

Many-level multilevel structural equation modeling: An efficient evaluation strategy

Joshua N. Pritikin

Virginia Commonwealth University

Michael D. Hunter

University of Oklahoma Health Sciences Center

Timo von Oertzen

Universität der Bundeswehr, München

Timothy R. Brick

Pennsylvania State University

Steven M. Boker

University of Virginia

Author Note

Joshua N. Pritikin, Department of Psychiatry, Virginia Commonwealth University; Michael D. Hunter, Department of Pediatrics, University of Oklahoma Health Sciences Center; Timo von Oertzen, Universität der Bundeswehr, München, Department für Psychologie; Timothy R. Brick, Human Development and Family Studies, Pennsylvania State University; Steven M. Boker, Department of Psychology, University of Virginia.

M.H. assisted in debugging and contributed the example application. T.O., T.B., and S.B. conceived Rampart and contributed equally to its conceptual design. This research was supported in part by the National Institute of Health (R01-DA018673) and a Jefferson Trust Big Data fellowship. We acknowledge the **OpenMx** team for helpful feedback.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health or the Jefferson Trust.

Correspondence concerning this article should be addressed to Joshua N. Pritikin, Virginia Institute for Psychiatric and Behavioral Genetics, Virginia Commonwealth University, PO Box 980126, 800 E Leigh St, Biotech One, Suite 1-133, Richmond, VA 23219. E-mail: jpritikin@pobox.com

Abstract

Structural equation models are increasingly used for clustered or multilevel data in cases where mixed regression is too inflexible. However, when there are many levels of nesting, these models can become difficult to estimate. We introduce a novel evaluation strategy, Rampart, that applies an orthogonal rotation to the parts of a model that conform to commonly met requirements. This rotation dramatically simplifies fit evaluation in a way that becomes more potent as the size of the data set increases. We validate and evaluate the implementation using a 3-level latent regression simulation study. Then we analyze data from a state-wide child behavioral health measure administered by the Oklahoma Department of Human Services. We demonstrate the efficiency of Rampart compared to other similar software using a latent factor model with a 5-level decomposition of latent variance. Rampart is implemented in OpenMx, a free and open source software.

Keywords: Relational database theory; big data; multilevel models; hierarchical linear models, open-source software

Many-level multilevel structural equation modeling: An efficient evaluation strategy

Introduction

As hypotheses become more elaborate, data need to be collected on more than one level. For example, a hypothesis about the effect of a teacher on her students cannot be tested without collecting data on both. These data are multilevel because there is not a 1-to-1 relationship between students and teacher. Since there are fewer teachers than students, teachers are regarded as the *upper* level and students as the *lower* level (see Figure 1).

The theoretical basis for maximum likelihood (ML) analysis of 2-level data is well-researched and many software implementations are available (S. H. du Toit & M. du Toit, 2008; Lee & Poon, 1998; B. O. Muthén, 1994). However, as the number of levels increase, existing methods experience difficulty. Here we introduce *Rampart* (Pritikin, 2016), an efficient evaluation strategy for many-level multilevel structural equation models. We start with a brief review of relational database theory and describe why the naïve approach to ML evaluation exhibits poor performance. Next, the similarities and differences between conditional probability and the relational join operator are clarified. Rampart’s `OpenMx` model specification is introduced by comparison with `lme4`’s formula model specification language. Details of the Rampart algorithm are given. To validate the implementation, we include a 3-level simulation study. Then we analyze data from a state-wide child behavioral health measure administered by the Oklahoma Department of Human Services. Performance on a 3-level model is compared among `OpenMx`, `Mplus`, `lme4`, and `nlme`. Finally, we exhibit a novel latent factor model with a 5-level decomposition of latent variance. We are unaware of other software, besides `OpenMx`, that can efficiently estimate this model.

Relational Database Theory

Although relational databases have been in wide use since at least the 1980s, it seems necessary to review the rudiments of relational theory for the statistical community as statistical models are only recently gaining the capability to model complex data. Data with complex structure are often stored in relational databases. In preparation for storage, data are typically 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 recovery of the relationships between the data tables by the join of primary and foreign keys (e.g., Maier, 1983).

Formally, the relational join operator can be defined as follows. Let R and S be tables (or data frames) that contain rows. A row is a single unit of data, like the data for one teacher or one student. Following standard relational database theory, the join operator (\bowtie) is defined as,

$$R \bowtie(F) S \equiv \{r \cup s \wedge r \in R \wedge s \in S \wedge F(r \cup s)\} \quad (1)$$

where F is a boolean valued function. Without loss of generality, here F tests whether primary and foreign keys match. We will omit F and write $\bowtie(k)$ where k is the name of the key. An example join of employee and department tables is given in Figure 2. The result of the join of two tables can itself be joined against another table allowing an unlimited number of tables to be joined together.

Two more terms are useful to describe data structure, *nested* and *crossed*. The distinction between nested and crossed data is useful because nested data are easier to statistically evaluate than crossed data. Data are nested when each lower level unit is associated with exactly one upper level unit and there are only associations between adjacent levels. When data are not nested then they are crossed. One set

of crossed associations need not be organized in relation to other crossed associations. Crossed associations may partition data 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 single teachers but some will have two or more teachers. Students with more than one teacher are regarded as crossed.

A model of these data ought to account for the multilevel structure and permit multivariate covariance modeling. Two popular approaches, univariate multilevel regression and structural equations modeling (SEM) each offer one but not both of these capabilities. Multilevel regression is typically limited to a single response variable whereas a SEM is often limited to at most 3-level models. Some effort has been expended to combine the flexibility of both approaches (Goldstein & McDonald, 1988; Krull & MacKinnon, 2001; McDonald, 1993; Mehta & Neale, 2005; Muthen, 1997; Raudenbush & Sampson, 1999). However, these early attempts have not seen wide use because model evaluation time rapidly becomes intractable as the number of levels increase.

A Closer Look at Multilevel Covariance

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 specify our relationships in terms of latent factors because we cannot measure the constructs of interest directly. We incorporate varying (a.k.a. random; Gelman, 2005) 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, perhaps entire states, because within states we allow that everybody may have some effect on everybody else.

It may be helpful to sketch out more concretely the structure of our hypothetical multilevel covariance matrix. To keep things simple, assume that the data are nested (not crossed). We introduce the *direct sum* operator,

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

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

to conveniently construct these matrices. Suppose we build a covariance model \mathbf{S} for a particular student. A classroom of s students will have covariance matrix

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_{1,1} & \mathbf{T}_{1,2} \\ \mathbf{T}_{2,1} & \bigoplus_{i=1}^s \mathbf{S}_i \end{pmatrix}. \quad (2)$$

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

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{1,2} \\ \mathbf{H}_{2,1} & \bigoplus_{i=1}^t \mathbf{T}_i \end{pmatrix} \quad (3)$$

and a district of h schools will have covariance matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} \\ \mathbf{D}_{2,1} & \bigoplus_{i=1}^h \mathbf{H}_i \end{pmatrix}. \quad (4)$$

Suppose we have data \mathbf{y} from many school districts. 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). With some regularity assumptions, the log Gaussian density can be written as,

$$\ell(\mathbf{y}|\boldsymbol{\theta}) = \sum_i \left[-\frac{1}{2} [K \log(2\pi) + \log(|\boldsymbol{\Sigma}|)] - \frac{1}{2} (\boldsymbol{\mu} - \mathbf{y}_i)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{y}_i) \right]. \quad (5)$$

The bottleneck in the evaluation of Equation 5 is the matrix inverse of the model implied covariance matrix $\boldsymbol{\Sigma}$. Gauss-Jordan matrix inverse requires $O(n^3)$ operations and the covariance matrix can quickly become very large. For example, if we only have 3 outcomes of interest at the student level and 1 variable at all the other levels then the dimension of \mathbf{D} still becomes large even with a modest number of units at each level. If there are 10 students per teacher, 5 teachers per school, and 6 schools per district, then S is 3×3 , T is 31×31 , H is 156×156 , and D is 937×937 . Although this matrix is very sparse, the highest level unit can relate to all the lowest level units leaving meager opportunity for truly independent blocks. To fit multilevel models quickly, it is essential to analyze the structure of this matrix and devise some way to reduce its dimension.

