# Fast Estimation of Nested Multilevel Structural Equation Models

Joshua Nathaniel Pritikin
Charlottesville, VA

M.A. Psychology, University of Virginia, 2013
B.S. Psychology, University of Oregon, 2009

A Dissertation presented to the Graduate Faculty
of the University of Virginia in Candidacy for the Degree of
Doctor of Philosophy

Department of Psychology

University of Virginia

May, 2016 (anticipated)

Abstract

*Not written yet.*

# Contents

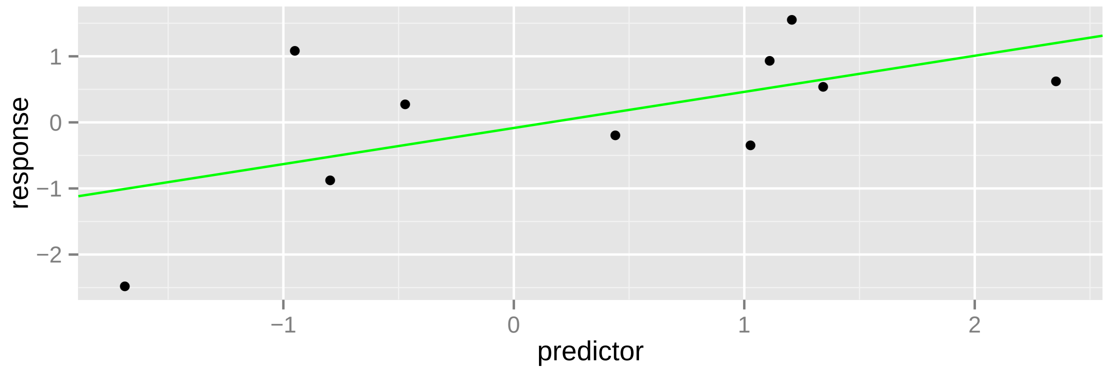


*Figure 1*. Data are shown as points with the least squared residual regression line.

## Introduction

Many non-statisticians have an intuitive notion of variability of a indicator and association between two indicators. We cannot entertain causal theories without these notions. When an infant learns that crying will cause her parents to offer her water, food, and a diaper change, these statistical engines are probably at work. Not all processes are best described by a Gaussian distribution. However, the non-Gaussian part is often confined to the outer vertices of a casual graph while the central part of the graph remains Gaussian. The Gaussian distribution is of central importance in statistics and causal reasoning (Pearl, 2000; Voelkle & Oud, 2013).

## Gaussian Models

Let parameter vector $\boldsymbol{\theta} \equiv \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ with $\boldsymbol{\mu}$ as a $K$ dimensional mean vector (1st moment) and $\boldsymbol{\Sigma}$ as a $K \times K$ covariance matrix (2nd moment). For data $\boldsymbol{y}$ and with some regularity assumptions, the Gaussian log density can be written as,

$$\ell(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_i \left[ -\frac{1}{2} \left[ K \log(2\pi) + \log(|\boldsymbol{\Sigma}|) \right] - \frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{y}_i)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \boldsymbol{y}_i) \right]. \qquad (1)$$

It is no overstatement to say that this model has a rich history in the annals of statistics.

Similar to the way that some countries that were slow to implement a wired phone system have skipped directly to wireless phones, we are now at a stage of Gaussian development where great swaths of less productive detours can be skipped. The history of the Gaussian model has grown sprawling and convoluted. Diverse special purpose models once conceived independently can now be re-expressed as variations of a general model. We introduce the general model with a judicious review of the essential building blocks.

### Linear Regression

In the 1870s, Galton and colleagues devised linear regression (Stanton, 2001). Linear regression answers questions of the form, given $n$ independent measurements

Table 1
*Example data for linear regression.*

| predictor | response |
|---|---|
| -0.95 | 1.08 |
| 1.34 | 0.54 |
| -0.47 | 0.28 |
| 1.11 | 0.93 |
| 1.21 | 1.55 |
| 1.03 | -0.34 |
| -1.69 | -2.48 |
| 0.44 | -0.19 |
| 2.35 | 0.62 |
| -0.80 | -0.88 |

of predictor $x$ and response $y$, what approximation to

$$\boldsymbol{y} = \alpha + \beta \boldsymbol{x} \tag{2}$$

minimizes the squared residual.[1] The solution is

$$\beta = \frac{\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y})}{\mathrm{Var}(\boldsymbol{x})} \tag{3}$$

$$\alpha = \bar{x} - \bar{y}\beta. \tag{4}$$

For example, given data in Table 1 ($n = 10$),

$$\beta = \frac{0.89}{1.62} = 0.55 \tag{5}$$

$$\alpha = 0.11 + -0.36\beta = -0.08. \tag{6}$$

The data and regression line are plotted in Figure 1.

Developed in the olden days before computers, regression was originally framed in terms of squared residuals because computational simplicity was the overriding concern. The modern day statistical engine, Bayes' Theorem (Equation 16), had been disseminated in 1763 but would not blossom until Fisher conceived the method of maximum likelihood in the 1920s. Fortuitously, if we specify a Gaussian model for the data and assume that the residual is independently, identically, and normally distributed then the least squared residual criterion identifies the the same estimates as would be found using Fisher's modern maximum likelihood approach.

## Analysis of Variance (ANOVA)

Analysis of variance is concerned with detection of group differences. The simplest version was formally introduced by Fisher in the 1920s. Like linear regression,

[1] We use the term *residual* instead of *error* because the connotations of *error* are not always appropriate.

Table 2
*Example data for one-way analysis of variance with groups 1 and 2 in columns.*

| 1 | 2 |
|---|---|
| 1.65 | -0.96 |
| 0.07 | 2.13 |
| 0.59 | 1.06 |
| -0.85 | -1.55 |
| 0.36 | -0.53 |
| 1.25 | 0.97 |
| 1.65 | 0.67 |
| -0.73 | 0.94 |
| 0.54 | -0.85 |
| 0.67 | -0.77 |

ANOVA was originally framed in terms of squared differences instead of in terms of Bayes' Theorem. Suppose we want to determine if two groups are different on some measure $y$. An $F$ statistic can be obtained with,

$$SS_{between} = \sum_{j=1}^{2} (\bar{y}_j - \bar{y})^2 \tag{7}$$

$$SS_{within} = \sum_{j=1}^{2} \sum_{i=1}^{n_j} (y_{ij} - \bar{y}_j)^2 \tag{8}$$

$$F = \frac{SS_{between}/1}{SS_{within}/(N-2)}. \tag{9}$$

For example, given the data in Table 2,

$$SS_{between} = 0.83 \tag{10}$$

$$SS_{within} = 19.53 \tag{11}$$

$$F = \frac{0.83}{1.09} = 0.77. \tag{12}$$

While convenient for hand calculation, the method framed in terms of squared differences obscures the relationship between ANOVA and linear regression. The two models are almost the same (compare with Equation 2) except that here $x$ is a binary indicator of group membership,

$$\boldsymbol{y} = \alpha + \beta \boldsymbol{x}. \tag{13}$$

If we code group 2 as $x = 1$ then

$$\alpha = \bar{y}_1 = 0.52 \tag{14}$$

$$\beta = \bar{y}_1 - \bar{y}_2 = -0.41 \tag{15}$$

The $t$ value for the null hypothesis that $\beta = 0$ is not such a simple calculation, but it can be obtained with R to cross-check the magnitude of $F^{\frac{1}{2}} = 0.88$

```r
summary(lm(y~group, aovData))$coefficients['group2','t value']

## [1] -0.88
```

### The Mixed Model

Linear regression and ANOVA models introduce two different kinds of coefficients. In linear regression Equation 2, $\beta$ helps predict every observation whereas in ANOVA Equation 13, $\beta$ only helps predict a subset of observations. This is an important distinction. Historically, coefficients that help predict all observations are called *fixed effects* whereas the other type of coefficient has been called a *random effect*. These are unfortunate terminology. In the statistical literature, there are at least five definitions of these phrases, all which differ (Gelman, 2005). Moreover, in computer science, the term *random* is usually associated with draws from a uniform random number generator, not synonymous with *stochastic* which does not suppose a particular distribution. Here we follow Gelman (2005) and use the terms *constant* and *varying*. For example, the model $y_{ij} = \alpha_j + \beta x_{ij}$ has varying intercepts $\alpha_j$ and a constant slope $\beta$. Models with both kinds of coefficients are called *mixed* models.

As foreshadowed, the squared residuals or squared differences approach to model estimation imposes inconvenient restrictions. To perform ANOVA using squared differences, all combinations of conditions must have an equal number of samples and there is no simple way to cope with missing data. There are some ways to finesse the problem (e.g., Henderson, 1953), but a much more robust solution is to embrace Bayes' Theorem. Let $\boldsymbol{\theta}$ be a vector of model parameters. Bayes' Theorem is,

$$\Pr(\boldsymbol{\theta}|data) = \frac{\Pr(data|\boldsymbol{\theta})\Pr(\boldsymbol{\theta})}{\Pr(data)}. \tag{16}$$

Since $\Pr(data)$ does not depend on the parameters $\boldsymbol{\theta}$, we can omit it, leaving

$$\Pr(\boldsymbol{\theta}|data) \propto \Pr(data|\boldsymbol{\theta})\Pr(\boldsymbol{\theta}). \tag{17}$$

This equation is of such paramount importance that special names are assigned to each term. The density $\Pr(\boldsymbol{\theta})$ is the *prior*, $\Pr(data|\boldsymbol{\theta})$ is the *likelihood*, and $\Pr(\boldsymbol{\theta}|data)$ is the *posterior*. For even modestly complex models, the posterior $\Pr(\boldsymbol{\theta}|data)$ can be impractical to understand directly. To explore and summarize the posterior, two popular approaches are available. One approach is to sample from the posterior, typically using some kind of Markov-Chain Monte Carlo (MCMC) method (e.g., Plummer, 2013; Stan Development Team, 2014). From these samples, mean point estimates and their marginal distributions can be obtained. The second approach is to treat the likelihood or posterior as an arbitrary function and find its mode. This method was introduced by Fisher in the 1920s under the name *maximum likelihood* (Efron, 1998). Some controversy surrounds the prior $\Pr(\boldsymbol{\theta})$ (e.g., Gelman, 2008), but we have no qualms about it and consider *maximum likelihood* synonymous with *maximum posterior*. The MCMC approach can obtain posterior means that are more stable than posterior modes when the posterior has multiple peaks of nearly equal height. However, unresolved questions remain about how to infer MCMC convergence (Gelman & Shirley, 2011). Hence, the present article focuses on the mode instead of mean.

