Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Fitting generalized linear mixed-effects models using lme4

Steven C. Walker
McMaster University

Rune Haubo Bojesen Christensen
Technical University of Denmark

Douglas Bates
University of Wisconsin - Madison

Ben Bolker
McMaster University

Martin Mächler
ETH Zurich

Abstract

abstract goes here

Keywords: sparse matrix methods, generalized linear mixed models, penalized least squares, Cholesky decomposition.

1. Introduction

The **lme4** package for R can be used to fit a broad range of mixed-effects models. One of the main advantages of **lme4** over its predecessor, **nlme**, is that it can be used to fit both linear mixed models (LMMs) and generalized linear mixed models (GLMMs), which combine the flexibility of LMMs and generalized linear models (GLMs). In a companion paper, we have described the facilities in **lme4** for fitting linear mixed models (LMMs). Here we describe the facilities for fitting GLMMs.

2. Generalized Linear Mixed Models

Generalized linear mixed models (GLMMs) extend the class of generalized linear models

(GLMs) by allowing for both fixed and random effects. In a GLM, the length-n vector-valued response variable, $\mathcal{Y}$ has a distribution in the exponential family (e.g. normal, binomial, Poisson). The mean, $\boldsymbol{\mu}_{\mathcal{Y}}$, of $\mathcal{Y}$ depends on a linear predictor,

$$\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}. \tag{1}$$

where $\boldsymbol{\beta}$ is a p-dimensional coefficient vector and $\boldsymbol{X}$ is an $n \times p$ model matrix. The mapping from $\boldsymbol{\mu}_{\mathcal{Y}}$ to $\boldsymbol{\eta}$, which is called the *link function* and written,

$$\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\eta} = \boldsymbol{g}\left(\boldsymbol{\mu}_{\mathcal{Y}}\right), \tag{2}$$

is a *diagonal mapping* in the sense that there is a scalar function, g, such that the ith component of $\boldsymbol{\eta}$ is g applied to the ith component of $\boldsymbol{\mu}_{\mathcal{Y}}$. (The name "diagonal" reflects the fact that the Jacobian matrix, $\frac{d\eta}{d\mu'}$, of such a mapping will be diagonal.) The scalar link function must be invertible over its range. The vector-valued *inverse link* function, $\boldsymbol{g}^{-1}$, will be the scalar inverse link, g^{-1}, applied component-wise to $\boldsymbol{\eta}$.

In the GLMM case, the mean of the exponential family distribution of $\mathcal{Y}$ depends on an unobserved random vector, $\mathcal{B}$, of length q, called the random-effects coefficient vector. In particular, the conditional mean of $\mathcal{Y}$ given that $\mathcal{B} = \boldsymbol{b}$, writen $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{B}=\boldsymbol{b}}$, depends on the linear predictor,

$$\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{X}\boldsymbol{\beta}. \tag{3}$$

where $\boldsymbol{Z}$ is an $n \times q$ random-effects model matrix. Similar to the GLM case, the mapping from the conditional mean, $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{B}=\boldsymbol{b}}$, to the linear predictor, $\boldsymbol{\eta}$,

$$\boldsymbol{Z}\boldsymbol{b} + \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\eta} = \boldsymbol{g}\left(\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{B}=\boldsymbol{b}}\right), \tag{4}$$

The random vector $\mathcal{B}$, is assumed to be distributed multivariate normally,

$$\mathcal{B} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_\theta) \tag{5}$$

where $\boldsymbol{\Sigma}_\theta$ is the covariance matrix of $\mathcal{B}$, which depends on a vector of covariance parameters, θ.

The optimization routines of **lme4** never actually compute Σ_θ directly, and instead use the covariance factor, $\boldsymbol{\Lambda}_\theta$, which is a matrix squareroot of Σ_θ,

$$\boldsymbol{\Sigma}_\theta = \boldsymbol{\Lambda}_\theta\boldsymbol{\Lambda}_\theta'. \tag{6}$$

This characterization of the random-effects covariance structure allows us to write the linear predictor as

$$\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{\Lambda}_\theta\boldsymbol{u} + \boldsymbol{X}\boldsymbol{\beta}, \tag{7}$$

where the spherized random effects vector, $\boldsymbol{u}$, is a realization of the random vector, $\mathcal{U}$,

$$\mathcal{U} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_q) \tag{8}$$

where $\boldsymbol{I}_q$ is the identity matrix.

Common forms of the conditional distribution are Bernoulli, for binary responses, binomial for binary responses that are recorded as the number of trials and the number of successes, and Poisson, for count data. The combination of a distributional form and a link function is called a *family*. For distributional forms in the exponential family there is a *canonical link*. For Bernoulli or binomial forms the canonical link is the *logit* link function

$$\eta_i = \log\left(\frac{\mu_i}{1-\mu_i}\right); \tag{9}$$

for the Poisson distribution the canonical link is the natural logarithm.

The form of the distribution determines the conditional variance, $\mathrm{Var}(\mathcal{Y}|\mathcal{U}=\boldsymbol{u})$, as a function of the conditional mean and, possibly, a separate scale factor. (In most cases the conditional variance is completely determined by the conditional mean.)

Discuss prior weights more thoroughly; mention offsets Prior weights can also be incorporated in the sense . . .

Scale factor??? We know that they're inconsistently incorporated in the code, but are there theoretical issues that we don't understand???

The likelihood of the parameters, given the observed data, is now

$$L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) = \int_{\mathbb{R}^q} f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}},\boldsymbol{u})\,d\boldsymbol{u} \tag{10}$$

where, as in the case of linear mixed models, $f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}},\boldsymbol{u})$ is the unscaled conditional density of $\mathcal{U}$ given $\mathcal{Y}=\boldsymbol{y}_{\text{obs}}$. The notation here is a bit blurred because, although the joint distribution of $\mathcal{Y}$ and $\mathcal{U}$ is always continuous with respect to $\mathcal{U}$, it can be (and often is) discrete with respect to $\mathcal{Y}$. However, when we condition on the observed value $\mathcal{Y}=\boldsymbol{y}_{\text{obs}}$, the resulting function is continuous with respect to $\boldsymbol{u}$ so the unscaled conditional density is indeed well-defined as a density, up to a scale factor.

To evaluate the integrand in (10) we use the value of the `dev.resids` function in the GLM family. This vector, $\boldsymbol{d}(\boldsymbol{y}_{\text{obs}},\boldsymbol{u})$, with elements, $d_i(\boldsymbol{y}_{\text{obs}},\boldsymbol{u}), i=1,\ldots,n$, provides the deviance of a generalized linear model as

$$\sum_{i=1}^{n} d_i(\boldsymbol{y}_{\text{obs}},\boldsymbol{u}).$$

(We should note that there some confusion in R (and in its predecessor, S) about what exactly the deviance residuals for a family are. As indicated above, we will use this name for the value of the `dev.resids` function in the family. The signed square root of this vector, using the signs of $\boldsymbol{y}_{\text{obs}}-\mu$, is returned from `residuals` applied to a fitted model of class `"glm"` when `type="deviance"`, the default, is specified. Both are called "deviance residuals" in the documentation but, although they are related, they are not the same.) One advantage of using the pre-existing GLM family structure is that models with any family and link function can be fitted (although common families and link functions are hard-coded in C++ for computational speed), in contrast to previous versions of `lme4` and other R packages for GLMM fitting.