Model Specification

In `OpenMx`, the universal building block of statistical models is the `MxMatrix`. An `MxMatrix` is an object which contains five separate R matrix layers, all of the same size: The `values` matrix holds the starting (or estimated) values and is of

type double. The `labels` matrix is of type character and holds the name of each element of the matrix. Matrix elements that have the same name are constrained to be equal to one another. The `free` matrix is of type logical and if an element is `TRUE`, then that element is considered a free parameter during estimation. The `lbound` and `ubound` matrices are of type double and contain lower and upper bounds for the free parameters (Boker et al., 2011).

In `OpenMx`, joins were facilitated by the addition `joinKey` and `joinModel` to the `MxMatrix` object and the addition of `primaryKey` to `mxData`. `MxMatrix` objects are always contained in an `MxModel`. We will call this model the `MxMatrix`'s home model. When a join is performed, the specified `joinModel` is joined against the home model using the `joinKey` column in the home model to match against the `primaryKey` column in the `joinModel`. For `mxPath`, a more friendly interface for specifying `MxMatrix` objects, the join model is named in the `from` parameter (i.e., `from='joinModel.column'`). An example may better illustrate how this works.

A Mixed Model Translated to `OpenMx`

Some popular R packages that implement the mixed model (e.g., D. Bates, Mächler, Bolker, & Walker, 2015; Pinheiro, Bates, DebRoy, Sarkar, & R Core Team, 2016) follow a model specification syntax that evolved from the notation for conditional probability instead of the notation used by relational databases. Formula notation (Wilkinson & Rogers, 1973) for specifying a regression equation was augmented with a vertical bar clause. For example,

```
lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
```

The left part of the regression equation, up to the parenthesis enclosing the vertical bar, follows standard formula notation. The vertical bar clause is used to specify varying coefficients. The part after the vertical bar (`Subject`) names a factor (a column in the data frame) that partitions the data set. The formula before the

vertical bar (`Days`) is joined to the base model according to this factor. Since some researchers with statistical training are familiar with the vertical bar notation but not with relational databases, it is worth emphasizing a difference between the two. The join operator (Equation 1) is commutative. That is,

$$student \bowtie (student.teacherID) teacher \quad (6)$$

$$teacher \bowtie (student.teacherID) student \quad (7)$$

are equivalent. In contrast, $P(Noah|Jane)$ and $P(Jane|Noah)$ are almost certainly different quantities (refer to Figure 1). The way the vertical bar is used in formula notation involves both ideas. The formula inside the parentheses on the left is conditional on the partitioning factor given on the right. Data must have already been joined into a single table, but the partitioning factor could also be regarded as the key on which the data were joined.

Since the formula-style specification is popular, it is hoped that the translation from it to an equivalent `OpenMx` model will be an easy way for readers to quickly grasp `OpenMx` model specification. While specification of `OpenMx` models is more laborious than formula notation, `OpenMx` makes assumptions explicit, and permits multivariate and latent variable models. We use the RAM parameterization (McArdle, 2005; McArdle & McDonald, 1984). The RAM model consists of 4 matrices, traditionally called \mathbf{A} (asymmetric), \mathbf{S} (symmetric), \mathbf{F} (filter), and \mathbf{M} (mean). The RAM matrices are related to the model's Gaussian distribution by,

$$\boldsymbol{\mu} = \mathbf{F}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{M} \quad (8)$$

$$\boldsymbol{\Sigma} = \mathbf{F}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{S}(\mathbf{I} - \mathbf{A})^{-T}\mathbf{F}^T. \quad (9)$$

```
1 SubjectData <- unique(sleepstudy$Subject)
```

```
2
```

```

3 bySubj <- mxModel(
4   model="bySubj", type="RAM",
5   latentVars=c("slope", "intercept"),
6   mxData(data.frame(Subject=SubjectData),
7     type="raw", primaryKey = "Subject"),
8   mxPath(from=c("intercept", "slope"), arrows=2, values=1),
9   mxPath(from="intercept", to="slope", arrows=2,
10    values=.25, labels="cov1"))
11
12 sleepModel <- mxModel(
13   model="sleep", type="RAM", bySubj,
14   manifestVars="Reaction", latentVars = "Days",
15   mxData(sleepstudy, type="raw"),
16   mxPath(from="one", to="Reaction", arrows=1, free=TRUE),
17   mxPath(from="one", to="Days", arrows=1,
18     free=FALSE, labels="data.Days"),
19   mxPath(from="Days", to="Reaction", arrows=1, free=TRUE),
20   mxPath(from="Reaction", arrows=2, values=1),
21   mxPath(paste0('bySubj.', c('intercept', 'slope')),
22     'Reaction', arrows=1, free=FALSE, values=c(1,NA),
23     labels=c(NA, "data.Days"), joinKey="Subject"))

```

We create an `mxModel` to contain the per-Subject model (line 3).

Traditionally, the mixed model does not permit response observations in upper levels. Hence, upper levels in this example only contain latent variables (line 5).

The `Subject` model's data contains no observations, only primary keys (line 6).

Conceptually, we would like to allow a per-Subject coefficient for `intercept` and `slope`. It may be surprising that this is accomplished by estimating the variance of those varying coefficients and not the coefficients themselves (line 8). We estimate the covariance between varying `intercept` and `slope` (line 10).

We include the upper level model as a submodel of the base model (line 13). Figure 3b pictorially describes this nesting structure for multilevel models. `OpenMx` treats this as equivalent to a more parallel model structure as depicted in Figure 3a. and other possible organizations are discussed in . The constant coefficients are specified starting at line 16. The predictor `Days` is included in the model as a definition variable (a value provided by the analyst) to function as a zero variance regression (line 18). This warrants a brief digression.

In SEM, it is customary to assume a parametric distribution for both predictor and response variables. In contrast, regression models only assume a parametric distribution for the residuals; no distributional assumption is made about predictors. There are pros and cons to both approaches.

A major advantage of assuming a distribution for predictors is that there is accounting for measurement error (Westfall & Yarkoni, 2016) and missing data are less of a problem (e.g., Enders & Bandalos, 2001). However, when predictors are not missing and have no measurement error then modeling predictors adds extra parameters for little gain. For example, a script from the `OpenMx` test suite, `UnivariateRandomInterceptWide.R`, implements a single predictor univariate random intercept model. The standard regression approach estimates 4 parameters (residual variance, intercept, constant regression coefficient, and varying intercept variance), but `UnivariateRandomInterceptWide.R` also estimates the mean and variance of predictor X , adding 2 parameters for a total of 6. The parameters that are common among these two models have matching estimates, so why estimate an extra 2 parameters unless they are of substantive interest? For optimal performance, the analyst should think carefully about whether a predictor needs to be parametrically modeled or can be included in the model as a zero variance regression.

The connections between the per-`Subject` and base models are set up at line 21. An executable version of this code is available in the Appendix. While the `OpenMx` is not as succinct as `lmer`, the `OpenMx` model could easily be extended to incorporate multivariate data such as digit span in addition to reaction time. Another `lmer` translation example using the `Orthodont` data set is available in the `OpenMx` test suite. All mixed models can be similarly translated into `OpenMx` models. Each vertical bar clause is implemented with a latent `mxModel` to specify extra variance to account for the varying coefficients. These latent `OpenMx` models are joined to the corresponding constant coefficients in the base model using fixed loadings (typically 1.0).

Upper to lower level transition matrices are of type `MxMatrix` and can take advantage of the usual `OpenMx` capabilities. A transition matrix can contain free parameters, definition variables, or populated values using square bracket notation. Or for maximum flexibility, transition matrices can be specified as the result of an `mxAlgebra`, an arbitrary algebraic expression.

Efficient Evaluation

We trace through the steps involved in our novel evaluation strategy for nested multilevel structure. We review how the Gaussian distribution is invariant to orthogonal rotation, show how to use the QR decomposition algorithm to create a specific axis rotation, and introduce the novel Rampart rotation to dramatically improve independence in multilevel covariance matrices. Rampart performance benefits and limitations are described. To validate the implementation, we include a simulation study.