A desire for models with arbitrary combinations of constant and varying coefficients without onerous restrictions on model form culminated in a maximum likelihood estimation method for the mixed model (Hartley & Rao, 1967). For column vector of observations $\boldsymbol{Y}$, covariates $\boldsymbol{X}$ associated with constant coefficients $\boldsymbol{\beta}$, covariates $\boldsymbol{Z}$ associated with varying coefficients $\boldsymbol{u}$, and column vector of residuals $\boldsymbol{e}$, the mixed model can be written as,

$$\boldsymbol{Y} = \underbrace{\boldsymbol{X\beta}}_{\text{constant}} + \underbrace{\boldsymbol{Zu} + \boldsymbol{e}}_{\text{varying}}. \tag{18}$$

To better appreciate the flexibility of this model, we suspend our presentation here without discussion of the distributional assumptions. A mixed model is often specified as a regression formula. A weakness of regression formulae are that they only specify the model for the first moment ($\boldsymbol{\mu}$ of Equation 1). Specification of the second moment ($\boldsymbol{\Sigma}$ of Equation 1) must be either taken as a well known default or given in following text. As an alternative, both moments of a model can be specified simultaneously with a path diagram.

## Path Diagrams

In the 1970s, two different model specifications languages emerged, LISREL (Jöreskog & Van Thillo, 1972) and COSAN (McDonald, 1978). In the process of reconciling these two different ways of modeling, the Reticular Action Model (RAM) was distilled (McArdle, 2005). Although LISREL, COSAN, and RAM offer equivalent expressive power, the RAM is the most parsimonious of the three. Moreover, there is a one-to-one correspondence between the RAM and intuitive path diagrams. In contrast to regression formulae, RAM path diagrams incorporate specifications of both the first and second moments.

The RAM model consists of 4 matrices, traditionally called $\boldsymbol{A}$ (asymmetric), $\boldsymbol{S}$ (symmetric), $\boldsymbol{F}$ (filter), and $\boldsymbol{M}$ (mean). The RAM matrices are related to the model's Gaussian distribution by,

$$\boldsymbol{\mu} = \boldsymbol{F}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{M} \tag{19}$$

$$\boldsymbol{\Sigma} = \boldsymbol{F}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{A})^{-T}\boldsymbol{F}^T. \tag{20}$$

These equations may appear daunting, but note that when $\boldsymbol{A}$ is zero and $\boldsymbol{F}$ is the identity then $\boldsymbol{\mu} = \boldsymbol{M}$ and $\boldsymbol{\Sigma} = \boldsymbol{S}$. So what is the purpose of $\boldsymbol{A}$ and $\boldsymbol{F}$? The $\boldsymbol{A}$ matrix comes into play in the specification of regression relationships. Our linear regression (Equation 2) can be diagrammed as in Figure 2. The multivariate generalization of Equation 4 is implemented by the products that involve $(\boldsymbol{I} - \boldsymbol{A})^{-1}$.

The $\boldsymbol{F}$ matrix is used to filter out variables from the model, permitting these variables to be latent (not measured). Latent variables were devised by Spearman

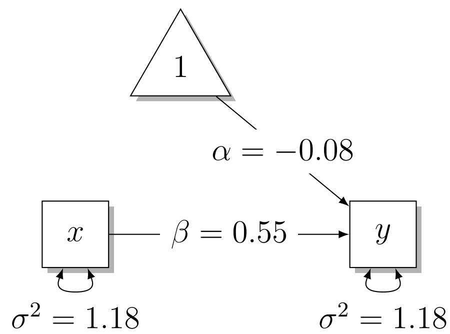


*Figure 2*. Equation 2 drawn as a RAM path diagram. The triangle acts like an observed variable that is always 1. The squares denote observed variables. Single-headed arrows are regressions and double-headed arrows are variances. The diagram takes up more space on the page compared to Equation 2, but it also makes the covariance model explicit, showing that the homogeneous variance is freely estimated.

Table 3
*Example data for latent factor model.*

| x1 | x2 | x3 |
|---|---|---|
| -0.85 | -0.10 | -0.48 |
| -1.19 | -0.24 | -0.87 |
| -0.74 | -0.29 | -0.21 |
| -1.41 | -0.52 | -1.08 |
| -2.31 | -2.27 | -2.33 |
| 2.38 | 1.88 | 2.16 |
| 1.35 | 0.93 | 1.04 |
| 0.79 | 0.21 | 0.58 |

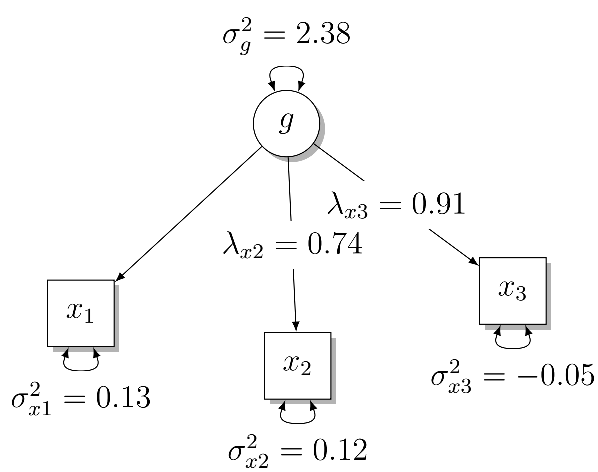


*Figure 3*. A latent factor model given the observed data in Table 3. Latent factors are drawn with a circle, but otherwise play the same role in the diagram as observed data. The regression from $g$ to $x_1$ is unlabeled, and therefore, has a fixed loading of 1.

in the early 1900s (P. Lovie & A. D. Lovie, 1996). For example, Figure 3 exhibits a latent factor model with 3 observed indicators. To clarify how this model works, the corresponding RAM matrices are given along with the model expected covariance $\mathbf{\Sigma}$,

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{21}$$

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \lambda_{x2} \\ 0 & 0 & 0 & \lambda_{x3} \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{22}$$

$$\boldsymbol{S} = \begin{pmatrix} \sigma^2_{x1} & 0 & 0 & 0 \\ 0 & \sigma^2_{x2} & 0 & 0 \\ 0 & 0 & \sigma^2_{x3} & 0 \\ 0 & 0 & 0 & \sigma^2_g \end{pmatrix} \tag{23}$$

$$\begin{aligned} \mathbf{\Sigma} &= \boldsymbol{F}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{A})^{-T}\boldsymbol{F}^T = \\ &\begin{pmatrix} \sigma^2_g + \sigma^2_{x1} & \sigma^2_g\lambda_{x2} & \sigma^2_g\lambda_{x3} \\ \sigma^2_g\lambda_{x2} & \sigma^2_g\lambda^2_{x2} + \sigma^2_{x2} & \sigma^2_g\lambda_{x2}\lambda_{x3} \\ \sigma^2_g\lambda_{x3} & \sigma^2_g\lambda_{x2}\lambda_{x3} & \sigma^2_g\lambda^2_{x3} + \sigma^2_{x3} \end{pmatrix}. \end{aligned} \tag{24}$$

There are 6 parameters. Since the observed covariance matrix has 6 non-redundant entries, this model is just specified. In modeling, latent factors can be treated as if they represent regular observed scores. If factor scores are desired then there are various ways to estimate them (e.g., Estabrook & Neale, 2013). In summary, latent factors are an ingenious user interface. Without the RAM parameterization, it would be an impractical burden to learn how to specify Equation 24.

A Gaussian parameterization that is well suited for estimation of latent factors and regressions is often called a *structural equation model* (SEM; Fan, 1997). We regard the RAM as a SEM parameterization of the Gaussian model. To review, using the RAM we can specify constant coefficients (1st and 2nd moment) in covariance or regression form with respect to observed variables or latent factors. Originally, the RAM did not provide any special support for varying coefficients. Recently, at least one proposal to extend the RAM to arbitrarily varying coefficients has been advanced (Curran & Bauer, 2007). Circles, traditionally used to represent latent factors, were re-purposed to represent varying coefficients. This makes sense because varying coefficients are a more general concept than latent factors. A latent factor is equivalent to a coefficient varying by individual with constant loadings to indicators. At this stage, it may be difficult to judge the merit of Curran and Bauer's proposal due to the potential diverse uses of varying coefficients. To better focus our user interface concerns, we introduce a major application of varying coefficients, multilevel structure.

| | Jane | Joe | |
|---|---|---|---|
| Teachers | | | upper |
| Students | Noah<br>Sophia<br>Liam<br>Emma | Jacob<br>Olivia<br>Mason<br>Isabella | lower |

*Figure 4*. Students nested within teachers. For example, Noah is Jane's student and Jacob is Joe's student. A different model would be needed to accommodate students that spent some proportion of their time with each teacher.

## Multilevel structure

The simple aggregation of observations (Equation 1) is contingent on the assumption that observations are independent. For example, students within a single classroom may exhibit independent performance. However, students drawn from two different classrooms may exhibit some classroom specific effect. Across classrooms, we can no longer consider the individual student as an independent unit of analysis (Kenny & Judd, 1986).

Data with complex structure is often stored in relational databases. Typically, data is normalized into *first normal form*, eliminating redundant or repeating data. Primary keys are assigned to uniquely identify entities. Foreign keys refer to primary keys, allowing the relationships between the data tables to be recovered by the join of primary and foreign keys. Data are considered multilevel when an independent unit of analysis must span across two or more normalized database tables. For example, data on students and teachers would be stored in at least two tables. These data must be stored in separate tables because there is not a 1-to-1 relationship between students and teachers. Since there are fewer teachers than students, teachers are regarded as the *upper* level and students as the *lower* level (see Figure 4).

To describe model structure when there are more than 2 levels we need to introduce two more terms, *nested* and *crossed*. Data are *nested* when the each lower level partitions within its upper level. When data are not nested then they are *crossed*. Crossed varying coefficients need not be organized in relation to other varying coefficients. Crossed coefficients may partition observations in arbitrary ways. For example, suppose a school reassigns some of its students to different classrooms halfway through the year. If we study the whole year, some students will have a single teachers but some will have two teachers. Students with two teachers involved a cross assignment of varying coefficients. The distinction between nested and crossed data is useful because nested data are easier to analyze than crossed data.

Modeling multilevel data is one of the major applications of varying coefficients. Suppose the focus of our analysis is students. We want to estimate a few constant regression coefficients to learn how student performance depends on socioeconomic status and some intervention. We would like to specify our relationships in terms of latent factors because we cannot measure any of the constructs of interest directly. However, we need to incorporate varying coefficients in the model to properly account for teacher effects within a school, school effects within a district, and district effects within a state. If we proceed along these lines, the independent units of analysis are the highest level units, states.