The likelihood can now be expressed as

$$L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) = \int_{\mathbb{R}^q} \exp\left(-\frac{\sum_{i=1}^{n} d_i(\boldsymbol{y}_{\text{obs}},\boldsymbol{u}) + \|\boldsymbol{u}\|^2}{2}\right)(2\pi)^{-q/2}\,d\boldsymbol{u} \tag{11}$$

As for linear mixed models, we simplify evaluation of the integral (10) by determining the value, $\tilde{\boldsymbol{u}}_{\beta,\theta}$, that maximizes the integrand. When the conditional density, $\mathcal{U}|\mathcal{Y}=\boldsymbol{y}_{\text{obs}}$, is multivariate Gaussian, this conditional mode will also be the conditional mean. However, for most families used in GLMMs, the mode and the mean need not coincide so we use the more general term and call $\tilde{\boldsymbol{u}}_{\beta,\theta}$ the *conditional mode*. We first describe the numerical methods for determining the conditional mode using the Penalized Iteratively Reweighted Least Squares (PIRLS) algorithm then return to the question of evaluating the integral (10).

2.1. Determining the conditional mode

The iteratively reweighted least squares (IRLS) algorithm is an incredibly efficient method of determining the maximum likelihood estimates of the coefficients in a generalized linear model. We extend it to a *penalized iteratively reweighted least squares* (PIRLS) algorithm for determining the conditional mode, $\tilde{\boldsymbol{u}}_{\beta,\theta}$. This algorithm has the form

1. Given parameter values, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, and starting estimates, $\boldsymbol{u}_0$, evaluate the linear predictor, $\boldsymbol{\eta}$, the corresponding conditional mean, $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{U}=\boldsymbol{u}}$, and the conditional variance. Establish the weights as the inverse of the variance. We write these weights in the form of a diagonal weight matrix, $\boldsymbol{W}$, although they are stored and manipulated as a vector.

2. Solve the penalized, weighted, nonlinear least squares problem

$$\arg\min_{\boldsymbol{u}} \left(\left\| \boldsymbol{W}^{1/2} \left(\boldsymbol{y}_{\text{obs}} - \boldsymbol{\mu}_{\mathcal{Y}|\mathcal{U}=\boldsymbol{u}} \right) \right\|^2 + \|\boldsymbol{u}\|^2 \right) \tag{12}$$

3. Update the weights, $\boldsymbol{W}$, and check for convergence. If not converged, go to step 2.

We use a Gauss-Newton algorithm with an orthogonality convergence criterion (Bates and Watts 1988, §2.2.3) to solve the penalized, weighted, nonlinear least squares problem in step 2. At the ith iteration we determine an increment, $\boldsymbol{\delta}_i$, as the solution to the penalized, weighted, linear least squares problem

$$\boldsymbol{\delta}_i = \arg\min_{\boldsymbol{\delta}} \left\| \begin{bmatrix} \boldsymbol{W}^{1/2} \left(\boldsymbol{y}_{\text{obs}} - \boldsymbol{\mu}_i \right) \\ \boldsymbol{u}_i \end{bmatrix} - \begin{bmatrix} \boldsymbol{W}^{1/2} \boldsymbol{M}_i \boldsymbol{Z} \boldsymbol{\Lambda}_\theta \\ \boldsymbol{I}_q \end{bmatrix} \boldsymbol{u} \right\|^2 \tag{13}$$

where $\boldsymbol{u}_i$ is current value of $\boldsymbol{u}$, $\boldsymbol{\mu}_i$ is the corresponding conditional mean of $\mathcal{Y}|\mathcal{U} = \boldsymbol{u}_i$ and $\boldsymbol{M}_i$ is the Jacobian matrix of the vector-valued inverse link, evaluated at $\boldsymbol{\mu}_i$. That is

$$\boldsymbol{M}_i = \left. \frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}'} \right|_{\boldsymbol{\eta}_i}, \tag{14}$$

which will be a diagonal matrix so, as for the weights, we store and manipulate the Jacobian as a vector.

The minimizer, $\boldsymbol{\delta}_i$, of (13) satisfies

$$\boldsymbol{P} \left(\boldsymbol{\Lambda}_\theta' \boldsymbol{Z}' \boldsymbol{M}_i \boldsymbol{W} \boldsymbol{M}_i \boldsymbol{Z} \boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q \right) \boldsymbol{P}' \boldsymbol{\delta}_i = \boldsymbol{\Lambda}_\theta' \boldsymbol{Z}' \boldsymbol{M}_i \boldsymbol{W} (\boldsymbol{y}_{\text{obs}} - \boldsymbol{\mu}_i) - \boldsymbol{u}_i \tag{15}$$

which we solve using the sparse Cholesky factor. At convergence, the factor, $\boldsymbol{L}_{\beta,\theta}$, satisfies

$$\boldsymbol{L}_{\beta,\theta} \boldsymbol{L}_{\beta,\theta}' = \boldsymbol{P} \left(\boldsymbol{\Lambda}_\theta' \boldsymbol{Z}' \boldsymbol{M} \boldsymbol{W} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q \right) \boldsymbol{P}' \tag{16}$$

As we show in the next section, the matrix $(\boldsymbol{L}_{\beta,\theta} \boldsymbol{L}_{\beta,\theta}')^{-1}$ is a Laplace approximation of the covariance matrix for the spherized random effects, conditional on the observed data. This fact is useful for constructing a nonlinear objective function for finding the approximate maximum likelihood estimates of θ and β.

2.2. Evaluating the likelihood for GLMMs using the Laplace approximation

A second-order Taylor series approximation to $-2\log[f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u})]$ based at $\tilde{\boldsymbol{u}}$ provides an approximation of the unscaled conditional density as a multiple of the density for the multivariate Gaussian $\mathcal{N}(\tilde{\boldsymbol{u}}, \boldsymbol{L}\boldsymbol{L}')$. The change of variable

$$\boldsymbol{u} = \tilde{\boldsymbol{u}} + \boldsymbol{L}\boldsymbol{z} \tag{17}$$

provides

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) &= \int_{\mathbb{R}^q} f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u})\, d\boldsymbol{u} \\ &\approx \tilde{f}\,|\boldsymbol{L}| \int_{\mathbb{R}^q} e^{-\|\boldsymbol{z}\|^2/2}\,(2\pi)^{-q/2}\, d\boldsymbol{z} \\ &= \tilde{f}\,|\boldsymbol{L}| \end{aligned} \tag{18}$$

or, on the deviance scale,

$$-2\ell(\boldsymbol{\beta}, \boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) \approx \sum_{i=1}^{n} d_i(\boldsymbol{y}_{\text{obs}}, \tilde{\boldsymbol{u}}) + \|\tilde{\boldsymbol{u}}\|^2 + \log(|\boldsymbol{L}|^2) + \frac{q}{2}\log(2\pi) \tag{19}$$

Decomposing the deviance for simple models

A special, but not uncommon, case is that of scalar random effects associated with levels of a single grouping factor, $\boldsymbol{h}$. In this case the dimension, q, of the random effects is the number of levels of $\boldsymbol{h}$ — i.e. there is exactly one random effect associated with each level of $\boldsymbol{h}$. We will write the vector of variance-covariance parameters, which is one-dimensional, as a scalar, θ. The matrix $\boldsymbol{\Lambda_\theta}$ is a multiple of the identity, $\theta\boldsymbol{I}_q$, and $\boldsymbol{Z}$ is the $n \times q$ matrix of indicators of the levels of $\boldsymbol{f}$. The permutation matrix, $\boldsymbol{P}$, can be set to the identity and $\boldsymbol{L}$ is diagonal, but not necessarily a multiple of the identity.

