PSYCHOMETRIKA—VOL. 87, NO. 3, 1173–1193
SEPTEMBER 2022
https://doi.org/10.1007/s11336-022-09840-2

COMPUTATION AND APPLICATION OF GENERALIZED LINEAR MIXED MODEL DERIVATIVES USING *LME4*

TING WANG

AMERICAN BOARD OF FAMILY MEDICINE

BENJAMIN GRAVES

UNIVERSITY OF MISSOURI

YVES ROSSEEL

GHENT UNIVERSITY

EDGAR C. MERKLE

UNIVERSITY OF MISSOURI

Maximum likelihood estimation of generalized linear mixed models (GLMMs) is difficult due to marginalization of the random effects. Derivative computations of a fitted GLMM's likelihood are also difficult, especially because the derivatives are not by-products of popular estimation algorithms. In this paper, we first describe theoretical results related to GLMM derivatives along with a quadrature method to efficiently compute the derivatives, focusing on fitted *lme4* models with a single clustering variable. We describe how psychometric results related to item response models are helpful for obtaining the derivatives, as well as for verifying the derivatives' accuracies. We then provide a tutorial on the many possible uses of these derivatives, including robust standard errors, score tests of fixed effect parameters, and likelihood ratio tests of non-nested models. The derivative computation methods and applications described in the paper are all available in easily obtained R packages.

Key words: generalized linear mixed model, score test, lme4, derivatives, merDeriv.

Maximum likelihood estimation of generalized linear mixed models (GLMMs; e.g., Stroup, 2012) is notoriously complicated due to the fact that random effects are integrated out of the model likelihood. In general, the integrals cannot be solved analytically, which means that we must use numerical methods to approximate the integrals. Along with model estimation, these issues make it difficult to apply other statistical methods to estimated GLMMs, because the required pieces of the estimated model are not generally available. For example, consider the computation of "robust" (Huber-White) standard errors (e.g., Huber, 1967; White, 1980), as applied to GLMM. In addition to the model's maximum likelihood estimates, we require first and second partial derivatives of the model's likelihood function. These derivatives also require integral approximations, which do not necessarily arise as by-products of the model estimation algorithm.

Of primary importance for this paper, the partial derivatives do not arise as by-products of model estimation via the *lme4* package (Bates, Mächler, Bolker, & Walker, 2015). This package uses a penalized, iteratively re-weighted least squares (PIRLS) algorithm that indirectly maximizes the marginal likelihood by optimizing a second function that involves conditional random effects

This work was supported by NSF grant 1460719.

Correspondence should be made to Edgar C. Merkle, University of Missouri, Columbia, MO, USA. Email: merklee@missouri.edu

(conditional on random effect (co-)variances; Bates, 2021). Although this conditional approach bypasses the difficult integration, it also loses the ability to produce the likelihood derivatives of interest. This makes it difficult to apply many relevant methods that are already implemented within the R ecosystem, including sandwich estimators from package *sandwich* (Zeileis, 2004, 2006; Zeileis, Köll, & Graham, 2020), score-based tests from *strucchange* (Zeileis et al., 2002), model-based recursive partitioning from *partykit* 2015, and Vuong tests from *nonnest2* (Hothorn & Zeileis, 2018). These packages all rely on partial derivatives of the model likelihood function (evaluated at the maximum likelihood estimates, *after* model estimation), which to date have not been available for GLMMs estimated by *lme4*. So the overall goal of this work is to connect existing statistical methods with GLMMs estimated by *lme4*. The paper's contributions toward this goal include (i) theoretical background on GLMM derivatives, as well as a quadrature method that capitalizes on the fact that we are dealing with estimated models; (ii) a general-purpose implementation of the methods via the *merDeriv* package; and (iii) a tutorial on how these derivatives can be used in applied research settings, including a variety of R examples.

Our derivations are informed by previous results from both statistics and psychometrics, which include diverse motivations for the GLMM. In particular, the statistics community often views the GLMM as an extension of the linear mixed model, whereas the psychometrics community additionally considers connections between the GLMM and item response theory (IRT) models (e.g., De Boeck et al., 2011; Doran et al., 2007). The latter connections are seldom noticed in the statistics literature, though Skrondal and Rabe-Hesketh (2004) is noteworthy in that LMMs, GLMMs, and IRT models are included within a larger latent variable framework. We describe below how IRT results can help us obtain derivatives of the GLMM likelihood function with respect to both fixed parameters and random effect hyperparameters (e.g., random effect variances) after model estimation.

In the following sections, we first fix notation and define the GLMM. We then present theoretical results related to derivatives of the GLMM likelihood function, including a quadrature method that can be applied to estimated models. Next, we provide a tutorial on the application of these results to GLMMs estimated via *lme4*. This is accomplished with the help of R package *merDeriv* (Wang & Merkle, 2018), which implements the methods described here, combined with other packages like *mirt* (Chalmers, 2012), *sandwich*, *nonnest2*, and *strucchange*. Finally, we discuss potential future extensions of our work.

1. Theoretical Background

Our presentation of the GLMM follows the *lme4* framework of Bates et al. (2015), which facilitates the R applications presented later. This framework encompasses a variety of GLMMs from the exponential family, with binomial models being especially popular. The framework does not allow for products between free parameters and random effects, which becomes important when we discuss relationships between GLMMs and IRT models below (also see De Boeck et al., 2011; Doran et al., 2007).

1.1. Model and Notation

Let $\boldsymbol{y}_i$ be a vector containing the response variable for the ith cluster, each entry of which is assumed to follow a specific probability distribution (e.g., binomial or Poisson). The sample size of cluster i is denoted as n_i, so the total sample size across all I clusters is given as $N = \sum_{i=1}^{I} n_i$. Let $\boldsymbol{X}_i$ be the $n_i \times p$ design matrix corresponding to fixed effects for cluster i; $\boldsymbol{\beta}$ is the fixed effect vector of length p; $\boldsymbol{Z}_i$ is the $n_i \times q$ design matrix corresponding to random effects for cluster i; and $\boldsymbol{u}_i$ is the random effect vector of length q. Then, the model can be written as

$$E(\boldsymbol{y}_i|\boldsymbol{u}_i, \boldsymbol{\Lambda_\theta}) = \boldsymbol{\mu}_i|\boldsymbol{\Lambda_\theta}, \boldsymbol{u}_i \tag{1}$$

$$\boldsymbol{\mu}_i = g^{-1}(\boldsymbol{\eta}_i|\boldsymbol{\Lambda_\theta}, \boldsymbol{u}_i) \tag{2}$$

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

$$\boldsymbol{b}_i = \boldsymbol{\Lambda_\theta}\boldsymbol{u}_i \tag{4}$$

$$\boldsymbol{u}_i \sim N(\boldsymbol{0}, \boldsymbol{I}_q). \tag{5}$$

The above equations express the idea that the bounded support of the expected value of $\boldsymbol{y}_i$ can be transformed to an unbounded support of the linear combination $\boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i$ through the link function $g()$. The random effects are in $\boldsymbol{b}_i$, which equals $\boldsymbol{\Lambda_\theta}\boldsymbol{u}_i$. The vector $\boldsymbol{u}_i$ follows the standard normal distribution $N(\boldsymbol{0}, \boldsymbol{I}_q)$, with $\boldsymbol{\Lambda_\theta}$ being the relative covariance factor, which can be seen as the Cholesky decomposition of the usual random effect covariance matrix $\boldsymbol{G}$. Reparameterizing $\boldsymbol{b}_i$ as the product of the relative covariance factor and standard normal distribution makes it easier to compare GLMM to IRT. We provide further discussion of this comparison in the next section.