The bottleneck in the evaluation of Equation 1 is the matrix inverse of the model implied covariance matrix $\mathbf{\Sigma}$. Gauss-Jordan matrix inverse requires $O(n^3)$ operations. To fit multilevel models quickly, it is essential to analyze the structure of this matrix and devise some way to reduce its dimension or complexity. Before we discuss optimization techniques, it will be helpful to sketch out more concretely the structure of our hypothetical multilevel student model covariance matrix. To keep things simple, assume the data are nested (not crossed). We introduce the *direct sum* operator,

$$\boldsymbol{B}_1 \bigoplus \boldsymbol{B}_2 = \begin{pmatrix} \boldsymbol{B}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_2 \end{pmatrix}$$

$$\bigoplus_{i=1}^{k} \boldsymbol{B}_i = \begin{pmatrix} \boldsymbol{B}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_2 & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{B}_k \end{pmatrix}.$$

Suppose we build covariance model $\boldsymbol{S}$ for a particular student. A classroom of $s$ students will have covariance matrix

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{T}_{1,1} & \boldsymbol{T}_{1,2} \\ \boldsymbol{T}_{2,1} & \bigoplus_{i=1}^{s} \boldsymbol{S}_i \end{pmatrix}. \tag{25}$$

That is, each student is independent of other students, $\boldsymbol{T}_{1,1}$ is square, and $\boldsymbol{T}_{1,2}$ and $\boldsymbol{T}_{2,1}$ are rectangular. The quadrants labeled with $\boldsymbol{T}$ represent classroom or teacher relationships with each student. This pattern continues as we move up levels. A school of $t$ classrooms will have covariance matrix

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} \\ \boldsymbol{H}_{2,1} & \bigoplus_{i=1}^{t} \boldsymbol{T}_i \end{pmatrix} \tag{26}$$

and a district of $h$ schools will have covariance matrix

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{1,1} & \boldsymbol{D}_{1,2} \\ \boldsymbol{D}_{2,1} & \bigoplus_{i=1}^{h} \boldsymbol{H}_i \end{pmatrix}. \tag{27}$$

Without working out the exact size of such a covariance matrix, it should be clear that it can be very large and very sparse. Fortunately, a great deal of prior research has focused on optimization of the mixed model for precisely this kind of covariance structure. Moreover, we can leverage the mixed model even when the model is specified using RAM notation because the two models offer similar expressive power (Goldstein & McDonald, 1988; Jennrich & Schluchter, 1986). The translation details from multilevel RAM to the mixed model is a solved problem (e.g., Mehta, 2013), but will be worked out again as part of this dissertation.

## Covariance

Although the user interface is low level and inconvenient compared to RAM, the mixed model is important because a great deal of research has gone into its efficient estimation (e.g., Bates & DebRoy, 2004; Harville, 1977; Lindstrom & Bates, 1990; Searle, Casella, & McCulloch, 1992; Wolfinger, Tobias, & Sall, 1994). Recent work has generalized the mixed model to non-Gaussian distributions (Rabe-Hesketh, Skrondal, & Pickles, 2004; Skrondal & Rabe-Hesketh, 2004), but we restrict our focus to Gaussian models. More detailed expositions of the mixed model are available from many sources (e.g., Bates, Mächler, Bolker, & Walker, 2014a; West, Welch, & Galecki, 2014). The essentials are as follows.

In matrix notation, for column vector of observations $\boldsymbol{Y}$, covariates $\boldsymbol{X}$ associated with constant coefficients $\boldsymbol{\beta}$, covariates $\boldsymbol{Z}$ associated with varying coefficients $\boldsymbol{u}$, and column vector of residuals $\boldsymbol{e}$, the mixed model can be written as,

$$\boldsymbol{Y} = \underbrace{\boldsymbol{X\beta}}_{\text{constant}} + \underbrace{\boldsymbol{Zu} + \boldsymbol{e}}_{\text{varying}}. \tag{28}$$

We assume $\boldsymbol{u}$ and $\boldsymbol{e}$ are normally distributed with

$$\mathrm{E}\begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{e}\end{pmatrix} = \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{0}\end{pmatrix} \tag{29}$$

$$\mathrm{Cov}\begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{e}\end{pmatrix} = \begin{pmatrix}\boldsymbol{G} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}\end{pmatrix}. \tag{30}$$

Design matrix $\boldsymbol{X}$ is not estimated. Matrix $\boldsymbol{Z}$ can be used in two distinct ways: as a design matrix for varying coefficients (not estimated) or as factor loadings for latent factors (as free parameters) (Skrondal & Rabe-Hesketh, 2004, p. 107). Let $\boldsymbol{\theta_Z}$, $\boldsymbol{\theta_G}$, and $\boldsymbol{\theta_R}$ refer to the free parameters in $\boldsymbol{Z}$, $\boldsymbol{G}$, and $\boldsymbol{R}$, respectively.

Although Equation 28 builds intuition, it actually describes the distribution of $\boldsymbol{Y}$ conditional on a particular realization of $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{G})$. The unconditional distribution is

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{e} \tag{31}$$

where

$$\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{ZGZ}^T + \boldsymbol{R}). \tag{32}$$

Univariate models typically use $\boldsymbol{R} = \sigma^2\boldsymbol{I}$. Independent units of analysis in multivariate models typically use a block diagonal $\boldsymbol{R}$ with each block as the independent unit. Once covariance components $\boldsymbol{R}$ and $\boldsymbol{G}$ are estimated, analytic solutions are available for constant $\hat{\boldsymbol{\beta}}$ and varying $\hat{\boldsymbol{u}}$ coefficients (Henderson Jr, 1982),

$$\begin{pmatrix}\boldsymbol{X^T\hat{R}^{-1}X} & \boldsymbol{X^T\hat{R}^{-1}Z}\\ \boldsymbol{Z^T\hat{R}^{-1}X} & \boldsymbol{Z^T\hat{R}^{-1}Z} + \boldsymbol{\hat{G}^{-1}}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{\beta}}\\ \hat{\boldsymbol{u}}\end{pmatrix} = \begin{pmatrix}\boldsymbol{X^T\hat{R}^{-1}Y}\\ \boldsymbol{Z^T\hat{R}^{-1}Y}\end{pmatrix}. \tag{33}$$

That is, varying coefficients $\boldsymbol{u}$ are not estimated directly but are an analytic function of the covariance. The solutions of Equation 33 can be written as,

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\hat{\boldsymbol{V}}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\hat{\boldsymbol{V}}^{-1}\boldsymbol{Y} \tag{34}$$

$$\hat{\boldsymbol{u}} = \hat{\boldsymbol{G}}\boldsymbol{Z}^T\hat{\boldsymbol{V}}^{-1}\left(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right) \tag{35}$$

where

$$\boldsymbol{V} \equiv \boldsymbol{Z}\hat{\boldsymbol{G}}\boldsymbol{Z}^T + \hat{\boldsymbol{R}}. \tag{36}$$

Let $\boldsymbol{\theta} \equiv \{\boldsymbol{\theta}_{\boldsymbol{Z}}, \boldsymbol{\theta}_{\boldsymbol{G}}, \boldsymbol{\theta}_{\boldsymbol{R}}\}$. The -2 log-likelihood of $n$ independent observations is,

$$-2\ell(\boldsymbol{\beta}, \boldsymbol{\theta}) = nk\log(2\pi) + \log|\boldsymbol{V}| + (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{V}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{37}$$

where $k$ is the size of $\boldsymbol{V}$. This likelihood can be simplified by plugging Equation 34 in for $\boldsymbol{\beta}$ (using provisional estimates). The resulting profile -2 log-likelihood is,

$$-2\ell(\boldsymbol{\theta}) = nk\log(2\pi) + \log|\boldsymbol{V}| + \boldsymbol{r}^T\boldsymbol{V}^{-1}\boldsymbol{r} \tag{38}$$

where

$$\boldsymbol{r} = \boldsymbol{Y} - \boldsymbol{X}\left[\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{Y}\right]. \tag{39}$$

This likelihood does not take into account the loss of degrees of freedom from constant coefficients $\boldsymbol{\beta}$ in the estimation of covariance parameters $\boldsymbol{\theta}$. Uncorrected, covariance parameters tend to exhibit bias. A solution was proposed to obtain unbiased covariance parameters estimates (known as REML; Patterson & Thompson, 1971). However, when REML is used, the likelihood ratio test cannot be used for constant coefficients $\boldsymbol{\beta}$ (West et al., 2014, p. 35). Fortunately, the addition of a Wishart prior to the likelihood corrects bias even more accurately than REML (Chung, Gelman, Rabe-Hesketh, Liu, & Dorie, 2015). The addition of a Bayesian prior is an elegant solution that corrects for bias without impairing the likelihood ratio test.

Nested varying coefficients produce a sparse covariance matrix with a pattern amenable to an efficient inverse (Goldstein & McDonald, 1988). However, crossed varying coefficients create less predictable covariance patterns. Sparse matrix algebra is probably required to compute inverses efficiently for arbitrarily crossed models (Fellner, 1987).

## A Potential Optimization

The mixed model offers more flexibility than required by many data sets with nested multilevel structure. von Oertzen and Hackett (n.d.) proposed a method of refactoring the model to reduce the dimension of the model covariance matrix $\boldsymbol{\Sigma}$. Suppose we have a model like Figure 5(a) in RAM parameterization. This model has fixed loadings from $x_1$ to the other indicators. Furthermore, $x_2$, $x_3$, and $x_4$ are structurally identical. Such a model can be transformed into a much simpler refactored model. With a suitable transformation of the data, the model in Figure 5(b) is equivalent to the model in Figure 5(a). The advantage of the model in Figure 5(b) is

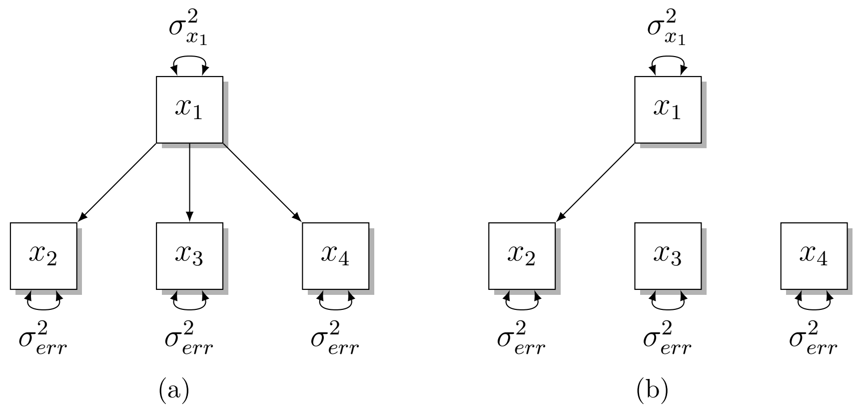


*Figure 5*. Original model (a) and refactored model (b). These models are equivalent with suitably transformed data.

that $x_3$ and $x_4$ are no longer attached to the model and do not add to the dimension of the model covariance matrix $\mathbf{\Sigma}$.