Because each element of $\boldsymbol{\mu}$ depends on only one element of $\boldsymbol{u}$ and the elements of $\mathcal{Y}$ are conditionally independent, given $\mathcal{U} = \boldsymbol{u}$, the conditional densities of the $u_j, j = 1, \ldots, q$ given $\mathcal{Y} = \boldsymbol{y}_{\text{obs}}$ are independent. We partition the indices $1, \ldots, n$ as $\mathbb{I}_j, j = 1, \ldots, q$ according to the levels of $\boldsymbol{h}$. That is, the index i is in $\mathbb{I}_j$ if $h_i = j$. This partitioning also applies to the deviance residuals in that the ith deviance residual depends only on u_j when $i \in \mathbb{I}_j$.

Writing the univariate conditional densities as

$$f_j(\boldsymbol{y}_{\text{obs}}, u_j) = \exp\left(-\frac{\sum_{i\in\mathbb{I}_j} d_i(\boldsymbol{y}_{\text{obs}}, u_j) + u_j^2}{2}\right)(2\pi)^{-1/2} \tag{20}$$

we have

$$f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u}) = \prod_{j=1}^{q} f_j(\boldsymbol{y}_{\text{obs}}, u_j) \tag{21}$$

and

$$L(\boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{y}_{\text{obs}}) = \prod_{j=1}^{q} \int_{\mathbb{R}} f_j(\boldsymbol{y}_{\text{obs}}, u) \, du \tag{22}$$

We consider this special case both because it occurs frequently and because, for some software, it is the only type of GLMM that can be fit. Also, in this particular case we can graphically assess the quality of the Laplace approximation by comparing the actual integrand to its approximation.

Consider the `cbpp` data on contagious bovine pleuropneumonia (CBPP) incidence according to season and herd, available in the **lme4** package (see 4.1 for more details).

```
str(cbpp)

'data.frame': 56 obs. of  4 variables:
 $ herd      : Factor w/ 15 levels "1","2","3","4",..: 1 1 1 1 2 2 2..
 $ incidence: num  2 3 4 0 3 1 1 8 2 0 ...
 $ size      : num  14 12 9 5 22 18 21 22 16 16 ...
 $ period    : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 1 2..
```

and the model

```
print(m1 <- glmer(cbind(incidence, size-incidence) ~ period + (1|herd),
                  cbpp, binomial), corr=FALSE)

Generalized linear mixed model fit by maximum likelihood ['glmerMod']
 Family: binomial ( logit )
Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
   Data: cbpp
     AIC       BIC    logLik deviance
  194.05    204.18    -92.03   184.05
Random effects:
 Groups Name        Std.Dev.
 herd   (Intercept) 0.642
Number of obs: 56, groups: herd, 15
Fixed Effects:
(Intercept)      period2      period3      period4
     -1.398       -0.992       -1.128       -1.580
```

This model has been fit by minimizing the Laplace approximation to the deviance. We can assess the quality of this approximation by evaluating the unscaled conditional density at $u_j(z) = \tilde{u_j} + z/\boldsymbol{L}_{j,j}$ and comparing the ratio, $f_j(\boldsymbol{y}_{\text{obs}}, u)/(\tilde{f_j}\sqrt{2\pi})$, to the standard normal density, $\phi(z) = e^{-z^2/2}/\sqrt{2\pi}$, as shown in Figure 1. *consider Q-Q plots to emphasize deviations from normality?* As we can see from this figure, the univariate integrands are very close to the standard normal density, indicating that the Laplace approximation to the deviance is a good approximation in this case.

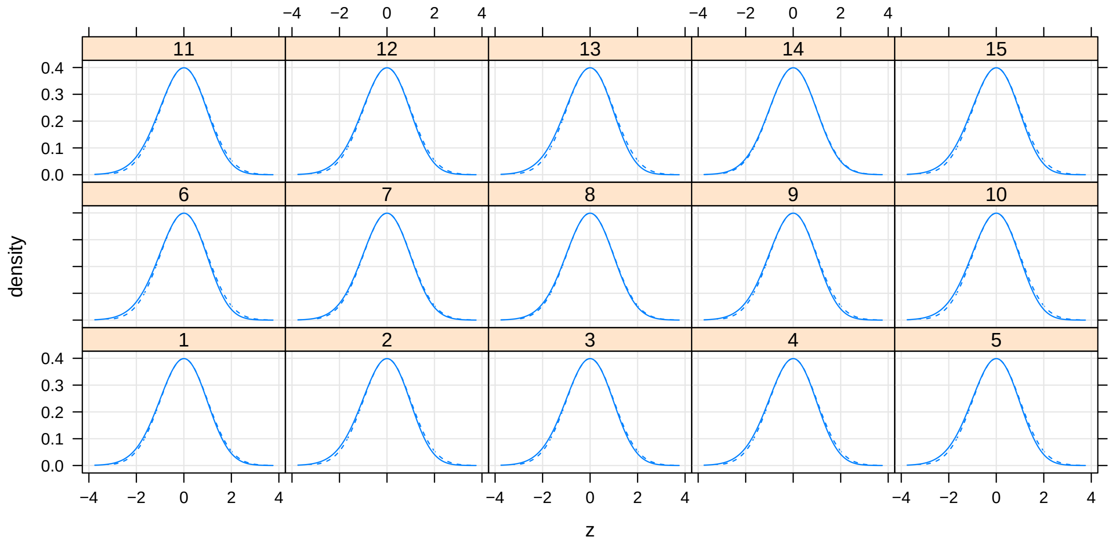


Figure 1: Comparison of univariate integrands (solid line) and standard normal density function (dashed line)

3. Adaptive Gauss-Hermite quadrature for GLMMs

When the integral (10) can be expressed as a product of low-dimensional integrals, we can use Gauss-Hermite quadrature to provide a closer approximation to the integral. Univariate Gauss-Hermite quadrature evaluates the integral of a function that is multiplied by a "kernel" where the kernel is a multiple of e^{-z^2} or $e^{-z^2/2}$. For statisticians the natural candidate is the standard normal density, $\phi(z) = e^{-z^2/2}/\sqrt{(2\pi)}$. A kth-order Gauss-Hermite formula provides knots, $z_i, i = 1, ..., k$, and weights, $w_i, i = 1, \ldots, k$, such that

$$\int_{\mathbb{R}} t(z)\phi(z)\, dz \approx \sum_{i=1}^{k} w_i t(z_i)$$

The function `GHrule` in **lme4** (based on code in the **SparseGrid** package) provides knots and weights relative to the standard normal kernel for orders k from 1 to 25. For example,

```
GHrule(5)

          z       w  ldnorm
[1,] -2.857 0.01126 -5.0001
[2,] -1.356 0.22208 -1.8378
[3,]  0.000 0.53333 -0.9189
[4,]  1.356 0.22208 -1.8378
[5,]  2.857 0.01126 -5.0001
```