Following the above notation, the model's log-likelihood (marginal over random effects) can be expressed as

$$\ell = \sum_{i=1}^{I} \ell_i = \sum_{i=1}^{I} \log \int f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i) f_{\boldsymbol{u}_i}(\boldsymbol{u}_i) d\boldsymbol{u}_i, \tag{6}$$

where I represents the number of total clusters. We further define the following "across-cluster" matrices:

$$\boldsymbol{y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_i, \ldots, \boldsymbol{y}_I\} \tag{7}$$

$$\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_i, \ldots, \boldsymbol{X}_I\} \tag{8}$$

$$\boldsymbol{Z} = \{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_i, \ldots, \boldsymbol{Z}_I\} \tag{9}$$

$$\boldsymbol{b} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_i, \ldots, \boldsymbol{b}_I\}. \tag{10}$$

1.2. GLMM Scores

One of the most popular IRT models is the two-parameter logistic model (e.g., Embretson & Reise, 2000; Lord & Novick, 1968), which can be viewed as a binomial GLMM with logit link function. Consider an IRT model parameterized as $\text{logit}^{-1}(p_{ij}) = \alpha_j\theta_i - \beta_j$, with each item j's difficulty described by β_j and discrimination described by α_j. The alternative parameterization as $\alpha_j(\theta_i - \beta_j)$ is also applicable, but less convenient for comparison. In the former parameterization, the IRT β_j parameters are similar to the negative of the GLMM fixed parameter $\boldsymbol{\beta}$. The IRT α_j parameters are then similar to the relative covariance factor in the GLMM, with the *lme4* package requiring the covariance factor to be equal for all items. This means that we cannot fit a 2PL model in *lme4*, though other GLMM software such as SAS PROC NLMIXED may allow for 2PL estimation.

In the context of IRT, Glas (1992; 1998; 1999) utilized an identity from Louis (1982) to obtain first derivatives of the marginal log-likelihood (marginal over person parameters θ_i). This identity can be used to show that the first derivative of the marginal log-likelihood with respect to difficulty and discrimination parameters equals an expected value involving first derivatives of the conditional likelihood (conditioned on person proficiency). That is, we can obtain derivatives of the marginal likelihood by taking an expected value that involves the conditional likelihood.

The same idea can be applied to GLMM (McCulloch & Neuhaus, 2001), where conditioning on person proficiency is replaced with conditioning on random effects. In the next sections, we will formalize these GLMM score derivations. Please note that, throughout this paper, *scores* refer to first derivatives of the clusterwise log-likelihood function with respect to some model parameters. They are different from *factor scores* and from *scoring* in psychometrics, which involve prediction of a model's random parameters.

1.2.1. Fixed Effect Scores Drawing on derivations by Glas as well as by McCulloch and Neuhaus (2001), the GLMM score with respect to the fixed effect parameter $\boldsymbol{\beta}$ can be expressed in the following form:

$$\frac{\partial \ell_i}{\partial \boldsymbol{\beta}} = \frac{\displaystyle\int \frac{\partial \log f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i)}{\partial \boldsymbol{\beta}} f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i) f_{\boldsymbol{u}_i}(\boldsymbol{u}_i) d\boldsymbol{u}_i}{f_{\boldsymbol{y}_i}(\boldsymbol{y}_i)}, \tag{11}$$

where $f_{\boldsymbol{y}_i}(\boldsymbol{y}_i) = \displaystyle\int f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i) f_{\boldsymbol{u}_i}(\boldsymbol{u}_i) d\boldsymbol{u}_i$.

The first term in the numerator of Equation (11) can be seen as the score of a Generalized Linear Model (GLM), which can be expressed in matrix form as

$$\frac{\partial \log f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i)}{\partial \boldsymbol{\beta}} = \boldsymbol{X}_i^T \boldsymbol{D}_i^{-1} \boldsymbol{V}_i^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_i), \tag{12}$$

where $\boldsymbol{D}_i$ and $\boldsymbol{V}_i$ are $n_i \times n_i$ diagonal matrices with diagonal entries as $\frac{\partial(\eta_t|u_t)}{\partial(\mu_t|u_t)}$ and $a(\phi_t)\mathrm{Var}(\mu_t|u_t)$, respectively. The t subscript indexes an observation within cluster i, 1, 2, ..., n_i. Further, the $a(\phi_t)$ function is unique to each distribution from the exponential family. For example, $a(\phi_t) = 1$ for the binomial distribution and for the Poisson distribution. The value of $a(\phi_t)$ for other exponential family distributions can be found in, e.g., McCullagh and Nelder (1989). Many of the relevant derivations are also supplied by the R `family()` function. Note that, if we use the canonical link function, $\frac{\partial(\eta_t|u_t)}{\partial(\mu_t|u_t)}$ and $\mathrm{Var}(\mu_t|u_t)$ will cancel out. This feature creates a shortcut for distributions using the canonical link.

The second term in the numerator of Equation (11) is the distribution of the GLM given $\boldsymbol{u}_i$. We use the following matrix form to express all distributions belonging to the exponential family:

$$f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i) = \exp\left(\boldsymbol{y}_i^T \boldsymbol{A}_i \boldsymbol{\kappa}_i - \boldsymbol{1}^T \boldsymbol{A}_i h(\boldsymbol{\kappa}_i) + c(\boldsymbol{y}_i, \boldsymbol{\psi}_i)\right), \tag{13}$$

where $\boldsymbol{A}_i$ is a $n_i \times n_i$ diagonal matrix with diagonal element as $\frac{1}{a(\phi_t)}$; $\boldsymbol{\kappa}_i$ is the vector of canonical parameters; $\boldsymbol{1}$ is a $n_i \times 1$ vector with each entry as 1; $h(\boldsymbol{\kappa}_i)$ is an $n_i \times 1$ vector defined by applying the distribution-specific function $h()$ to each element of $\boldsymbol{\kappa}_i$; and $c(\boldsymbol{y}_i, \boldsymbol{\psi}_i)$ is an $n_i \times 1$ vector of remaining terms not depending on $\boldsymbol{\kappa}_i$, with $\boldsymbol{\psi}_i$ containing scale parameters. For exponential distributions, these terms can also be found in McCullagh and Nelder (1989) or in the R `family()` functions.

The above results based on generalized linear models are straightforward, while the difficulty involves the integration over $\boldsymbol{u}$. In the same spirit, the denominator can be viewed as the integration of the GLM distribution over the random variable $\boldsymbol{u}$. Both integrals have no closed form for GLMMs. We discuss use of quadrature to approximate the integrals below, after describing derivatives of random effect hyperparameters.

1.2.2. Random Effect Hyperparameter Scores Following the same type of derivation, the scores w.r.t. the random effect hyperparameters can be seen as the scores w.r.t. parameters in the $\boldsymbol{\Lambda_\theta}$ matrix. The derivation can thus be expressed as:

$$\frac{\partial \ell_i}{\partial \boldsymbol{\Lambda_\theta}} = \frac{\displaystyle\int \frac{\partial \log f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i)}{\partial \boldsymbol{\Lambda_\theta}} f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i) f_{\boldsymbol{u}_i}(\boldsymbol{u}_i) d\boldsymbol{u}_i}{f_{\boldsymbol{y}_i}(\boldsymbol{y}_i)}, \tag{14}$$

where $\frac{\partial \log f_{\boldsymbol{y}_i|\boldsymbol{u}_i}(\boldsymbol{y}_i|\boldsymbol{u}_i)}{\partial \boldsymbol{\Lambda_\theta}}$ equals $\boldsymbol{u}_i^T \frac{\partial \boldsymbol{\Lambda_\theta}}{\partial \theta} \boldsymbol{Z}_i^T (\boldsymbol{y}_i - \boldsymbol{\mu}_i)$, with $\frac{\partial \boldsymbol{\Lambda_\theta}}{\partial \theta}$ as a matrix composed of 1s (corresponding to a particular random effect hyperparameter θ) and 0s (not corresponding to a particular random effect hyperparameter θ). This derivation is similar to the score derivation for the IRT discrimination parameter. An equivalent approach is to rearrange terms using the trace operator (e.g., Petersen & Pedersen, 2012), which results in the expression $\text{Tr}\left((\boldsymbol{Z}_i^T(\boldsymbol{y}_i - \boldsymbol{\mu}_i)\boldsymbol{u}_i^T)^T \frac{\partial \boldsymbol{\Lambda_\theta}}{\partial \theta}\right)$.