Topological Sort

Once a relational SEM is specified, each row must be assigned to a location in a model-wide covariance matrix (Goldstein & McDonald, 1988). There are many

possible assignments of rows to covariance locations. One type of ordering that offers a computational advantage is a topological sort. We can regard a relational SEM as a directed graph. If we add the restriction that cycles are not allowed then we can sort the graph by dependency. Units without dependency on other units come first and then dependent units. For example, refer to Figure 4. This ordering allows us to compute the model expected mean unit-wise instead of model-wise.

Gaussian Density Rotation

An intuitive argument is given in Figure 5. Here we work through the equations to show exactly how an orthogonal rotation \mathbf{Q} cancels out of the Gaussian likelihood. The $-2 \log$ density of a single observation \mathbf{x} from the K dimensional Gaussian distribution is,

$$K \log(2\pi) + \log(|\Sigma|) + (\boldsymbol{\mu} - \mathbf{x})^T \Sigma^{-1} (\boldsymbol{\mu} - \mathbf{x}). \quad (10)$$

Suppose we want to apply an orthogonal rotation \mathbf{Q} to \mathbf{x} . The rotated density is,

$$K \log(2\pi) + \log(|\mathbf{Q}\Sigma\mathbf{Q}^T|) + (\mathbf{Q}(\boldsymbol{\mu} - \mathbf{x}))^T \mathbf{Q}\Sigma^{-1}\mathbf{Q}^T (\mathbf{Q}(\boldsymbol{\mu} - \mathbf{x})). \quad (11)$$

We know that $|\mathbf{Q}\Sigma\mathbf{Q}^T|$ is equal to $|\Sigma|$ because $|\mathbf{Q}\Sigma\mathbf{Q}^T| = |\mathbf{Q}||\Sigma||\mathbf{Q}^T| = |\mathbf{Q}||\mathbf{Q}^T||\Sigma| = 1|\Sigma|$. For the term on the right, we can expand the transpose, regroup, and use the fact that $\mathbf{Q}^{-1} = \mathbf{Q}^T$,

$$(\mathbf{Q}(\boldsymbol{\mu} - \mathbf{x}))^T \mathbf{Q}\Sigma^{-1}\mathbf{Q}^T (\mathbf{Q}(\boldsymbol{\mu} - \mathbf{x})) \quad (12)$$

$$\left((\boldsymbol{\mu} - \mathbf{x})^T \mathbf{Q}^T \right) \mathbf{Q}\Sigma^{-1}\mathbf{Q}^T (\mathbf{Q}(\boldsymbol{\mu} - \mathbf{x})) \quad (13)$$

$$(\boldsymbol{\mu} - \mathbf{x})^T (\mathbf{Q}^T \mathbf{Q}) \Sigma^{-1} (\mathbf{Q}^T \mathbf{Q}) (\boldsymbol{\mu} - \mathbf{x}) \quad (14)$$

$$(\boldsymbol{\mu} - \mathbf{x})^T I \Sigma^{-1} I (\boldsymbol{\mu} - \mathbf{x}) \quad (15)$$

$$(\boldsymbol{\mu} - \mathbf{x})^T \Sigma^{-1} (\boldsymbol{\mu} - \mathbf{x}). \quad (16)$$

QR Decomposition

QR decomposition is a versatile procedure that can be used to accomplish a variety of goals. QR decomposition expresses a matrix \mathbf{A} as the product of orthogonal matrix \mathbf{Q} and upper triangular matrix \mathbf{R} . The matrix \mathbf{A} must be m -by- n with $m \geq n$. Here we describe how to use the QR decomposition algorithm to create an orthogonal axis rotation that we can plug into the Gaussian density (Equation 11). Hence, \mathbf{A} will always be m -by- m (square) and full rank. Let \mathbf{x} be an arbitrary column vector of \mathbf{A} of length $|\alpha|$. One Householder reflection consists of,

$$\mathbf{u} = \mathbf{x} + \text{sign}(x_1)\alpha [1, 0, \dots, 0]^T \quad (17)$$

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|} \quad (18)$$

$$\mathbf{Q} = \mathbf{I} - 2\mathbf{v}\mathbf{v}^T. \quad (19)$$

In Equation 17, we choose the sign to increase the magnitude of the first entry of \mathbf{x} . This ensures the length of \mathbf{u} is at least α . Vector \mathbf{u} can be regarded as the average of the direction of \mathbf{x} and the target axis. Vector \mathbf{v} is the reflection pivot. The obtained \mathbf{Q} will zero out all except the first row of \mathbf{x} such that,

$$\mathbf{QA} = \begin{bmatrix} \alpha_1 & \star & \dots & \star \\ 0 & & & \\ \vdots & \mathbf{A}' & & \\ 0 & & & \end{bmatrix}. \quad (20)$$

The process is repeated on \mathbf{A}' until \mathbf{QA} is upper triangular, generating a series of rotations $\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_m$.

To illustrate the process, let us perform a rotation to an arbitrary basis,

$$\mathbf{A} = \begin{bmatrix} 2.28 & & \\ 1.50 & 1.01 & \\ 1.31 & 2.28 & 0.86 \end{bmatrix}. \quad (21)$$

We place the basis vectors in the lower triangle because the QR algorithm is blind to the upper triangle. The first reflection obtains,

$$\mathbf{x}_1 = \begin{bmatrix} 2.28 \\ 1.50 \\ 1.31 \end{bmatrix} \quad (22)$$

$$\alpha_1 = \|\mathbf{x}_1\| = 3.03 \quad (23)$$

$$\mathbf{u} = \mathbf{x}_1 + \text{sign}(x_{1,1})\alpha_1 [1, 0, \dots, 0]^T = \begin{bmatrix} 5.31 \\ 1.50 \\ 1.31 \end{bmatrix} \quad (24)$$

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|} = \begin{bmatrix} 0.94 \\ 0.26 \\ 0.23 \end{bmatrix} \quad (25)$$

$$\mathbf{Q}_1 = \mathbf{I} - 2\mathbf{v}\mathbf{v}^T = \begin{bmatrix} -0.75 & -0.50 & -0.43 \\ -0.50 & 0.86 & -0.12 \\ -0.43 & -0.12 & 0.89 \end{bmatrix}. \quad (26)$$

As expected, \mathbf{Q}_1 zeros all but the first entry of the first column of \mathbf{A} ,

$$\mathbf{Q}_1\mathbf{A} = \begin{bmatrix} -3.03 & -1.49 & -0.37 \\ & 0.59 & -0.11 \\ & 1.91 & 0.77 \end{bmatrix}.$$

We continue with the second reflection,

$$\mathbf{x}_2 = \begin{bmatrix} 0.59 \\ 1.91 \end{bmatrix} \quad (27)$$

$$\alpha_2 = \|\mathbf{x}_2\| = 2 \quad (28)$$

$$\mathbf{u} = \mathbf{x}_2 + \text{sign}(x_{2,1})\alpha_2 [1, 0, \dots, 0]^T = \begin{bmatrix} 2.59 \\ 1.91 \end{bmatrix} \quad (29)$$

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|} = \begin{bmatrix} 0.80 \\ 0.59 \end{bmatrix} \quad (30)$$

$$\mathbf{Q}_2 = \mathbf{I} - 2\mathbf{v}\mathbf{v}^T = \begin{bmatrix} 1.00 & & \\ & -0.29 & -0.96 \\ & -0.96 & 0.29 \end{bmatrix}. \quad (31)$$

\mathbf{Q}_2 is 2-by-2, but we fill it with the identity matrix to expand it back to m -by- m . \mathbf{A} is fully decomposed. We obtain,

$$\mathbf{Q} = \mathbf{Q}_2\mathbf{Q}_1 = \begin{bmatrix} -0.75 & -0.50 & -0.43 \\ 0.56 & -0.14 & -0.82 \\ 0.35 & -0.86 & 0.38 \end{bmatrix} \quad (32)$$

$$\mathbf{R} = \mathbf{Q}_2\mathbf{Q}_1\mathbf{A} = \begin{bmatrix} -3.03 & -1.49 & -0.37 \\ & -2.00 & -0.70 \\ & & 0.33 \end{bmatrix} \quad (33)$$

However, this \mathbf{Q} is the inverse of what we want. We want the rotation from the identity axis to the axis described by \mathbf{A} . Hence, the desired rotation is \mathbf{Q}^T . With a deeper understanding of axis rotation, we have the tools we need to describe the Rampart rotation.

