Maximum Likelihood Estimation for Serially Correlated Longitudinal Data

by

Charles E. McCulloch
Biometrics Unit and Statistics Center,

Cornell University
Ithaca, NY 14853
U.S.A.

BU-1337-M May, 1996

SUMMARY

We describe and compare algorithms for calculating maximum likelihood estimates for longitudinal data which arise from a generalized linear model. Our models can accommodate virtually unlimited correlation structures but are more efficient with random effects and autoregressive structures. We consider a Monte Carlo Newton-Raphson and a stochastic approximation algorithm for fitting these models.

Keywords and phrases: generalized linear mixed models, Markov chain Monte Carlo

1. INTRODUCTION

Generalized linear models have enjoyed widespread use due to their flexibility in modeling independent observations from a wide variety of distributions and incorporating possibly nonlinear links between the mean of the response and the predictors. In the past decade research has focused on extending those methods to accommodate correlated responses (e.g., Zeger and Liang, 1986; Schall, 1991; McCulloch, 1996). In this paper we describe simulation based approaches to calculating fully parametric maximum likelihood estimates (MLEs) for a wide class of generalized linear models for serially correlated data.

Because of the computational difficulties of ML estimation even for simple models (e.g., Albert, 1991; Le, Leroux and Puterman, 1992) there has been little work on non-normal data. Chan and Ledholter (1995) describe a Monte Carlo EM algorithm for count data which is similar to the methods developed in Section 3. However, much work (e.g. Zeger and Qaqish, 1988) has centered on alternatives to ML estimation.

Section 2 describes a model which allows virtually unlimited correlation structures in the serial measurements. Section 3 describes two methods of calculating MLEs, a Monte Carlo version of Newton-Raphson and a method based on stochastic approximation. Section 4 describes details for a logit-normal model and Section 5 offers conclusions.

2. THE MODEL

We consider the following class of models. $\mathbf{Y}$ will denote the observed data vector and we will describe the correlation structure through a vector $\mathbf{e}$. Conditional on $\mathbf{e}$, we assume that the elements of $\mathbf{Y}$ are independent and drawn from a distribution which, for simplicity of exposition, we take to have canonical link and constant scale function. To complete the specification, we assume a distribution for $\mathbf{e}$, depending on parameters α:

$$\begin{aligned} &f_{Y_i|e}(y|\beta,\mathbf{e}) = \exp\{y\eta_i - c(\eta_i) + d(y)\} \\ &\mathbf{e} \sim f_e(\mathbf{e}|\alpha). \end{aligned} \tag{1}$$

Here, $g(E[Y_i|\mathbf{e}]) = \eta_i = \mathbf{x}_i'\beta + e_i$, with $\mathbf{x}_i'$ being the *ith* row of $\mathbf{X}$, the model matrix for the fixed effects and $g(\cdot)$ being the link function. The likelihood for (1) is given by

$$L(\beta,\alpha) = \int\prod_{i=1}^{n} f_{Y_i|e}(y|\beta,\mathbf{e}) f_e(\mathbf{e}|\alpha) d\mathbf{e}, \tag{2}$$

which often cannot be evaluated in closed form. Our goal is to develop algorithms to calculate fully parametric MLEs based on (2).

3. FITTING METHODS

In this section we develop two simulation based methods for calculating MLEs, a Monte-Carlo Newton-Raphson (MCNR) approach and an approach based on stochastic

approximation. Central to either is the ability to simulate random draws from $f_{e|Y}(\mathbf{e}|\beta,\alpha,\mathbf{Y})$, so we first describe that.