In this dissertation, we will explore how far we can take this approach and the exact model restrictions required to apply it. Suppose the indicators in Figure 5 were not just indicators but entire factor models. For example, suppose $x_1$ was a teacher factor model with $x_2$, $x_3$, and $x_4$ as student factor models. The same transformation can be applied to obtain an equivalent model with only 1 student and the remainder of the students unlinked from the model. Furthermore, this can be applied recursively from the lowest level to the highest level entities. For nested multilevel data, the reduction in the dimension of the model covariance matrix $\mathbf{\Sigma}$ is dramatic. This extension of von Oertzen and Hackett (n.d.) to a nested multilevel model was devised in an unpublished manuscript with contributions from Steve Boker and Tim Brick.

The most notorious aspect of this algorithm is its name. It was called *pre-processed maximum likelihood* in the title of von Oertzen and Hackett (n.d.). However, the phrase *pre-processed* is remarkably non-specific. Furthermore, there is nothing about the algorithm that requires maximum likelihood as a fit function as opposed to, say, least squared residual. Hence, none of the elements of the original name provide helpful semantic cues. Tentatively, we propose *rampart*. The name *rampart* lexically emphasizes the connection with the RAM parameterization of covariance. Colloquially, a rampart is a wall built for defense. The Rampart algorithm partitions, or places a wall between, repeated identical elements to defend against poor performance.

## Inference

Due to our reliance on maximum likelihood, large sample theory provides a number of ready tools for inference such as the Wald test (including the sandwich estimator), the likelihood ratio test (including profile likelihood confidence intervals), the bootstrap, and the jackknife (Pawitan, 2001; Pek & Wu, in press; White, 1982). What is less clear is size of the sample required for the null distribution of these tests to converge to their theoretical distributions. Another important consideration is the type of parameter. For constant coefficients, adjustments are advised to improve calibration of the false positive rate (e.g., Manor & Zucker, 2004). Variance components can be divided into two cases. When the null hypothesis does not involve a parameter space boundary then standard asymptotic results apply. An example is a test between heterogeneous and homogeneous residual variance. The second case arises when a parameter space boundary is involved. This commonly occurs in the test of whether to include a varying coefficient because varying coefficients are not tested directly but by restriction of their variance (and covariances) to zero (e.g., Crainiceanu & Ruppert, 2004).

Much prior research on small sample inference in mixed models is confined to univariate mixed models without latent factors. It is unclear whether prior research will generalize to more complex models. More simulation studies are needed to provide guidance about how best to perform inference on these models.

## Application

In order to demonstrate the efficacy of the mixed model and Rampart algorithm, we will reanalyze data from a facial expression tracking experiment (Boker et al., 2009). When two people engage in conversation, prior research indicates that the style of their head movements tends to become more similar. In this experiment, confederates engaged in conversation with naive participants over a video conferencing system. However, naive participants did not see the unfiltered confederates but a computer generated avatar. To produce a convincing portrayal, confederates' facial expressions were meticulously tracked in real-time. The portrayals were sufficiently convincing that no naive participants guessed that the computer generated faces were not unmodified live video.

In a crossed experimental design, damping was applied to confederate facial expressions, vocal inflections, and head movements. Confederates were familiar with the nature of the manipulations and their probable effects, but were blind to the order and timing. The head movements of both participants in the conversation were motion tracked at 81.6 Hz. The dependent variables were anterior-posterior (A-P) and lateral head angle. These correspond to nods of affirmation (pitch) and head shakes of disagreement (yaw), respectively. Vigor of angular velocity was taken as a metric. Based on prior research, it was hypothesized that women would nod and shake their head with greater vigor than men. In addition, it was hypothesized that each of the manipulations would increase the vigor of nods and shakes. The notion of vigor was operationalized as the root mean square of the angular velocity during a condition.

For each 1 minute condition, there were 4860 velocity measurements ($81.6 \cdot 60 \approx 4860$). Conversations were described as lasting 8 minutes (Boker et al., 2009, p. 3488) with a different condition every minute. However, conversations ranged from 6 to 10 minutes with a median of 9 minutes. Conditions always lasted 1 whole minute so conversations shorter than 8 minutes did not include all conditions and conversations longer than 8 minutes included some repeated conditions.

To ensure an accurate re-analysis, the original analyses were reproduced (Tables 4 and 5). There was a varying intercept per naive participant, but all confederates were

Table 4
*Original linear mixed effects regression of A-P RMS angular velocity.*

| | Value | Std.Error | DF | t-value | p-value |
|---|---|---|---|---|---|
| (Intercept) | 10.01 | 0.52 | 780.00 | 19.23 | 0.00 |
| selfSex | -3.93 | 0.25 | 780.00 | -15.55 | 0.00 |
| otherSex | -1.77 | 0.27 | 780.00 | -6.57 | 0.00 |
| isConfed | -0.36 | 0.18 | 780.00 | -1.99 | 0.05 |
| dampHead | 0.57 | 0.19 | 780.00 | 3.07 | 0.00 |
| dampFace | 0.45 | 0.19 | 780.00 | 2.43 | 0.02 |
| dampVoice | -0.04 | 0.18 | 780.00 | -0.20 | 0.84 |
| otheryRotFV | -0.01 | 0.04 | 780.00 | -0.39 | 0.70 |
| confedByOtherSex | -2.40 | 0.51 | 780.00 | -4.73 | 0.00 |
| confedByDampHead | -0.04 | 0.37 | 780.00 | -0.12 | 0.91 |
| confedByDampFace | 0.39 | 0.37 | 780.00 | 1.05 | 0.29 |
| confedByDampVoice | 0.35 | 0.37 | 780.00 | 0.94 | 0.35 |

Table 5
*Original linear mixed effects regression of lateral RMS angular velocity.*

| | Value | Std.Error | DF | t-value | p-value |
|---|---|---|---|---|---|
| (Intercept) | 176.37 | 22.95 | 780.00 | 7.69 | 0.00 |
| selfSex | -60.91 | 9.64 | 780.00 | -6.32 | 0.00 |
| otherSex | -31.86 | 9.67 | 780.00 | -3.29 | 0.00 |
| isConfed | -21.02 | 6.73 | 780.00 | -3.12 | 0.00 |
| dampHead | 14.19 | 6.75 | 780.00 | 2.10 | 0.04 |
| dampFace | 8.21 | 6.76 | 780.00 | 1.21 | 0.22 |
| dampVoice | 4.40 | 6.75 | 780.00 | 0.65 | 0.51 |
| otherxRotFV | -0.30 | 0.03 | 780.00 | -8.78 | 0.00 |
| confedByOtherSex | -49.65 | 18.98 | 780.00 | -2.62 | 0.01 |
| confedByDampHead | -4.81 | 13.47 | 780.00 | -0.36 | 0.72 |
| confedByDampFace | 6.30 | 13.50 | 780.00 | 0.47 | 0.64 |
| confedByDampVoice | 10.89 | 13.49 | 780.00 | 0.81 | 0.42 |

assumed to produce equally vigorous head movements. All minutes were assumed independent for model simplicity. Hence, the original model violated the assumption of independent observations since minutes involving the same participants should be more similar than minutes involving different participants. Furthermore, these models are a little awkward to interpret because both conversation participants are modeled in the same regression equation with `isConfed` used to switch between them. This model was loosely based on the Actor-Partner Interdependence Model (Cook & Kenny, 2005). A regression style model was chosen because, at the time, no software was available to conveniently specify a multivariate mixed model (S. Boker, personal communication, March 2015). Before proceeding, we note that the RMS statistics are skewed and leptokurtic. The distribution can be improved by a log transformation. The same analyses using log transformed responses are exhibited in Tables 6 and 7.

Table 6
*Original linear mixed effects regression of log A-P RMS angular velocity.*

| | Value | Std.Error | DF | t-value | p-value |
|---|---|---|---|---|---|
| (Intercept) | 2.77 | 0.10 | 780.00 | 27.82 | 0.00 |
| selfSex | -0.66 | 0.03 | 780.00 | -19.97 | 0.00 |
| otherSex | -0.40 | 0.04 | 780.00 | -11.19 | 0.00 |
| isConfed | -0.07 | 0.02 | 780.00 | -3.06 | 0.00 |
| dampHead | 0.07 | 0.02 | 780.00 | 2.84 | 0.00 |
| dampFace | 0.05 | 0.02 | 780.00 | 2.23 | 0.03 |
| dampVoice | -0.01 | 0.02 | 780.00 | -0.38 | 0.70 |
| lnotheryRotFV | -0.19 | 0.03 | 780.00 | -5.37 | 0.00 |
| confedByOtherSex | -0.40 | 0.07 | 780.00 | -5.90 | 0.00 |
| confedByDampHead | 0.00 | 0.05 | 780.00 | 0.01 | 0.99 |
| confedByDampFace | 0.04 | 0.05 | 780.00 | 0.91 | 0.36 |
| confedByDampVoice | 0.04 | 0.05 | 780.00 | 0.89 | 0.37 |

Table 7
*Original linear mixed effects regression of log lateral RMS angular velocity.*

| | Value | Std.Error | DF | t-value | p-value |
|---|---|---|---|---|---|
| (Intercept) | 5.81 | 0.24 | 780.00 | 24.01 | 0.00 |
| selfSex | -0.78 | 0.08 | 780.00 | -9.18 | 0.00 |
| otherSex | -0.38 | 0.09 | 780.00 | -4.34 | 0.00 |
| isConfed | -0.02 | 0.06 | 780.00 | -0.34 | 0.73 |
| dampHead | 0.16 | 0.06 | 780.00 | 2.70 | 0.01 |
| dampFace | 0.11 | 0.06 | 780.00 | 1.90 | 0.06 |
| dampVoice | 0.03 | 0.06 | 780.00 | 0.45 | 0.65 |
| lnotherxRotFV | -0.36 | 0.03 | 780.00 | -10.96 | 0.00 |
| confedByOtherSex | -0.34 | 0.17 | 780.00 | -2.06 | 0.04 |
| confedByDampHead | 0.02 | 0.12 | 780.00 | 0.21 | 0.83 |
| confedByDampFace | 0.09 | 0.12 | 780.00 | 0.75 | 0.45 |
| confedByDampVoice | 0.07 | 0.12 | 780.00 | 0.55 | 0.58 |

## Proposed Work

Two areas of work are proposed. Software development is proposed to provide mixed and Rampart optimized nested multilevel modeling capability in OpenMx. Once these improved tools are available, we will proceed to reanalyze the data from Boker et al. (2009).

### The Mixed Model

Free software that implements the univariate mixed model without latent factors is readily available (Bates, Mächler, Bolker, & Walker, 2014b). Within OpenMx, we will recreate a translation from a nested multilevel RAM specification to the mixed model. As mentioned, a great deal of effort has been invested in optimization of mixed models. However, an efficient implementation is not our focus and we will strive for an implementation that is merely correct and leave optimization for future work.

### Rampart

During 2013, the Rampart nested multilevel optimization was prototyped in OpenMx. A proof-of-concept test script is exhibited in Appendix B along with a discussion of the new features. This early work will be brought forward and reintegrated with the current code base. Furthermore, a number changes are required to bring this code up to a level of quality and robustness expected by applied researchers. The user interface will be made more intuitive, constraints on the shape of the data will be relaxed, and full support will be extended to definition variables.

- Instead of placing all the levels in a container model, Ryne Estabrook suggested that each lower level `mxModel` could nest within the upper level `mxModel`. This change would likely improve the ergonomics of model specification.
- The prototype was limited to the situation where there are exactly the same number of lower level units for each upper level unit. Such perfectly balanced data are unlikely to occur in practice. For example, the lowest level of data from Boker et al. (2009) does not meet this criterion because conversations ranged from 6 to 10 minutes. We anticipate no mathematical barrier to easing this restriction, but the code is yet to be written.
- The prototype did not consider definition variables. Definition variables are an important OpenMx feature that users expect to be implemented consistently throughout OpenMx. We will add complete support for definition variables.

### Inference

As discussed, inference for the mixed model is a complex and evolving area. We will rely on the general capabilities of OpenMx to obtain asymptotic results for arbitrary exponential models. While Wald tests are of limited interest in a nested multilevel context, the Rampart optimization can also be applied to speed estimation of latent trajectory or latent differential equation models. In this context, it would be useful if SEs obtained from the Rampart model could be transformed to match SEs that would be obtained without use of the Rampart optimization. Since our focus is mainly on the nested multilevel case, we argue that this work is beyond the scope of this dissertation.

### Reanalysis of Boker et al. (2009)

We will use the newly implemented nested multilevel support in OpenMx to reanalyze Boker et al. (2009) with a better account of data dependence. Our model will nest minutes within trials and trials within confederate / naive participant pairs. This is not a perfect solution because trials with different participant pairs will be treated as independent observations even when one of the participants is the same

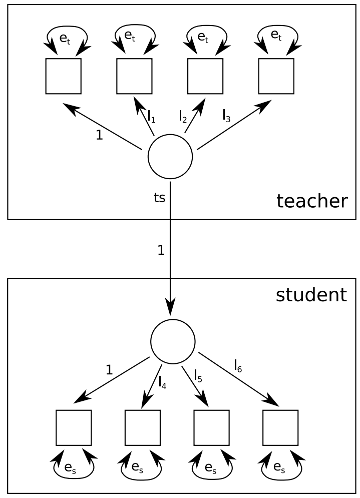


*Figure 6*. Estimate the influence *ts* of a teacher's latent factor on a student's latent factor. Although the cross-level loading is fixed to 1, we can still estimate the regression weight in terms of teachers (`ts`).

in both observations. A nested multilevel model is not quite flexible enough to fully account for the dependence in the data. To properly account for the dependence, a crossed design is necessary. We leave a crossed design to future work.

**Detailed Work Plan**

1. Implement the mixed model (Equation 38) within OpenMx to the extent that univariate models obtain estimates that are within some epsilon (i.e. $10^{-2}$) of the estimates obtained by `lme4`. The implementation will be validated using models given in the example section of the `lmer` manual page, perhaps with reduced sample size. These models include (1) `lmer(Reaction   Days + (Days | Subject), sleepstudy)` and (2) `lmer(distance   age + (age|Subject) + (0+nsex|Subject) + (0 + nsexage|Subject), data=Orthodont)`.

2. Adjust OpenMx to permit the specification of nested multilevel RAM using nested `mxModels` (see Figure 7). Fixed regression paths will be permitted from containing models (upper level) to contained models (lower level). For example, see line 107 of Appendix B. The main difference between Appendix B and what is proposed here is a change in user interface to better reflect the nesting of models (Figure 7b). The importance of user interface for the attraction of applied researchers is not to be underestimated. This will contribute to the novelty of this dissertation.

3. Translate nested multilevel RAM into the mixed parameterization. The model diagrammed in Figure 6 will be used to validate the translation. Similar estimates (i.e. within $10^{-2}$) should be obtained in either parameterization. One challenge is that existing OpenMx examples such as `MultilevelUniRandomSlopeInt.R` (Appendix C) and `UnivariateRandomInterceptWide.R` (Appendix D), do not

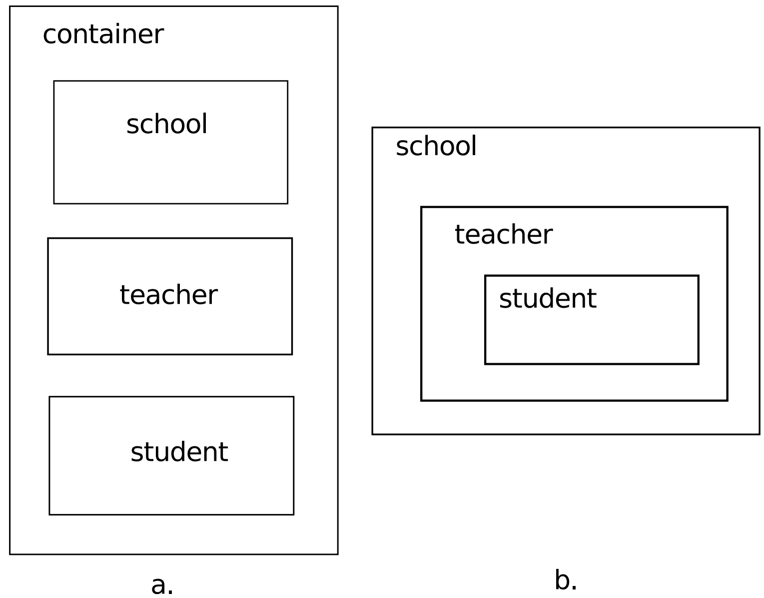


*Figure 7*. Two equivalent model specifications for students nested within teachers nested within schools. Each rectangle corresponds to an `mxModel`. The prototype uses organization (a) to specify nested multilevel models. We propose to replace (a) with (b).

specify a distribution for varying coefficients ($\boldsymbol{G}$ in Equation 30) or use Henderson's Equation 33, but estimate the varying coefficients directly as part of model estimation. The two parameterizations (mixed and in these examples) seem asymptotically equivalent but exhibit differences for small samples. This difference will be eliminated by specifying a distribution for varying coefficients. The two aforementioned OpenMx tests that involve varying coefficients will be modified to match the mixed model. That is, the value of the likelihood function will match to within $10^{-2}$ for any given parameter vector.

The Rampart transformation does not profile out varying coefficients but reorganizes them by transforming the RAM $\boldsymbol{A}$ matrix (Equations 19 and 20) and corresponding data. The value of the likelihood function is unaffected by this transformation. Some work is required to determine how to carry the distribution on varying coefficients through the Rampart transformation. Once adjusted, the Rampart and original likelihood functions will match (within $10^{-3}$) for any given parameter vector.

4. Revive the Rampart work from 2013 and generalize Rampart to unbalanced data. A generalized Rampart will generate a separate model for each upper level unit and glue them together with equality constraints. To validate this work, `UnivariateRandomInterceptWide.R` will be rephrased to make it easy to enable or disable Rampart. The number of upper level units and lower level units will be manipulated with the expectation that the use of Rampart will not change obtained estimates (within $10^{-3}$).

5. In OpenMx, definition variables can be assigned to any fixed coefficient. The mixed model and Rampart will be analyzed with definition variables in mind. Any restrictions on use of definition variables will be documented. Categories of uses that are supported will be enumerated. Each type of use will be validated by creating a minimal test case.

6. To validate the algorithm implementations, a simulation study of parameter recovery accuracy will be conducted using naive RAM, mixed, and Rampart approaches. We will consider Figure 6 and the same model extended to 3 levels (nested multilevel latent univariate regression). Two sets of true parameters ($\theta_1$ and $\theta_2$) will be randomly drawn. One hundred Monte Carlo replications will be run for each of the 24 conditions (Algorithm $\times$ Levels $\times$ $\theta$). For each replication, a data set will be generated from the true parameters and the model optimized against this data set. Monte Carlo bias and variance of estimates will be reported by algorithm, model, and parameter set. With the OpenMx implementation of the mixed model being unoptimized, optimization elapse time will not be compared intensively. We estimate that this study is small enough to complete using the investigator's laptop within 24 hours. We expect that all algorithms will obtain similar results (equal to within $10^{-1}$ bias and $10^{-1}$ parameter variance).

7. In preparation to reanalyze Boker et al. (2009) with a nested multilevel design, a simulation study will be conducted to better understand the effect of model choice. Our new model will nest conditions within trials and trials within pairs of participants. This design does not eliminate all of the dependency between the highest level units because some confederates will participate in many pairs. However, this is probably the best design that can conform to a nested multilevel framework.

   Two models will be compared, the original mixed model (Tables 6 and 7) and a nested multilevel Rampart-optimized model. The goal of this simulation study is to quantify how well each model can recover effects similar those found by Boker et al. (2009) encoded into a data set by simulation. The number of conditions will be large. For example, the effect of head damping on naive participant head movement vigor in the A-P axis is one effect. For this particular effect, we will simulate either no effect, the average effect found in Boker et al. (2009) or the maximum effect (large effect). This approach will be applied to all predictors and both signs (positive and negative) (see Appendix E).

   Given the number of potential conditions, a fully crossed experimental design will be impractical. Instead, we will sample a configuration of effects, generate data, and fit both models to the same data. Five hundred replications will be conducted with a fresh configuration of effects for every sample. We anticipate that this simulation can be completed using the investigator's laptop within 48 hours. Since no attempt was made to calibrate the null distributions of the Wald statistics, models will be scored using signal detection theory. We will summarize results as the area under the ROC curve for each model. Since the generating model (both theoretically and in Appendix E) is more similar to a nested multilevel design than the original model, we are confident that the nested multilevel design will exhibit more power than the model used in Boker et al. (2009).

8. With the confidence gained in the ability of the nested multilevel model to accurately recover effects, we will proceed to reanalyze the data from Boker et al. (2009) and discuss the findings.

## References

Bates, D. & DebRoy, S. (2004). Linear mixed models and penalized least squares. *Journal of Multivariate Analysis*, *91*(1), 1–17.

Bates, D., Mächler, M., Bolker, B. M., & Walker, S. (2014a, June). Fitting Linear Mixed-Effects Models using lme4. *ArXiv e-prints*. arXiv: 1406.5823

Bates, D., Mächler, M., Bolker, B. M., & Walker, S. (2014b). lme4: Linear mixed-effects models using Eigen and S4. ArXiv e-print; submitted to *Journal of Statistical Software*. Retrieved from http://arxiv.org/abs/1406.5823

Boker, S. M., Cohn, J. F., Theobald, B.-J., Matthews, I., Brick, T. R., & Spies, J. R. (2009). Effects of damping head movement and facial expression in dyadic conversation using real–time facial expression tracking and synthesized avatars. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1535), 3485–3495.

Chung, Y., Gelman, A., Rabe-Hesketh, S., Liu, J., & Dorie, V. (2015). *Weakly informative prior for point estimation of covariance matrices in hierarchical models.* Manuscript submitted for publication.

Cook, W. L. & Kenny, D. A. (2005). The Actor-Partner interdependence model: A model of bidirectional effects in developmental studies. *International Journal of Behavioral Development*, *29*(2), 101–109.

Crainiceanu, C. & Ruppert, D. (2004). Likelihood ratio tests in linear mixed models with one variance component. *Journal of the Royal Statistical Society, Series B*, *66*, 165–185.

Curran, P. J. & Bauer, D. J. (2007). Building path diagrams for multilevel models. *Psychological Methods*, *12*(3), 283–297.

Efron, B. (1998). R. A. Fisher in the 21st Century. *Statistical Science*, 95–114.

Estabrook, R. & Neale, M. C. (2013). A comparison of factor score estimation methods in the presence of missing data: Reliability and an application to nicotine dependence. *Multivariate Behavioral Research*, *48*(1), 1–27.

Fan, X. (1997). Canonical correlation analysis and structural equation modeling: what do they have in common? *Structural Equation Modeling: A Multidisciplinary Journal*, *4*(1), 65–79. doi:10.1080/10705519709540060

Fellner, W. H. (1987). Sparse matrices, and the estimation of variance components by likelihood methods. *Communications in Statistics-Simulation and Computation*, *16*(2), 439–463.

Gelman, A. (2005). Analysis of variance–why it is more important than ever. *The Annals of Statistics*, *33*(1), 1–53.

Gelman, A. (2008). Objections to Bayesian statistics. *Bayesian Analysis*, *3*(3), 445–449.

Gelman, A. & Shirley, K. (2011). Inference from simulations and monitoring convergence. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 163–174). CRC press.

Goldstein, H. & McDonald, R. P. (1988). A general model for the analysis of multilevel data. *Psychometrika*, *53*(4), 455–467.

Hartley, H. O. & Rao, J. N. (1967). Maximum-likelihood estimation for the mixed analysis of variance model. *Biometrika*, *54*(1-2), 93–108.

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, *72*(358), 320–338.

Henderson Jr, C. R. (1982). Analysis of covariance in the mixed model: Higher-level, nonhomogeneous, and random regressions. *Biometrics*, 623–640.

Henderson, C. R. (1953). Estimation of variance and covariance components. *Biometrics*, *9*(2), 226–252.

Jennrich, R. I. & Schluchter, M. D. (1986). Unbalanced repeated-measures models with structured covariance matrices. *Biometrics*, 805–820.

Jöreskog, K. G. & Van Thillo, M. (1972). LISREL: A general computer program for estimating a linear structural equation system involving multiple indicators of unmeasured variables. [Computer software].

Kenny, D. A. & Judd, C. M. (1986). Consequences of violating the independence assumption in analysis of variance. *Psychological Bulletin*, *99*(3), 422.

Lindstrom, M. J. & Bates, D. (1990). Nonlinear mixed effects models for repeated measures data. *Biometrics*, 673–687.

Lovie, P. & Lovie, A. D. (1996). Charles Edward Spearman, F.R.S. (1863-1945). *Notes and Records of the Royal Society*, *50*(1), 75–88. doi:10.1098/rsnr.1996.0007

Manor, O. & Zucker, D. M. (2004). Small sample inference for the fixed effects in the mixed linear model. *Computational Statistics & Data Analysis*, *46*, 801–817. doi:10.1016/j.csda.2003.10.005

McArdle, J. J. (2005). The development of the RAM rules for latent variable structural equation modeling. In A. Maydeu-Olivares & J. J. McArdle (Eds.), *Contemporary advances in psychometrics* (pp. 225–273). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

McDonald, R. P. (1978). A simple comprehensive model for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology*, *31*(1), 59–72.

Mehta, P. (2013). xxM: N-Level structural equation modeling [Computer software].

Patterson, H. D. & Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika*, *58*(3), 545–554.

Pawitan, Y. (2001). *In all likelihood: Statistical modelling and inference using likelihood*. Oxford University Press.

Pearl, J. (2000). *Causality: Models, reasoning and inference*. Cambridge University Press.

Pek, J. & Wu, H. (in press). Profile likelihood-based confidence intervals and regions for structural equation models. *Psychometrika*.

Plummer, M. (2013). *JAGS version 3.4.0 user manual*. Retrieved from http://mcmc-jags.sourceforge.net/

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2004). Generalized multilevel structural equation modeling. *Psychometrika*, *69*(2), 167–190.

Searle, S. R., Casella, G., & McCulloch, C. E. (1992). *Variance components*. John Wiley & Sons, Inc.

Skrondal, A. & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. CRC Press.

Stan Development Team. (2014). *Stan modeling language users guide and reference manual, version 2.5.0*. Retrieved from http://mc-stan.org/

Stanton, J. M. (2001). Galton, Pearson, and the peas: A brief history of linear regression for statistics instructors. *Journal of Statistics Education*, *9*(3).

Voelkle, M. C. & Oud, J. H. (2013). Continuous time modelling with individually varying time intervals for oscillating and non-oscillating processes. *British Journal of Mathematical and Statistical Psychology*, *66*(1), 103–126.

von Oertzen, T. & Hackett, D. C. (n.d.). *Analytical solution of maximum likelihood in structural equation models with fixed loadings*. Manuscript submitted for publication.

West, B. T., Welch, K. B., & Galecki, A. T. (2014). *Linear mixed models: A practical guide using statistical software*. CRC Press.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, 1–25.

Wolfinger, R., Tobias, R., & Sall, J. (1994). Computing Gaussian likelihoods and their derivatives for general linear mixed models. *SIAM Journal on Scientific Computing*, *15*(6), 1294–1310.

## Appendix A
### Multivariate model for Boker et al. (2009)

```
library(ggplot2)
library(OpenMx)
library(dplyr)
load("3level.rda")

#mxOption(NULL, "Default optimizer", "CSOLNP")

# prep info for per participant "random" latents
uniqueNaive <- length(unique(by_pair$naiveID))
uniqueConfed <- length(unique(by_pair$confedID))

flat <- left_join(left_join(by_naive, by_pair), by_trial)
for (col in c('pairID')) {
  flat[[col]] <- NULL
```

```
}

for (who in c('naive','confed')) {
  id <- paste0(who, "ID")
  index <- paste0(who, "Index")
  flat[[index]] <- match(flat[[id]], unique(flat[[id]]))
}

flat$sameSex <- as.numeric(flat$naiveSex == flat$confedSex)

manifest <- c('cx','cy','nx','ny')

# ------------------------------------------------------------------

flatModel1 <- mxModel(
  model="flat1", type="RAM",
  manifestVars=manifest, latentVars=c(),
  mxData(observed=as.data.frame(flat), type="raw"),
  # unaccounted for variance
  mxPath(manifest, arrows=2, values=1),
  # covariances
  mxPath(c('cx'), c('cy'), arrows=2),
  mxPath(c('nx'), c('ny'), arrows=2),
  # overall intercept
  mxPath("one", manifest))
flatModel1 <- mxRun(flatModel1)

#------------------------------------------

#uniqueConfed <- 1   # assume all confeds are equal

mkModel2 <- function(latent=c()) {
  # add per-person latents
  flatModel2 <- mxModel(
    model="flat2", type="RAM",
    manifestVars=manifest,
    latentVars=c("confedX", "confedY", "naiveX", "naiveY", latent),
    mxData(observed=as.data.frame(flat), type="raw"),
    # covariances
    mxPath(c('cx'), c('cy'), arrows=2),
    mxPath(c('nx'), c('ny'), arrows=2),
    # per confederate
    mxMatrix(name="PerConfederate", nrow=uniqueConfed, ncol=4,
```

```
                 free=TRUE, dimnames=list(unique(flat$confedID)[1:uniqueConfed],
                                          c("xM", "xVar", "yM", 'yVar'))),
    # per naive participant
    mxMatrix(name="PerNaive", nrow=uniqueNaive, ncol=4,
                 free=TRUE, dimnames=list(unique(flat$naiveID),
                                          c("xM", "xVar", "yM", 'yVar'))),
    mxAlgebra(PerNaive[data.naiveIndex,], name="CurNaive"),
    # per participant variance
    mxPath(c("confedX", "confedY", "naiveX", "naiveY"), arrows=2, free=FALSE,
           labels=c("CurConfed[1,2]", "CurConfed[1,4]",
                    "CurNaive[1,2]", "CurNaive[1,4]")),
    # per particpant mean
    mxPath("one", c("confedX", "confedY", "naiveX", "naiveY"), free=FALSE,
           labels=c("CurConfed[1,1]", "CurConfed[1,3]",
                    "CurNaive[1,1]", "CurNaive[1,3]")),
    # connect latents to manifests
    mxPath(c("confedX", "confedY", "naiveX", "naiveY"),
           manifest, free=FALSE, values=1))

  flatModel2$PerConfederate$values[,c('xVar','yVar')] <- 1
  flatModel2$PerNaive$values[,c('xVar','yVar')] <- 1
  if (uniqueConfed==1) {
    flatModel2 <- mxModel(
        flatModel2,
        mxAlgebra(PerConfederate[1,], name="CurConfed"))
  } else {
    flatModel2 <- mxModel(
        flatModel2,
        mxAlgebra(PerConfederate[data.confedIndex,], name="CurConfed"))
  }
  flatModel2
}

flatModel2 <- mxModel(model=mkModel2(), mxComputeGradientDescent())

flatModel2 <- mxRun(flatModel2)  # 1403.605

#----------------------------------------------------------

latent <- c()
for (cond in c("Head", "Face", "Voice", "naiveSex",
               "confedSex")) {  #, "sameSex"
  latent <- c(latent, paste0(manifest, cond))
```

```r
}

flatModel3 <- mxModel(
  model=mkModel2(latent), name="flat3",
  # means
  mxPath("one", latent),
  # variance
  mxPath(latent, arrows=2, values=0, free=FALSE)
  )

for (cond in c("Head", "Face", "Voice")) {
  flatModel3 <- mxModel(
    flatModel3,
    mxPath(paste0(manifest, cond), manifest, free=FALSE,
           labels=paste0("data.damp", cond)))
}

for (cond in c("naiveSex", "confedSex")) {  #, "sameSex"
  flatModel3 <- mxModel(
    flatModel3,
    mxPath(paste0(manifest, cond), manifest, free=FALSE,
           labels=paste0("data.", cond)))
}

flatModel3 <- mxModel(model=flatModel3, mxComputeGradientDescent())

flatModel3 <- mxRun(flatModel3)

# ------------------------------------------------------------

mxCompare(flatModel3, c(flatModel2, flatModel1))

save(flatModel3, flatModel2, flatModel1, file="flatModel.rda")
```

## Appendix B
## Multilevel syntax as of June 2013

```r
library(OpenMx)
library(mvtnorm)
library(plyr)

set.seed(1)

#more.noise <- 0
```

```
more.noise <- 1

gen.data <- function(n) {
  data.cov <- matrix(c(1, .2, .2, 1), byrow=TRUE, nrow=2)
  latent <- rmvnorm(n, mean=c(0,0), sigma=data.cov)
  colnames(latent) <- c("A","B")
  latent <- as.data.frame(latent)
  df <- data.frame(C=latent$A + latent$B,
                   D=latent$A - latent$B)
  if (more.noise) {
    df$C <- df$C + rnorm(1, sd=more.noise)
    df$D <- df$D + rnorm(1, sd=more.noise)
  }
  df
}

fanout <- 20

school.data <- cbind(id=1:fanout, gen.data(fanout))
teacher.data <- cbind(schoolId=1:fanout, id=seq(1,fanout^2),
                      gen.data(fanout^2))
student.data <- cbind(teacherId=seq(1,fanout^2), id=seq(1,fanout^3),
                      gen.data(fanout^3))

stack.data <- function (key, upper, lower) {
  lower <- ddply(lower, key, function(slice) {
    id <- unique(slice[[key]])
    parent <- upper[upper$id == id,]
    slice$C <- slice$C + parent$C
    slice$D <- slice$D + parent$C
    slice
  })
}
teacher.data <- stack.data("schoolId", school.data, teacher.data)
student.data <- stack.data("teacherId", teacher.data, student.data)

manifests<-c("C","D")
latents<-c("A","B")
student <- mxModel("student",
                   type="RAM",
                   manifestVars = manifests,
                   latentVars = latents,
                   mxPath(from="A",to=c("C","D"), free=c(FALSE,FALSE),
```

```
                              value=c(1,1) , arrows=1,
                              label=c("A_TO_C","A_TO_D") ),
                     mxPath(from="B",to=c("C","D"), free=c(FALSE,FALSE),
                              value=c(1,-1) ,
                              arrows=1, label=c("B_TO_C","B_TO_D") ),
                     mxPath(from="A",to=c("A","B"), free=c(TRUE,TRUE),
                              value=c(1,0) , arrows=2,
                              label=c("VAR_A","COV_A_B") ),
                     mxPath(from="B",to=c("B"), free=c(TRUE), value=c(1),
                              arrows=2, label=c("VAR_B") ),
                     mxPath(from="C",to=c("C"), free=as.logical(more.noise),
                              value=more.noise, arrows=2, label=c("VAR_C") ),
                     mxPath(from="D",to=c("D"), free=as.logical(more.noise),
                              value=more.noise, arrows=2, label=c("VAR_D") ),
                     mxPath(from="one", to=c(manifests, latents),
                                value=0, free=FALSE)
);

relabel <- function(m, prefix) {
  for (mat in c("A","S")) {
    lab <- m@matrices[[mat]]@labels
    lab[!is.na(lab)] <- paste0(prefix, lab[!is.na(lab)])
    m@matrices[[mat]]@labels <- lab
  }
  m
}

teacher <- relabel(mxModel(student, name="teacher"), "tea_")
school <- relabel(mxModel(student, name="school"), "sch_")
student <- relabel(student, "st_")

school <- mxModel(school, mxData(school.data, type="raw",
                                  primaryKey="id"))
teacher <- mxModel(teacher,
                     mxData(teacher.data, type="raw",
                            primaryKey="id",
                            foreignKeys=list(c('schoolId', 'school.id'))))

student <- mxModel(student,
                     mxData(student.data, type="raw",
                            primaryKey="id",
                            foreignKeys=list(c('teacherId', 'teacher.id'))))
```

```
93
94 sonly <- mxModel(student, type="RAM", name="sonly")
95 sonly <- mxRun(sonly)
96 summary(sonly)
97
98 district <- mxModel("district",
99                     type="RAM",
100                    school, teacher, student,
101
102                  mxPath(from="school.C", to=c("teacher.A"),
103                         free=FALSE, value=1),
104                  mxPath(from="teacher.C", to="student.A",
105                         free=FALSE, value=1),
106
107                  mxExpectationRAM(H=paste0("H",1:2), HomerTransform=TRUE),
108                  mxFitFunctionML()
109 )
110
111 district <- mxRun(district)
112 district@submodels[[1]]@matrices$S@values
113 district@submodels[[2]]@matrices$S@values
114 district@submodels[[3]]@matrices$S@values
```

This OpenMx script demonstrates a number of extensions used to facilitate multilevel modeling. `mxData` takes a primary key (line 83) and a foreign key (line 87). This mirrors the usual arrangement in relational databases and gives the relationship between levels. `mxPath` is extended to permit across-level regressions (line 102). In keeping with the tradition of inscrutable single letter matrix names, the 'H' matrices denote cross level regressions (line 107). A container model is used to gather all the relevant information together (line 98).

## Appendix C
## MultilevelUniRandomSlopeInt.R

```
#
#   Copyright 2007-2015 The OpenMx Project
#
#   Licensed under the Apache License, Version 2.0 (the "License");
#   you may not use this file except in compliance with the License.
#   You may obtain a copy of the License at
#
#        http://www.apache.org/licenses/LICENSE-2.0
#
#   Unless required by applicable law or agreed to in writing, software
#   distributed under the License is distributed on an "AS IS" BASIS,
#   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#   See the License for the specific language governing permissions and
#   limitations under the License.

require(OpenMx)
require(nlme)

```

```
# Multilevel Long Format Test
# Author: Steve Boker
# Date: Sun Nov 29 14:06:07 EST 2009


# This script is used to test the multilevel long format functionality using definition variables
#     as indices.
totalOccasions <- 100
totalSubjects <- 10
set.seed(42) # repeatibility
tID <- rep(1:totalSubjects, each=totalOccasions)
trueX <- rep(rnorm(totalOccasions, mean=0, sd=2), each=totalSubjects) + rnorm(totalOccasions*totalSubjects, mean=
trueB <- rep(rnorm(totalSubjects, mean=.8, sd=.3), each=totalOccasions)
tDataFrame <- data.frame(ID=tID, X=trueX, Y=trueB*trueX + rnorm(totalOccasions*totalSubjects,mean=0, sd=.1),trueB=
summary(tDataFrame)

manifestVars <- c("X", "Y")
numSubjects <- length(unique(tDataFrame$ID))


multilevelModel2 <- mxModel("Multilevel_2",
    mxMatrix("Full", nrow=numSubjects, ncol=2,
        values=c(.2,0),
        free=c(TRUE, TRUE),
        name="Rand",
        byrow=TRUE
    ),
    mxMatrix("Full", 2, 2,
        labels=c(NA,  NA,
                "randrow[1,1]", NA),
        free=FALSE,
        name="A",
        byrow=TRUE
    ),
    mxMatrix("Symm", 2, 2,
        values=c(.9,0,.9),
        free=c(T,
               F, T),
        labels=c("varX",
                 NA, "varY"),
        name="S",
        byrow=TRUE
    ),
        mxMatrix("Full", 2, 2,
                 values=c(1,0,
                                0,1),
                     free=FALSE,
                     byrow=TRUE, name="F"),
    mxMatrix("Iden", 2, name="I"),
    mxAlgebra(F %*% solve(I-A) %*% S %*% t(solve(I-A)) %*% t(F),
        name="R",
        dimnames = list(manifestVars, manifestVars)
    ),
    mxMatrix("Full", nrow=1, ncol=length(manifestVars),
        values=0,
        free=FALSE,
        labels=c(NA,"randrow[1,2]"),
        dimnames=list(NULL, manifestVars),
        name="M"
    ),
        mxAlgebra(Rand[data.ID,], name="randrow"),
    mxFitFunctionML(),mxExpectationNormal(covariance="R", means="M"),
    mxData(tDataFrame, type="raw")
)

# ----------------------------------
# Fit the  model and examine the summary results.

multilevelModel2Fit <- mxRun(multilevelModel2)

summary(multilevelModel2Fit)

lmeOut <- lme(Y~X, random= ~ X | ID, data=tDataFrame)

```

```
cbind(multilevelModel2Fit$output$estimate[1:numSubjects],
        lmeOut$coef$random$ID[,2] + lmeOut$coef$fixed[2],
        trueB[seq(1,totalOccasions*(totalSubjects), by=totalOccasions)])

mean(multilevelModel2Fit$output$estimate[1:numSubjects])

omxCheckCloseEnough(mean(multilevelModel2Fit$output$estimate[1:numSubjects]),
    lmeOut$coef$fixed[2],
    0.001)

omxCheckCloseEnough(mean(multilevelModel2Fit$output$estimate[(1:numSubjects)+(1*numSubjects)]),
    lmeOut$coef$fixed[1],
    0.001)
```

# Appendix D
# UnivariateRandomInterceptWide.R

```
#
#   Copyright 2007-2015 The OpenMx Project
#
#   Licensed under the Apache License, Version 2.0 (the "License");
#   you may not use this file except in compliance with the License.
#   You may obtain a copy of the License at
#
#        http://www.apache.org/licenses/LICENSE-2.0
#
#   Unless required by applicable law or agreed to in writing, software
#   distributed under the License is distributed on an "AS IS" BASIS,
#   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#   See the License for the specific language governing permissions and
#   limitations under the License.

# ----------------------------------------------------------------------------------------------------
# Program: UniRandomIntTest-120815.R
#  Author: Steve Boker
#    Date: Wed Aug 15 10:50:12 CEST 2012
#
# This program simulates some univariate multilevel data with random intercepts only,
#   fits it with lme(), fits a naive wide format multilevel OpenMx model and
#   checks the results
#
# ----------------------------------------------------------------------------------------------------
# Revision History
#   Steve Boker -- Wed Aug 15 10:50:14 CEST 2012
#      Created UniRandomIntTest-120815.R
#
# ----------------------------------------------------------------------------------------------------

# ----------------------------------------------
# Read libraries and set options.

options(width=110)
library(nlme)
library(OpenMx)

# ----------------------------------------------
# Set constants.

sdLevelOneE <- sqrt(.2)
sdIntercepts <- sqrt(.5)
sdX <- sqrt(1)

N <- 400     # number of participants
P <- 100     # number of observations per participant
b0 <- .5     # Fixed effect intercept
b1 <- .8     # Fixed effect slope

set.seed(1)

# ----------------------------------------------
# Simulate the data.

```

```
X <- rnorm(N*P, 0, sd=sdX)
ID <- rep(1:N, each=P)
b0i <- b0 + rnorm(N, 0, sd=sdIntercepts)
Y <- rep(b0i, each=P) + b1*X + rnorm(N*P, 0, sd=sdLevelOneE)

SimUniRandomIntFrame <- data.frame(ID, X, Y)

# -----------------------------------
# Test with lme().

lmeOut <- summary(lme(Y ~ X, random= list(~ 1 | ID), data=SimUniRandomIntFrame))

# For lme4, use:
# lmerOut <- lmer(Y ~ X + (1 | ID), data=SimUniRandomIntFrame)

# -----------------------------------
# Set constants.

theIDs <- unique(SimUniRandomIntFrame$ID)
totalN <- length(theIDs)
totalVars <- 2

maxP <- 0
for (tID in theIDs) {
    tLen <- length(SimUniRandomIntFrame$ID[SimUniRandomIntFrame$ID==tID])
    if (tLen > maxP)
        maxP <- tLen
}

# -----------------------------------
# Wide-format the data frame from tall format.

wideMatrix <- matrix(NA, nrow=totalN, ncol=1 + (maxP*totalVars))
colnames(wideMatrix) <- c("ID", paste("Y",1:maxP, sep=""), paste("X",1:maxP, sep=""))
i <- 1
for (tID in theIDs) {
    wideMatrix[i, 1] <- tID
    tY <- SimUniRandomIntFrame$Y[SimUniRandomIntFrame$ID==tID]
    wideMatrix[i, 2:(length(tY)+1)] <- tY
    tX <- SimUniRandomIntFrame$X[SimUniRandomIntFrame$ID==tID]
    wideMatrix[i, (2+maxP):(length(tY)+1+maxP)] <- tX
    i <- i + 1
}
wideFrame <- data.frame(wideMatrix)

manifestNames <- colnames(wideFrame)[2:dim(wideFrame)[2]]
xNames <- paste("X",1:maxP, sep="")
yNames <- paste("Y",1:maxP, sep="")
latentNames <- c("b0i")

# -----------------------------------
# Build the OpenMx wide model.

OpenMxModelUniRandomIntModel1 <- mxModel("OpenMxModelUniRandomIntModel1",
        type="RAM",
        manifestVars=manifestNames,
    latentVars=latentNames,
    mxPath(from=xNames, to=yNames, connect="single", arrows=1, free=TRUE, values=.2, labels="b1"),
    mxPath(from=xNames, to=xNames, connect="single", arrows=2, free=TRUE, values=.8, labels="vX"),
    mxPath(from=yNames, to=yNames, connect="single", arrows=2, free=TRUE, values=.8, labels="eY"),
    mxPath(from=latentNames, to=yNames, arrows=1, free=FALSE, values=1),
    mxPath(from=latentNames, to=latentNames, connect="single", arrows=2, free=TRUE, values=.8, labels="vb0i"),
    mxPath(from="one", to=c(xNames), arrows=1, free=TRUE, values=1, labels="mX"),
    mxPath(from="one", to=c(latentNames), arrows=1, free=TRUE, values=1, labels="mb0i"),
    mxData(observed=wideFrame, type="raw")
)

# -----------------------------------
# Fit the model and examine the summary results.

OpenMxModelUniRandomIntModel1Fit <- mxRun(OpenMxModelUniRandomIntModel1)

summary(OpenMxModelUniRandomIntModel1Fit)
```

```
omxCheckCloseEnough(lmeOut$coefficients$fixed[1], mxEval(mb0i, model=OpenMxModelUniRandomIntModel1Fit), 0.001)

omxCheckCloseEnough(lmeOut$coefficients$fixed[2], mxEval(b1, model=OpenMxModelUniRandomIntModel1Fit), 0.001)

omxCheckCloseEnough(lmeOut$sigma, mxEval(sqrt(eY), model=OpenMxModelUniRandomIntModel1Fit), 0.001)

omxCheckCloseEnough(sd(c(lmeOut$coefficients$random$ID)), mxEval(sqrt(vb0i), model=OpenMxModelUniRandomIntModel1F

if (0) {
  omxCheckCloseEnough(lmeOut$coefficients$fixed, fixef(lmerOut), 1e-4)
  omxCheckCloseEnough(lmeOut$sigma, sigma(lmerOut), 1e-4)
  omxCheckCloseEnough(c(lmeOut$coefficients$random$ID), ranef(lmerOut)$ID[[1]], 1e-4)
}
```

# Appendix E
## similar.R

```
library(nlme)
library(dplyr)
load("3level.rda")
load("tFrame.rda")

tFrame$lnselfyRotFV <- log(tFrame$selfyRotFV)
tFrame$lnselfxRotFV <- log(tFrame$selfxRotFV)
tFrame$lnotheryRotFV <- log(tFrame$otheryRotFV)
tFrame$lnotherxRotFV <- log(tFrame$otherxRotFV)

headAPlme2 <- lme(lnselfyRotFV ~ selfSex + otherSex + isConfed +
                    dampHead + dampFace + dampVoice + lnotheryRotFV +
                    confedByOtherSex + confedByDampHead +
                    confedByDampFace + confedByDampVoice,
                  random= ~ 1 | naiveID, data=tFrame)

headLlme2 <- lme(lnselfxRotFV ~ selfSex + otherSex + isConfed +
                   dampHead + dampFace + dampVoice + lnotherxRotFV +
                   confedByOtherSex + confedByDampHead +
                   confedByDampFace + confedByDampVoice,
                 random= ~ 1 | naiveID, data=tFrame)

xConst5 <- fivenum(abs(fixef(headAPlme2)[-1]))
yConst5 <- fivenum(abs(fixef(headLlme2)[-1]))

flat <- left_join(left_join(by_naive, by_pair), by_trial)
flat$sameSex <- as.numeric(flat$naiveSex == flat$confedSex)

# c := confed, n := naive, x := lateral, y := A-P
manifest <- c('cx','cy','nx','ny')

resample <- function(df, id) {
```

```
  for (cx in 1:ncol(df)) {
    if (any(colnames(df)[cx] == id)) next
    orig <- df[[cx]]
    df[[cx]] <- rnorm(length(orig), mean=mean(orig), sd = sd(orig))
  }
  df
}

cint <- summarize(group_by(flat, confedID), cx=mean(cx), cy=mean(cy))
cint <- resample(cint, "confedID")
nint <- summarize(group_by(flat, naiveID), nx=mean(nx), ny=mean(ny))
nint <- resample(nint, "naiveID")
pint <- summarize(group_by(flat, pairID),
                  cx=mean(cx), cy=mean(cy), nx=mean(nx), ny=mean(ny))
pint <- resample(pint, "pairID")
pvar <- summarize(group_by(flat, pairID),
                  cx=var(cx), cy=var(cy), nx=var(nx), ny=var(ny))

constEffect <- NULL
for (cond in c("Head","Face","Voice","naiveSex","confedSex","sameSex")) {
  coef <- c(sample(c(0, xConst5[3], xConst5[5]), 2, replace=TRUE),
            sample(c(0, yConst5[3], yConst5[5]), 2, replace=TRUE))
  names(coef) <- NULL
  constEffect <- rbind(
    constEffect, data.frame(cond=cond, part=manifest, coef=coef))
}
constEffect$coef <- (constEffect$coef *
                       sample(c(-1,1), nrow(constEffect), replace=TRUE))

for (fx in 1:nrow(flat)) {
  f1 <- as.list(flat[fx,])
  c1 <- cint[cint$confedID == f1$confedID,]
  n1 <- nint[nint$naiveID == f1$naiveID,]
  p1 <- pint[pint$pairID == f1$pairID,]
  noise <- as.list(pvar[pint$pairID == f1$pairID,])
  int <- c(cx=mean(c1$cx, p1$cx) + rnorm(1, sd=sqrt(noise$cx)),
           cy=mean(c1$cy, p1$cx) + rnorm(1, sd=sqrt(noise$cy)),
           nx=mean(n1$nx, p1$nx) + rnorm(1, sd=sqrt(noise$nx)),
           ny=mean(n1$ny, p1$ny) + rnorm(1, sd=sqrt(noise$ny)))
  for (cond in c("naiveSex", "confedSex", "sameSex")) {
    if (f1[[cond]]) {
      int <- int + constEffect[constEffect$cond == cond, 'coef']
    }
```

```
  }
  for (cond in c("Head","Face","Voice")) {
    dcond <- paste0("damp", cond)
    if (f1[[dcond]]) {
      int <- int + constEffect[constEffect$cond == cond, 'coef']
    }
  }
  flat[fx, manifest] <- int
}

# analyze a random dataset (for example)

tFrame$lnselfyRotFV <- c(flat$cy, flat$ny)
tFrame$lnselfxRotFV <- c(flat$cx, flat$nx)

headAPlme3 <- lme(lnselfyRotFV ~ selfSex + otherSex + isConfed +
                    dampHead + dampFace + dampVoice + lnotheryRotFV+
                    confedByOtherSex + confedByDampHead +
                    confedByDampFace + confedByDampVoice,
                  random= ~ 1 | naiveID, data=tFrame)

headLlme3 <- lme(lnselfxRotFV ~ selfSex + otherSex + isConfed +
                   dampHead + dampFace + dampVoice + lnotherxRotFV +
                   confedByOtherSex + confedByDampHead +
                   confedByDampFace + confedByDampVoice,
                 random= ~ 1 | naiveID, data=tFrame)
```