Rampart Rotation

Let us take a close look at the model in Figure 6. This model is identified with only two teachers. With only 8 observations, the matrices are compact enough to investigate the full model. First we examine the model implied covariance (Equation 9). Our model has no latent variables so the \mathbf{F} matrix is set to the identity. Parameters are assigned arbitrary values.

$$\mathbf{A} = \begin{bmatrix} 1.07 \\ 1.07 \\ 1.07 \end{bmatrix} \quad (34)$$

$$\mathbf{S} = \begin{bmatrix} 0.29 & & & \\ & 0.70 & & \\ & & 0.70 & \\ & & & 0.70 \end{bmatrix} \quad (35)$$

$$\mathbf{\Sigma} = (\mathbf{I} - \mathbf{A})^{-1} \mathbf{S} (\mathbf{I} - \mathbf{A})^{-T} = \begin{bmatrix} 0.29 & 0.31 & 0.31 & 0.31 \\ 0.31 & 1.04 & 0.33 & 0.33 \\ 0.31 & 0.33 & 1.04 & 0.33 \\ 0.31 & 0.33 & 0.33 & 1.04 \end{bmatrix} \quad (36)$$

We obtain a 4-by-4 covariance matrix instead of 8-by-8 since both sets of teacher-and-students have the same model. However, this efficiency gain of grouping by independence does not help much if we add more students. A classroom with a few hundred students with many observations per student requires a large covariance matrix. Observe that $\mathbf{\Sigma}$ follows the structure described in Equation 2, 0.29 corresponds to $T_{1,1}$, 0.31 to $T_{1,2}$ and $T_{2,1}$, 1.04 to S_i , and the rest of the entries are covariances between students induced by having the same teacher.

Observe that λ , the regression from teacher to student, is a single parameter that is some function of the mean of the students. This is true regardless of the number of students. Instead of dispersing the information about the mean across all the students, suppose we could rotate the data such that the mean was already computed and readily available. In fact, we can.

Let us use a QR decomposition find an orthogonal rotation to column basis vectors,

$$\begin{bmatrix} 1.00 & 2.00 \\ 1.00 & -1.00 & 1.00 \\ 1.00 & -1.00 & -1.00 \end{bmatrix}. \quad (37)$$

These vectors are not normalized to unit length to make it easier to understand the construction. The first vector obtains a value proportional to the mean. The remaining basis vectors consist of an arbitrary orthogonal contrast, Helmert contrasts in this case. QR decomposition obtains

$$\mathbf{Q}^T = \begin{bmatrix} -0.58 & -0.58 & -0.58 \\ 0.82 & -0.41 & -0.41 \\ & -0.71 & 0.71 \end{bmatrix}. \quad (38)$$

We apply this rotation to the 3 student values associated with the first teacher,

$$\mathbf{Q}^T \begin{bmatrix} 0.69 \\ -2.03 \\ -0.98 \end{bmatrix} = \begin{bmatrix} 1.34 \\ 1.79 \\ 0.74 \end{bmatrix}. \quad (39)$$

The mean of the first 3 students is -0.77 . The value obtained (1.34) is $-\sqrt{3}$ times the mean. The wrong sign is due to rotational indeterminacy. We can take $-\mathbf{Q}^T$ instead of \mathbf{Q}^T . The $\sqrt{3}$ factor results from the need to preserve the length of the

many levels such that most of the nonzero regions in a very large multilevel covariance structure (e.g., Equation 4) become independent. Note that the rotated \mathbf{A}^* matrix (Equation 41) is only used to compute the covariance (Equation 9). Although \mathbf{A} also appears in the computation of the expected means (Equation 8), this equation uses the unrotated \mathbf{A} . The residuals are rotated, not (somehow) the predicted means (refer to Equation 11).

To extend this univariate approach to multiple indicators per students, we rotate each indicator independently. Since the orthogonal contrasts are identical and in the same order for each indicator, not only is the variance preserved but also the covariance! Hence, there is no limit on the complexity of the student model. The only requirement is that all student models must be identical and have the same single parent.

Sufficient Statistic Formula for the Gaussian Density

A challenge with evaluation of the Gaussian density (Equation 5) is that, taking the naïve approach, the covariance dimension is the total number of observations in the model, potentially a very large number. One common way to speed up evaluation of the Gaussian likelihood function is to use the sufficient statistic formula. Suppose we have data of N independent observations of K -variate units. Let $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ be the model expected mean vector and covariance matrix, respectively. Let \mathbf{m} and \mathbf{S} be the mean vector and covariance matrix of the data, respectively. The sufficient statistic formula is,

$$-2 \log L(\text{data}|\boldsymbol{\theta}) = NK \log(2\pi) + N \log(|\boldsymbol{\Sigma}|) + (N-1)\text{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}) + N(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{m}). \quad (43)$$

The derivation of this formula is given in many textbooks and omitted here (e.g., Bollen, 1989). The advantage of this formula is that the maximum dimension of the covariance matrix is K regardless of the number of units N . However, this formula is only applicable when the units are independent and identical (including identical missingness patterns). Fortunately, Rampart dramatically improves the prospects for application of the sufficient statistic formula. Most of the lowest level units are rotated such that the expected mean is zero and with an identical expected covariance. These units, regardless of number, can be evaluated in constant time per evaluation.

Rampart and Definition Variables

To apply Rampart, the upper to lower level transition matrix must be exactly the same for all lower level units. Constant transition matrices, possibly with free parameters, pose no difficulty. However, no attempt is made to check whether this condition holds when the transition matrix is an `mxAlgebra` or contains square bracket populated values. If definition variables appear in the transition matrix then an attempt is made to group them by value. Another common use for definition variables is to specify zero variance regressions. Since these regressions do not affect the covariance, units that differ only in mean structure are Rampart rotated and evaluated using the Gaussian log density (Equation 5). That is, Σ^{-1} is computed once for all i and then we reuse Σ^{-1} for the quadratic form $(\boldsymbol{\mu} - \mathbf{y}_i)^T \Sigma^{-1} (\boldsymbol{\mu} - \mathbf{y}_i)$ over each i .

Latent Regression Parameter Recovery Simulation Study

To validate the accuracy of Rampart, a parameter recovery simulation study was conducted on a 3-level latent regression model (Figure 8). In addition, the first student indicator was set to missing with 20% probability. The simulation study focused on the correctness of Rampart, comparing Rampart with the simple

application of Equation 5. Elapsed time was not compared between evaluation approaches.

Two sets of true parameters (θ_1 and θ_2) were randomly chosen and data generated. Random numbers of students were assigned to each class and random numbers of teachers assigned per school. Parameter θ_1 was associated with 7 schools, 38 teachers, and 293 students. Parameter θ_2 was associated with 7 schools, 37 teachers, and 296 students. This was the smallest 3-level data set that we found empirically identified for most replications.

Two hundred Monte Carlo replications were run for each condition (Algorithm $\times \theta$). For each replication, data were generated from the true parameters. The number of units, which lower level units were linked to which upper level units, and data missingness patterns were identical for all replications. The model was optimized against these data to obtain ML estimate $\hat{\theta}$, using the true parameters θ_{true} as starting values. For R replications, Monte Carlo bias and variance are

$$MC_{bias} \equiv \left[R^{-1} \sum_{r=1}^R \hat{\theta}_r \right] - \theta_{true} \quad (44)$$

$$MC_{var} \equiv \text{Var}(\hat{\theta}). \quad (45)$$

After every replication, the information matrix was estimated by 2-iteration Richardson extrapolation of the central difference. The condition number of the information matrix is the maximum singular value divided by the minimum singular value and provides a rough gauge of the stability of a solution (Luenberger & Ye, 2008, p. 239). Replications were excluded from further analysis when the condition number of the information matrix was greater than 5 median absolute deviations from the median. For θ_1 , 190 trials converged. The maximum absolute difference in bias, variance, and deviance were 2.58×10^{-7} , 1.66×10^{-7} , and 1.18×10^{-8}

respectively. For θ_2 , 172 trials converged. The maximum absolute difference in bias, variance, and deviance were 3.21×10^{-7} , 8.49×10^{-7} , and 1.33×10^{-8} respectively.