1.2.3. Reparameterization As mentioned above, $\boldsymbol{\Lambda_\theta}$ is a Cholesky decomposition of the usual variance–covariance matrix $\boldsymbol{G}$, so our derivations are taken with respect to the Cholesky decomposition. In order to obtain the scores with respect to the variance–covariance parameters contained in $\boldsymbol{G}$, we utilize the chain rule:

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{G}} &= \frac{\partial \ell}{\partial \boldsymbol{\Lambda_\theta}} \frac{\partial \boldsymbol{\Lambda_\theta}}{\partial \boldsymbol{G}} &\quad (15)\\ &= \frac{\partial \ell}{\partial \boldsymbol{\Lambda_\theta}} \left\{ \frac{\partial \boldsymbol{\Lambda_\theta}}{\partial (\boldsymbol{\Lambda_\theta}\boldsymbol{\Lambda_\theta}^T)} \right\} &\quad (16)\\ &= \frac{\partial \ell}{\partial \boldsymbol{\Lambda_\theta}} \left\{ \frac{\partial (\boldsymbol{\Lambda_\theta}\boldsymbol{\Lambda_\theta}^T)}{\partial \boldsymbol{\Lambda_\theta}} \right\}^{-1}. &\quad (17)\end{aligned}$$

For the entry in row i and column j of $\boldsymbol{\Lambda_\theta}$, we have that

$$\frac{\partial (\boldsymbol{\Lambda_\theta}\boldsymbol{\Lambda_\theta}^T)}{\partial \boldsymbol{\Lambda}_{\boldsymbol{\theta}ij}} = \boldsymbol{\Lambda_\theta} \boldsymbol{J}_{ji} + \boldsymbol{J}_{ij} \boldsymbol{\Lambda_\theta}^T, \tag{18}$$

where $\boldsymbol{J}_{ij}$ is a matrix with entry (i, j) equal to 1 and 0 elsewhere. The derivatives with respect to all unique, nonzero entries of $\boldsymbol{\Lambda_\theta}$ can be computed in this manner to obtain the desired scores.

As an alternative to variances and covariances, users may wish to parameterize the model via standard deviations and correlations. The scores with respect to standard deviations and correlations can be obtained by applying another chain rule to the above scores that are taken with respect to $\boldsymbol{G}$. For example, assume a GLMM with two correlated random effects. In the variance–covariance parameterization, we would have parameters σ_0^2, σ_1^2, and σ_{01}, while, in the standard deviation-correlation parameterization, we would have parameters σ_0, σ_1, and ρ. Derivatives for the latter parameterization are:

$$\begin{aligned}\frac{\partial \ell}{\sigma_0} &= \frac{\partial \ell}{\partial \sigma_0^2} \frac{\partial \sigma_0^2}{\partial \sigma_0} &\quad (19)\\ &= \frac{\partial \ell}{\partial \sigma_0^2}(2\sigma_0) &\quad (20)\end{aligned}$$

$$\frac{\partial \ell}{\partial \sigma_1} = \frac{\partial \ell}{\partial \sigma_1^2}\frac{\partial \sigma_1^2}{\partial \sigma_1} \tag{21}$$

$$= \frac{\partial \ell}{\partial \sigma_1^2}(2\sigma_1) \tag{22}$$

$$\frac{\partial \ell}{\partial \rho} = \frac{\partial \ell}{\partial \rho\sigma_0\sigma_1}\frac{\partial \rho\sigma_0\sigma_1}{\partial \rho} \tag{23}$$

$$= \frac{\partial \ell}{\partial \sigma_{01}}(\sigma_0\sigma_1). \tag{24}$$

1.2.4. Quadrature All the derivatives above involve integrals that marginalize over the model random effects $\boldsymbol{u}$. These integrals do not have closed forms, requiring numerical methods for approximation. The method implemented in R package *merDeriv* is a simplified version of multivariate adaptive Gauss–Hermite quadrature (Liu & Pierce, 1994; Naylor & Smith, 1982), with the simplifications being based on the fact that we are computing derivatives *after* model estimation. This means that we already have information about posterior modes and variances of random effects from *lme4*, and we can make use of this information in place of the "adaptive" part of the algorithm. Merkle et al. (2019) recently used a similar method to compute marginal versions of Bayesian information criteria (see especially their Appendix C), with that method being based on earlier methods described by Pinheiro and Bates (1995) and Rabe-Hesketh et al. (2005). While it would be possible to simply use a traditional adaptive quadrature method here, we would have to use it separately for each case in the data (because we seek to compute casewise derivatives). This would be much slower and infeasible for many datasets, as compared to our quadrature method described here.

Focusing on the GLMM framework, the integrals from Equations (11) and (14) are both of the form

$$\int g(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\omega}) f_{\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\omega}}(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\omega}) f_{\boldsymbol{u}|\boldsymbol{\omega}}(\boldsymbol{u}|\boldsymbol{\omega}) d\boldsymbol{u}, \tag{25}$$

where $g()$ differs depending on the integral, and $\boldsymbol{\omega}$ is a vector of model parameters excluding the random effects $\boldsymbol{u}$. This conditioning on $\boldsymbol{\omega}$ is implicit in earlier sections but was excluded to simplify notation.

For a single clustering variable with I levels, the clusters i are independent. Therefore, the above equation can be written as

$$\prod_{i=1}^{I} \int g(\boldsymbol{y}_i|\boldsymbol{u}_i,\boldsymbol{\omega}) f_{\boldsymbol{y}_i|\boldsymbol{u}_i,\boldsymbol{\omega}}(\boldsymbol{y}_i|\boldsymbol{u}_i,\boldsymbol{\omega}) f_{\boldsymbol{u}_i|\boldsymbol{\omega}}(\boldsymbol{u}_i|\boldsymbol{\omega}) d\boldsymbol{u}_i. \tag{26}$$

To compute scores, we are interested in the elements of the above product: the integral for each cluster i. For M quadrature points, we use Gauss–Hermite quadrature to approximate the integral for cluster i by:

$$\sum_{m=1}^{M} w_{im}^{*} g(\boldsymbol{y}_i|\boldsymbol{a}_{im}^{*},\boldsymbol{\omega}) f_{\boldsymbol{y}_i|\boldsymbol{u}_i,\boldsymbol{\omega}}(\boldsymbol{y}_i|\boldsymbol{a}_{im}^{*},\boldsymbol{\omega}). \tag{27}$$

That is, the integral is approximated by a weighted sum of function evaluations, where the functions are evaluated at different random effect values represented by $\boldsymbol{a}_{im}^{*}$, $m = 1, \ldots, M$. For a random effect of dimension d, the quadrature locations and weights are computed by

$$\boldsymbol{a}^*_{im} = \tilde{\boldsymbol{b}}_i + \tilde{\boldsymbol{C}}_i \times \boldsymbol{a}_m \tag{28}$$

$$w^*_{im} = w_m \times (2\pi)^{d/2} \times \det(\tilde{\boldsymbol{C}}_i) \times \exp(0.5 \times \boldsymbol{a}_m \boldsymbol{a}'_m) \times \boldsymbol{\phi}(\boldsymbol{a}^*_{im} | \boldsymbol{0}, \hat{\boldsymbol{G}}) \tag{29}$$

where $\tilde{\boldsymbol{b}}_i$ are the posterior modes of random effects for cluster i, $\tilde{\boldsymbol{C}}_i$ is the Cholesky factor of the *conditional* covariance matrix of the random effects for cluster i (obtained from the *lme4* function `ranef()`), $\boldsymbol{\phi}()$ is the normal density function, and $\hat{\boldsymbol{G}}$ is the estimated covariance matrix of the random effects (obtained from the *lme4* function `VarCorr()`). Finally, $\boldsymbol{a}_m$ and w_m are the usual Gauss–Hermite locations and weights, respectively.

1.2.5. Second Derivatives While we have focused on first derivatives, the Louis (1982) identity can also aid in computation of second derivatives, leading to the model Hessian and information matrix. We do not present the equations here because, for models estimated via `glmer()` (but not `lmer()`), a Hessian is already computed and stored in the resulting model object (specifically in the `optinfo` slot). According to the *lme4* documentation, this Hessian is computed using a finite difference approach. The *merDeriv* package provides a convenience function to access this Hessian, and we use it in our applications later.

By default, the *lme4* Hessian is parameterized via the Cholesky decomposition of random effects. The Hessian based on the standard deviation/correlation parameterization can alternatively be obtained via the `devfun2()` function in *lme4*, which uses the profile likelihood. The Hessian for the variance/covariance parameterization is then related to the latter option, through the chain rule mentioned earlier. The *merDeriv* package incorporates these computations and enables researchers to request the parameterization of interest via the `ranpar` argument (taking possible values of `"var"`, `"sd"`, or `"theta"`). This is illustrated in the tutorials below.

2. Tutorial on the Derivatives' Uses in R

We now provide a tutorial on R package *merDeriv*, which can carry out the computations described above and which can be used to solve applied problems. As we go, we provide snippets of code that illustrate how *merDeriv* interacts with other packages, which readers can adapt to other models and datasets. We first provide some evidence that *merDeriv* operates in the manner expected, by comparing a Rasch model estimated via *lme4* to a Rasch model estimate via *mirt* (Chalmers, 2012). We then consider a variety of other applications.

2.1. Verifying the Computations

Before using the scores from *merDeriv* in GLMM applications, we use the relationship between GLMM and IRT to verify the correctness of the quadrature implementation. We specifically compare the score computations to those of package *mirt* (Chalmers, 2012), which estimates many types of item response models. We make use of the fact that the Rasch model can be estimated as a generalized linear mixed model, which was illustrated by De Boeck et al. (2011). We also make use of the fact that *mirt* has its own, independent quadrature method for score computation, which was used by Schneider et al. (2020) to apply Vuong tests to item response models.

```r
## mirt:
library("mirt")
ls7 <- expand.table(LSAT7)
mirtmod <- mirt(ls7[,1:5], 1, itemtype = "Rasch", SE = TRUE)

## reshape data and fit with glmer():
library("reshape2")
ls7$person <- 1:nrow(ls7)
ls7long <- melt(ls7, id = "person")
lme4mod <- glmer(value ~ -1 + variable + (1 | person),
                 family = binomial, data = ls7long, nAGQ = 5L)

## score calculation:
mirtsc <- estfun.AllModelClass(mirtmod)
lme4sc <- estfun.glmerMod(lme4mod, ranpar = "var")
```

FIGURE 1.
Code to fit Rasch models using *mirt* and *lme4*, then calculate scores

2.1.1. Method For comparing the two score computation algorithms, we use the LSAT7 data (Bock & Lieberman, 1970) included with *mirt*. This dataset includes the item responses (correct/incorrect) of 1,000 individuals across 5 items of the LSAT.

The code in Fig. 1 shows how a Rasch model can be fit to the data using both *mirt* and *lme4*. For *mirt*, we require the LSAT7 data to be arranged in wide format, where each row is a person and each column is an item. If we then rearrange the data to be in long format, as shown in Fig. 1, we can fit the Rasch model via *lme4*. We use the `nAGQ` argument to employ adaptive quadrature during *lme4* model estimation, avoiding the `glmer()` default, `nAGQ=1`, which uses the Laplace approximation. The quadrature leads to a more accurate approximation of the model log-likelihood, which in turn leads to maximum likelihood estimates that tend to be closer to the true maximum of the likelihood. The *mirt* package employs a fixed quadrature method with 61 quadrature points.

2.1.2. Results As shown at the bottom of Fig. 1, scores for the two models are obtained via their respective `estfun()` functions. The function for *mirt* models is included directly within the *mirt* package, whereas the function for *lme4* models is included in *merDeriv*. Both functions output a score matrix, where rows index people and columns index model parameters. For the `glmer` model, we use the `ranpar` argument so that the *merDeriv* scores involve the variance–covariance parameterization, which matches the *mirt* output.

In comparing the two sets of scores, we arrive at Fig. 2. The x-axis depicts scores from *merDeriv*, the y-axis depicts scores from *mirt*, and each point is a particular score. We see that the values are nearly exactly equal for *mirt* and for *merDeriv*, falling directly on the identity line. One can also compare the parameter variance–covariance matrix of *merDeriv* and of *mirt*, using the `vcov()` method of each package. That comparison, not shown, exhibits agreement similar to the score comparison. These provide evidence that the *merDeriv* code is performing as expected. Now that we have obtained this evidence, we move on to illustrate practical uses of the scores in GLMM applications.

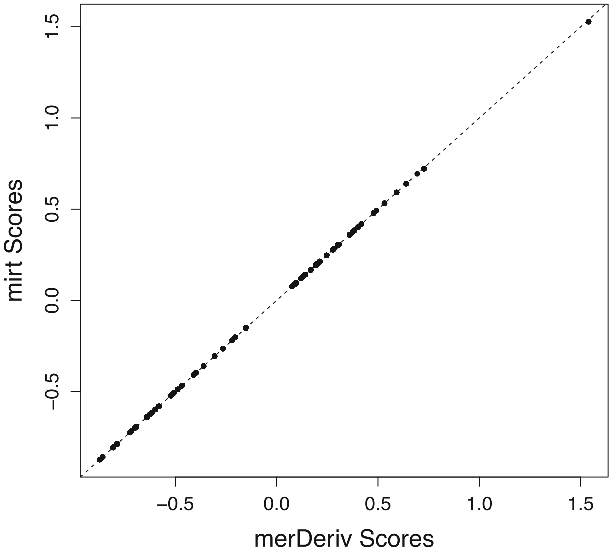

FIGURE 2.
Comparison of Rasch model scores from *mirt* and from *merDeriv*.

TABLE 1.
Comparison between Rasch model standard errors reported by *lme4* and robust standard errors reported by *sandwich* The β columns correspond to item difficulties, while the σ^2 column corresponds to person (intercept) variance.

	β_1	β_2	β_3	β_4	β_5	σ^2
lme4	0.1004	0.0811	0.0913	0.0787	0.1037	0.1300
sandwich	0.0996	0.0814	0.0898	0.0785	0.1058	0.1311

2.2. *Huber-White estimator*

Let $\boldsymbol{\omega}$ be the model parameter vector, which in a GLMM would contain fixed effect parameters and random effect (co-)variances. Then the Huber-White (e.g., Huber, 1967; White, 1980) sandwich estimator of the covariance matrix of $\boldsymbol{\omega}$ is

$$\boldsymbol{V}(\hat{\boldsymbol{\omega}}) = (\boldsymbol{A})^{-1}\boldsymbol{B}(\boldsymbol{A})^{-1}, \tag{30}$$

where $\boldsymbol{A}$ is the negative expectation of the model Hessian and $\boldsymbol{B}$ is the covariance matrix of scores (see Wang & Merkle, 2018, for further discussion in the context of linear mixed models). The score computations described in the previous sections facilitate computation of this $\boldsymbol{B}$ matrix. The square root of the diagonal elements of $\boldsymbol{V}$ are then typically called "robust standard errors."

Robust standard errors are used to address model misspecifications such as unmodeled dependence between observations or deviations from normality. While random effects are typically used in GLMMs to account for dependence between observations, the Huber-White estimator can be used on top of a GLMM to account for further model misspecifications. Further, Stroup and Claassen (2020) recently provided evidence that quadrature can lead to downward-biased variance estimates in GLMMs, resulting in inflated Type I error rates. The Huber-White estimator may be considered in light of this result.

```
library("sandwich")
sandwich(lme4mod, bread. = bread.glmerMod, meat. =
         meat(lme4mod, level = 2))
```

FIGURE 3.
Example code for calculating Huber-White standard errors.

We can easily compute Huber-White standard errors using the scores from the previous section, paired with the *sandwich* package, as shown in Fig. 3. In that figure, the `bread.glmerMod()` and `meat()` functions come from *merDeriv*, while `sandwich()` comes from the *sandwich* package. Applying this result to the Rasch model estimated in the previous section, we obtain the results in Table 1. For this particular application, the *lme4* standard errors and *sandwich* standard errors are virtually equal, likely due to the large sample size (large by GLMM standards, at least).

2.3. *Score Tests*

Researchers have long been familiar with score tests, also known as Lagrange multiplier tests, that can be used as alternatives to the likelihood ratio test or to the Wald test (e.g., Engle, 1984; Glas, 1992; 1998; 1999). In typical score test applications, a constrained model is fit to data, then first derivatives of the likelihood function are used to test whether or not some constraint should be relaxed. In contrast, the likelihood ratio test requires us to estimate two models (a constrained model and an unconstrained model), and the Wald test requires us to estimate only the unconstrained model.

This score test framework has expanded to a class of “parameter instability” tests, where we test whether an estimated model’s parameters differ with respect to unmodeled auxiliary variables (with different test statistics being used for continuous, ordinal, or discrete auxiliary variables). Zeileis and Hornik (2007) summarized much previous work on this topic, developing a family of score-based tests that can be used within an M-estimation framework (of which maximum likelihood estimation is a special case). They also developed R package *strucchange* (Zeileis et al., 2002), which can be used to compute the test statistics so long as a model’s scores and Hessian are available. The family of score-based tests has subsequently been studied in the context of many specific types of models, including linear mixed models (Wang et al., 2020), structural equation models (Merkle & Zeileis, 2013; Merkle et al., 2014), and item response models (Komboz et al., 2018; Strobl et al., 2015; Wang et al., 2018). The developments in the current paper make it possible to apply score-based tests to GLMMs, yielding test statistics for GLMMs that have been unavailable up to now. The score computations described above can be used to construct the cumulative scores, which are further used to compute test statistics (Merkle & Zeileis, 2013; Merkle et al., 2014).

In this section, we show how scores can be used to test fixed effect parameters that are not directly included in a GLMM model. This is potentially useful in situations where a model with the fixed effect included does not converge, which often happens in applied mixed modeling (see Barr et al., 2013; Matuschek et al. 2017). In these situations, if we can get a model to converge *without* some fixed effect of interest, it is possible to apply score-based tests to the fitted model in order to test the omitted fixed effect. While the more popular approach here is to drop random effects (as opposed to fixed effects) from the model, dropping fixed effects may be useful in instances where, e.g., the random effect variances are all large, yet the model still exhibits convergence problems.

2.3.1. *Method*

We use data from 500 respondents on the Nerdy Personality Attributes Scale (NPAS), a personality test designed for personal entertainment on the Open Source Psychometrics

```r
## Model that has problems with convergence:
m1 <- glmer(answer ~ -1 + item*ext + (1 | subject),
            data = npas.sampled, family = binomial)

## Model with only main effects, which converges:
m2 <- glmer(answer ~ -1 + item + ext + (1 | subject),
            data = npas.sampled, family = binomial,
            control = glmerControl(optimizer='bobyqa'))

## Score test:
ext <- with(npas.sampled,
            as.numeric(tapply(ext, subject, head, 1)))

sc1 <- sctest(m2, fit=NULL, scores=estfun.glmerMod, order.by=ext,
              parm=1:26, functional='maxLMo')
```

FIGURE 4.
Models of the NPAS data. The first model has issues with non-convergence, leading us to the simpler, second model. A score test is then used to study the interaction.

Project website (Open Source Psychometrics Project, n.d.). The questionnaire consists of 26 items that attempt to define the concept of “nerdiness.” Responses were originally measured on 5-point Likert scales, but we converted them to binary responses for this example (where 0 corresponds to 3 or less and 1 corresponds to 4 or 5). The items ask about different aspects of nerdiness, including hobbies and interests that are usually associated with nerds, social interactions, personality traits, and academic or intellectual endeavors. The data also include various demographic variables and other personality measures assessing the “Big Five” personality factors.

Here, we assess whether item responses vary across extraversion, while also accounting for inherent item differences (which would be called “item difficulties” in an IRT context). The *lme4* syntax for this model is shown at the top of Fig. 4, where the variable names are generally self-explanatory. Note that inclusion of the interaction term (`item*ext`) automatically includes main effects of both item and extraversion, in addition to the interaction. This GLMM can be viewed as a person-by-covariate item response model, falling into the class of explanatory item response models considered by De Boeck and Wilson (2004) and De Boeck et al. (2011).

2.3.2. Results The first model in Fig. 4 did not converge, even after making changes to the optimizer and its settings. We could have experimented further, perhaps finding some combination of settings that led to a converged model and that would render the score test unnecessary. But each attempted model estimation took about ten minutes, so we could easily have spent hours tweaking the settings. In contrast, the score test could be immediately applied to a simpler model that converged more easily.

Our simpler model was the second model in Fig. 4, which used the `bobyqa` optimizer (Powell, 2009) instead of the default `Nelder_Mead`. In estimating this second model, we capitalize on the fact that score tests require only a “constrained” model, which here assumes that responses to items do not vary across levels of extraversion. We can then obtain a score test statistic for the interaction without directly including the interaction in the model.

To obtain a test statistic for this interaction, we carry out the score test using the code at the bottom of Fig. 4. This makes use of the `sctest()` function found in the R package *strucchange* as well as `estfun.glmerMod()` function found in *merDeriv*. It simultaneously tests all 26 item parameters for fluctuations with respect to extraversion, which is similar to including an

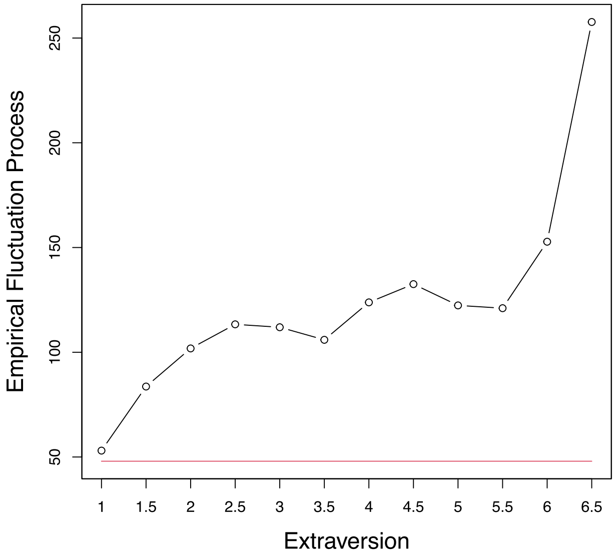

FIGURE 5.
M-fluctuation test for NPAS data. This graph presents item parameter fluctuation across varying levels of extraversion. Peaks of the graph suggest extraversion cutpoints that isolate individuals with similar item parameters.

item × extraversion interaction. Results of this test are visualized in Fig. 5, which shows how the scores fluctuate across different values of extraversion (x-axis). We can see that there is significant parameter fluctuation in items across levels of extraversion, because the black line goes above the "critical value" that is depicted by the red line. The peaks in the black line, around extraversion values of 2.5 and 4.5, suggest cutpoints for subgroups of individuals that exhibit similar item parameters. The test provides information about the nature of the interaction that was not easily obtained by including extraversion in the model, due to model convergence problems.

2.4. Vuong Tests

Scores also play a role in Vuong tests (Vuong, 1989), which can be used to compare nested and non-nested models to one another. In the nested case, the tests can be viewed as extensions of the traditional likelihood ratio test, which, unlike the traditional likelihood ratio test, make no assumptions about the more complex model being correct. In the non-nested case, the tests provide a formal way of comparing the fits of the two models. The scores described in this paper can be used in tandem with package *nonnest2* (Merkle & You, 2018) to compare GLMMs, providing new capabilities for comparing models with different predictor variables and different random effects. Specifically, our score computations are used to compute the null distribution of the test statistic, which is a weighted sum of chi-square distributions. Further descriptions of the tests and applications to psychometric models can be found in Merkle, You, and Preacher (2016) and in Schneider et al. (2020). An illustration involving GLMMs is provided here.

2.4.1. Method The data used for this example comes from the SPISA data set, which can be found in the R package *psychotree* (Strobl et al., 2015). The data is a subsample of 1,075 Bavarian university students who took an online, general knowledge quiz called "Studentenpisa" administered by a German weekly news magazine (Trepte & Verbeet, 2010). The quiz consists of 45 items on 5 topics, and we focus here on a subset of nine questions dealing with natural science.

```
mod1 <- glmer(response ~ -1 + item + agecent + gender + (1 | pnum),
              data = spisa, family = binomial,
              control = glmerControl(optimizer='bobyqa'))

mod2 <- glmer(response ~ -1 + item + semester + elite + (1 | pnum),
              data = spisa, family = binomial,
              control = glmerControl(optimizer='bobyqa'))
```

FIGURE 6.
Code for the non-nested models to be compared using the Vuong test. The first model uses age and gender as potential predictors, while the second model uses number of semesters at the university and elite university status.

The data set includes several covariates such as age, gender, semester of university enrollment, and elite university status.

Using a similar item response model as in the previous example, we construct two non-nested models with different covariates. These models are based on a common reduced model that only contains item and subject effects. The first model uses age and gender as covariates, while the second model uses semester of university enrollment and whether the student's university has been granted "elite" status or not. The code for these models is shown in Figure 6. Similar to the previous application, the models here did not immediately converge, and we switched optimizers in order to attain convergence. Following model estimation, we obtained scores and compared the two models using a Vuong test computed via the R package *nonnest2* (Merkle & You, 2018).

2.4.2. Results The *nonnest2* code and output for the Vuong test is shown in Figure 7. First, we create a convenience function, `vcg()`, to compute the full parameter covariance matrix (including random effect variances/covariances) for each of the models. This function, along with functions from *merDeriv* for calculating the likelihoods and scores, is then sent to `vuongtest()`.

The output from the function first shows a variance test, which provides information about whether the non-nested models are distinguishable from each other via the observed dataset. From this, we reject the hypothesis that the models are indistinguishable from one another. We then move on to the non-nested likelihood ratio test to examine whether one model fits better than the other. For our example, we conclude that neither model fits better than the other.

Figure 8 shows that the *nonnest2* functionality can also be used to test nested models, by adding the `nested = TRUE` argument. We first fit a simple Rasch model to the data, with this model being nested in the two considered previously. We then compute test statistics comparing this model to the second model from Fig. 6. The two test statistics in the output can each be used to compare the nested models, providing two alternatives to the traditional likelihood ratio test. Here, we conclude that the full model including the "semester" and "elite" predictors fits better than the simple Rasch model without those predictors.

2.5. Poisson GLMMs

Of course, the GLMM framework is not limited solely to binomial models, and our derivations extend to other exponential family models. In this section, we illustrate extensions to the Poisson GLMM using the epilepsy data set (Thall & Vail, 1990) found in the package *brms* (Bürkner, 2018).

2.5.1. Method The data consist of 236 observations of seizure counts from 59 people across 4 time periods. Covariates include study group (treatment vs control), participant age, and a base

```
vcg <- function(obj) vcov(obj, full = TRUE)

vuongtest(mod1, mod2, ll1 = llcont.glmerMod, ll2 = llcont.glmerMod,
          score1 = estfun.glmerMod, score2 = estfun.glmerMod,
          vc1 = vcg, vc2 = vcg)


Model 1
 Class: glmerMod
 Call: glmer(formula = response ~ -1 + item + agecent + gender + (1 | ...

Model 2
 Class: glmerMod
 Call: glmer(formula = response ~ -1 + item + semester + elite + (1 | ...

Variance test
  H0: Model 1 and Model 2 are indistinguishable
  H1: Model 1 and Model 2 are distinguishable
    w2 = 0.033,   p = 6.35e-07

Non-nested likelihood ratio test
  H0: Model fits are equal for the focal population
  H1A: Model 1 fits better than Model 2
    z = -0.356,   p = 0.639
  H1B: Model 2 fits better than Model 1
    z = -0.356,   p = 0.3611
```

FIGURE 7.
Code to run Vuong test for comparing two non-nested models. The models are able to be distinguished from each other, but one model does not have better fit over the other.

rate seizure count across 8-weeks (standardized). For our initial model, we predict number of seizures using the patient's base rate (`zBase`), treatment group indicator (`Trt`), and visit number (`visit`). We allow the intercept and `visit` slope to vary by participant, with these two random effects being correlated. The *lme4* code for this model is at the top of Fig. 9.

Because this model includes multiple random effects, *lme4* requires that we use the Laplace approximation (`nAGQ = 1`) for estimation. We can still choose a larger number of quadrature points for score computation after model estimation, however, which provides more precise approximations of these quantities. We can also use extra quadrature points to compute the model's log-likelihood (via the *merDeriv* command `llcont.glmerMod()`), which potentially yields a log-likelihood that is more precise than the log-likelihood that is output by *lme4*.

2.5.2. Results We first used *merDeriv* to repeatedly compute the log-likelihood and the standardized gradient of the estimated Poisson GLMM, using one to ten quadrature points per dimension (the gradient is obtained by summing scores across people). Some of those results are shown in Fig. 10, where the left panel displays results for the log-likelihood and the right panel displays results for the standardized gradient of a single model parameter (the fixed intercept). We see that, for small numbers of quadrature points, both of the displayed quantities are unstable. The log-likelihood varies by about a tenth of a point, while the standardized gradient varies by much more. Both quantities stabilize around five quadrature points, however, suggesting that we should use at least that many points in practice (while also considering total computation time). We also remark that the log-likelihood reported by *lme4* is the value in the left panel at 1 quadrature point, which is somewhat different from the "stabilized" value at larger numbers of quadrature

```
mod3 <- glmer(response ~ -1 + item + (1 | pnum), data = spisa,
              family = binomial,
              control = glmerControl(optimizer='bobyqa'))

vuongtest(mod2, mod3, nested = TRUE,
          ll1 = llcont.glmerMod, ll2 = llcont.glmerMod,
          score1 = estfun.glmerMod, score2 = estfun.glmerMod,
          vc1 = vcg, vc2 = vcg)


Model 1
 Class: glmerMod
 Call: glmer(formula = response ~ -1 + item + semester + elite + (1 | ...

Model 2
 Class: glmerMod
 Call: glmer(formula = response ~ -1 + item + (1 | pnum), data = spisa, ...

Variance test
  H0: Model 1 and Model 2 are indistinguishable
  H1: Model 1 and Model 2 are distinguishable
    w2 = 0.017,   p = 0.000109

Robust likelihood ratio test of distinguishable models
  H0: Model 2 fits as well as Model 1
  H1: Model 1 fits better than Model 2
    LR = 18.680,   p = 9.07e-05
```

FIGURE 8.
Code for testing fit of two nested models. The full model has better fit than the reduced model.

```
## linear effect of visit number:
epilepsy$visit <- as.numeric(epilepsy$visit)

## Poisson model:
poimod <- glmer(count ~ zBase * Trt * visit + (visit | patient),
                data = epilepsy, family = poisson)

## Robust standard errors:
rse <- sandwich(poimod, bread. = bread.glmerMod,
                meat. = meat(poimod, level = 2))

## Score-based test with 5 quadrature points:
age <- with(epilepsy, tapply(Age, patient, head, 1))
efg5 <- function(...) estfun.glmerMod(..., nAGQ = 5)

poisc <- sctest(poimod, fit = NULL, scores = efg5,
                order.by = age, parm = 3, functional = 'maxLMo')
```

FIGURE 9.
Code to fit a Poisson GLMM predicting the number of seizures in epileptic patients, then compute robust standard errors and a score test statistic.

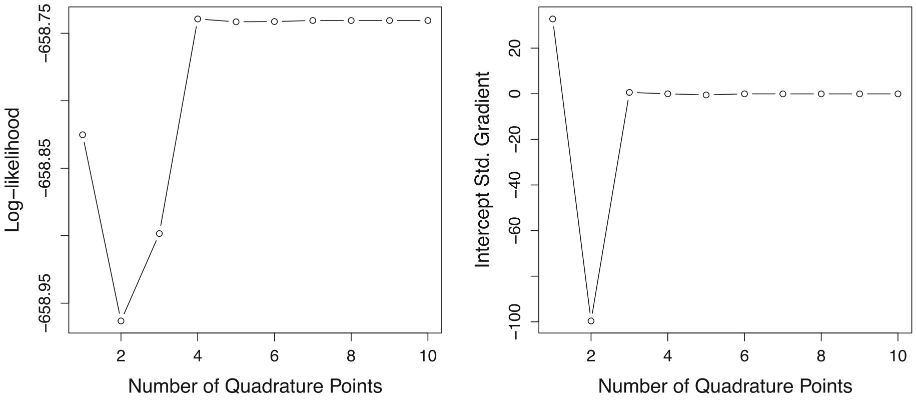

FIGURE 10.
Log-likelihood and standardized gradient of the Poisson mixed model, by number of quadrature points used. The standardized gradient shown is that of the model's fixed intercept parameter.

TABLE 2.
Comparison of model generated standard errors to robust standard errors for Poisson model.

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
lme4	0.1575	0.1465	0.2194	0.0468	0.1915	0.0404	0.0656	0.0518
sandwich	0.2464	0.1588	0.2300	0.0771	0.1554	0.0452	0.0679	0.0412

points. We can obtain a more accurate approximation of the fitted model's log-likelihood using the methods described here, and this approximation could influence some likelihood ratio tests or other statistics that rely on the model's log-likelihood.

We now illustrate how methods from the previous sections can be applied to the Poisson GLMM. We first calculate robust standard errors using the code in the middle of Fig. 9, with Table 2 showing the results. The table shows that, for the model considered here, the Huber-White standard errors are generally larger.

Similarly to the previous section on score-based tests, we next examine the Poisson GLMM parameter fluctuation across an extraneous variable. In this example, we assess the stability of the treatment main effect across patient age, which provides information about whether the treatment efficacy varies for patients of different ages. The score test is carried out via the code at the bottom of Fig. 9, which is similar to that used in the score test section above. The test statistic here (not shown) indicates that the parameter fluctuation is not significant, suggesting that the treatment effect does not fluctuate across the range of age. Figure 11 contains the parameter fluctuation across values age, with the critical value being the red horizontal line.

As can be seen, our methods work for other exponential family GLMMs, with the code remaining very similar. In the General Discussion below, we provide further detail about models that our methods cannot handle, as well as future extensions.

3. General Discussion

In this paper, we have provided technical details on computing derivatives of the GLMM likelihood function, with a focus on models estimated via package *lme4*. We then showed how the

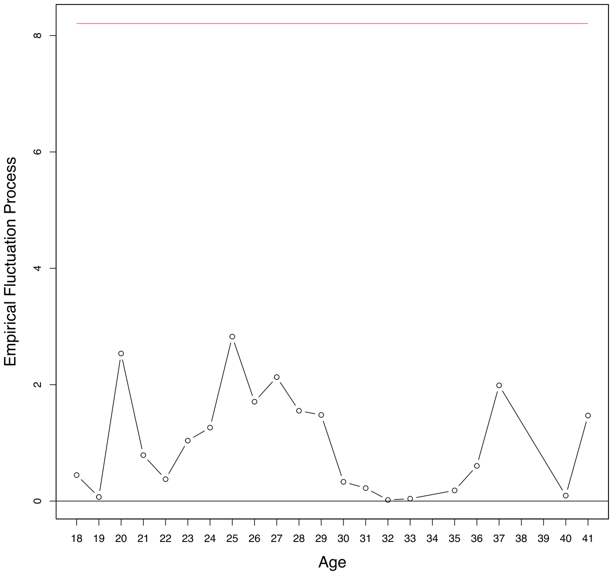

FIGURE 11.
Graph of M-Fluctuation test for Poisson model. Model parameters are stable across the range of age.

derivatives can be used in various manners: to obtain robust standard errors, to test predictors that were not included in the estimated model, and to carry out Vuong tests of non-nested GLMMs. All of these applications used the GLMM derivatives in concert with other R packages, illustrating how the R infrastructure can be combined to obtain new statistics that were difficult or impossible to obtain previously.

3.1. Computational Issues

The quadrature implementation described in this paper can be used to obtain derivatives of the marginal likelihood function for many models with random effects. This method is especially relevant because the conditional random effects $\boldsymbol{b}$ and corresponding components in the variance–covariance matrix $\boldsymbol{G}$ are often employed in the model estimation process, in place of derivatives (for example, Cai, 2010a, 2010b; Bauer & Curran, 2004). Therefore, the derivatives based on the marginal distribution are often not available, or at least not easy to obtain. Our quadrature method took advantage of the fact that the model was already estimated, so that the predicted modes of the random parameters were available.

Another integral approximation method is the Laplace approximation, which is equivalent to Gauss–Hermite quadrature with one quadrature point (McCulloch & Neuhaus, 2005). Thus, the Laplace approximation is less accurate than Gauss–Hermite quadrature with multiple points, but also less computationally intensive and more flexible (Stroup, 2012). Additionally, it is possible to use derivatives associated with the pseudo maximum likelihood function, which is a transformation of the y response variable into $y^{\star}$, which conditions on the random effect (Stroup, 2012). The scores are then related to a simpler GLM, with such a procedure being implemented in SAS (Schabenberger, 2005). However, the scores based on this pseudo likelihood are not always applicable because the estimates can be problematic, such as when y follows a two-parameter exponential family distribution or sparse Bernoulli distribution (Nelder & Lee, 1992). Finally, numerical methods and Monte Carlo can be flexibly applied to many types of derivative computations, but they are often too slow to be practical. In all, these remarks indicate that there is not a single, superior method for all scenarios. The quadrature method described here is flexible and appears to work well enough for many types of models.

3.2. Additional Applications

There exist other relevant applications that are worth exploring in more detail, including use of the derivatives in GLMM trees. GLMM trees are part of a model-based recursive partitioning framework that has been developed by Zeileis and colleagues (Hothorn & Zeileis, 2015). The goal of the framework is to split a dataset into homogeneous subsamples based on auxiliary variables, where each subsample exhibits different values of model parameters. To accomplish this, a tree is constructed via the following steps

1. Fit the model of interest to the data in the current node of the tree.
2. Conduct a score-based test for each auxiliary variable.
3. Split the current node into two nodes, based on the auxiliary variable with the largest test statistic.
4. Repeat steps 1–3 for the two nodes that were just created.

This procedure is continued until the score-based tests indicate no parameter instabilities with respect to any auxiliary variables (or until a minimal subsample size is reached).

Fokkema, Smits, Zeileis, Hothorn, and Kelderman (2018) recently applied model-based recursive partitioning to GLMMs. But, due to the difficulty of obtaining scores associated with GLMMs, they developed an alternative procedure where only fixed effect parameters were allowed to vary across subgroups. The developments in the current paper make it possible to apply the original, model-based recursive partitioning procedure to GLMMs, allowing us to detect new types of GLMM heterogeneity in a tree-based framework.

In addition to trees, scores may be used to study heterogeneity through "on the fly" tests of residual covariance structures in GLMMs. These developments could reduce computation time by testing multiple covariance structures after fitting a single model, as opposed to requiring estimation of one model per covariance structure. Such tests can be facilitated by the `coeftest()` function of package *lmtest* (Zeileis & Hothorn, 2002), though some *merDeriv* extensions may be necessary before this works.

3.3. Limitations

While the derivations in this paper work for general, exponential family models, two-parameter distributions such as the gamma and inverse Gaussian are additionally complicated by estimation of the extra dispersion parameter. The current *merDeriv* implementation does not currently handle some of these models, nor does it handle the quasi-Poisson or quasi-binomial families (which are not based on formal likelihood functions). Additionally, the applications in this paper take advantage of the fact that we focused on models with a single clustering variable. Researchers often consider three-level models and models with crossed or partially crossed random effects, though, which utilize multiple clustering variables. The derivations in this paper generally work for those models, allowing us to obtain scores for each case in the data (i.e., for each row of the data). But most of the applications in this paper require a way to split observations into independent groups, which is often impossible when we have multiple clustering variables. For example, individuals in separate groups under one clustering variable may appear in the same group under another clustering variable, leading to different forms of dependence between different pairs of individuals' scores. In contrast, when there is only one clustering variable, we know that individuals in one group are independent of individuals in other groups.

For models with multiple clustering variables, it may be possible to de-correlate scores after the fact, using an appropriately specified covariance matrix (Zeileis, 2004; Zeileis et al., 2020) or a self-normalization technique that is commonly used in time series research (Shao & Zhang, 2010; Zhang, et al., 2011). This would allow us to split observations into uncorrelated groups, which may be sufficient for applications. Alternatively, Rasbash and Goldstein (1994) describe methods for re-specifying a model with crossed random effects to be a fully hierarchical model, in which case it may be possible to directly use the results described in this paper. None of these solutions is trivial, and we hope to further study them in the future. We aspire to a future version of *merDeriv* that is able to handle all of the models that *lme4* can estimate.

Computational Note

All results were obtained using the R system for statistical computing (R Core Team, 2020), version 4.1.2, employing the add-on package *merDeriv* 0.2-3 for derivative computations and *lme4* 1.1-27.1 (Bates et al., 2015) for fitting of the mixed models. Code to reproduce the results in the paper is available at https://osf.io/58ruw/.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language, 68*, 255–278.

Bates, D. (2021). Computational methods for mixed models. lme4 Package Vignette. Retrieved from https://cran.r-project.org/web/packages/lme4/vignettes/Theory.pdf

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Bauer, D. J., & Curran, P. J. (2004). The integration of continuous and discrete latent variable models: Potential problems and promising opportunities. *Psychological Methods, 9*(1), 3–29.

Bock, R. D., & Lieberman, M. (1970). Fitting a response model for n dichotomously scored items. *Psychometrika, 35*, 179–198.

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal, 10*(1), 395–411. https://doi.org/10.32614/RJ-2018-017

Cai, L. (2010). High-dimensional exploratory item factor analysis by a Metropolis-Hastings Robbins-Monro algorithm. *Psychometrika, 75*(1), 33–57.

Cai, L. (2010). A two-tier full-information item factor analysis model with applications. *Psychometrika, 75*(4), 581–612.

Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software, 48*(6), 1–29. https://doi.org/10.18637/jss.v048.i06

De Boeck, P., Bakker, M., Zwitser, R., Nivard, M., Hofman, A., Tuerlinckx, F., & Partchev, I. (2011). The estimation of item response models with the lmer function from the lme4 package in R. *Journal of Statistical Software, 39*(12), 1–28.

De Boeck, P., & Wilson, M. (2004). *Explanatory item response models: A generalized linear and nonlinear approach*. New York: Springer.

Doran, H., Bates, D., Bliese, P., & Dowling, M. (2007). Estimating the multilevel Rasch model: With the lme4 package. *Journal of Statistical Software, 20*(2), 1–18. https://doi.org/10.18637/jss.v020.i02

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Mahwah, NJ: Erlbaum Associates.

Engle, R. F. (1984). Wald, likelihood ratio, and Lagrange multiplier tests in econometrics. In Z. Griliches & M. D. Intriligator (Eds.), *Handbook of econometrics.* Elsevier.

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., & Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods, 50,* 2016–2034. Retrieved from http://link.springer.com/article/10.3758/s13428-017-0971-x

Glas, C. A. W. (1992). A Rasch model with a multivariate distribution of ability. *Objective measurement: Theory into practice, 1*, 236–258.

Glas, C. A. W. (1998). Detection of differential item functioning using Lagrange multiplier tests. *Statistica Sinica, 8*(3), 647–667.

Glas, C. A. W. (1999). Modification indices for the 2-PL and the nominal response model. *Psychometrika, 64*, 273–294.

Hothorn, T., & Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research, 16,* 3905–3909. Retrieved from http://jmlr.org/papers/v16/hothorn15a.html

Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 221–233).

Komboz, B., Strobl, C., & Zeileis, A. (2018). Tree-based global model tests for polytomous Rasch models. *Educational and Psychological Measurement, 78*(1), 128–166.

Liu, Q., & Pierce, D. A. (1994). A note on Gauss-Hermite quadrature. *Biometrika, 81*, 624–629.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological), 44*(2), 226–233.

Matuschek, H., Kliegl, R., Vasishth, S., Baayen, H., & Bates, D. (2017). Balancing Type I error and power in linear mixed models. *Journal of Memory and Language, 94*, 305–315.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. Boca Raton, FL: Chapman & Hall.

McCulloch, C. E., & Neuhaus, J. M. (2001). *Generalized linear mixed models*. Wiley.

McCulloch, C. E., & Neuhaus, J. M. (2005). Generalized linear mixed models. In P. Armitage & T. Colton (Eds.), *Encyclopedia of Biostatistics*. https://doi.org/10.1002/0470011815.b2a10021

Merkle, E. C., Fan, J., & Zeileis, A. (2014). Testing for measurement invariance with respect to an ordinal variable. *Psychometrika, 79*, 569–584.

Merkle, E. C., Furr, D., & Rabe-Hesketh, S. (2019). Bayesian comparison of latent variable models: Conditional versus marginal likelihoods. *Psychometrika, 84*, 802–829.

Merkle, E. C., & You, D. (2018). nonnest2 : Tests of non-nested models [Computer software manual]. Retrieved from https://cran.r-project.org/package=nonnest2 (R package version 0.5- 2)

Merkle, E. C., You, D., & Preacher, K. J. (2016). Testing nonnested structural equation models. *Psychological Methods, 21*(2), 151–163.

Merkle, E. C., & Zeileis, A. (2013). Tests of measurement invariance without subgroups: A generalization of classical methods. *Psychometrika, 78*, 59–82.

Naylor, J. C., & Smith, A. F. M. (1982). Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society C, 31*, 214–225.

Nelder, J., & Lee, Y. (1992). Likelihood, quasi-likelihood and pseudolikelihood: some comparisons. *Journal of the Royal Statistical Society: Series B (Methodological), 54*(1), 273–284.

Open Source Psychometrics Project. (n.d.). Open psychology data: Raw data from online [Computer software manual]. Retrieved 2017-10-17, from https://openpsychometrics.org/_rawdata/

Petersen, K. B., & Pedersen, M. S. (2012). The matrix cookbook. Technical University of Denmark. Retrieved from http://www2.imm.dtu.dk/pubdb/p.php3274 (Version 20121115)

Pinheiro, J. C., & Bates, D. M. (1995). Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational Graphics and Statistics, 4*, 12–35.

Powell, M. J. D. (2009). The BOBYQA algorithm for bound constrained optimization without derivatives. Report No. DAMTP 2009/NA06, Centre for Mathematical Sciences, University of Cambridge, UK. http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.

R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2005). Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics, 128*(2), 301–323.

Rasbash, J., & Goldstein, H. (1994). Efficient analysis of mixed hierarchical and cross-classified random structures using a multilevel model. *Journal of Educational and Behavioral Statistics, 19*, 337–350.

Schabenberger, O. (2005). Introducing the GLIMMIX procedure for generalized linear mixed models. SUGI 30 Proceedings, 196.

Schneider, L., Chalmers, R. P., Debelak, R., & Merkle, E. C. (2020). Model selection of nested and non-nested item response models using Vuong tests. *Multivariate Behavioral Research, 55*, 664–684.

Shao, X., & Zhang, X. (2010). Testing for change points in time series. *Journal of the American Statistical Association, 105*(491), 1228–1240.

Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation modeling*. Boca Raton, FL: Chapman & Hall.

Strobl, C., Kopf, J., & Zeileis, A. (2015). Rasch trees: A new method for detecting differential item functioning in the Rasch model. *Psychometrika, 80*(2), 289–316. https://doi.org/10.1007/s11336-013-9388-3

Stroup, W. W. (2012). *Generalized linear mixed models: Modern concepts, methods and applications*. USA: CRC Press.

Stroup, W. W., & Claassen, E. (2020). Pseudo-likelihood or quadrature What we thought we knew, what we think we know, and what we are still trying to figure out. *Journal of Agricultural. Biological and Environmental Statistics, 25*(4), 639–656.

Thall, P. F., & Vail, S. C. (1990). Some covariance models for longitudinal count data with overdispersion. *Biometrics, 46*(3), 657–671.

Trepte, S., & Verbeet, M. (Eds.). (2010). Allgemeinbildung in Deutschland–erkenntnisse aus dem SPIEGEL Studentenpisa-Test. Wiesbaden: VS Verlag.

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica, 57*, 307–333.

Wang, T., & Merkle, E. C. (2018). merDeriv: Derivative computations for linear mixed effects models with application to robust standard errors. *Journal of Statistical Software, 87*(1), 1–16. https://doi.org/10.18637/jss.v087.c01

Wang, T., Merkle, E. C., Anguera, J. A., & Turner, B. M. (2020). Score-based tests for detecting heterogeneity in linear mixed models. *Behavior Research Methods, 53*, 216–231.

Wang, T., Strobl, C., Zeileis, A., & Merkle, E. C. (2018). Score-based tests of differential item functioning via pairwise maximum likelihood estimation. *Psychometrika, 83*(1), 132–155.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica: Journal of the Econometric Society, 48*(4), 817–838.

Zeileis, A. (2004). Econometric computing with HC and HAC covariance matrix estimators. *Journal of Statistical Software, 11*(10), 1–17. Retrieved from http://www.jstatsoft.org/v11/i10/

Zeileis, A. (2006). Object-oriented computation of sandwich estimators. *Journal of Statistical Software, 16*(9), 1–16. https://doi.org/10.18637/jss.v016.i09

Zeileis, A., & Hornik, K. (2007). Generalized M-fluctuation tests for parameter instability. *Statistica Neerlandica, 61*, 488–508.

Zeileis, A., & Hothorn, T. (2002). Diagnostic checking in regression relationships. *R News, 2*(3), 7–10. Retrieved from https://CRAN.R-project.org/doc/Rnews/

Zeileis, A., Köll, S., & Graham, N. (2020). Various versatile variances: An object-oriented implementation of clustered covariances in R. *Journal of Statistical Software, 95*(1), 1–36. https://doi.org/10.18637/jss.v095.i01

Zeileis, A., Leisch, F., Hornik, K., & Kleiber, C. (2002). strucchange: An R package for testing for structural change in linear regression models. *Journal of Statistical Software, 7*(2), 1–38. Retrieved from http://www.jstatsoft.org/v07/i02/

Zhang, X., Shao, X., Hayhoe, K., & Wuebbles, D. J. (2011). Testing the structural stability of temporally dependent functional observations and application to climate projections. *Electronic Journal of Statistics, 5*, 1765–1796.

Manuscript Received: 23 OCT 2020
Final Version Received: 5 NOV 2021
Published Online Date: 3 FEB 2022