The choice of the value of k depends on the behavior of the function $t(z)$. If $t(z)$ is a polynomial of degree $k - 1$ then the Gauss-Hermite formula for orders k or greater provides an exact

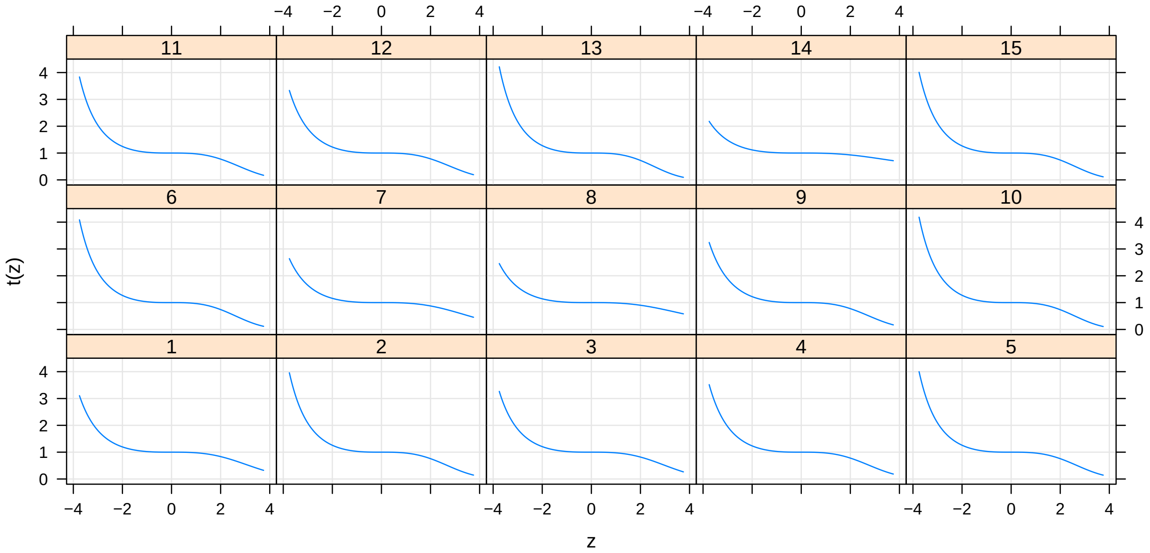


Figure 2: The function $t(z)$, which is the ratio of the normalized unscaled conditional density to the standard normal density, for each of the univariate integrals in the evaluation of the deviance for model `m1`. These functions should behave like low-order polynomials.

answer. The fact that we want $t(z)$ to behave like a low-order polynomial is often neglected in the formulation of a Gauss-Hermite approximation to a quadrature. The quadrature knots on the u scale are chosen as

$$u_{i,j}(z) = \tilde{u_j} + z_i/\boldsymbol{L}_{j,j}, \quad i = 1, \ldots, k;\ j = 1, \ldots, q \tag{23}$$

exactly so that the function $t(z)$ should behave like a low-order polynomial over the region of interest, which is to say the region where quadrature knots with large weights are located. The term "adaptive Gauss-Hermite quadrature" reflects the fact that the approximating Gaussian density is scaled and shifted to provide a second order approximation to the logarithm of the unscaled conditional density.

Figure 2 shows $t(z)$ for each of the unidimensional integrals in the likelihood for the model `m1` at the parameter estimates.

4. Examples

4.1. CBPP

The `?cbpp` help page describes the CBPP data set (Lesnoff, Laval, Bonnet, Abdicho, Workalemahu, Kifle, Peyraud, Lancelot, and Thiaucourt 2004) as follows:

> Contagious bovine pleuropneumonia (CBPP) is a major disease of cattle in Africa, caused by a mycoplasma. This dataset describes the serological incidence of CBPP in zebu cattle during a follow-up survey implemented in 15 commercial herds located in the Boji district of Ethiopia. The goal of the survey was to study the within-herd spread of CBPP in newly infected herds. Blood samples were quarterly collected from all animals of these herds to determine their CBPP status. These data were used to compute the serological incidence of CBPP (new cases occurring during a given time period). Some data are missing (lost to follow-up).

Lesnoff *et al.* (2004) estimated the effects of different treatments using (1) ordinary logistic regression incorporating a variance-inflation factor, also known as a quasi-binomial model ("logistic regression" is sometimes used specifically to describe analyses of Bernoulli responses, but in this case there are multiple trials per observation [cows that could become seropositive], and so a dispersion or scale parameter can be estimated); (2) a GLMM implemented in `lme4`; and a (3) Markov chain Monte Carlo algorithm *[CITE Zeger and Karim 1991] as L+2004 do?*, which as they state allows for a non-parametric rather than a Normal model for the random effects. *I can't find any evidence of what they* actually *did in the paper: there is no code or description of the MCMC algorithm. Z&K point out that you can use a Normal distribution for the random effects, or use rejection sampling to allow a non-Normal distribution, but we don't know what L+2004 actually used . . . is it worth asking them?* The authors did not find any significant effects of treatment, ascribing the null results to "a lack of power in the statistical analyses or to a quality problem for the medications used (and more generally, for health-care delivery in the Boji district)."

There appears to be a typo in Table 1; herd 6 in the table has {N,INC} pairs (i.e. initial size and percentage incidence of P1={14,14}; P2={[blank],12}; P3={25,[blank]}; P4={9,44}). Herd 1 in the `cbpp` *data set, which seems to be the corresponding one, has {size, incidence} pairs: P1={2,14}; P2={3,12}; P3={4,9}; P4={0,5}. This corresponds to incidences of 14, 25, 44, 0 which would match the values in the table if we assume that there are a couple of spurious blanks in the table,* and *that there was zero incidence in P4 for this herd. We should check with (1) DMB (where did he originally get these data?) and (2) the authors (we want to ask them about their MCMC algorithm anyway . . .)*

This is the simple example.

- *describe the problem in enough detail.*
- *fit parameters; show effects of AGQ number; describe accessors (prediction, simulation, confidence intervals)*
- *use PB to address finite-size inference issues*

We model proportional incidence as a binomial response depending on the fixed effects of period, treatment and average herd size; to account for repeated measures we fit a model with a random effect of herd. In principle we might be curious about a treatment by period interaction, but a model incorporating such a treatment would clearly be overfitting the data set. With 56 observations, we should be fitting at most 5–6 parameters (Harrell 2001); the model with period, treatment, and average herd size already has 7 parameters, and adding a treatment × period interaction would bring the total to 13.