Application

Many statistical software packages have been published. In order to compare the performance of Rampart with existing methods, we examined the most popular software packages that might permit ML fitting of a 5-level SEM. EQS, `xxM`, and `Stata/GLLAMM` almost offered the sought functionality, but not quite.

EQS (Bentler, 2006, Chapter 11) offers many-level SEM using a two-stage estimation (Chou, Bentler, & Pentz, 2000). However, this simpler analytical approach cannot offer the theoretically optimal properties of ML estimation. Hence, we gave no further attention to EQS.

Available only as a binary for the Microsoft Windows operating system, an R extension `xxM` offers many-level multilevel SEM (Mehta, 2013). During the development of Rampart, comparisons were made across different software to ensure that the same estimates were obtained for a variety of example models. Examples were drawn from `Mplus` (L. K. Muthén & B. O. Muthén, 2010, Chapter 9) and `xxM`. `OpenMx`, `xxM`, and `Mplus` agreed on all ML solutions except for one model. `xxM` had difficulty with `Mplus` example 9.23 “Three-level growth model with a continuous outcome and one covariate on each of the three levels.” Some of the parameters at level 2 seemed stuck at their starting values. We contacted author Paras Mehta about this defect in July 2016, but still have not received a resolution. Although `xxM` is free to download, it is not open-source. Therefore, we could not determine whether the difficulty was caused by an error in our model specification or a bug in `xxM`. Due to our doubt about the correctness of the implementation, we reluctantly excluded `xxM` from our performance comparison.

In theory, `Stata/GLLAMM` can evaluate many-level SEM models. However,

elapsed time for model evaluation is expected to be large. Adaptive quadrature is used to integrate out the random effects. This approach scales exponentially with the number of quadrature points: q^r for q quadrature points and r random effects (Rabe-Hesketh, Skrondal, & Pickles, 2004). `Stata/GLLAMM` was not included in our performance comparison given that its performance was unlikely to be competitive.

To demonstrate Rampart's performance we examined a large-scale study of child behavioral health with `OpenMx` 2.7, `Mplus` 7.3, `lme4` 1.1.12, and `nlme` 3.1.128. As part of an ongoing state contract between the University of Oklahoma Health Sciences Center and the Oklahoma Department of Human Services (OKDHS), each child in foster care receives a monthly screening across a broad spectrum of behavioral health outcomes. For children aged between 4 and 17 years, a primary component of this screening is the pediatric symptom checklist (PSC; Jellinek et al., 1988).

Between May of 2015 and December of 2016, the PSC was administered 14,436 times on 6,076 children in OKDHS custody by 1,280 caseworkers. Workers were spread over 83 county offices and managed by 34 district offices. The goal of this example is merely to decompose the sources of variation in the PSC total score due to each of these 5 nested levels: occasions, children, workers, counties, and districts. As a precursor to the 5-level model, a 3-level variance decomposition was specified in `lme4`, `nlme`, `Mplus`, and `OpenMx`. The 3-level example was chosen primarily for demonstration purposes. `Mplus` cannot estimate models with more than three levels, and `lme4` could not estimate any model on these data with more than three levels¹. So, it is not intended to be the best model for these data, but rather a preliminary model that will be extended later.

The `lme4` syntax for this 3-level model is

```
psc3 <- lmer(PSC_TOTAL ~ 1 +
```

¹A 4-level model in `lme4` was attempted, but after 30 minutes of running and using 24 GB of RAM on a 64-bit Windows machine, R crashed.

```
(1 | workerId/childId),
data=ds)
```

The `nlme` syntax is similar. Full parameter estimates are reported in the left half of Table 1. As can be seen in Table 1, there is broad agreement between the estimates from `lme4`, `nlme`, and `OpenMx`, but the upper-level variance estimates from `Mplus` appear to overestimate the child-level variance and underestimate the worker-level variance. The estimation time from `OpenMx` on this example is comparable to that of `lme4` and `nlme` but about ten times faster than `Mplus`. Because the parameter estimates in the 3-level model are quite similar to those in the 5-level model, we will only interpret those from the 5-level example.

The `nlme` syntax for a 5-level variance decomposition of the PSC is,

```
psc5 <- lme(PSC_TOTAL ~ 1,
  random= ~1 | districtId/countyId/workerId/childId,
  data=ds)
```

This model was run in `nlme` and `OpenMx`. The parameter estimates are shown in the right half of Table 1. Again, `nlme` and `OpenMx` obtain approximately the same parameter estimates. In this case, there is evidence of a slight performance advantage for `OpenMx`. The largest source of variation is across children. The within-child (and thus, across time) variation is captured by the residual variance. There also appears to be an important amount of variability at the worker level. The district level seems to have more variation than the county level.

The previous examples of 3- and 5-level models showed that Rampart in `OpenMx` has similar performance to dedicated mixed effect software that can only estimate univariate models. However, Rampart also applies equally well to multivariate outcomes. The original version of the PSC is known to have three subscales: attention problems (e.g., has trouble paying attention), internalizing

behavior problems (e.g., worries a lot), and externalizing behavior problems (e.g., teases others). Specific to this population, three additional items were added to assess any trauma symptoms (e.g., gets very upset when reminded of traumatic events). Thus, the PSC as administered in this sample has four subscales which relate to a common overall factor. Hence, a factor model was built at the lowest level, and then the variance of this factor was decomposed according to the same structure as the 5-level variance decomposition of the PSC sum score that was used previously.

Table 2 shows the results of estimating the 5-level factor variance decomposition of the PSC. The scale of the latent PSC variable was set by fixing the factor loading on the Attention subscale to 1.0. The Internalizing and Trauma subscales have somewhat lower factor loadings compared to the Attention and Externalizing loadings. The factor mean is now on the scale of the Attention score instead of the PSC sum score that was reported in Table 1. The overall pattern of variance across the levels is maintained across both examples. The largest component of variance is due to variation across children with substantial contributions from both the time (residual) and worker levels. As before, there is relatively little contribution of variance from the county and district levels, but there may be some evidence that there is more variation due to different districts than to different counties. The variation at the county and district levels may be due to differences in training practices, regional variation in the interpretation of PSC items, and differences in policies surrounding PSC administration particular to individual offices.

This 5-level factor variance decomposition is not readily possible in standard univariate mixed effects programs (e.g., `nlme` and `lme4`). Because of the number of levels, this model is not possible in `Mplus`. We acknowledge that 4-level models can be fit in a 3-level program when the lowest level is made with a wide data structure

(L. K. Muthén & B. O. Muthén, 2010, Chapter 9). However, this is computationally inefficient and generally works best for a univariate outcome. To run the univariate example using the PSC sum score as a 4-level model, we would need 21 variables at the lowest level because several children were observed 21 times. To run the multivariate example as a 4-level model in this way would require $4 \times 21 = 84$ variables on the lowest level. `Mplus` was about ten times slower than `OpenMx` on the univariate example. We infer based on this observation that `Mplus` would perform poorly on an 84-variate example, and still would not match the 5-level factor variance decomposition shown here. In summary, the 5-level univariate variance decomposition shows that Rampart can match the performance of dedicated mixed effect programs on large, heavily nested data. The 5-level factor variance decomposition shows that Rampart can then exceed these programs, and other multilevel SEM programs, by fitting more complicated variance structures for more outcomes across more levels than are possible with other software.

Discussion

Rampart, a novel approach to speed evaluation of nested multilevel structure, is introduced. Translation from an `lme4` model formula into an equivalent `OpenMx` model serves as a didactic example to introduce `OpenMx`-style model specification. A latent regression parameter recovery simulation study was conducted to demonstrate the correctness of the Rampart implementation. Among similar software examined, `EQS`, `xxM`, `Stata/GLLAMM`, `Mplus`, `lme4`, and `nlme`, none could surpass Rampart for factor analysis with 5- and more-level latent variance decomposition. Rampart works at the granularity of a set of 2 or more lower level models associated with a single upper level model. Missing data and definition variables can be accommodated by partitioning the data into identical missingness patterns and identical model specifications, then applying the algorithm to each partition.

The Rampart rotation requires that lower level units be associated with exactly one upper level unit. For many data sets, this is an onerous restriction. However, it is not clear how to relax this restriction. More research is needed to determine whether models for crossed data can be rotated in such a way as to increase evaluation efficiency or if some other transformation may be more fruitful.