3.1 *A Metropolis algorithm for simulating from* $f_{e|Y}$

The algorithm we use is a Metropolis algorithm so we need to specify the distribution, $h_e(e)$, from which candidate draws are made and an acceptance function for deciding whether to retain the old draw or accept the candidate one. We first consider the acceptance function so let $\mathbf{e}$ denote the previous draw and suppose we generate a new value e_k^* for the k*th* component of $\mathbf{e}$ from the candidate distribution. Letting $\mathbf{e}^* = (e_1, e_2, \ldots, e_{k-1}, e_k^*, e_{k+1}, \ldots, e_n)$ we then accept $\mathbf{e}^*$ with probability $A_k(\mathbf{e},\mathbf{e}^*)$ and otherwise retain $\mathbf{e}$. Here $A_k(\mathbf{e},\mathbf{e}^*)$ is given by

$$A_k(\mathbf{e},\mathbf{e}^*) = \min\left\{1, \frac{f_{e|Y}(\mathbf{e}^*|\mathbf{Y})h_e(\mathbf{e})}{f_{e|Y}(\mathbf{e}|\mathbf{Y})h_e(\mathbf{e}^*)}\right\}. \tag{3}$$

The second term in braces in (3) can be rewritten:

$$\begin{aligned}\frac{f_{e|Y}(\mathbf{e}^*|\mathbf{Y})h_e(\mathbf{e})}{f_{e|Y}(\mathbf{e}|\mathbf{Y})h_e(\mathbf{e}^*)} &= \frac{\prod f_{Y_i|e}(y_i|\mathbf{e}^*)f_e(\mathbf{e}^*)h_e(\mathbf{e})}{\prod f_{Y_i|e}(y_i|\mathbf{e})f_e(\mathbf{e})h_e(\mathbf{e}^*)} \\ &= \frac{f_{Y_k|e}(y_k|e_k^*)f_{e_k|e_{-k}}(e_k^*|\mathbf{e}_{-k}^*)f_{e_{-k}}(\mathbf{e}_{-k}^*)h_e(\mathbf{e})}{f_{Y_k|e}(y_k|e_k)f_{e_k|e_{-k}}(e_k|\mathbf{e}_{-k})f_{e_{-k}}(\mathbf{e}_{-k})h_e(\mathbf{e}^*)},\end{aligned} \tag{4}$$

where $\mathbf{e}_{-k} = (e_1, e_2, \ldots, e_{k-1}, e_{k+1}, \ldots, e_n) = \mathbf{e}_{-k}^*$. Upon choosing $h_e(e) = f_{e_k|e_{-k}}(e_k|\mathbf{e}_{-k})$ equation (4) simplifies to $f_{Y_k|e}(y_k|e_k^*) / f_{Y_k|e}(y_k|e_k)$. So, to generate value from $f_{e|Y}$ we consider the elements of $\mathbf{e}$ one at a time, generate candidate draws from the conditional distribution of that element given the rest and accept it if the ratio of the conditional densities is high enough.

In many situations, $f_{e_k|e_{-k}}(e_k|\mathbf{e}_{-k})$ takes a simple form and hence will be easy to generate from. For example, in an autoregressive process of order 1, the conditional distribution of e_k only depends on e_{k-1}. Spatial processes with a dependence structure which only included a neighborhood of nearby values would experience a similar simplification. We now consider the use of this Metropolis step.

3.2 *Monte Carlo Newton-Raphson*

Whenever the marginal density of $\mathbf{Y}$ is formed as a mixture as in (2), the ML equations can be written as

$$E\left[\frac{\partial \ln f_{\mathbf{Y}|\mathbf{e}}(\mathbf{Y}|\mathbf{e},\beta)}{\partial \beta}\middle|\mathbf{Y}\right]=0 \tag{5a}$$

$$E\left[\frac{\partial \ln f_{\mathbf{e}}(\mathbf{e},\alpha)}{\partial \alpha}\middle|\mathbf{Y}\right]=0 \tag{5b}$$

Equation (5a) suggests (McCulloch, 1996) a Monte-Carlo version of Newton-Raphson (MCNR) or scoring which would take the form:

$$\beta^{(m+1)} = \beta^{(m)} + \mathrm{E}[\mathbf{X}'\mathbf{W}(\theta^{(m)},\mathbf{e})\mathbf{X}|\mathbf{y}]^{-1}\mathbf{X}'(\mathrm{E}[\mathbf{W}(\theta^{(m)},\mathbf{e})\frac{\partial\eta}{\partial\mu}\Big|_{\theta=\theta^{(m)}}(\mathbf{y}-\mu(\beta^{(m)},\mathbf{e}))|\mathbf{y}]),$$

where $\mu_i(\beta,\mathbf{e}) = \mathrm{E}[Y_i|\mathbf{e}]$, $W(\theta,\mathbf{e})^{-1} = diag\{(\partial\eta_i / \partial\mu_i)^2 \mathrm{var}(Y_i|\mathbf{e})\}$ and $\partial\eta / \partial\mu$ is a diagonal matrix with entries $\partial\eta_i / \partial\mu_i$.