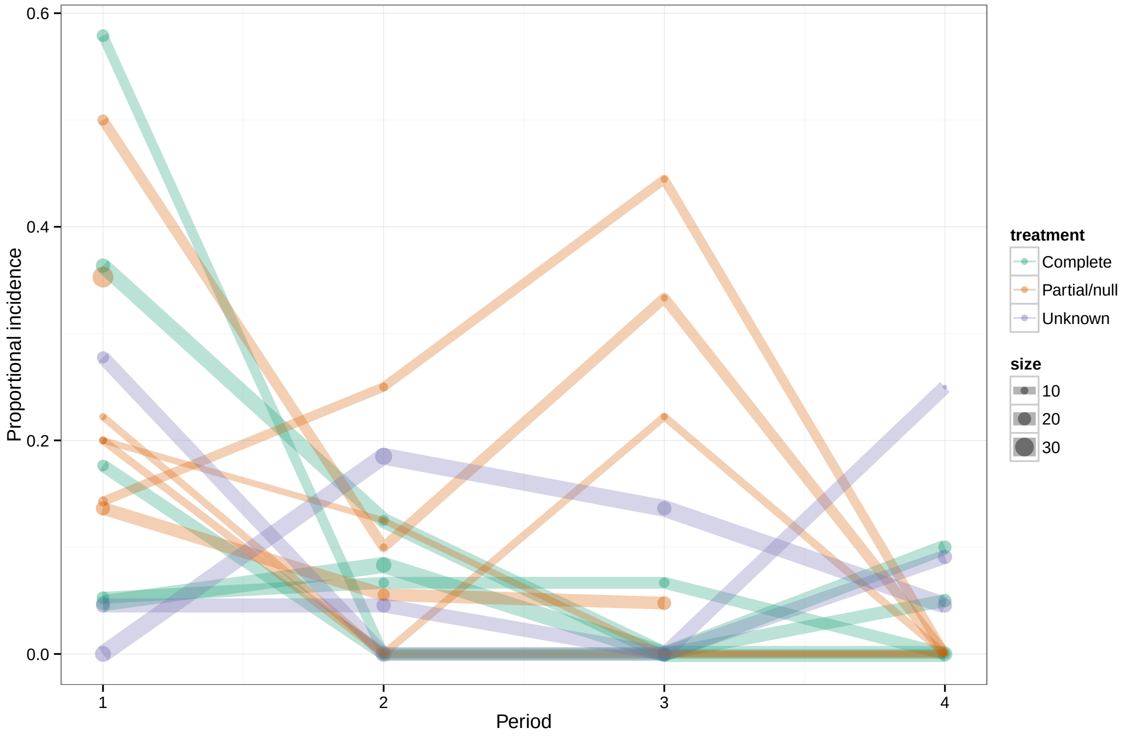


Figure 3: Incidence (proportion of cows becoming seropositive per observation period) vs. period. Colours show treatment category for each herd; point sizes and line widths show size (seronegative cows at the start of each period) and average herd size respectively.

Here we illustrate that, as in `glm`, we can specify a binomial response by a proportion and use the `weights` argument to specify the sample size, instead of the slightly more typical `cbind(successes,failures)` format. We use the `bobyqa` optimizer instead of the default Nelder-Mead optimizer because `glmer` warns us that the gradients of the converged model are worryingly large. *We can get rid of this if we change the default to bobyqa before we submit the paper . . . should we comment on how to check the gradient elements?*

```
gm1 <- glmer(incidence/size ~ period + treatment + avg_size + (1 | herd),
             family = binomial,
             data = cbpp2, weights = size,
             control=glmerControl(optimizer="bobyqa"))
```

It turns out that it will also be worthwhile to consider adding an observation-level random effect to the model, which we can do by creating a new factor based on observation number and using `update()` on the previous model:

```
cbpp2 <- transform(cbpp2,obs=factor(seq(nrow(cbpp2))))
gm2 <- update(gm1,.~.+(1|obs))  ## herd and observation-level REs
gm3 <- update(gm1,.~.-(1|herd)+(1|obs))  ## observation-level REs only
```

Model summary

The first part of the summary just reiterates the family and link function used, the model formula, and gives various summary statistics (log-likelihood etc.), as well as quantiles of the scaled (Pearson) residuals:

```
Generalized linear mixed model fit by maximum likelihood ['glmerMod']
 Family: binomial ( logit )
Formula: incidence/size ~ period + treatment + avg_size + (1 | herd)
   Data: cbpp2
Weights: size
Control: glmerControl(optimizer = "bobyqa")

     AIC      BIC   logLik deviance df.resid
   197.8    214.0    -90.9    181.8       48

Scaled residuals:
   Min     1Q Median     3Q    Max
-2.231 -0.797 -0.373  0.469  2.756
```

Outside of `summary()`, these quantities are also accessible via standard accessors (`AIC()`, `BIC()`, `logLik()`, `deviance()`). *Do we want to take the opportunity to sort out the deviance-vs-likelihood mess, as in `https://github.com/lme4/lme4/issues/100`?*

The next chunk of `summary()` describes the random effects and the number of levels associated with each grouping factor (the latter is useful for debugging random-effects formulae):

```
Random effects:
 Groups Name        Variance Std.Dev.
 herd   (Intercept) 0.312    0.558   
Number of obs: 56, groups: herd, 15
```

This information is also accessible via `VarCorr()`, which returns a list of variance-covariance matrices (the `print` method for `VarCorr` objects allows control of whether the variance, or standard deviation, or both, are printed).

Next come the estimates of the fixed effects, along with Wald estimates of the standard error, Z statistic, and p-value:

```
Fixed effects:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.00560    0.70484   -1.43  0.15366
period2          -0.98630    0.30539   -3.23  0.00124
period3          -1.12518    0.32593   -3.45  0.00056
period4          -1.56114    0.42913   -3.64  0.00027
treatmentComplete -0.37620    0.50230   -0.75  0.45388
treatmentUnknown  -0.68319    0.64503   -1.06  0.28953
avg_size         -0.00614    0.04555   -0.13  0.89285
```

One can use `coef(summary())` to retrieve this information, and optionally format it with `printCoefmat()`.