With axis rotation firmly situated in continuous space, Rampart is limited to continuous indicators. It is not clear whether Rampart can be adapted to ordinal probit indicators (e.g., Mehta, Neale, & Flay, 2004) or generalized categorical response models. Item parceling is one way that ordinal indicators can currently be accommodated (e.g., Matsunaga, 2008). However, many researchers have cautioned that parceling adds nuisance variability (e.g., Nasser & Wisenbaker, 2003; Sterba & MacCallum, 2010).

While large sample inference can rely on the asymptotic results of large sample theory, much prior research on small sample inference is limited to the mixed model (i.e., univariate with no latent factors). It is unclear whether prior research on small sample inference generalizes to relational SEM. There could be complications because relational SEM models do not take into account the loss of degrees of freedom from constant coefficients (Patterson & Thompson, 1971). Most research to date on addressing this bias has focused on the mixed model where there is a clear delineation between constant and varying coefficients. Due to the efficiency of Rampart, it is now feasible to create relational SEM models that are nested many levels deep with some response observations at each level. It is not clear whether the distinction between constant and varying coefficients applies in the circumstance where a middle level coefficient is somewhat varying and somewhat constant. Inspired by Bayesian sampling methods, the use of a Wishart prior to correct bias in an ML point estimation context seems like a promising line of investigation (Chung, Gelman, Rabe-Hesketh, Liu, & Dorie, 2015). More research

is needed to establish whether this approach can profitably be applied to relational SEM or whether a different approach is more suitable.

The join operator in `OpenMx` supports one-to-many relationships but omits support for unlimited many-to-many relationships such as can be recorded in a relational database using a linking table. For example, a *classroom membership* table might contain foreign keys for both teachers and students. A linking table facilitates many-to-many relationships: a teacher with many students and a student with many teachers. Although there is no problem with linking tables from the standpoint of the join operator, it is problematic from a modeling point of view because the maximum number of teachers per student is not fixed. Definition variables are a simple way to parameterize models using data. Some kind of more intricate parameterization mechanism might be devised to connect an arbitrary number of units together in a default way without requiring the analyst to specify explicitly how, for instance, a 5 teacher and 6 student model ought to look.

A conspicuous missing feature in Rampart is the ability to estimate varying slopes (a.k.a. random slopes), latent interactions, or quadratic terms (e.g., Kelava, Nagengast, & Brandt, 2014). This is a glaring deficiency given the popularity of moderation models (e.g., Baron & D. A. Kenny, 1986). Fortunately, it seems likely that some approaches may mesh well with Rampart and permit estimation of quadratic effects without interfering with Rampart rotation (e.g., Klein & Moosbrugger, 2000).

While Rampart was developed in the context of ML point estimation, there is nothing in the algorithm specific to ML. Rampart could offer similar efficiency gains in the context of Bayesian SEM (e.g., B. Muthén & Asparouhov, 2012). We look toward the Stan community (Carpenter et al., 2016) for fruitful developments along these lines in the future.

Our software implementation is part of `OpenMx`. `OpenMx` is a free and

open-source software originally designed for structural equation modeling. **OpenMx** runs inside the R statistical programming environment (R Core Team, 2014) in all major computing environments. To help organize a community around the project, the **OpenMx** team maintains a web site <http://openmx.psyc.virginia.edu> that hosts binary and source versions of the software and several forms of tutorials and reference documentation (Neale et al., 2016). **OpenMx**, is now capable of estimating relational SEM models efficiently using Rampart rotation. Multivariate SEM models of large data sets, such as entire school districts, may have been considered intractable due to the required estimation time. With Rampart, these data sets may now be revisited and estimated with relative efficiency.

References

- Baron, R. M. & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, *51*(6), 1173.
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48.
doi:10.18637/jss.v067.i01
- Bentler, P. M. (2006). *EQS 6 structural equations program manual*. Encino, CA.
- Boker, S. M., Neale, M. C., Maes, H., Wilde, M., Spiegel, M., Brick, T. R., . . . Bates, T., et al. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika*, *76*(2), 306–317.
- Bollen, K. A. (1989). *Structural equations with latent variables*. New York: Wiley.
- Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., . . . Riddell, A. (2016). Stan: A probabilistic programming language. *Journal of Statistical Software*, *20*.
- Chou, C.-P., Bentler, P. M., & Pentz, M. A. (2000). A two-stage approach to multilevel structural equation models: Application to longitudinal data. In T. D. Little, K. U. Schnabel, & J. Baumert (Eds.), *Modeling longitudinal and multilevel data: Practical issues, applied approaches, and specific examples* (pp. 33–49). Mahwah, NJ: Lawrence Erlbaum Associates Publishers.
- Chung, Y., Gelman, A., Rabe-Hesketh, S., Liu, J., & Dorie, V. (2015). Weakly informative prior for point estimation of covariance matrices in hierarchical models. *Journal of Educational and Behavioral Statistics*, *40*(2), 136–157.
doi:10.3102/1076998615570945
- du Toit, S. H. & du Toit, M. (2008). Multilevel structural equation modeling. In J. de Leeuw & E. Meijer (Eds.), *Handbook of multilevel analysis* (pp. 435–478). Springer.

- Enders, C. K. & Bandalos, D. L. (2001). The relative performance of full information maximum likelihood estimation for missing data in structural equation models. *Structural Equation Modeling*, 8(3), 430–457.
- Gelman, A. (2005). Analysis of variance—why it is more important than ever. *The Annals of Statistics*, 33(1), 1–53.
- Gilbert, P. & Varadhan, R. (2012). *numDeriv: Accurate numerical derivatives*. R package version 2012.9-1. Retrieved from <http://CRAN.R-project.org/package=numDeriv>
- Goldstein, H. & McDonald, R. P. (1988). A general model for the analysis of multilevel data. *Psychometrika*, 53(4), 455–467.
- Jellinek, M. S., Murphy, J. M., Robinson, J., Feins, A., Lamb, S., & Fenton, T. (1988, February). Pediatric symptom checklist: Screening school-age children for psychosocial dysfunction. *The Journal of Pediatrics*, 112(2), 201–209. doi:10.1016/s0022-3476(88)80056-8
- Kelava, A., Nagengast, B., & Brandt, H. (2014). A nonlinear structural equation mixture modeling approach for nonnormally distributed latent predictor variables. *Structural Equation Modeling: A Multidisciplinary Journal*, 21(3), 468–481. doi:10.1080/10705511.2014.915379
- Klein, A. & Moosbrugger, H. (2000). Maximum likelihood estimation of latent interaction effects with the lms method. *Psychometrika*, 65(4), 457–474.
- Krull, J. L. & MacKinnon, D. P. (2001). Multilevel modeling of individual and group level mediated effects. *Multivariate Behavioral Research*, 36(2), 249–277.
- Lee, S.-Y. & Poon, W.-Y. (1998). Analysis of two-level structural equation models via EM type algorithms. *Statistica Sinica*, 749–766.
- Luenberger, D. G. & Ye, Y. (2008). *Linear and nonlinear programming*. Springer-Verlag.
- Maier, D. (1983). *The theory of relational databases*. Computer Science Press.

- Matsunaga, M. (2008). Item parceling in structural equation modeling: A primer. *Communication Methods and Measures*, 2(4), 260–293.
- 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.
- McArdle, J. J. & McDonald, R. P. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology*, 37(2), 234–251.
- McDonald, R. P. (1993). A general model for two-level data with responses missing at random. *Psychometrika*, 58(4), 575–585.
- Mehta, P. D. (2013). xxM: N-Level structural equation modeling [Computer software].
- Mehta, P. D. & Neale, M. C. (2005). People are variables too: Multilevel structural equations modeling. *Psychological Methods*, 10(3), 259.
- Mehta, P. D., Neale, M. C., & Flay, B. R. (2004). Squeezing interval change from ordinal panel data: Latent growth curves with ordinal outcomes. *Psychological methods*, 9(3), 301.
- Muthén, B. (1997). Latent variable modeling of longitudinal and multilevel data. *Sociological methodology*, 27(1), 453–480.
- Muthén, B. O. (1994). Multilevel covariance structure analysis. *Sociological Methods & Research*, 22(3), 376–398.
- Muthén, B. & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological methods*, 17(3), 313.
- Muthén, L. K. & Muthén, B. O. (2010). *Mplus user's guide: Statistical analysis with latent variables: User's guide*. Muthén & Muthén.