Equation (5b) involves only the distribution of **e** and can be chosen to be easy to solve. A MCNR algorithm can now be constructed as follows:

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

2. Simulate N values, $\mathbf{e}^{(1)},\mathbf{e}^{(2)},\ldots,\mathbf{e}^{(N)}$, from $f_{e|Y}(\mathbf{e}|\beta^{(m)},\alpha^{(m)},\mathbf{Y})$ using the Metropolis algorithm described in Section 3.1

3. a. Calculate

$$\beta^{(m+1)} = \beta^{(m)} + \hat{\mathrm{E}}[\mathbf{X}'\mathbf{W}(\theta^{(m)},\mathbf{e})\mathbf{X}|\mathbf{y}]^{-1}\mathbf{X}'(\hat{\mathrm{E}}[\mathbf{W}(\theta^{(m)},\mathbf{e})\frac{\partial\eta}{\partial\mu}\Big|_{\theta=\theta^{(m)}}(\mathbf{y}-\mu(\beta^{(m)},\mathbf{e}))|\mathbf{y}]), \tag{6}$$

where the hats denote expectations approximated using the values from 2.

b. Choose $\alpha^{(m+1)}$ to maximize $\frac{1}{N}\sum_{k=1}^{N}\ln f_e(\mathbf{e}^{(k)}|\alpha^{(m)})$,

c. Set m=m+1.

4. If convergence is achieved, declare $\beta^{(m+1)}$ and $\alpha^{(m+1)}$ to be MLEs, otherwise return to step 2.

3.3 *Stochastic approximation*

A different approach to fitting these models has been suggested recently by Gu and Lin (1995) through the use of a stochastic approximation (SA) algorithm, though the basic idea of using SA to find MLEs is certainly older (e.g., Moyeed and Baddeley, 1991; Ruppert, 1991). The basic concept is to write $f_{Y,e}$ as $f_Y f_{e|Y}$. We can then easily derive

$$\frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial\theta} = \frac{\partial \ln f_{\mathbf{Y}}(\mathbf{Y}|\alpha,\beta)}{\partial\theta} + \frac{\partial \ln f_{\mathbf{e}|\mathbf{Y}}(\mathbf{e}|\mathbf{Y},\alpha,\beta)}{\partial\theta}. \quad (7)$$

We are interested in finding the root of the likelihood equation, e.g., where $\frac{\partial \ln f_{\mathbf{Y}}(\mathbf{Y}|\alpha,\beta)}{\partial\theta} = 0$. SA algorithms are methods of finding roots of regression equations so we need to rewrite (7) as a regression equation. Write $\mu(\theta)$ for the score function, $\frac{\partial \ln f_{\mathbf{Y}}(\mathbf{Y}|\alpha,\beta)}{\partial\theta}$, to emphasize we are regarding it as a function of θ and that it is not a function of $\mathbf{e}$ and note that $E\left[\frac{\partial \ln f_{\mathbf{e}|\mathbf{Y}}(\mathbf{e}|\mathbf{Y},\alpha,\beta)}{\partial\theta}\right]$ is zero for fixed $\mathbf{Y}$ when $\mathbf{e} \sim f_{\mathbf{e}|\mathbf{Y}}(\mathbf{e}|\mathbf{Y},\alpha,\beta)$ by the usual score identity. Hence $\frac{\partial \ln f_{\mathbf{e}|\mathbf{Y}}(\mathbf{e}|\mathbf{Y},\alpha,\beta)}{\partial\theta}$ can be regarded as a mean-zero, "error" term in the regression equation, $\frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial\theta} = \mu(\theta) + error$. Thus, plugging the random values of $\mathbf{e}$ into $\frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial\theta}$ gives the "data" for performing the regression.