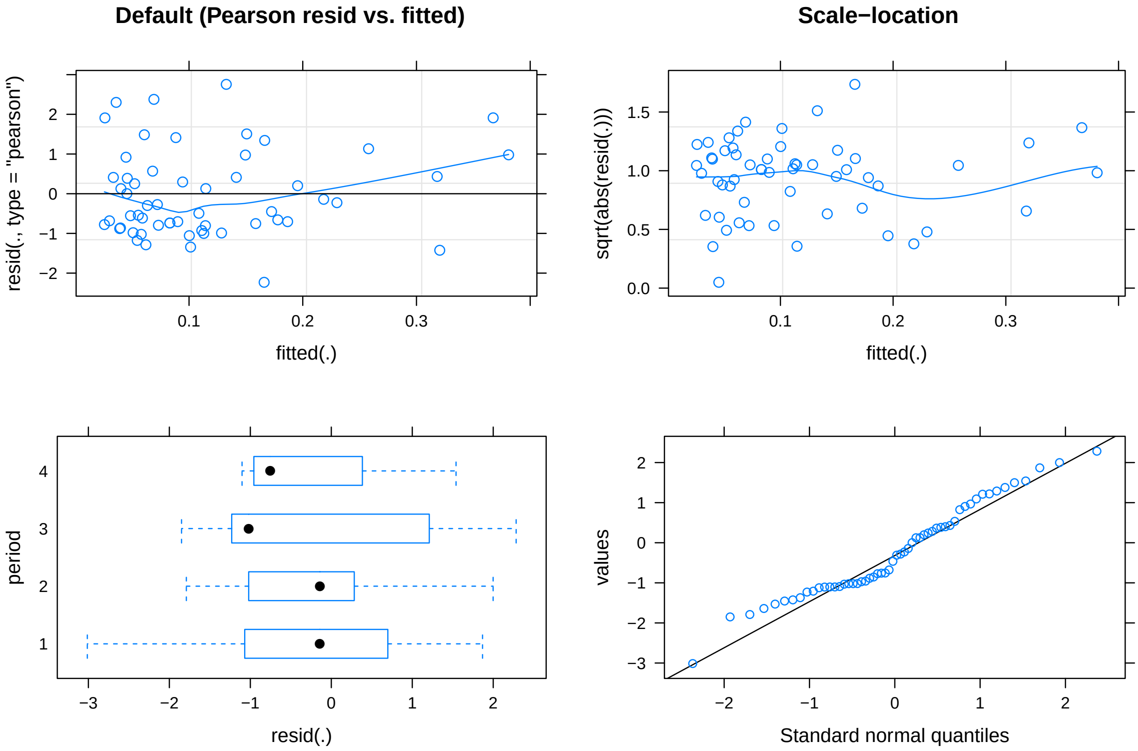
The last component of `summary()` gives the estimated correlations among the fixed-effect parameters, which can be useful for assessing multicollinearity (it can also be overwhelming: it is suppressed by default for models with more than 20 fixed-effect parameters, and can also be suppressed by using `print(summary(.),correlation=FALSE)`).

```
Correlation of Fixed Effects:
            (Intr) perid2 perid3 perid4 trtmnC trtmnU
period2     -0.126                                   
period3     -0.121  0.266                            
period4     -0.078  0.201  0.186                     
trtmntCmplt  0.288 -0.015 -0.019 -0.058              
trtmntUnknw  0.433 -0.053 -0.045 -0.043  0.587       
avg_size    -0.912  0.022  0.024  0.017 -0.545 -0.649
```

Diagnostics

A reasonable range of graphical diagnostic tools is available for `merMod` objects, although not quite as wide as for simpler (non-mixed) models. The plot methods in the `lme4` package are inspired by those in the `nlme` package, using `lattice` plots to provide a reasonable blend of convenience and flexibility.

The following code produces a standard range of plots (Figure 4). At present, `lme4` does not offer other standard diagnostic tools such as influence measures or standardized residuals, due to technical difficulties in computing the hat matrix. *can we fix this?*



Default (Pearson resid vs. fitted)

Scale−location

Figure 4: Graphical diagnostics *needs work!*

```
## basic residual plot
plot(gm1)
## scale-location plot
plot(gm1,sqrt(abs(resid(.)))~fitted(.),type=c("p","smooth"))
## boxplot of residuals grouped by a categorical predictor
plot(gm1,period~resid(.))
## Q-Q plot
qqmath(gm1)
```

The `ranef()` accessor extracts the conditional modes *can we call these "estimates"??*; the argument `condVar=TRUE` additionally extracts the variances of the conditional modes, which are stored as an attribute labelled `"postVar"` — a three-dimensional array that gives the variance-covariance matrix of the conditional modes for each level of the grouping variable. The plotting methods `dotplot()` and `qqmath()` return lists of graphical objects showing *caterpillar plots* (ordered values of the random effects with confidence bars); in the case of the Q-Q plot (`qqmath`) the y-axis shows corresponding values of the standard normal quantiles (Figure 5).

Having checked the diagnostics, we would now like to compare the three models we have fitted. Inspecting the `VarCorr` components, we see that when we fit both herd- and observation-level random effects, the among-herd variance is estimated as zero. The appropriate procedure at this point (e.g. whether one drops non-significant terms, or those with small scaled magnitudes, or those that worsen the AIC or BIC of the model) depends on the goals of the

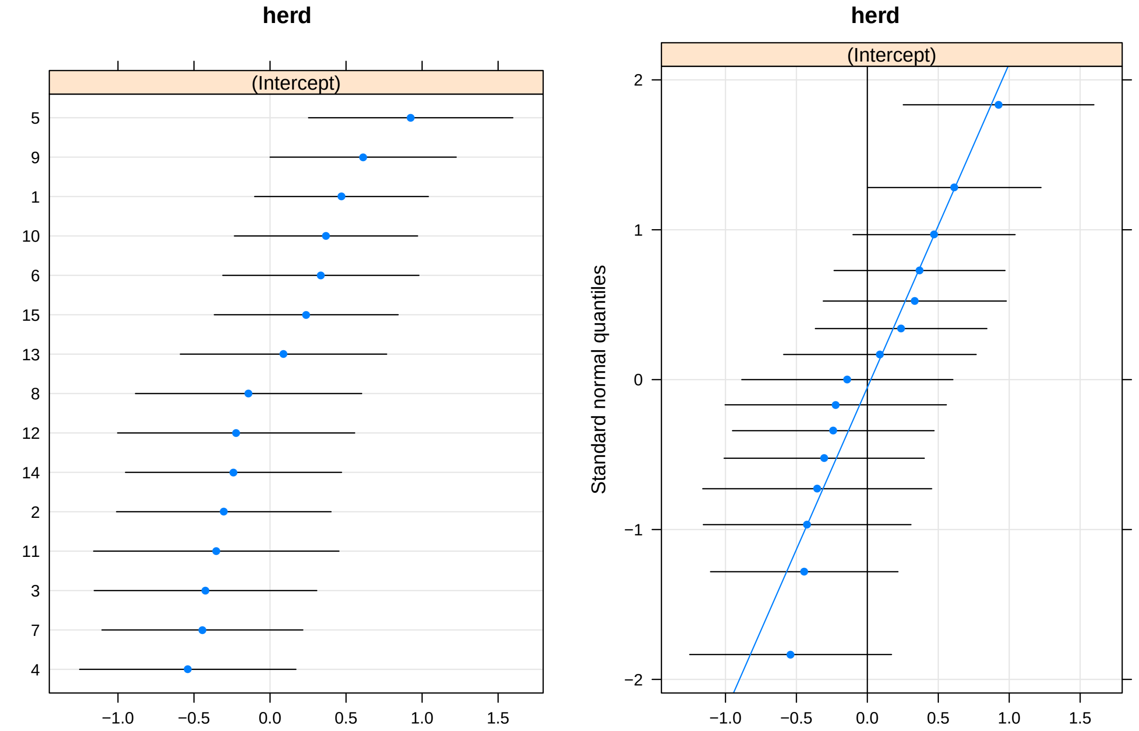


Figure 5: Graphical display of random effects. *Left*: conditional modes ±1.96 × conditional mode, ordered by magnitude. *Right*: quantile-quantile plot, with linear regression line overlaid.