- Nasser, F. & Wisenbaker, J. (2003). A Monte Carlo study investigating the impact of item parceling on measures of fit in confirmatory factor analysis. *Educational and Psychological Measurement*, *63*(5), 729–757.
- Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R., . . . Boker, S. M. (2016). **OpenMx** 2.0: Extended structural equation and statistical modeling. *Psychometrika*, *81*(2), 535–549.
doi:10.1007/s11336-014-9435-8
- Patterson, H. D. & Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika*, *58*(3), 545–554.
- Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & R Core Team. (2016). *nlme: linear and nonlinear mixed effects models*. R package version 3.1-124.
Retrieved from <http://CRAN.R-project.org/package=nlme>
- Pritikin, J. N. (2016). *Unbelievably fast estimation of nested multilevel structural equation models* (Doctoral dissertation, University of Virginia, Charlottesville, VA).
- R Core Team. (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from <http://www.R-project.org>
- Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2004). Generalized multilevel structural equation modeling. *Psychometrika*, *69*(2), 167–190.
- Raudenbush, S. W. & Sampson, R. (1999). Assessing direct and indirect effects in multilevel designs with latent variables. *Sociological Methods & Research*, *28*(2), 123–153.
- Richardson, L. F. (1911). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society*

of London. Series A, Containing Papers of a Mathematical or Physical Character, 210, 307–357.

- Schaalje, G. B., McBride, J. B., & Fellingham, G. W. (2002). Adequacy of approximations to distributions of test statistics in complex mixed linear models. *Journal of Agricultural, Biological, and Environmental Statistics*, 7(4), 512–524.
- Sterba, S. K. & MacCallum, R. C. (2010). Variability in parameter estimates and model fit across repeated allocations of items to parcels. *Multivariate Behavioral Research*, 45(2), 322–358.
- Tarjan, R. E. (1976). Edge-disjoint spanning trees and depth-first search. *Acta Informatica*, 6(2), 171–185.
- Westfall, J. & Yarkoni, T. (2016). Statistically controlling for confounding constructs is harder than you think. *PLoS ONE*, 11(3). doi:10.1371/journal.pone.0152719
- Wilkinson, G. N. & Rogers, C. E. (1973). Symbolic description of factorial models for analysis of variance. *Applied Statistics*, 392–399.

Table 1

Variance decomposition of the Pediatric Symptom Checklist in common mixed effects programs. Code to take advantage of shared memory parallel processing was disabled reduce the variance in estimation time.

	3-level				5-level	
	lme4	nlme	Mplus	OpenMx	nlme	OpenMx
District Variance	-	-	-	-	0.913	0.874
County Variance	-	-	-	-	0.096	0.122
Worker Variance	9.235	9.235	1.063	9.231	8.247	8.202
Child Variance	32.255	32.255	39.914	32.234	32.257	32.239
Residual Variance	17.946	17.946	17.922	17.960	17.939	17.955
Intercept	10.103	10.103	10.003	10.116	10.115	10.125
Estimation Time (seconds)	11.14	2.21	53.40	4.66	15.68	12.28

Table 2

Estimates from a 5-level variance decomposition with a factor model of the Pediatric Symptom Checklist at the lowest level. Standard errors (SEs) are derived from an information matrix approximated by finite differences with Richardson extrapolation (Gilbert & Varadhan, 2012; Richardson, 1911). No attempt was made to correct the SEs for the structure of the data (e.g., Schaalje, McBride, & Fellingham, 2002).

Matrix	To	From	Estimate	SE
A	Attention	PSC	1.000	-
A	Internalizing	PSC	0.585	0.005
A	Externalizing	PSC	1.224	0.007
A	Trauma	PSC	0.303	0.003
S	Attention	Attention	3.358	0.057
S	Internalizing	Internalizing	3.076	0.042
S	Externalizing	Externalizing	4.284	0.077
S	Trauma	Trauma	1.558	0.020
S	PSC	PSC	0.558	0.030
M	1	PSC	3.254	0.069
childModel.S	childVar	childVar	3.315	0.092
workerModel.S	workerVar	workerVar	0.818	0.082
countyModel.S	countyVar	countyVar	0.005	0.046
districtModel.S	districtVar	districtVar	0.085	0.047

	Jane	Joe	
Teachers			upper
Students	Noah Sophia Liam Emma	Jacob Olivia Mason Isabella	lower

Figure 1. Students nested within teachers. For example, Noah is Jane's student and Jacob is Joe's student. There is a one-to-many relationship between teachers and students. A different model would be needed to accommodate students that spent some proportion of their time with each teacher.

Employee	Dept
Harry	Sales
Sally	Finance
George	Finance
Harriet	Sales

Dept	Manager
Sales	George
Finance	Harriet
Production	Charles

Employee \bowtie (Dept) Manager

Employee	Dept	Manager
Harry	Sales	George
Sally	Finance	Harriet
George	Finance	Harriet
Harriet	Sales	George

Figure 2. An employee table (a.k.a relation or data frame) and manager table are given (upper tables). The employee and manager tables are joined by department (lower table). For example, observe the first Employee, Harry, is in the Sales department. The manager of the Sales department is George. Therefore, the first line of the join table is Harry, Sales, and George.

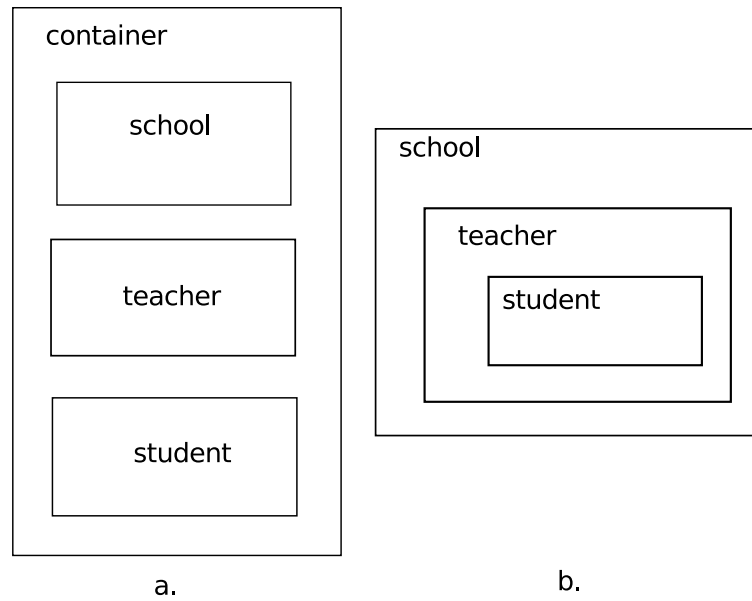


Figure 3. Two equivalent model specifications for students nested within teachers nested within schools. Each rectangle corresponds to an `mxModel`. An early prototype used organization (a) to specify nested multilevel models. We finalized on (b) for `mxPath` specified models. Scheme (b) may seem backwards, but it offers the advantage that each submodel is also a valid model. This is because, for strictly nested data, outer models cannot depend on inner models. For example, a school cannot depend on a teacher and a teacher cannot depend on a student. This structure is only required for `mxPath` specified models. No particular model nesting is required for `mxMatrix` specified models.