To implement a SA algorithm we use the Metropolis algorithm of Section 3.1 to generate a sequence of $\mathbf{e}^{(i)} \sim f_{\mathbf{e}|\mathbf{Y}}(\mathbf{e}|\mathbf{Y},\alpha,\beta)$ and use them to form data $\frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}^{(i)}|\alpha,\beta)}{\partial\theta}$. One can then apply a multivariate version of a SA algorithm. Ruppert (1991) provides a nice review. A SA algorithm for this problem would generally take the following form

$$\theta^{(m+1)} = \theta^{(m)} - a_m \frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}^{(m)}|\alpha,\beta)}{\partial\theta}, \quad (8)$$

where a_m is chosen to decrease slowly to zero. Ideally a_m also incorporates information about the derivative of $\dfrac{\partial \ln f_{\mathbf{Y}}(\mathbf{Y}|\alpha,\beta)}{\partial\theta}$ (with respect to θ) at the root, but this is rarely known in practice.

We considered three forms of SA algorithms. One used (8) with

$$a_m = \frac{a}{(m+k)^{\alpha}}\left(\hat{E}\left[\frac{\partial^2 \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial\theta\partial\theta'}\right]\right)^{-1},$$

where $\hat{E}$ denotes an estimate of the expectation (for some details see the next Section) and k and α are predetermined constants. The other versions of SA were similar but formed the estimate not directly from the last iteration of (8) but instead by using data from a majority of the values from the iteration scheme. This follows suggestions of Frees and Ruppert (1990) and Ruppert (1991). The estimates are formed either by fitting a straight line to the data (plotting $\theta^{(m+1)}$ versus $\dfrac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}^{(m)}|\alpha,\beta)}{\partial\theta}$) and solving for the root or by simply averaging the $\theta^{(m)}$ values from (8).

4. AN ILLUSTRATION

We give some of the details for a balanced data, logit-normal model. This would be useful for modelling binary, serially correlated data. Let $\mathbf{Y}_i$ denote the data for the *ith* subject with the $\mathbf{Y}_i$ independent. Next, and conditional on $\mathbf{e}_i$, $Y_{ij} \sim$ Bernoulli(p_{ij}) with

$$\ln(p_{ij}/(1-p_{ij})) = \mathbf{x}_{ij}'\beta + \mathbf{e}_i \quad \text{and}$$
$$\mathbf{e}_i \sim N(\mathbf{0},\alpha),$$

where $\mathbf{x}_{ij}'$ is the ij*th* row of the model matrix, **X**. The use of the Metropolis algorithm of Section 3.1 in either a MCNR or SA algorithm takes the following form.

1. Generate e_{ij}^* from the conditional distribution of e_{ij} given e_{ik} (k≠j) and using the current values of α and β.

2. Accept the new e_{ij}^* with probability $A_k(\mathbf{e},\mathbf{e}^*)$, where $A_k(\mathbf{e},\mathbf{e}^*)$ is given by

$$\min\left\{1, \exp\{y_{ij}(e_{ij}^* - e_{ij})\}\frac{1+\exp\{\mathbf{x}_{ij}'\beta + e_{ij}\}}{1+\exp\{\mathbf{x}_{ij}'\beta + e_{ij}^*\}}\right\}.$$

If the new value is not accepted, then retain the old value.

The MCNR algorithm consists of the following steps.

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

2. Simulate N values, $\mathbf{e}^{(1)},\mathbf{e}^{(2)},...,\mathbf{e}^{(N)}$, from $f_{e|Y}(\mathbf{e}|\beta^{(m)},\alpha^{(m)},\mathbf{Y})$ using the Metropolis algorithm described above.

3. a. Calculate

$$\beta^{(m+1)} = \beta^{(m)} + \hat{\mathbf{E}}[\mathbf{X}'\mathbf{W}(\beta^{(m)},\mathbf{e})\mathbf{X}|\mathbf{y}]^{-1}\mathbf{X}'(\mathbf{y} - \hat{\mathbf{E}}[\mathbf{p}(\beta^{(m)},\mathbf{e}))|\mathbf{y}]), \qquad (9)$$

where the hats denote expectations approximated using the values from 2, $\mathbf{p}(\beta,\mathbf{e})$ has entries $(1+\exp(-(\mathbf{x}_{ij}'\beta+\mathbf{e}_i))^{-1}$ and $\mathbf{W}(\beta,\mathbf{e})$ is a diagonal matrix with entries $\mathbf{p}(\beta,\mathbf{e})(1-\mathbf{p}(\beta,\mathbf{e}))$.

b. Choose $\alpha^{(m+1)}$ to maximize $\frac{1}{N}\sum_{k=1}^{N}\ln f_e(\mathbf{e}^{(k)}|\alpha^{(m)})$,

c. Set m=m+1.

4. If convergence is achieved, declare $\beta^{(m+1)}$ and $\alpha^{(m+1)}$ to be MLEs, otherwise return to step 2.

Step 3b. involves the normal distribution, so the maximizing values depend on the structure assumed. In our example we will assume no structure for α and, since the data are complete and balanced, $\alpha^{(m+1)}$ will simply be the sample variance-covariance matrix of the $\mathbf{e}^{(i)}$. Other variance-covariance structures can be handled by deriving the corresponding ML equations.

The SA algorithm simplifies in the case of the mixture model (2) since β only enters the conditional distribution of $\mathbf{Y}$ given $\mathbf{e}$ and α only enters the distribution of $\mathbf{e}$. We therefore have

$$\frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial\beta} = \frac{\partial \ln f_{\mathbf{Y}|\mathbf{e}}(\mathbf{Y}|\mathbf{e},\beta)}{\partial\beta} \text{ and} \qquad (10)$$

$$\frac{\partial \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial\alpha} = \frac{\partial \ln f_{\mathbf{e}}(\mathbf{e}|\alpha)}{\partial\alpha}.$$

The other ingredient we need to use (8) is an equation for a_m. The optimal a_m would be of the form (Ruppert, 1991) $a_m = \frac{a}{(m+k)^{\alpha}}\left(\frac{\partial\mu(\theta)}{\partial\theta}\right)^{-1}\Big|_{\theta=\hat{\theta}_{ML}}$ but this is impossible

since we do not know the MLE, $\hat{\theta}_{ML}$. However, we can approximate the derivative at the current estimates since

$$\frac{\partial \mu(\theta)}{\partial \theta} = \frac{\partial^2 \ln f_{\mathbf{Y}}(\mathbf{Y}|\alpha,\beta)}{\partial \theta \partial \theta'} = E\left[\frac{\partial^2 \ln f_{\mathbf{Y},\mathbf{e}}(\mathbf{Y},\mathbf{e}|\alpha,\beta)}{\partial \theta \partial \theta'}\right],$$

where the expectation is taken with respect to $f_{\mathbf{e}|\mathbf{Y}}$ and using (7). This is the rationale behind equation (9).

Using (10), the iteration for β in the SA algorithm is given by

$$\beta^{(m+1)} = \beta^{(m)} + \frac{a}{(m+k)^{\alpha}} \hat{\mathbf{E}}[\mathbf{X}'\mathbf{W}(\beta^{(m)},\mathbf{e})\mathbf{X}|\mathbf{y}]^{-1}\mathbf{X}'(\mathbf{y} - \mathbf{p}(\beta^{(m)},\mathbf{e}^{(m)})), \qquad (11)$$

where the hat denotes an approximation to the expectation, $\mathbf{p}(\beta,\mathbf{e})$ has entries $(1+\exp(-(\mathbf{x}_{ij}'\beta + \mathbf{e}_i))^{-1}$ and $\mathbf{W}(\beta,\mathbf{e})$ is a diagonal matrix with entries $\mathbf{p}(\beta,\mathbf{e})(1-\mathbf{p}(\beta,\mathbf{e}))$. A similar set of equations can be derived for α, though a possible problem is keeping $\alpha^{(m+1)}$ positive semi-definite. An alternative is to use an iteration more like step 3b of MCNR.

It remains to estimate the expectation in (11). We propose using the average of all the simulated values to estimate $E\left[\mathbf{W}(\beta^{(m)},\mathbf{e})|\mathbf{Y}\right]$, e.g., $\hat{E}\left[\mathbf{W}(\beta^{(m)},\mathbf{e})|\mathbf{Y}\right] = \frac{1}{N}\sum_i \mathbf{W}(\beta^{(m)},\mathbf{e}^{(i)})$.

5. DISCUSSION

To compare MCNR and SA we focus on (6) and (11), i.e., the iterations for β. They are very similar with the main difference being the multiplier which precedes the increment from $\beta^{(m+1)}$ to $\beta^{(m)}$. Minor differences are the fact that a single simulated value is used in SA and hence the $\mathbf{p}(\beta^{(m)},\mathbf{e}^{(m)})$ term is calculated with that single value rather than averaged over a number of values. We approximate the second derivative matrix $\hat{\mathbf{E}}[\mathbf{X}'\mathbf{W}(\beta^{(m)},\mathbf{e})\mathbf{X}|\mathbf{y}]$ similarly for both, though in SA the average is over all the $\mathbf{e}^{(i)}$ values while in MCNR it is just for the values simulated at that iteration.

The multiplier $\frac{a}{(m+k)^{\alpha}}$ decreases the step size as the iterations increase in SA. This eventually serves to eliminate the stochastic error involved in the Metropolis step. To achieve a corresponding reduction using MCNR, the simulation size would have to be increased as the iterations increase in order to eliminate the simulation noise.

SA seems to have advantages in that it can use all of the simulated data to calculate estimates and it uses the simulated values one at a time. A theoretical advantage

of SA is that convergence proofs are worked out for many cases. Practical details of the implementation of both SA and MCNR need to be worked out.

Acknowledgments

Thanks to David Ruppert for numerous discussions on stochastic approximation algorithms.

REFERENCES

Albert, P.S. (1991). A two-state Markov mixture model for a time series of epileptic seizure counts. *Biometrics* 47: 1371-1381.

Chan, K.S. and Ledholter, J. (1995). Monte Carlo EM estimation for time series models involving counts. *Journal of the American Statistical Association,* 90: 242-252.

Frees, E.W. and Ruppert, D. (1990) Estimation following a sequentially designed experiment. *Journal of the American Statistical Association,* 85: 1123-1129.

Gu, M. and Lin, S.. (1995). A stochastic approximation algorithm for maximum likelihood estimation with incomplete data. Technical report, Department of Mathematics and Statistics, McGill University.

Le, N.D., Leroux, B.G., and Puterman, M.L. (1992). Exact likelihood evaluation in a Markov mixture model for time series of seizure counts. *Biometrics* 48: 317-324.

McCulloch, C.E. (1996). Maximum likelihood algorithms for generalized linear mixed models. Accepted for publication in *Journal of the American Statistical Association.*

Moyeed, R.A. and Baddeley, A.J. (1991). Stochastic approximation of the MLE for a spatial point process. *Scandinavian Journal of Statistics* 18: 39-50.

Ruppert, D. (1991). Stochastic approximation. pp. 503-529 in *Handbook of Sequential Analysis*, (Ghosh, B.K. and Sen, P.K. eds.). Dekker, New York.

Schall, R. (1991), "Estimation in Generalized Linear Models with Random Effects," *Biometrika*, 78, 719-727.

Zeger, S.L. and Liang, K.-Y. (1986). Longitudinal data analysis for discrete and continuous outcomes. *Biometrics* 42: 121-130.

Zeger, S.L. and Qaqish, B. (1988). Markov regression models for time series: A quasi-likelihood approach. *Biometrics* 44: 1019-1031.