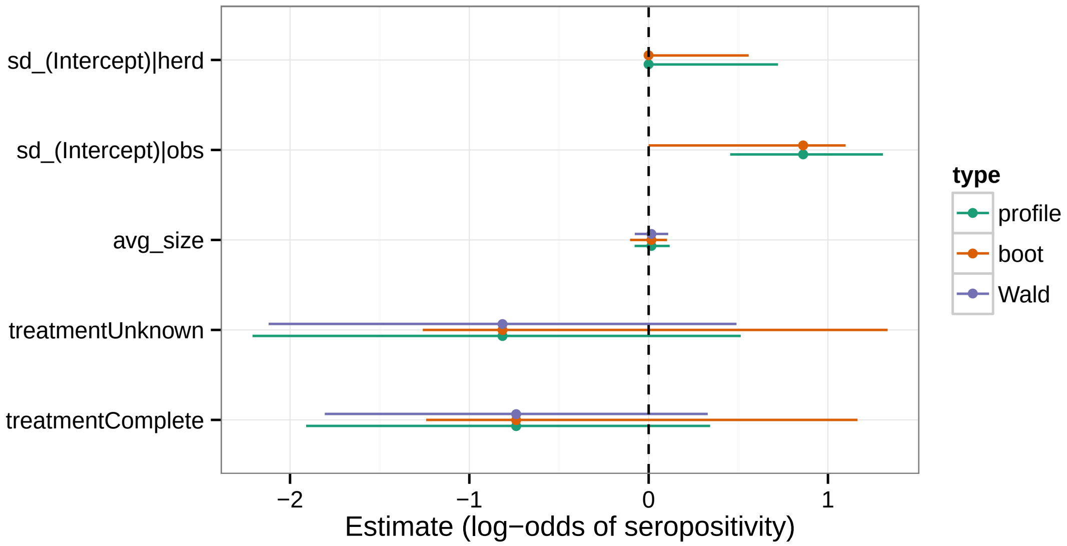


Figure 6: CBPP example: comparison of point and confidence interval estimation for different methods

analysis and one's philosophy of model-building. One might either stick with the full model, or continue with the reduced model with observation-level random effects only (as it has exactly the same likelihood as the full model but uses an additional parameter, it would be chosen according to either an information-theoretic or a hypothesis-testing model selection framework). Here we will start by computing likelihood profiles and confidence intervals for the model incorporating both random effects; although it has the same point estimates and maximum likelihood as the reduced model, confidence intervals that incorporate non-local information (i.e. profile- or parametric bootstrap-based) will give different, more conservative results for the full model. *demonstrate?*

The `profile` method computes profile likelihoods. The computation can be slow, since complete profiling for a model with p random- and fixed-effect parameters requires fitting p profiles, each of which requires many $p-1$-dimensional optimizations. *say any more about internal computations/strategy for profiling? Or add to a vignette somewhere?* The `profile` method returns an object of class `thpr` — a data frame containing the profiles, augmented with attributes containing interpolation splines for each parameter profile and their inverses (using `splines::interpSpline` and `splines::backSpline`); the latter are used for plotting profiles and computing confidence intervals. An `as.data.frame` method adds `.focal` and `.par` variables to the data frame, useful for customized plots.

Profiles can be used for univariate (`xyplot`) and bivariate (`splom`) profile plots, and to compute profile confidence intervals (`confint`). (`confint` applied to a `glmer` fit will first fit the profile, then use it to compute profile confidence intervals. Given the computational cost of profiling, it makes sense to compute and save the profile as an intermediate step if one plans to do anything other than computing confidence intervals.)

```
drop1(gm2,scope=~treatment+avg_size,test="Chisq")

Single term deletions

Model:
incidence/size ~ period + treatment + avg_size + (1 | herd) +
    (1 | obs)
          Df AIC   LRT Pr(Chi)
<none>       190
treatment  2 188 2.037    0.36
avg_size   1 188 0.109    0.74
```

4.2. Contraception

This is the complex example. DB has used this example a lot, e.g. slide set 4 in the recent Lawrence presentations.

- *describe the problem in enough detail.*
- *fit parameters; show effects of AGQ number; describe accessors (prediction, simulation, confidence intervals)*
- *use PB to address finite-size inference issues*

5. Future directions

Cute stuff at the end to make ourselves feel better/future plans:

- *anything inherited from lmer framework: Julia, pureR? improved linear algebra? special-case optimizations?*
- *holes: AGQ for vector-valued (?) RE, condVar for vector-valued RE*
- *reinstating mcmcsamp?*
- *flexLambda*
- *NB, by nesting or by augmented parameter vector*
- *zero-inflation by E-M*

References

Bates DM, Watts DG (1988). *Nonlinear Regression Analysis and Its Applications.* Wiley, Hoboken, NJ. ISBN 0-471-81643-4.

Harrell F (2001). *Regression Modeling Strategies*. Springer. ISBN 0387952322.

Lesnoff M, Laval G, Bonnet P, Abdicho S, Workalemahu A, Kifle D, Peyraud A, Lancelot R, Thiaucourt F (2004). "Within-herd spread of contagious bovine pleuropneumonia in Ethiopian highlands." *Preventive Veterinary Medicine*, **64**(1), 27–40. ISSN 0167-5877. doi:10.1016/j.prevetmed.2004.03.005. URL http://www.sciencedirect.com/science/article/pii/S0167587704000856.

6. Appendix: derivation of PIRLS

We seek to maximize the unscaled conditional log density for a GLMM over the conditional modes, $\boldsymbol{u}$. This problem is very similar to maximizing the log-likelihood for a GLM, about which there is a large amount of work. The standard algorithm for dealing with this kind of problem is iteratively reweighted least squares (IRLS). Here we modify IRLS by incorporating a penalty term that accounts for variation in the random effects, which we call penalized iteratively reweighted least squares (PIRLS).

The unscaled conditional log-density takes the following form,

$$f(\boldsymbol{u}) = \log p(\boldsymbol{y}, \boldsymbol{u}|\boldsymbol{\beta}, \boldsymbol{\theta}) = \boldsymbol{\psi}^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{a}^\top \boldsymbol{\phi} + \boldsymbol{c} - \frac{1}{2}\boldsymbol{u}^\top \boldsymbol{u} - \frac{q}{2}\log 2\pi \tag{24}$$

where $\boldsymbol{\psi}$ is the n-by-1 canonical parameter of an exponential family, $\boldsymbol{\phi}$ is the n-by-1 vector of cumulant functions, $\boldsymbol{c}$ an n-by-1 vector of normalizing constants, and $\boldsymbol{A}$ is an n-by-n diagonal matrix of prior weights, $\boldsymbol{a}$. Both $\boldsymbol{a}$ and $\boldsymbol{c}$ could depend on a dispersion parameter, although we ignore this possibility for now.

The canonical parameter, $\boldsymbol{\psi}$, and vector of cumulant functions, $\boldsymbol{\phi}$, depend on a linear predictor,

$$\boldsymbol{\eta} = \boldsymbol{o} + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\Lambda}_\theta \boldsymbol{u} \tag{25}$$

where $\boldsymbol{o}$ is an n-by-1 vector of *a priori* offsets. The specific form of this dependency is specified by the choice of the exponential family (e.g. binomial). Furthermore, the mean, $\boldsymbol{\mu}$, of this distribution is a function of $\boldsymbol{\eta}$, where this function is standardly referred to as the inverse link function.