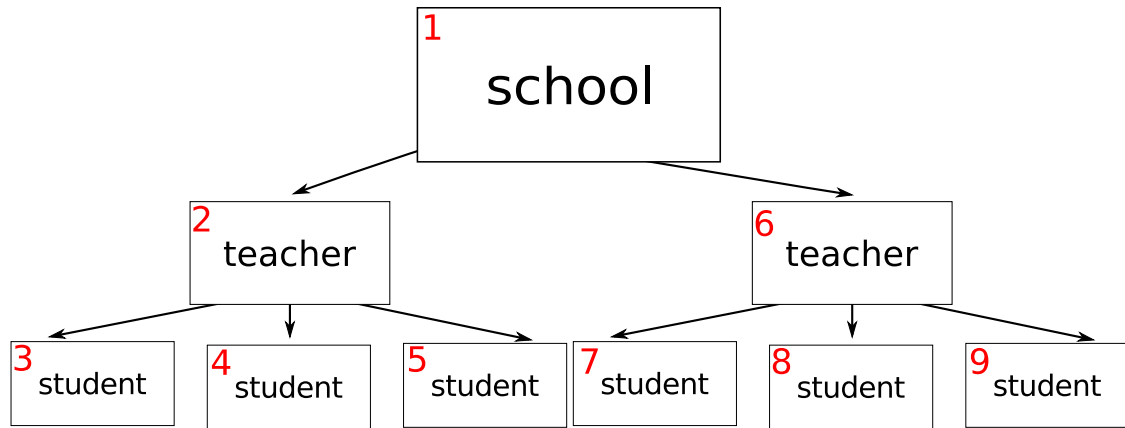


Figure 4. Topological sort is accomplished by depth-first search (Tarjan, 1976) in the opposite direction of the arrows starting from each of the lowest level units (students in this example). Units are assigned a location (the number in the upper left) as soon as all the units that they depend upon are assigned a location. This algorithm is linear in time with the number of units.

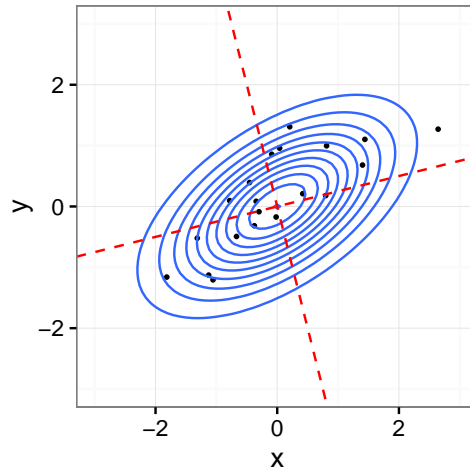


Figure 5. Observations (represented by points) in a Gaussian density. The likelihood of these points is unaffected by axis rotation. For example, the axis could be rotated to the tilted dashed lines without affecting the likelihood.

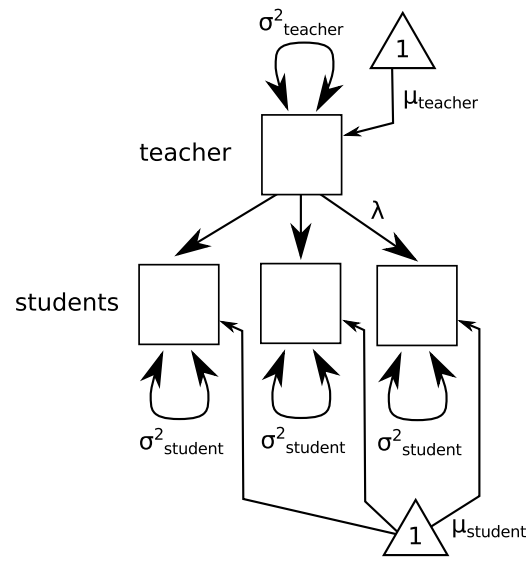


Figure 6. A simple multilevel model with 5 parameters: $\sigma^2_{teacher}$, $\mu_{teacher}$, $\sigma^2_{student}$, $\mu_{student}$, and λ . A teacher's three students have exactly the same model implied distribution.

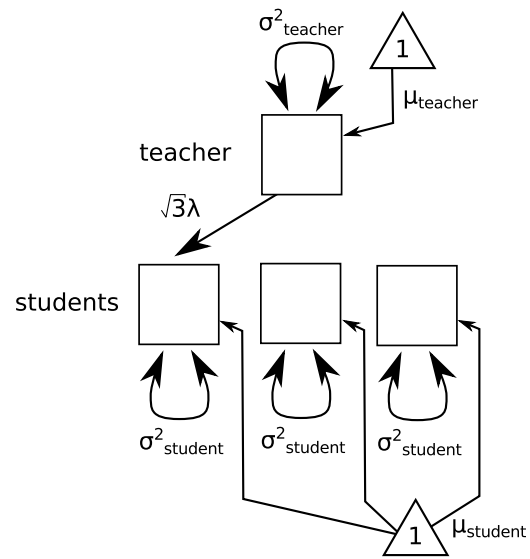


Figure 7. Figure 6 after Rampart rotation is applied to unlink all but one student from the teacher. Note that the student data (not shown) requires a corresponding rotation to preserve the value of the likelihood.

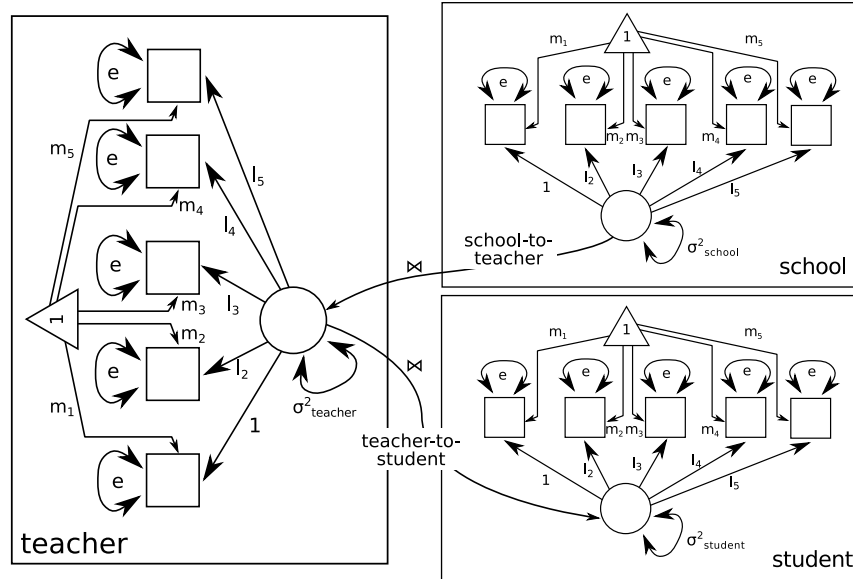


Figure 8. A 3-level latent regression model. All levels use an identical 5 indicator factor model with the loading to the first indicator fixed to 1.0, freely estimated means, free factor variance, and homogeneous error variance. Regressions are estimated from school to teacher and from teacher to school. There are 11 parameters per level and 2 between level regressions for a total of 35 parameters. Indicator error variance does not need to be homogeneous. More complex error structures are possible, but were not included in this study. Manifest indicators are not shared by levels, but are unique to their level. For example, teacher indicators might include *level of education* and *years of service*.

Appendix

lmer sleepstudy example

```

1 library(lme4)
2 fm1 <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy, REML=FALSE)
3
4 library(OpenMx)
5
6 if (is.factor(sleepstudy$Subject)) {
7   subjnum <- unclass(sleepstudy$Subject)
8   sleepstudy$Subject <- as.integer(levels(sleepstudy$Subject)[ subjnum ])
9 }
10
11 bySubj <- mxModel(
12   model="bySubj", type="RAM",
13   latentVars=c("slope", "intercept"),
14   mxData(data.frame(Subject=unique(sleepstudy$Subject)),
15         type="raw", primaryKey = "Subject"),
16   mxPath(from=c("intercept", "slope"), arrows=2, values=1),
17   mxPath(from="intercept", to="slope", arrows=2, values=.25, labels="cov1"))
18
19 sleepModel <- mxModel(
20   model="sleep", type="RAM", bySubj,
21   manifestVars="Reaction", latentVars = "Days",
22   mxData(sleepstudy, type="raw", sort=FALSE),
23   mxPath(from="one", to="Reaction", arrows=1, free=TRUE),
24   mxPath(from="one", to="Days", arrows=1, free=FALSE, labels="data.Days"),
25   mxPath(from="Days", to="Reaction", arrows=1, free=TRUE),
26   mxPath(from="Reaction", arrows=2, values=1),
27   mxPath(paste0('bySubj.', c('intercept', 'slope')),
28         'Reaction', arrows=1, free=FALSE, values=c(1,NA),
29         labels=c(NA, "data.Days"), joinKey="Subject"))
30
31 m1 <- mxRun(sleepModel)
32
33 omxCheckCloseEnough(logLik(m1), logLik(fm1), 1e-6)

```