGHT[glob.c],i,0)=(float)0.;
                    }
    else { /*MEL( glob.Eta[glob.c], i, 0 ) = eta_i;*/
             loglike = -(float)10000;
             printf("**");
             goto loop;
                    }
           }
 }/* end loop through individuals */

 /* ----------------------------------------*/
 /* Calculation numerical integration bounds*/
 /*----------------------------------------*/

 if(glob.HLF_eps==0) { /*NON-ROBUST*/
  a=gi*-5.;
  b=gi*+5.;
  ubd=&b;
  do{b=b+1.;
   ubd=&b;
  }while(likelihood_(ubd)>=(float).0000001);
  do{a=a-1.;
   lbd=&a;
  }while(likelihood_(lbd)>=(float).0000001);
}else{/*ROBUST*/
  a=gi*-5.;
  b=gi*+5.;
  do{b=b+1.;
   ubd=&b;
  }while(likelihlf_(ubd)>=(float).0000001);
  do{a=a-1.;
   lbd=&a;
  }while(likelihlf_(lbd)>=(float).0000001);
}

 /* numerically evaluate likelihood for individual c*/
 cutl=-glob.HLF_k;
 cutu=glob.HLF_k;
```

```
        like_i=0.;
        if(glob.HLF_eps==0) { /*NON-ROBUST*/
         quad_(&a,&b,results,&m,&error,&npots,&checker,likelihood_);
         like_i=results[m-1];}
        else { /*ROBUST*/
        quad_(&a,&cutl,results,&m,&error,&npots,&checker,likelihlf_);
        like_i=results[m-1];
        quad_(&cutl,&cutu,results,&m,&error,&npots,&checker,likelihood_);
        like_i+=results[m-1];
        quad_(&cutu,&b,results,&m,&error,&npots,&checker,likelihlf_);
        like_i+=results[m-1];
        like_i=(1.- glob.HLF_eps)*like_i;
                        }
        if(like_i<=(float)1e-45) { printf("**");}
        loglike = loglike + log(like_i);
        MEL( indlike, glob.c, 0 ) = log(like_i);

          if(printflag==1){
   wfile=fopen("ROBLIKE.out","a");
   fprintf(wfile,"like_i = %f, log(like_i) = %f for individual %i\n", like_i,
   log(like_i), glob.c);
   fclose(wfile);}
   destroy_matrix(glob.Mu[glob.c]);
   destroy_matrix(glob.Lbda[glob.c]);
   destroy_matrix(glob.PWGHT[glob.c]);
   destroy_matrix(glob.LWGHT[glob.c]);
    }/* c */
   destroy_matrix(ILWGHT);


    loop: loglike=-(float)1.*loglike;
    /* For use with nlm since it is a minimizer !!!! */
    *S_logL = loglike;
     to_S( indlike, S_indlike);

}/* end of routine */

/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
double get_HLF_k( eps )

  double eps;
{
  double k;

  if(eps==.001) k=2.88;
  if(eps==.002) k=2.70;
  if(eps==.005) k=2.46;
  if(eps==.01) k=2.27;
  if(eps==.02) k=2.07;
  if(eps==.05) k=1.81;
  if(eps==.1) k=1.62;
 return k;
}

/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
double get_lambda( ind,num, obs,prob)

  double obs,prob;
  int ind, num;
{
```

```
  double lbda, dev, res, ares, abdev;
     FILE    *wfile;

  /*Weights are based on Pearson Residuals*/
    res=(obs-prob)/sqrt(prob*(1-prob));

    ares=abs(res);
    if(ares<=glob.cflag){ /*Huber's Psi Function*/
         lbda=0.;
    }else { if(res<=0){lbda=-1.*(1.+(glob.cflag/res));}
    else{lbda=1.-(glob.cflag/res);}}
    return lbda;
}

/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
double get_pweight( ind,num, obs,prob)

  double obs,prob;
  int ind, num;
{
  double weight, dev, res, ares;
       FILE    *wfile;
  /*Weights are based on Pearson Residuals*/
  /*if(prob==0.) {printf("expected =0, problem!");}*/
  if(prob<=0.) {weight=0.;}
  else if(prob>=1.) {weight=0.;}
  else{
    res=(obs-prob)/sqrt(prob*(1-prob));
    ares=abs(res);
    if(ares<=glob.cflag){ /*Huber's Psi Function*/
         weight=1.;
    }else{
         weight=glob.cflag/ares;
    }
  }
 return weight;
}


/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */

double  likelihood_(x)
double  *x;
{
    double  ret_val, rl, res, lambdai;
    double  yi, etai, gi,ps, wi, ui;
    int     Ni, i;
    FILE    *wfile;
    rl=*x;
    ret_val=0.;

    Ni=glob.Y[glob.c]->nrows;
      for(i=0; i < Ni; i++){ /*loop thru observations for this individual*/
       res=0.;
       yi = MEL( glob.Y[glob.c], i, 0 );
       etai = MEL( glob.Eta[glob.c], i, 0 );
       gi = MEL( glob.gamma, i, 0 );
       wi=MEL(glob.LWGHT[glob.c],i,0);
       ui=MEL(glob.PWGHT[glob.c],i,0);
       lambdai = MEL(glob.Lbda[glob.c],i,0);
```

```
        if(wi==0.){res=0.;}
        else{
        ps = antilogit(etai+gi*r1);
        lambdai = MEL( glob.Lbda[glob.c], i, 0);
        res = ui*wi*((yi-lambdai)*(etai+gi*r1) + log(1.-ps));}
        ret_val=ret_val+res;
       } /*i*/
     ret_val= exp(ret_val)*(float)1. / sqrt((float)6.2831853071795862) *
     exp(-(r1 * r1)/(float)2.);
    return ret_val;
} /* likelihood_ */

/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */

double  likelihlf_(x)
double  *x;
{
    double  ret_val, r1, res, k, lambdai;
    double  yi, etai, gi,ps, wi, ui,test;
    int     Ni, i;
    FILE    *wfile;
    r1=*x;
    ret_val=0.;
    Ni=glob.Y[glob.c]->nrows;
    k=glob.HLF_k;
       for(i=0; i < Ni; i++){ /*loop thru observations for this individual*/
        res=0.;
        yi = MEL( glob.Y[glob.c], i, 0 );
        etai = MEL( glob.Eta[glob.c], i, 0 );
        gi = MEL( glob.gamma, i, 0 );
        wi=MEL(glob.LWGHT[glob.c],i,0);
        ui=MEL(glob.PWGHT[glob.c],i,0);
        lambdai = MEL(glob.Lbda[glob.c],i,0);
        if(wi==0.){res =0.;}
        else{

        ps = antilogit(etai+gi*r1);
        res = ui*wi*((yi-lambdai)*(etai+gi*r1) + log(1.-ps));}
        ret_val=ret_val+res;
          } /*i*/
      ret_val= exp(ret_val)*(float)1. / sqrt((float)6.2831853071795862) *
        exp(-pow(k,2)/(float)2.) * pow(abs(k/r1),pow(k,2));
      return ret_val;
} /* likelihlf_ */
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */


double antilogit( x )

  double x;
{
  double out;

  out = exp( x )/(1.0+exp( x ) );

  return out;

}
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
```

```
double logit( x )

  double x;
{
  double out;

  out = log( x ) - log( 1.-x );
  if(out<=-100.) {printf("We are in trouble , out = %f\n",out);}
  return out;

}
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */

double HLF_f( x,eta,sigma,kpar )

  double x, eta, sigma, kpar;
{
  double out;

     if( abs(x)<=kpar) {
        out = 1.;
          }else{
        out = pow((kpar/abs(x)),pow(kpar,2.));
        out = out*exp(-pow(kpar/sqrt(2.),2.)) / exp(-pow(x/sqrt(2.),2.));
        }
  return out;

}
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
```

A.2 convol.c

```
/* ---------------------------------------------------------------- */
/*  convol.c (to be used with rob_like.c)                           */
/*  Solves the deconvolution equation - J.Mills 15/12/99            */
/* ---------------------------------------------------------------- */
/*                                                                  */
/* ---------------------------------------------------------------- */
#include "f2c.h" /* required because this problem includes a
                    re-implementation of Patterson's QUAD
                    numerical integration routine, originally
                    written in fortran - must compile with -lm -lf2c*/

#include "chanmatstruct.h"
#include "chanmatfuns.h"
#include "math.h"


#define cfree free

#define ETA_TOLERANCE 1e-5
#define ETA_MAX_ITER  50
#define ETA_EPS 1e-7

extern double f1_(),f2_(),f3_(),f4_(),f5_(),f6_();
extern double f1hlf_(),f2hlf_(),f3hlf_(),f4hlf_(),f5hlf_(),f6hlf_();
extern double coneq_(), robconeq_();
extern int quad_();
extern double antilogit();
```

```
extern double zerofind();

/* Common Block Declarations */

struct {
    double HLF_eps;
    double HLF_k;
    double etai;
    double sigma;
    double mu;
    double upbd;
    double lobd;
} com;

#define com_1 com

/*Main program alias*/
void convol(eta, mu, sigma, HLF_k, HLF_eps, flagg)
  double  *eta;
  double  mu, sigma, HLF_k, HLF_eps;
  int     *flagg;
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
{
  double     upper, lower;
  double     a,b;
  double     *lbd;
  double     *ubd;
  double     cutl,cutu;
  double     epsil=(float)1e-6;
  int        npts, k, check;
  double     answr, result[8];
  double     etai, dmuidetai, new_mui, delmu, ps;
  double     dmuids, detads;
  double     d1, d2, int2, int3, int4, x;
  double     RELERR;
  int        eta_converge, s, count, NPTS, NLIMIT, IFAIL;
  int        counter;
  FILE       *wfile;
  double     root;

  /* Parameter Initialization */
  com_1.HLF_eps=HLF_eps;
  com_1.HLF_k=HLF_k;
  com_1.etai=*eta;
  com_1.sigma=sigma;


  /* Calculate bounds of integration */

  if(HLF_eps==0) { /*NON-ROBUST*/
         a=sigma*-5.;
         b=sigma*+5.;
         ubd=&b;
         do{b=b+1.;
          ubd=&b;
         }while(coneq_(ubd)>=(float).0000001);
         do{a=a-1.;
          lbd=&a;
         }while(coneq_(lbd)>=(float).0000001);
       }else{/*ROBUST*/
```

```
        a=sigma*-5.;
        b=sigma*+5.;
        do{b=b+1.;
         ubd=&b;
        }while(f1h1f_(ubd)>=(float).0000001);
        do{a=a-1.;
         lbd=&a;
        }while(f1h1f_(lbd)>=(float).0000001);
      }


com_1.upbd = b; /*upper bound of integration*/
com_1.lobd = a; /*lower bound of integration*/


/* etai = deltai in Heagerty's notation */
eta_converge = 0;
count = 0;

cutl=-HLF_k;
cutu=HLF_k;


/* Computing eta  (delta_ij)  */
/*---------------------------*/
/* We need adaptive bounds for the root finder */
com_1.mu=0.;
upper=sigma*4.;
lower=sigma*-4.;
counter=0;
/* forcing non-robust convolution */
/*com_1.HLF_eps=0;*/
if(com_1.HLF_eps==0){/*NON-ROBUST*/
        do{upper=upper+1.;
        }while(coneq_(upper)<=(float).9999999);
        do{lower=lower-1.;
        }while(coneq_(lower)>=(float)0.0000001);
        upper = -1.*lower;
}else{/*ROBUST*/
       do{upper=upper+1.;
          counter=counter + 1;
          if(counter>=15) {goto next;}
       }while(robconeq_(upper)<=(float).999933);
       next: do{lower=lower-1.;
       }while(robconeq_(lower)>=(float)0.0000001);
}

/* Now consider the appropriate mu value */
com_1.mu=mu;

if(com_1.HLF_eps==0) { /*NON-ROBUST*/
    root=zerofind(lower,upper,coneq_,1e-6);

 }   else { /*ROBUST*/
    root=zerofind(lower,upper,robconeq_,1e-6); }


  *eta =root;
  if(root== upper){
    *flagg=1;}
```

```
    if(root==lower){
      *flagg=1;}

  return;

}
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
/*     =====     Convolution Equation        ============================== */
double coneq_(delta)
double delta;
{
    double ret_val, rhs, r1, ps;
    double     a, b;
    int        npts, k, check;
    double     result[8];
    double     epsil=(float)1e-6;
    FILE       *wfile;

    a=com_1.lobd;
    b=com_1.upbd;
    com_1.etai = delta;
    quad_(&a,&b,result,&k,&epsil,&npts,&check,f1_);
    rhs=result[k-1];
    ret_val=rhs-com_1.mu;
    return ret_val;
} /* coneq_ */

/*     =====   Robust  Convolution Equation  ============================== */
double robconeq_(delta)
double delta;
{
    double ret_val, rhs, r1, ps;
    double     a,b;
    double     cutl,cutu;
    int        npts, k, check;
    double     result[8];
    double     epsil=(float)1e-6;
     FILE   *wfile;

    com_1.etai = delta;
    cutl=-com_1.HLF_k;
    cutu=com_1.HLF_k;
    a=com_1.lobd;
    b=com_1.upbd;

    quad_(&a,&cutl,result,&k,&epsil,&npts,&check,f1hlf_);
    rhs=result[k-1];
    quad_(&cutu,&b,result,&k,&epsil,&npts,&check,f1hlf_);
    rhs+=result[k-1];
    quad_(&cutl,&cutu,result,&k,&epsil,&npts,&check,f1_);
    rhs+=result[k-1];
    rhs=(1.-com_1.HLF_eps)*rhs;
    ret_val=rhs-com_1.mu;
    return ret_val;
} /* robconeq_ */


/*     =====     First Integrand Function    ============================== */
double f1_(x)
double *x;
```

```
{
    double ret_val, r1, ps;

    r1 = *x;
    ps = antilogit(com_1.etai+com_1.sigma*r1);
    ret_val = (float)1. / sqrt((float)6.2831853071795862) *
       exp(-1.*(r1 * r1)/(float)2.)*ps;
    return ret_val;
} /* f1_ */

double f1hlf_(x)
double *x;
{
    double ret_val, r1, ps;

    r1 = *x;
    ps = antilogit(com_1.etai+com_1.sigma*r1);
    ret_val = ps* (float)1. / sqrt((float)6.2831853071795862) *
       exp(-1.*(com_1.HLF_k * com_1.HLF_k)/(float)2.) *
        pow(abs(com_1.HLF_k/r1),pow(com_1.HLF_k,2.));
    return ret_val;
} /* f1hlf_ */


/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */

int quad_(a, b, result, k, epsil, npts, icheck, f)
double *a, *b, *result;
int    *k;
double *epsil;
int    *npts, *icheck;
double (*f) ();
{
    /* Initialized data */

    static int fl[7] = { 2,3,5,9,12,14,1 };
    static int fh[7] = { 2,4,8,16,17,17,0 };
    static int kl[11] = { 1,1,1,1,1,3,5,9,5,9,12 };
    static int kh[11] = { 1,2,4,8,16,3,6,17,5,9,17 };
    static int kx[8] = { 0,1,2,3,4,5,8,11 };
    static double p[305] = { (float)-.11111111111111111111,(float)
            .22540333075851662296,(float).55555555555555555556,(float)
            .00647209421402969791,(float)-.00928968790944433705,(float)
            .03950873129197971 6579,(float).10465622602646726519,(float)
            .56575625065319744 2,(float).40139741477596222291,(float)
            5.223046896961622e-5,(float)1.712103096175e-4,(float)
            -7.248300161538928 98e-4,(float)-7.017801099209042e-5,(float)
            .00616803678724497 77899,(float).01700171962994026 0339,(float)
            .11154076712774300 11,(float).09292719531512453 7686,(float)
            .37889705326277359705,(float).17151190913639138079,(float)
            .77661331357103311837,(float).21915685840158749 64,(float)
            6.82166534792e-9,(float)1.2667409859336e-7,(float)
            5.95659763678371 65e-6,(float)1.392330106826e-8,(float)
            -6.629407564902392e-5,(float)-7.04395804282302e-7,(float)
            -3.45182053392 41e-8,(float)-8.14486910996e-9,(float)
            9.01875032332402340 38e-4,(float).00254478079156187441 54,(float)
            .01846885044625989313,(float).01644604985438781093 4,(float)
            .07034514257025994333,(float).03595710330712932209 7,(float)
            .16327406183113126449,(float).05697950949412335741 2,(float)
            .29750379350847292139,(float).07687962049900353104 3,(float)
```

```
.46868025635562437602,(float).093627109981264473617,(float)
.66884660674202316691,(float).10566989358023480974,(float)
.88751105686681337425,(float).11195687302095345688,(float)
3.71583e-16,(float)2.1237877e-13,(float)1.0522629388435e-9,(float)
1.748029e-15,(float)3.47571898301716e-7,(float)9.0312761725e-12,(
float)1.2558916e-14,(float)5.4591e-16,(float)
-7.2338395508691963e-6,(float)-1.69699579757977e-8,(float)
-8.54363907155e-11,(float)-1.228130093e-12,(float)-4.62334825e-14,
(float)-4.2244055e-15,(float)-8.8501e-16,(float)-4.0904e-16,(
float)1.2711187964238806027e-4,(float)3.6322148184553065969e-4,(
float).002793740627780409196,(float).002579049794685688 2724,(
float).011315242452570520059,(float).006115506822117246 3397,(
float).027817125251418203419,(float).010498246909621321898,(float)
.053657141626597094849,(float).015406750466559497802,(float)
.089628843042995707499,(float).020594233915912711149,(float)
.13609206180630952284,(float).025869679327214746911,(float)
.19305946804978238813,(float).031073551111687964 88,(float)
.26024395564730524132,(float).036064432780782572 64,(float)
.33709033997521940454,(float).040715510116944318934,(float)
.42280428994795418516,(float).044914531653632197414,(float)
.51638197305415897244,(float).048564330406673198716,(float)
.61664067580126965307,(float).051583253952048458777,(float)
.72225017797817568492,(float).053905499335266063927,(float)
.83176474844779253501,(float).055481404356559363988,(float)
.94365568695340721002,(float).056277699831254301273,(float)
1.041098e-16,(float)2.49472054598e-11,(float)5.5e-21,(float)
2.90412475995385e-8,(float)3.67282126e-14,(float)5.568e-19,(float)
-8.71176477376972025e-7,(float)-8.147324267441e-10,(float)
-8.830920337e-13,(float)-1.8018239e-15,(float)-7.0528e-18,(float)
-5.06e-20,(float)1.75696451084014199 61e-5,(float)
5.053609520786251762 5e-5,(float)4.012003280893167500 9e-4,(float)
3.77746646326984660 27e-4,(float).001683364681592607 4696,(float)
9.38369848542381500 79e-4,(float).004275895301592811 49,(float)
.001681142865421469 9063,(float).008504278821893867 6006,(float)
.002568764943794020 3731,(float).014628500401479628 89,(float)
.003572892783517299 6494,(float).022858485360294285 84,(float)
.004671050372114321 7474,(float).033362148441583432 91,(float)
.005843449875835639 5076,(float).046269993574238863589,(float)
.007072489995433555468,(float).061679602220407116 35,(float)
.008342838753968157 7056,(float).079659974529987579 27,(float)
.009641177729702536 6953,(float).100255100223059963 35,(float)
.010955733387837901648,(float).123486585515294730 26,(float)
.012275830560082770087,(float).149355550523164972024,(float)
.013591571009765546 79,(float).177843745635019592 62,(float)
.014893641664815182035,(float).208915066200151638 57,(float)
.016173218729577719942,(float).242516033619486362 06,(float)
.017421930159464173747,(float).278576914629901084 52,(float)
.018631848256138790186,(float).317012568908920771 91,(float)
.019795495048097499488,(float).357723357490240486 22,(float)
.020905851445812023852,(float).400596069757757107 02,(float)
.021956366305317824939,(float).445504867368067451 12,(float)
.022940964229387748761,(float).492312246466283397 85,(float)
.023854052106038540 08,(float).540869988010167667 12,(float)
.024690524744487676909,(float).591020178770111327 59,(float)
.025445769965464765813,(float).642596162168467847 62,(float)
.026115673376706097 68,(float).695423558443285956 66,(float)
.026696622927450359906,(float).749321269696516823 39,(float)
.027185513229624791819,(float).804102497288899846 07,(float)
.027579749566481873035,(float).859575768474398254,(float)
.027877251476613701609,(float).915545959916289116 29,(float)
```

```
.028076455793817246607,(float).97181535105025430566,(float)
.028176319033016602131,(float)3.326e-19,(float)1.14094770478e-12,(
float)2.952436056970351e-9,(float)5.1608328e-16,(float)
-1.10177219650597323e-7,(float)-5.8656987416475e-11,(float)
-2.3340340645e-14,(float)-1.24895e-17,(float)
2.4036202515353807 63e-6,(float)6.937936432410826717e-6,(float)
5.6003792945624240417e-5,(float)5.32752936697806 13125e-5,(float)
2.3950907556795267013e-4,(float)1.3575491094922871973e-4,(float)
6.1966197497641806982e-4,(float)2.4921240048299729402e-4,(float)
.001254385531904885 3002,(float)3.8974528447328229322e-4,(float)
.002194645504042725 4399,(float)5.5429531493037471492e-4,(float)
.003485854085109726 15,(float)7.402828042445033 3046e-4,(float)
.005168497199378999 4803,(float)9.453615168585253 8246e-4,(float)
.007278655717211384 6706,(float).00116748411742995 94077,(float)
.009848629599229840 8193,(float).00140490799565514 46427,(float)
.012907472045965932809,(float).00165611272815445 26052,(float)
.016481342421367271 24,(float).00191971297101387 24125,(float)
.020593718329137316189,(float).00219440692536383 88388,(float)
.025265540247597332 24,(float).00247895822666757 9307,(float)
.030515340497540768229,(float).00277219576459345 0994,(float)
.036359378430187867 48,(float).00307301843470257 83234,(float)
.042811783890139037259,(float).00338039799108692 03823,(float)
.049884702478705123 44,(float).00369337791702565 08183,(float)
.057588434808916940 19,(float).00401106872407502 33989,(float)
.065931563842274211999,(float).00433264096809298 28545,(float)
.074921067092924347 64,(float).00465731729975685 47773,(float)
.084562412844234959 36,(float).00498436456476553 86012,(float)
.094859641186738404 81,(float).00531308660518705 65663,(float)
.105815431664440977 14,(float).00564281810138444 41585,(float)
.117431159752658093 15,(float).00597291956550816 58049,(float)
.129706944451886094 14,(float).00630277344908575 87172,(float)
.142641689113767843 47,(float).00663178124290188 78941,(float)
.156233117327291398 95,(float).00695936140939042 29394,(float)
.170477805362598599 81,(float).00728494798055380 70639,(float)
.185371212344862586 56,(float).00760798966571905 65832,(float)
.200907709039158598 19,(float).00792794933429484 91103,(float)
.217080605881716983 6,(float).00824430376303286 80306,(float)
.233882180696239909 28,(float).00855654356130768 96192,(float)
.251303706383063397 18,(float).00886417320948249 42641,(float)
.269335478757818738 67,(float).00916671116356078 84067,(float)
.287966844637747965 4,(float).00946368999383006 52943,(float)
.307186230220885297 11,(float).00975465656363174 114611,(float)
.326981169769581520 79,(float).01003917204405684 0798,(float)
.347338334589982503 89,(float).01031681233094762 1682,(float)
.368243562288805769 59,(float).01058716790488519 7931,(float)
.389681886284813599 83,(float).01084984408933731 4099,(float)
.411637565552337458 57,(float).01110446113400692 6537,(float)
.434094114576345577 37,(float).01135065431598059 6602,(float)
.457034333501688509 51,(float).01158807403304395 2568,(float)
.480440338462542978 01,(float).01181638589083023 5763,(float)
.504293592081238539 83,(float).01203527078527956 263,(float)
.528574934128341123 07,(float).01224442498161198 5899,(float)
.553246612337971526 25,(float).01244356019071403 5263,(float)
.578342313373836699 93,(float).01263240364354207 8765,(float)
.603787193942384060 82,(float).01281069816387736 1967,(float)
.629577912049921769 86,(float).01297820223953739 9286,(float)
.655692658400561977 21,(float).01313469009196015 2836,(float)
.682109187931523316 82,(float).01327995174393053 065,(float)
.708804851481753318 03,(float).01341379308511009 8513,(float)
.735756627589073238 06,(float).01353603593495621 3614,(float)
```

```
            .76294115441017027278,(float).013646518102571291428,(float)
            .79033476175681880523,(float).013745093443001896632,(float)
            .81791350324074780175,(float).013831631909506428676,(float)
            .84565318851862189l3,(float).013906019601325461264,(float)
            .87352941562769803314,(float).013968158806516938516,(float)
            .90151760340188079791,(float).014017968039456608l,(float)
            .92959302395714482093,(float).014055382072649964277,(float)
            .95773083523463639678,(float).014080351962553661325,(float)
            .98590611358921753738,(float).014092845069160408355 };

    /* System generated locals */
    int    i__1, i__2;
    double r__1, r__2;

    /* Local variables */
    static double diff;
    static double  acum;
    static double work[17];
    static int    j;
    static double x, delta;
    static double pacum;
    static int    j1, j2, jh, jl, kk, ip;
    static double fncval;


/*     This quadrature program uses formulae due to T. N. L. Patterson, */
/*     Mathematics of computation, Volume 22, 1968, pages 847-856, as */
/*     modified by F. T. Krogh and W. V. Snyder, ACM Transactions on */
/*     Mathematical Software 17, 4 (December 1991) pp 457-461.  It is a */
/*     functional replacement for Algorithm 468, T. N. L. Patterson, */
/*     Communications of the ACM 16, 11 (November 1973) 694-699. */

/*     *****     Formal Arguments     ********************************** */
*/

/* Input: */
/*  A, B   Lower and upper limits of integration, respectively. */
/*  EPSIL  Relative accuracy required.  When the relative difference of */
/*         two successive formulae does not exceed EPSIL the last formula
*/
/*         computed is taken as the result. */
/*  F      A FUNCTION subprogram that evaluates the integrand at a given
*/
/*         abscissa.  F is invoked F(X).  F MUST BE MENTIONED IN AN */
/*         EXTERNAL STATEMENT IN THE CALLING PROGRAM. */
/* Output: */
/*  RESULT This array, which should be declared to have at least 8 */
/*         elements, holds the results obtained by the 1, 3, 7, etc. */
/*         point formulae.  The number of formulae computed depends on */
/*         EPSIL. */
/*  K      RESULT(K) holds the value of the integral to the specified */
/*         relative accuracy. */
/*  NPTS   Reports the number of integrand evaluations. */
/*  ICHECK On exit normally ICHECK=0.  However if convergence to the */
/*         accuracy requested is not achieved ICHECK=1 on exit. */


/*     *****     Local Variables     ********************************** */
*/
```

```
/* ACUM    is the accumulating estimate of the integral. */
/* DELTA   is B - A. */
/* DIFF    is 0.5 * DELTA. */
/* FH, FL  contain subscripts to indicate where to store integrand */
/*         samples in WORK. */
/* FNCVAL  is an integrand sample, or a sum of two symmetrically placed */
/*         integrand samples. */
/* IP      is a subscript used to index P. */
/* J       is a subscript and loop inductor. */
/* J1, J2  are bounds of the indexes at which integrand samples are */
/*         stored in WORK. */
/* JH, JL  are bounds for the loop that accumulates new function values */
/*         into the integral estimate. */
/* KH, KL  are bounds for indexes at which integrand samples are */
/*         retrieved from WORK in order to begin applying a quadrature */
/*         formula. */
/* KK      is a loop inductor and subscript. */
/* KX      is a list of bounds of subscripts into KH and KL.  KH is */
/*         indexed by K.  The set of indexes from which to retrieve */
/*         integrand samples is given by the set of bounds in KL and KH */
/*         indexed by KX(K-1)+1 to KX(K) inclusively. */
/* P       contains the coefficients necessary to the quadrature */
/*         formulae.  Their organization is described below in the */
/*         section on DATA statements. */
/* PACUM   is the previous value of ACUM. */
/* X       is the distance of the abscissa from the boundary of the */
/*         region . */
/* WORK    is space in which to store function values to be used in the */
/*         next formula. */


/*     *****     Data Statements     ********************************** */

    /* Parameter adjustments */
    --result;

    /* Function Body */

/*     In the comments below, F(K,I) refers to the function value */
/*     computed for the I'th node of the K'th formula.  The abscissae and */
/*     weights are stored in order according to the distance from the */
/*     boundary of the region, not from the center.  Since we store */
/*     1 - |abscissa|, the first "node" coefficient for each formula is */
/*     the smallest. */

/*     Corrections, nodes and weights for the 3-point formula. */

/*     Correction for F(1,1). */
/*     Node and weight for F(2,1). */

/*     Corrections, nodes and weights for the 7-point formula. */

/*     Corrections for F(1,1) and F(2,1). */
/*     Nodes and weights for F(3,1-2) */

/*     Corrections, nodes and weights for the 15-point formula. */

/*     Corrections for F(1,1), F(2,1), F(3,1-2). */
```

```
/*     Nodes and weights for F(4,1-4). */

/*     Corrections, nodes and weights for the 31-point formula. */

/*     Corrections for F(1,1), F(2,1), F(3,1-2), F(4,1-4). */
/*     Nodes and weights for F(5,1-8). */

/*     Corrections, nodes and weights for the 63-point formula. */

/*     Corrections for F(1,1), F(2,1), F(3,1-2), F(4,1-4), F(5,1-8). */
/*     Nodes and weights for F(6,1-16). */

/*     Corrections, nodes and weights for the 127-point formula. */

/*     Corrections for F(3,1), F(4,1-2), F(5,1-3), F(6,1-6). */
/*     Nodes and weights for F(7,1-32). */

/*     Corrections, nodes and weights for the 255-point formula. */

/*     Corrections for F(4,1), F(5,1), F(6,1-2), F(7,1-4). */
/*     Nodes and weights for F(8,1-64). */

/*     *****     Executable Statements     ****************************
*/

    *icheck = 0;
    delta = *b - *a;
    diff = delta * (float).5;
    ip = 1;
    jh = 0;

/*     Apply 1-point Gauss formula (Midpoint rule). */

    r__1 = *a + diff;
    fncval = (*f)(&r__1);
/*     Don't write '0.5*(b+a)' above if the radix of arithmetic isn't 2.
*/
    *npts = 1;
    work[0] = fncval;
    acum = fncval * delta;
    result[1] = acum;

    for (*k = 2; *k <= 8; ++(*k)) {

/*       Go on to the next formula. */

        pacum = acum;
        acum = (float)0.;

/*       Compute contribution to current estimate due to function */
/*       values used in previous formulae. */

        i__1 = kx[*k - 1];
        for (kk = kx[*k - 2] + 1; kk <= i__1; ++kk) {
            i__2 = kh[kk - 1];
            for (j = kl[kk - 1]; j <= i__2; ++j) {
                acum += (double) (p[ip - 1] * work[j - 1]);
                ++ip;
/* L10: */
            }
```

```
/* L20: */
        }

/*       Compute contribution from new function values. */

        jl = jh + 1;
        jh = jl + jl - 1;
        jl = fl[*k - 2];
        j2 = fh[*k - 2];
        i__1 = jh;
        for (j = jl; j <= i__1; ++j) {
            x = p[ip - 1] * diff;
            r__1 = *a + x;
            r__2 = *b - x;
            fncval = (*f)(&r__1) + (*f)(&r__2);
            *npts += 2;
            acum += (double) (p[ip] * fncval);
            if (jl <= j2) {
                work[jl - 1] = fncval;
                ++jl;
            }
            ip += 2;
/* L30: */
        }
        acum = (double) diff * acum + pacum * .5;
        result[*k] = acum;
        if ((r__1 = result[*k] - result[*k - 1], dabs(r__1)) <= (r__2 = *
                epsil * result[*k], dabs(r__2))) {
            goto L50;
        }
/* L40: */
    }
    *icheck = 1;
    *k = 8;
L50:
    return 0;
} /* quad_ */
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=- */
/*
 ************************************************************************
 *                          C math library
 * function ZEROIN - obtain a function zero within the given range
 *
 * Input
 *     double zeroin(ax,bx,f,tol)
 *     double ax;                       Root will be seeked for within
 *     double bx;                       a range [ax,bx]
 *     double (*f)(double x);           Name of the function whose zero
 *                                      will be seeked for
 *     double tol;                      Acceptable tolerance for the root
 *                                      value.
 *                                      May be specified as 0.0 to cause
 *                                      the program to find the root as
 *                                      accurate as possible
 *
 * Output
 *     Zeroin returns an estimate for the root with accuracy
 *     4*EPSILON*abs(x) + tol
 *
 * Algorithm
```

```
 *     G.Forsythe, M.Malcolm, C.Moler, Computer methods for mathematical
 *     computations. M., Mir, 1980, p.180 of the Russian edition
 *
 *     The function makes use of the bissection procedure combined with
 *     the linear or quadric inverse interpolation.
 *     At every step program operates on three abscissae - a, b, and c.
 *     b - the last and the best approximation to the root
 *     a - the last but one approximation
 *     c - the last but one or even earlier approximation than a that
 *             1) |f(b)| <= |f(c)|
 *             2) f(b) and f(c) have opposite signs, i.e. b and c confine
 *                the root
 *     At every step Zeroin selects one of the two new approximations, the
 *     former being obtained by the bissection procedure and the latter
 *     resulting in the interpolation (if a,b, and c are all different
 *     the quadric interpolation is utilized, otherwise the linear one).
 *     If the latter (i.e. obtained by the interpolation) point is
 *     reasonable (i.e. lies within the current interval [b,c] not being
 *     too close to the boundaries) it is accepted. The bissection result
 *     is used in the other case. Therefore, the range of uncertainty is
 *     ensured to be reduced at least by the factor 1.6
 *
 ************************************************************************
 */

#include "math.h"
#include<float.h>
#define EPSILON DBL_EPSILON

double zerofind(ax,bx,f,tol)              /* An estimate to the root      */
double ax;                               /* Left border | of the range   */
double bx;                               /* Right border| the root is seeked*/
double (*f)(double x);                   /* Function under investigation */
double tol;                              /* Acceptable tolerance         */
{
  double a,b,c;                          /* Abscissae, descr. see above  */
  double fa;                             /* f(a)                         */
  double fb;                             /* f(b)                         */
  double fc;                             /* f(c)                         */

  a = ax;  b = bx;  fa = (*f)(a);  fb = (*f)(b);
  c = a;   fc = fa;

  for(;;)              /* Main iteration loop  */
  {
    double prev_step = b-a;              /* Distance from the last but one*/
                                         /* to the last approximation    */
    double tol_act;                      /* Actual tolerance             */
    double p;                            /* Interpolation step is calcu- */
    double q;                            /* lated in the form p/q; divi- */
                                         /* sion operations is delayed   */
                                         /* until the last moment        */
    double new_step;                      /* Step at this iteration       */

    if( fabs(fc) < fabs(fb) )
    {                                    /* Swap data for b to be the    */
       a = b;  b = c;  c = a;            /* best approximation           */
       fa=fb;  fb=fc;  fc=fa;
    }
    tol_act = 2*EPSILON*fabs(b) + tol/2;
```

```
    new_step = (c-b)/2;

    if( fabs(new_step) <= tol_act || fb == (double)0 )
      return b;                                  /* Acceptable approx. is found  */

                        /* Decide if the interpolation can be tried     */
    if( fabs(prev_step) >= tol_act     /* If prev_step was large enough*/
       && fabs(fa) > fabs(fb) )        /* and was in true direction,   */
    {                                  /* Interpolatiom may be tried   */
       register double t1,cb,t2;
       cb = c-b;
       if( a==c )                      /* If we have only two distinct */
       {                               /* points linear interpolation  */
         t1 = fb/fa;                   /* can only be applied          */
         p = cb*t1;
         q = 1.0 - t1;
       }
       else                            /* Quadric inverse interpolation*/
       {
         q = fa/fc;  t1 = fb/fc;  t2 = fb/fa;
         p = t2 * ( cb*q*(q-t1) - (b-a)*(t1-1.0) );
         q = (q-1.0) * (t1-1.0) * (t2-1.0);
       }
       if( p>(double)0 )               /* p was calculated with the op-*/
         q = -q;                       /* posite sign; make p positive */
       else                            /* and assign possible minus to */
         p = -p;                       /* q                            */

       if( p < (0.75*cb*q-fabs(tol_act*q)/2)   /* If b+p/q falls in [b,c]*/
           && p < fabs(prev_step*q/2) )        /* and isn't too large  */
         new_step = p/q;                       /* it is accepted       */
                                       /* If p/q is too large then the */
                                       /* bissection procedure can     */
                                       /* reduce [b,c] range to more   */
                                       /* extent                       */
    }

    if( fabs(new_step) < tol_act )     /* Adjust the step to be not less*/
      if( new_step > (double)0 )       /* than tolerance               */
       new_step = tol_act;
      else
       new_step = -tol_act;

    a = b;  fa = fb;                   /* Save the previous approx.    */
    b += new_step;  fb = (*f)(b);      /* Do step to a new approxim.   */
    if( (fb > 0 && fc > 0) || (fb < 0 && fc < 0) )
    {                                  /* Adjust c for it to have a sign*/
      c = a;  fc = fa;                  /* opposite to that of b       */
    }
  }

}
```

A.3 sim-data.c

```
/* ---------------------------------------------------------------- */
/*                                                                  */
/*  Simulate Data From a Marginalized                               */
/*  Logistic-Normal Model                                           */
```

```
/*  ( Clustered Binary (Binomial) Data )                              */
/*                                                                    */
/*                                                J. Mills            */
/*                                                17/01/00            */
/*                                                                    */
/* ------------------------------------------------------------------ */

#include "chanmatstruct.h"
#include "chanmatfuns.h"

#define cfree free

#define HALFSTEP 1
#define PI 3.1415927
#define RIDGE 1e-1
#define NSAFE 15

extern double  matmaxabs(), antilogit(), logit();
extern void    convol();

extern MATRIX  *matantilogit(), *matxdiagasvec(), *luinv();

void sim_data( S_id, S_y, S_n, S_Pr, S_x, S_beta, S_z, S_ints,
               S_alpha, S_eta)

  double  *S_id, *S_y, *S_n, *S_Pr, *S_x, *S_beta, *S_z, *S_alpha, *S_eta;
  integer *S_ints;
{
  MATRIX  **Y, **N, **X, **Z, **Eta, **Mu, **Pr;
  MATRIX  *idin, *yin, *nin, *prin, *xin, *zin, *etain;
  MATRIX  *beta, *alpha, *gamma;
  MATRIX  *mu, *eta, *H, *Hi, *U, *Ui;
  MATRIX  *Di, *E;
  MATRIX  *delta;
  MATRIX  *beta_0, *mu_0, *lp;
  double  *z, *w;
  double  dmax, tolerance, lambda;
  double  y_i, n_i;
  double  eta_i, gi, ps, pm, p_q_s;
  double   log_like, logPi, Pi_s, logPi_s, logPi_indep;
  double  mu_i, detadb, detads, d11, d12, d22;
  double  HLF_k, HLF_eps;
  double  dEta_dTheta_j, dEta_dTheta_k, d2Eta_dTheta_jk, step;
  int     c, i, j, k, nobs, nclust, p, q, r, one, count, s, Ni;
  int     maxiter, flag;
  int     flagg;
  /*FILE    *wfile;*/

  one = 1;

  p = *(S_ints+0);
  q = *(S_ints+1);
  nobs = *(S_ints+2);
  nclust = *(S_ints+3);
  flagg = *(S_ints+4);

  from_S( S_beta, &p, &one, beta );
  make_permanent( beta );

  from_S( S_alpha, &q, &one, alpha );
```

```
make_permanent( alpha );

from_S( S_id, &nobs, &one, idin );
from_S( S_n, &nobs, &one, nin );
from_S( S_Pr, &nobs, &one, prin );
from_S( S_x, &nobs, &p, xin );
from_S( S_z, &nobs, &q, zin );
from_S( S_eta, &nobs, &one, etain );

N = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
X = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
Z = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
Eta = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
Mu = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
Pr = (MATRIX **)calloc( nclust, (unsigned)sizeof( struct matrix) );
make_permanent( idin );

split( nin, idin, N );
split( xin, idin, X );
split( zin, idin, Z );
split( etain, idin, Eta );
split( prin, idin, Pr);

destroy_matrix( idin );

for( c=0; c<nclust; c++ ){
  make_permanent( N[c] );
  make_permanent( X[c] );
  make_permanent( Z[c] );
  make_permanent( Eta[c] );
  make_permanent( Pr[c] );
  Mu[c] = create_matrix( Pr[c]->nrows, 1, EPHEMERAL );
}

HLF_k=1000.;
HLF_eps=0.;
  for( c=0; c<nclust; c++ ){

    Ni = Pr[c]->nrows;

      gamma = matexp( matmult( Z[c], alpha ) );

      make_permanent( gamma );


      lp = matmult( X[c], beta );
      for( i=0; i<Ni; i++ ) MEL( Mu[c], i, 0 ) = antilogit( MEL(lp,i,0) );
      destroy_matrix( lp );

      /* update deconvolution calcs */


      for( i=0; i<Ni; i++ ){

        eta_i = MEL( Eta[c], i, 0 );

        gi = MEL( gamma, i, 0 );

        mu_i = MEL( Mu[c], i, 0 );
```

```
          flag=0;

          convol( &eta_i, mu_i, gi, HLF_k, HLF_eps, &flag );
          if( flag==0 ){
            MEL( Eta[c], i, 0 ) = eta_i;
            ps= MEL(Pr[c],i,0);
            MEL( Pr[c], i, 0) = antilogit(eta_i+(ps * gi));
          }else{
           *(S_ints + 4) = flag;
          }

        }/* i */

            eta_i = MEL( Eta[c], i, 0 );

            gi = MEL( gamma, i, 0 );

    }/* c */


      destroy_matrix( gamma );


  to_S( beta, S_beta );

  to_S( alpha, S_alpha );

      count = 0;
    for(c=0;c<nclust;c++){
      Ni = Mu[c]->nrows;
      for(i=0;i<Ni;i++){
        *(S_eta + count ) = MEL( Eta[c], i, 0 );
        *(S_Pr + count ) = MEL(Pr[c],i,0);
        count++;
      }
    }
}/* end of routine */
```

A.4 sim-data.q

```
sim.data<-function(mix, corr,ni,nclust)
{
# Simulation Study for Thesis - J.E. Mills
# Simulate data from a Random Intercepts logistic-normal model
# or variants there of.

# Initializations:
#-----------------
        beta<-c(-2.,3.,1.)
        x<-gen.X.matrix(1,1,ni,nclust)
        sigma<-1.
        alpha<-log(sigma)
        p<-length(beta)
        id<-rep(1:nclust,rep(ni,nclust))
        if( ncol(x)!=p ){
          stop("x and beta do not conform!") }
        nobs <- nclust*ni
        y <-length(nobs)
```

```
        n <-rep(1,length(y))
        q <- length(alpha)
        z <-rep(1,length(y))
        if(q>1) stop("cannot deal with multidimensional alpha yet")
        if( is.null(ncol(z)) ){
          if( q>1) stop("z non-matrix but len(alpha)>1 !")
        }else{
          if( ncol(z)!=q ) stop("z and alpha do not conform!")
        }
        if( is.null( dimnames(x)[[2]] ) ){
                xnames<-paste( rep("beta",p), as.character(c(1:p)) )
        }else{
                xnames<-dimnames(x)[[2]]
        }
        if( is.null( dimnames(z)[[2]] ) ){
                znames<-paste( rep("alpha",q), as.character(c(1:q)) )
        }else{
                znames<-dimnames(z)[[2]]
        }
        flag<-0

# Generate random effects:
#-------------------------

# N(0,1):
        Pr<-rep(rnorm(nclust),rep(ni,nclust))

# (1-E)N(0,1)+EN(0,9):
if(mix==1){ #take E = .05
 con<-3
# Randomly select individuals to be contaminated -
 ind<-round(sort(runif(3,1,nclust)))
 for( i in 1:con){
      contam<-rnorm(1,mean=0,sd=9)
      first<-(ind[i]-1)*ni + 1
      last<-ind[i]*ni
      for (k in first:last) {
      Pr[k]<-contam }
    }
}

# Call C Program:
#----------------

        int.parms<-c(p,q,nobs,nclust,flag)
        eta <- as.vector( x\%*\%beta)
#
        z <- .C("sim_data",
                as.double(id),
                as.double(y),
                as.double(n),
                Pr=as.double(Pr),
                as.double(x),
                beta = as.double(beta),
                as.double(z),
                int.parms=as.integer(int.parms),
                alpha = as.double(alpha),
                eta = as.double(eta))

# On return from "C" Pr contains P(Yij=1|zi,Xij)
```

```
        beta<-as.vector(z[["beta"]])
        Pr<-as.vector(z[["Pr"]])
        y<-rbinom(length(Pr),1,Pr)
        names(beta)<-xnames
        alpha<-as.vector(z[["alpha"]])
        names(alpha)<-znames
        int.parms<-as.vector(z[["int.parms"]])
        flag<-int.parms[5]
        if(flag!=0) stop("Simulation did not complete successfully")
#
out <- list(    #title = "Simulated Data from a Logistic-Normal Model",
                #beta = beta,
                #alpha = alpha,
                y=y,
                id=id,
                x=x,
                eta = z[["eta"]],
                Pr = z[["Pr"]],
                flag=flag)

#class(out) <- "sim.data"
out
}
```

A.5 gen-X-matrix.q

```
gen.X.matrix<-function( nc, nd, ni, nclust)
{
# Simulation of X design matrix for use with Logistic-Normal Model - J.Mills
# nclust = number of clusters
      nobs<-nclust*ni
      x<-matrix(c(rep(1,nobs),
#     Time-independent
         rep(round(runif(nclust,min=1,max=5))/5,rep(ni,nclust)),
#     Time-dependent
         runif(1*nobs, min=0, max=1)),nrow=nobs)
#The above makes use of S-plus vector capabilities
      x
}
```

A.6 rob-like.q

```
rob.like<-function( id, y, x, beta, betain, z, alpha,
                            epsilon, lwflag, prflag, cflag, print)
{
#
# robust maximum likelihood for logistic-normal model -J.Mills
# random intercept std dev design
#
# betain is used to calculate prediction weights!

        n<-rep( 1, length(y) )
        nclust<-length( table( id ) )
# for storing individual likelihoods:
        indlike<-rep(-10,nclust)

        p <- length(beta)
```

```
        if( ncol(x)!=p ){
          stop("x and beta do not conform!") }
        nobs <- length(y)
        q <- length(alpha)
        if(epsilon!=.001 & epsilon!= .002 & epsilon!= .005 &
           epsilon!= .01 & epsilon!= .02 & epsilon!= .05 &
           epsilon!= .1 & epsilon!= 0.) {
        stop("epsilon must be one of .001,.002,.005,.01,.02,.05 or .1")}

        if( is.null(ncol(z)) ){
          if( q>1) stop("z non-matrix but len(alpha)>1 !")
        }else{
          if( ncol(z)!=q ) stop("z and alpha do not conform!")
        }
        if( is.null( dimnames(x)[[2]] ) ){
                xnames<-paste( rep("beta",p), as.character(c(1:p)) )
        }else{
                xnames<-dimnames(x)[[2]]
        }
        if( is.null( dimnames(z)[[2]] ) ){
                znames<-paste( rep("alpha",q), as.character(c(1:q)) )
        }else{
                znames<-dimnames(z)[[2]]
        }
        flag<-0

        logL <- 0.0
        int.parms <- c( p, q, nobs, nclust)
        eta <- as.vector( x%*%beta)
#
        z <- .C("rob_like",
                as.double(id),
                as.double(y),
                as.double(x),
                beta = as.double(beta),
                betainit = as.double(betain),
                as.double(z),
                alpha = as.double(alpha),
                epsilon = as.double(epsilon),
                lw = as.integer(lwflag),
                pr = as.integer(prflag),
                eta = as.double(eta),
                as.integer(int.parms),
                logL = as.double(logL),
                indlike = as.double(indlike),
                flag=as.integer(flag),
                c=as.double(cflag),
                print=as.integer(print) )
#
        beta<-as.vector(z[["beta"]])
#       names(beta)<-xnames
        alpha<-as.vector(z[["alpha"]])
#       names(alpha)<-znames
#
        indlike<-as.vector(z[["indlike"]])
out <- list(    title = "ML Estimation for Logistic-Normal Models",
                dispersion = "logistic-normal random effects",
                beta = beta,
                alpha = alpha,
                epsilon = epsilon,
```

```
#               eta = z[["eta"]],
                logL = z[["logL"]],
                indlike=indlike,
                tol=z[["tol"]],
                iter=z[["iter"]],
                flag=z[["flag"]] )
#
#class(out) <- "rob.like"
#out
}
```

Bibliography

[1] K. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover Publications, New York, 1972.

[2] H. Akaike. Information theory and an extension of the likelihood principle. In B. N. Petrov and F. Csaki, editors, *Proceedings of the Second International Symposium of Information Theory*. Budapest: Akademiai Kiado, 1973.

[3] A. Allen, R. Attenborough, L. Dodds, E. Luther, and J. Pole. *Perinatal Care in Nova Scotia 1988 to 1995*. The Reproductive Care Program of Nova Scotia, Nova Scotia, 1996.

[4] A. Azzalini. Logistic regression for autocorrelated data with application to repeated measures. *Biometrika*, 81:767–775, 1994.

[5] Y. Bishop, S. Feinberg, and P. Holland. *Discrete Multivariate Analysis*. The MIT Press, MA, 1975.

[6] J. G. Booth and J. P. Hobert. Maximizing generalized linear mixed models likelihoods with an automated monte carlo em algorithm. *Journal of the Royal Statistical Society, Series B.*, 61:265–285, 1999.

[7] N. E. Breslow and D. G. Clayton. Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, 88:9–25, 1993.

[8] E. Cantoni and E. Ronchetti. Robust Estimation and Inference for Generalized Linear Models. *submitted*, 1999.

[9] R. J. Carroll and S. Pederson. On robustness in the logistic regression model. *Journal of the Royal Statistical Society, Series B*, 55:693–706, 1993.

[10] D. R. Cox. Role of models in statistical analysis. *Statistical Science*, 5, 1990.

[11] J. R. Dale. Global cross-ratio models for bivariate discrete ordered responses. *Biometrics*, 42:909–917, 1986.

[12] P. J. Diggle, K. Liang, and S. L. Zeger. *Analysis of Longitudinal Data*. Oxford University Press, New York, 1994.

[13] L. Dodds. Prevalence of smoking among pregnant women in nova scotia from 1988 to 1992. *Canadian Medical Association Journal*, 152:185–190, 1995.

[14] G. M. Fitzmaurice and N. M. Laird. A likelihood-based method for analysing longitudinal binary response. *Biometrika*, 80:141–151, 1993.

[15] G. Forsythe, M. Malcolm, and C. Moler. *Computer methods for mathematical computations*, page 180, 1980.

[16] H. Goldstein. *The Design and Analysis of Longitudinal Studies*. Oxford University Press, New York, 1975.

[17] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, Inc., New York, 1986.

[18] F. R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69:383–393, 1974.

[19] F. R. Hampel. *Handouts for the Instructional Meeting on Robust Statistical Methods*. 1982.

[20] P. J. Heagerty. Marginally specified logistic-normal models for longitudinal binary data. *Biometrics*, 55:688–698, 1999.

[21] P. J. Heagerty and B. F. Kurland. Misspecified Maximum Likelihood Estimates and Generalized Linear Mixed Models. *Submitted*, 2000.

[22] P. J. Heagerty and S. L. Zeger. Marginalized regression models for cluster ordinal measurements. *Journal of the American Statistical Association*, 91:1024–1036, 1996.

[23] P. J. Heagerty and S. L. Zeger. Marginalized Multilevel Models and Likelihood Inference. *Statistical Science*, 2000.

[24] J. R. Hebel, N. L. Fox, and M. Sexton. Dose-response of birth weight to various measures of maternal smoking during pregnancy. *Journal of Clinial Epidemiology*, 41:483–489, 1988.

[25] N. J. Horton and S. R. Lipsitz. Review of software to fit generalized estimating equation regression models. *The American Statistician*, 53:160–169, 1999.

[26] P. J. Huber. *Robust Statistics*. John Wiley and Sons, Inc., New York, 1981.

[27] R. M. Huggins. A Robust Approach to the Analysis of Repeated Measures. *Biometrics*, 35:43–57, 1993.

[28] Ihaka and Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5:299–314, 1996.

[29] J. Krishnakumar and E. Ronchetti. Robust estimators for simultaneous equation models. *Journal of Econometrics*, 78:294–3141, 1997.

[30] F. T. Krogh and W. V. Snyder. Algorithm 699; a new representation of patterson's quadrature formulae. *ACM Transactions on Mathematical Software*, 17:457–461, 1991.

[31] H. R. Kunsch, L. A. Stefanski, and R. J. Carroll. Conditionally unbiased bounded-influence estimation in general regression models, will applications to generalized linear models. *Journal of the American Statistical Association*, 84:460–466, 1989.

[32] J. B. Lang and A. Agresti. Simultaneously modeling joing and marginal distributions of multivariate categorical responses. *Journal of the American Statistical Association*, 89:625–632, 1994.

[33] K. Liang and S. L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73:13–22, 1986.

[34] P. McCullagh and J. A. Nelder. Quasi-likelihood functions. *Annals of Statistics*, 11:59–67, 1983.

[35] C. McCulloch. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Society*, 89:330–335, 1997.

[36] S. Morgenthaler. Least-absolute-deviations fits for generalized linear models. *Biometrika*, 79:747–754, 1992.

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

[38] J. M. Neuhaus and J. D. Kalbfleisch. Between- and within- cluster covariate effects in the analysis of clustered data. *Biometrics*, 54:638–645, 1998.

[39] J. M. Neuhaus, J. D. Kalbfleisch, and W. W. Hauck. A Comparison of Cluster-Specific and Population-Averaged Approaches for Analyzing Correlated Binary Data. *International Statistical Review*, 59:25–35, 1991.

[40] T. N. Patterson. The optimal addition of points to quadrature formulae. *Math. Comput.*, 22:847–856, 1969.

[41] J. S. Preisser and B. F. Qaqish. Robust Regression for Clustered Data with Application to Binary Responses. *Biometrics*, 55:575–579, 1999.

[42] R. L. Prentice. Correlated binary regression with covariates specific to each binary observation. *Biometrics*, 44(1033–1048), 1988.

[43] A. M. Richardson and A. H. Welsh. Robust restricted maximum likelihood in mixed linear models. *Biometrics*, 51:1429–1439, 1995.

[44] E. Ronchetti. Robustness aspects of model choice. *Statistica Sinica*, 1997.

[45] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, Inc., USA, 1987.

[46] R. Stiratelli, N. M. Laird, and J. H. Ware. Random-effects model for several observations with binary response. *Journal of the American Statistical Association*, 40:961–971, 1984.

[47] R. W. Wedderburn. Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika*, 61:439–447, 1974.

[48] H. White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50:1–26, 1982.

[49] K.-H. Yuan and R. I. Jennrich. Asymptotics of estimating equations under natural conditions. *Journal of Multivariate Analysis*, 65:245–260, 1998.

[50] S. Zeger, K. Liang, and P. Albert. Models for Longitudinal Data: A Generalized Estimating Equation Approach. *Biometrics*, 44:1049–1060, 1988.

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

[52] S. L. Zeger and K. Liang. Longitudinal Data Analysis for Discrete and Continuous Outcomes. *Biometrics*, 42:121–130, 1986.