Our goal is to find the values of $\boldsymbol{u}$ that maximize the unscaled conditional density, for given $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ vectors. These maximizers are the conditional modes, which we require for the Laplace approximation and adaptive Gauss-Hermite quadrature. To do this maximization we use a variant of the Fisher scoring method, which is the basis of the iteratively reweighted least squares algorithm for generalized linear models. Fisher scoring is itself based on Newton's method, which we apply first.

6.1. Newton's method

To apply Newton's method, we need the gradient and the Hessian of the unscaled conditional log-likelihood. Following standard GLM theory (e.g. McCullagh and Nelder 1989), we use the chain rule,

$$\frac{dL(\boldsymbol{\beta}, \boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{u})}{d\boldsymbol{u}} = \frac{dL(\boldsymbol{\beta}, \boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{u})}{d\boldsymbol{\psi}} \frac{d\boldsymbol{\psi}}{d\boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}} \frac{d\boldsymbol{\eta}}{d\boldsymbol{u}}$$

The first derivative in this chain follow from basic results in GLM theory,

$$\frac{dL(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{\psi}} = (\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{A}$$

Again from standard GLM theory, the next two derivatives define the inverse diagonal variance matrix,

$$\frac{d\boldsymbol{\psi}}{d\boldsymbol{\mu}} = \boldsymbol{V}^{-1}$$

and the diagonal Jaccobian matrix,

$$\frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}} = \boldsymbol{M}$$

Finally, because $\boldsymbol{\beta}$ affects $\boldsymbol{\eta}$ only linearly,

$$\frac{d\boldsymbol{\eta}}{d\boldsymbol{u}} = \boldsymbol{Z}\boldsymbol{\Lambda}_\theta$$

Therefore we have,

$$\frac{dL(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}} = (\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{A}\boldsymbol{V}^{-1}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{u}^\top \tag{26}$$

This is very similar to the gradient for GLMs with respect to fixed effects coefficients, $\boldsymbol{\beta}$. The only difference induced by differentiating with respect to the random effects, $\boldsymbol{u}$, is the addition of the $\boldsymbol{u}^\top$ term.

Again we apply the chain rule to take the Hessian,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{u}} = \frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{\mu}}\frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}}\frac{d\boldsymbol{\eta}}{d\boldsymbol{u}} + \boldsymbol{I}_q \tag{27}$$

which leads to,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{u}} = \frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{\mu}}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q \tag{28}$$

The first derivative in this chain can be expressed as,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{\mu}} = -\boldsymbol{\Lambda}_\theta^\top\boldsymbol{Z}^\top\boldsymbol{M}\boldsymbol{V}^{-1}\boldsymbol{A} + \boldsymbol{\Lambda}_\theta^\top\boldsymbol{Z}^\top\left[\frac{d\boldsymbol{M}\boldsymbol{V}^{-1}}{d\boldsymbol{\mu}}\right]\boldsymbol{A}\boldsymbol{R} \tag{29}$$

where $\boldsymbol{R}$ is a diagonal residuals matrix with $\boldsymbol{y}-\boldsymbol{\mu}$ on the diagonal. The two terms arise from a type of product rule, where we first differentiate the residuals, $\boldsymbol{y}-\boldsymbol{\mu}$, and then the diagonal matrix, $\boldsymbol{M}\boldsymbol{V}^{-1}$, with respect to $\boldsymbol{\mu}$.

The Hessian can therefore be expressed as,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{u}} = -\boldsymbol{\Lambda}_\theta^\top\boldsymbol{Z}^\top\boldsymbol{M}\boldsymbol{A}^{1/2}\boldsymbol{V}^{-1/2}\left(\boldsymbol{I}_n - \boldsymbol{V}\boldsymbol{M}^{-1}\left[\frac{d\boldsymbol{M}\boldsymbol{V}^{-1}}{d\boldsymbol{\mu}}\right]\boldsymbol{R}\right)\boldsymbol{V}^{-1/2}\boldsymbol{A}^{1/2}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q \tag{30}$$

This result can be simplified by expressing it in terms of a weighted random-effects design matrix, $\boldsymbol{U} = \boldsymbol{A}^{1/2}\boldsymbol{V}^{-1/2}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta$,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{u}} = -\boldsymbol{U}^\top\left(\boldsymbol{I}_n - \boldsymbol{V}\boldsymbol{M}^{-1}\left[\frac{d\boldsymbol{V}^{-1}\boldsymbol{M}}{d\boldsymbol{\mu}}\right]\boldsymbol{R}\right)\boldsymbol{U} + \boldsymbol{I}_q \tag{31}$$

6.2. Fisher-like scoring

There are two ways to further simplify this expression for $\boldsymbol{U}^\top\boldsymbol{U}$. The first is to use the canonical link function for the family being used. Canonical links have the property that $\boldsymbol{V} = \boldsymbol{M}$, which means that for canonical links,

$$\frac{d^2 L(\boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{y}, \boldsymbol{u})}{d\boldsymbol{u} d\boldsymbol{u}} = -\boldsymbol{U}^\top \left(\boldsymbol{I}_n - \boldsymbol{I}_n \left[\frac{d\boldsymbol{I}_n}{d\boldsymbol{\mu}} \right] \boldsymbol{R} \right) \boldsymbol{U} + \boldsymbol{I}_q = \boldsymbol{U}^\top \boldsymbol{U} + \boldsymbol{I}_q \tag{32}$$

The second way to simplify the Hessian is to take its expectation with respect to the distribution of the response, conditional on the current values of the spherized random effects coefficients, $\boldsymbol{u}$. The diagonal residual matrix, $\boldsymbol{R}$, has expectation 0. Therefore, because the response only enters into the expression for the Hessian via $\boldsymbol{R}$, we have that,

$$E\left(\frac{d^2 L(\boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{y}, \boldsymbol{u})}{d\boldsymbol{u} d\boldsymbol{u}} | \boldsymbol{u} \right) = -\boldsymbol{U}^\top \left(\boldsymbol{I}_n - \boldsymbol{U}\boldsymbol{M}^{-1} \left[\frac{d\boldsymbol{V}^{-1}\boldsymbol{M}}{d\mu} \right] E(\boldsymbol{R}) \right) \boldsymbol{U} + \boldsymbol{I}_q = \boldsymbol{U}^\top \boldsymbol{U} + \boldsymbol{I}_q \tag{33}$$

6.3. Gauss-Markov

Affiliation:

Steven C. Walker
Department of Mathematics & Statistics
McMaster University
1280 Main Street W
Hamilton, ON L8S 4K1, Canada
E-mail: scwalker@math.mcmaster.ca

Rune Haubo Bojesen Christensen
Technical University of Denmark
Matematiktorvet
Building 324, room 220
2800 Kgs. Lyngby
E-mail: rhbc@dtu.dk

Douglas Bates
Department of Statistics, University of Wisconsin
1300 University Ave.
Madison, WI 53706, U.S.A.
E-mail: bates@stat.wisc.edu

Martin Mächler
Seminar für Statistik, HG G 16
ETH Zurich
8092 Zurich, Switzerland
E-mail: maechler@stat.math.ethz.ch

Benjamin M. Bolker
Departments of Mathematics & Statistics and Biology
McMaster University
1280 Main Street W
Hamilton, ON L8S 4K1, Canada
E-mail: bolker@mcmaster.ca

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd