EFFICIENT LIKELIHOOD ESTIMATION OF GENERALIZED STRUCTURAL EQUATION MODELS WITH A MIX OF NORMAL AND NONNORMAL RESPONSES

Nicholas J. Rockwood

loma linda university

This is a post-peer-review, pre-copyedit version of an article accepted for publication in *Psychometrika.* (DOI pending)

Correspondence should be sent to Nicholas J. Rockwood, Division of Interdisciplinary Studies, School of Behavioral Health, Loma Linda University, 11065 Campus St., Loma Linda, CA 92350. Email: nrockwood@llu.edu
Website: njrockwood.com

EFFICIENT LIKELIHOOD ESTIMATION OF GENERALIZED STRUCTURAL EQUATION MODELS WITH A MIX OF NORMAL AND NONNORMAL RESPONSES

Abstract

A maximum likelihood estimation routine is presented for a generalized structural equation model that permits a combination of response variables from various distributions (e.g., normal, Poisson, binomial, etc.). The likelihood function does not have a closed-form solution and so must be numerically approximated, which can be computationally demanding for models with several latent variables. However, the dimension of numerical integration can be reduced if one or more of the latent variables do not directly affect any nonnormal endogenous variables. The method is demonstrated using an empirical example and the full estimation details, including first-order derivatives of the likelihood function, are provided.

Key words: Structural equation modeling, latent variable modeling, generalized linear models, maximum likelihood estimation

1. Introduction

Structural equation modeling (SEM) has become one of the most widely used multivariate statistical modeling approaches within the social and behavioral sciences. Within the traditional SEM framework, the latent variables and the observed variables, conditional on the latent variables, are both assumed to be multivariate normally distributed. Consequently, the marginal distribution of the data is also multivariate normal, allowing for computationally efficient maximum likelihood (ML) estimation of the model parameters. Such models are implemented in an ever-growing suite of commercial software, such as LISREL (Jöreskog & Sörbom, 1996), EQS (Bentler, 2004), Mplus (L. K. Muthén & Muthén, 2017), and Stata (StataCorp, 2019a), as well as open source software, such as the lavaan (Rosseel, 2012), sem (Fox, Nie, & Byrnes, 2017), and OpenMx (Neale et al., 2016) packages within the R computing environment (R Core Team, 2019).

However, behavioral research often involves a variety of nonnormal response variables, including categorical and count outcomes. When the only type of categorical responses are ordinal (including dichotomous), the SEM is typically formulated assuming that there are multivariate normal latent responses underlying the discrete observations. Unfortunately, ML estimation of the model parameters assuming latent responses requires an intractable multidimensional integral, where the dimension of integration is equal to the number of endogenous categorical variables (each with their own latent response). The intractable integral can be avoided by using diagonally weighted least squares (DWLS; B. Muthén, 1984), a limited information approach, to estimate the model parameters.

Alternatively, several methodologists have advocated for a generalized latent variable modeling approach, where the conditional distribution of the data given the latent variables is modeled using a generalized linear model (GLM) formulation (Bartholomew & Knott, 1999; Moustaki & Knott, 2000; Rabe-Hesketh, Skrondal, & Pickles, 2004; L. K. Muthén & Muthén, 2017). This approach can be used for ordered and unordered categorical variables, count variables, and others. Because an underlying latent variable is not necessarily assumed for the observed variables, the DWLS estimation developed for ordered categorical variables is typically not applicable. Instead, ML estimation is most commonly used for such models, where the likelihood function contains a multidimensional integral with the dimension equal to the number of latent variables. This integral cannot be computed in closed-form and so must be numerically approximated, which can be computationally intensive for high dimensional problems.

Two of the most flexible and widely used general latent variable modeling frameworks are those implemented within the Stata package gllamm (Rabe-Hesketh et al., 2004) and Mplus (L. K. Muthén & Muthén, 2017). Although not exactly equivalent, there is large overlap between each of the frameworks, particularly in regard to the model discussed here. Both programs allow for ML estimation of generalized SEMs with a wide range of potential outcome distributions, including normal, binomial, Poisson (and extensions that account for overdispersion and zero-inflation), and others.

In addition to publishing a wide array of articles and books discussing the general modeling framework and interesting applications (e.g., Skrondal & Rabe-Hesketh, 2003; Rabe-Hesketh et al., 2004; Skrondal & Rabe-Hesketh, 2004; Zheng & Rabe-Hesketh, 2007), the developers of gllamm have also provided many of the technical details related to ML parameter estimation of generalized latent variable models as implemented in gllamm (Rabe-Hesketh, Skrondal, & Pickles, 2002, 2005; Skrondal & Rabe-Hesketh, 2009). These documents have likely contributed to the implementation of such methods into other commercial and open-source software programs. Indeed, Stata has since released a gsem function (StataCorp, 2019b) for generalized SEM that relies upon many of the technical details provide by the gllamm developers and colleagues. The developers of Mplus have also published several papers documenting recent advancements in Mplus and applications of such advancements (e.g., B. O. Muthén, 2002; Asparouhov, Masyn, & Muthen, 2006; B. Muthén & Asparouhov, 2008, 2011; B. O. Muthén, Muthén, & Asparouhov, 2017). However, these documents do not always contain the technical details necessary for direct software implementation by others. Two such details relevant to Mplus' computational advantage when estimating generalized SEMs are discussed here.

In gllamm and Stata's gsem function, the dimension of numerical integration required for ML estimation of generalized SEMs (and traditional SEMs) is equal to the number of latent variables in the model, regardless of the specific model structure and the distributions of the endogenous variables. In Mplus, on the other hand, the dimension of numerical integration required is equal to only those latent variables that directly affect nonnormal responses. Thus, for models that include a mix of normal and nonnormal responses, the estimation method in Mplus may be less computationally demanding than that in gllamm and gsem. Another notable advantage of Mplus (and the gsem function) relative to gllamm is the use of analytic derivatives rather than numerical derivatives, which can be computationally costly for models with many freely estimated parameters. Unfortunately, neither the dimension reduction technique nor the analytic derivatives for the generalized SEM implemented within Mplus have been publicly documented. This has likely stunted the implementation of such methods into other software packages.

The purpose of this paper is to provide all of the technical details necessary for efficient ML estimation of generalized SEMs. In Section 2, a generalized SEM framework incorporating both normal and nonnormal response variables is presented. An ML estimation routine for the model is developed within Section 3. For certain models it is shown that, by reformulating the likelihood function, a subset of the latent factors can be integrated out of the likelihood function analytically. This method, which appears to be comparable to the method implemented within Mplus, is an adaptation of the numerical integration dimension reduction technique proposed by du Toit and Cudeck (2009) in the context of nonlinear mixed effects models. Two example models are then fit within Section 4, where estimates obtained using Mplus, gsem, and the method introduced here are compared. The paper concludes with a discussion in Section 5. To aid in software implementation of such methods, the derivatives required for approximating and maximizing the log-likelihood function are presented in Appendices A and B, respectively. Finally, steps for reparameterizing the model to take full advantage of the dimension reduction technique are detailed in Appendix C.

2. Model

The generalized SEM that will be the focus here can be formulated by first starting with the standard SEM for conditionally normal endogenous variables. Let $\mathbf{y}_i$ denote a p-dimensional vector of observed continuous variables for individual i $(i = 1, ..., N)$ that are modeled as:

$$\mathbf{y}_i = \boldsymbol{\nu}_y + \boldsymbol{\Lambda}_y \boldsymbol{\eta}_i + \mathbf{K}_y \mathbf{x}_i + \boldsymbol{\varepsilon}_i, \tag{1}$$

where $\boldsymbol{\nu}_y$ is a p-dimensional vector of intercepts, $\boldsymbol{\Lambda}_y$ is a $p \times m$ factor loading matrix relating the m-dimensional vector of latent variables $\boldsymbol{\eta}_i$ to the observed dependent variables, $\mathbf{K}_y$ is a $p \times q$ matrix relating the q observed covariates $\mathbf{x}_i$ to the observed dependent variables, and $\boldsymbol{\varepsilon}_i$ is a p-dimensional vector of random errors where it is assumed that $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta})$. The latent variables $\boldsymbol{\eta}_i$ are modeled via the structural model:

$$\boldsymbol{\eta}_i = \boldsymbol{\alpha} + \mathbf{B}\boldsymbol{\eta}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\zeta}_i, \tag{2}$$

where $\boldsymbol{\alpha}$ is an m-dimensional vector of factor means or intercepts, $\mathbf{B}$ is an $m \times m$ matrix relating the latent variables to one another, $\boldsymbol{\Gamma}$ is an $m \times q$ matrix relating the observed covariates to the latent variables, and $\boldsymbol{\zeta}_i$ an an m-dimensional vector of random latent variable disturbances, such that $\boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ and $\mathrm{Cov}(\boldsymbol{\eta}_i, \boldsymbol{\varepsilon}_i) = \mathbf{0}$.

However, in addition to the p continuous and (conditionally) normally distributed observed variables $\mathbf{y}_i$, suppose there are k nonnormal (and potentially categorical) variables $\mathbf{u}_i = (u_{i1}, u_{i2}, ..., u_{ik})'$. Let $f_j(u_{ij}|\boldsymbol{\xi}_i)$ denote any legitimate probability mass or density function for u_{ij} $(j = 1, ..., k)$, which depends on the r-dimensional vector

$$\boldsymbol{\xi}_i = \boldsymbol{\nu}_u + \boldsymbol{\Lambda}_u \boldsymbol{\eta}_i + \mathbf{K}_u \mathbf{x}_i, \tag{3}$$

where $\boldsymbol{\nu}_u$ is $r \times 1$, $\boldsymbol{\Lambda}_u$ is $r \times m$, and $\mathbf{K}_u$ is $r \times q$. Since $\boldsymbol{\xi}_i$ depends on the random vector $\boldsymbol{\eta}_i$, so too does u_{ij}. Hence the u subscripts for $\boldsymbol{\nu}_u$, $\boldsymbol{\Lambda}_u$, and $\mathbf{K}_u$.

A generalized linear model can be formulated for the conditional distribution of u_{ij} by specifying the probability distribution of u_{ij} as a member of the exponential family and defining $g_j(E(u_{ij}|\boldsymbol{\xi}_i)) = \xi_{ij}$, where $g_j(\cdot)$ is an appropriate link function and ξ_{ij} is the jth element of $\boldsymbol{\xi}_i$. For example, if u_{ij} is dichotomous, the response may be modeled as a Bernoulli random variable with mean

$$E(u_{ij}|\boldsymbol{\xi}_i) = g_j^{-1}(\xi_{ij}) = \frac{\exp(\xi_{ij})}{1 + \exp(\xi_{ij})}, \tag{4}$$

resulting in a logistic regression. A probit regression could instead be specified by changing the link function $g_j(\cdot)$ to the probit link. Similarly, a Poisson regression model can be specified for a count outcome u_{ij} by specifying $f_j(u_{ij}|\boldsymbol{\xi}_i)$ as Poisson with mean $E(u_{ij}|\boldsymbol{\xi}_i) = \exp(\xi_{ij})$.

To complete the general model specification, it is assumed that the nonnormal responses $\mathbf{u}_i$ are independent of one another and $\mathbf{y}_i$ conditional on the latent variables $\boldsymbol{\eta}_i$,

$$f(\mathbf{y}_i, \mathbf{u}_i|\boldsymbol{\eta}_i) = f(\mathbf{y}_i|\boldsymbol{\eta}_i)f(\mathbf{u}_i|\boldsymbol{\eta}_i) = f(\mathbf{y}_i|\boldsymbol{\eta}_i)\prod_j f_j(u_{ij}|\boldsymbol{\eta}_i), \tag{5}$$

where $f_j(u_{ij}|\boldsymbol{\eta}_i)$ is used in place of $f_j(u_{ij}|\boldsymbol{\xi}_i)$ to highlight that u_{ij} is dependent on the latent variables $\boldsymbol{\eta}_i$ (by way of one or more elements of $\boldsymbol{\xi}_i$), which can be used to model dependency between $\mathbf{u}_i$ and $\mathbf{y}_i$.

2.1. Modeling more complex response types

The model for u_{ij} is actually more flexible than a typical generalized linear (latent) model specification in two important ways. First, there is no requirement that the distribution of u_{ij} must be a member of the exponential family. For example, if u_{ij} is censored continuous data, a Tobit regression model could be specified for $f(u_{ij}|\boldsymbol{\xi}_i)$. Alternatively, if u_{ij} is a multiple-choice test item, a three parameter item response theory (IRT) model could be specified. Second, as can be noted by the different dimensions of $\mathbf{u}_i$ and $\boldsymbol{\xi}_i$, the model allows for more complex processes to be modeled because the distribution of u_{ij} may depend on more than one element of $\boldsymbol{\xi}_i$. Two common scenarios in which it may be useful to allow u_{ij} to depend on multiple elements of $\boldsymbol{\xi}_i$ include the analysis of polytomous item response data and zero-inflated count data.

For example, suppose that u_{ij} is a zero-inflated count variable (i.e., the variable has an excess of zeros than would be expected under a typical count distribution such as Poisson or negative Binomial). A zero-inflated count model theorizes that a zero response may arise through one of two distinct processes. The first process corresponds to whether there is a chance of a non-zero response, and the second process corresponds to the count response given that it may be non-zero. A common example of a zero-inflated count variable is the number of alcoholic beverages consumed by an individual over a particular weekend. The zero/non-zero process would correspond to whether an individual consumes alcohol. All individuals who do not consume alcohol would have a zero probability of drinking that particular weekend, whereas alcohol consumers may drink one or more alcohol beverages. The count process would then correspond to the number of beverages consumed by those who actually drink alcohol. But just because an individual drinks alcohol, meaning they have non-zero probability of drinking one or more beverages, does not necessarily imply that their responses is non-zero. They may be an alcohol consumer, but abstained that particular weekend. Thus, a zero response may arise because either the individual is not a drinker or the individual is a drinker, but did not happen to drink that particular weekend.

Models for zero-inflated count data are used to better understand heterogeneity within each of these two distinct processes, and different variables may be predictors of each. A zero-inflated Poisson (ZIP; Lambert, 1992) model uses either a logit or probit regression for the zero/non-zero process and a Poisson regression for the count process. Alternatively, a negative Binomial distribution can be specified for the counts to allow for overdispersion. Both of these models are easy to formulate within the generalized SEM presented here. The ZIP model can be specified for u_{ij} as:

$$f(u_{ij} = z|\boldsymbol{\xi}) = \begin{cases} \xi_{1i} + (1 - \xi_{1i})\exp(-\xi_{2i}) & \text{if } z = 0 \\ (1 - \xi_{1i})\frac{\xi_{2i}^{u_{ij}}\exp(-\xi_{2i})}{u_{ij}!} & \text{if } z > 0 \end{cases}$$

where ξ_{1i} corresponds to the zero/non-zero process and ξ_{2i} corresponds to the count process. Because $\boldsymbol{\xi}_i$ is a function of $\boldsymbol{\eta}_i$, both zero-inflated IRT (Wang, 2010) and growth models (H. Liu, 2007) may be specified. The analysis of zero-inflated count data is just one of many potential examples that demonstrate the flexibility of the modeling framework, which is similar to the composite link and exploded likelihood formulation described by Rabe-Hesketh and Skrondal (2007). Sophisticated software routines could even permit user-defined probability distributions for $f(u_{ij}|\boldsymbol{\xi}_i)$.

3. Parameter estimation

ML estimation for the generalized SEM is discussed in this section. After presenting the likelihood function, the model is slightly reparameterized to distinguish between latent variables that only directly affect continuous, normally distributed endogenous variables and latent variables that affect one or more nonnormal endogenous variables. It is then demonstrated that the former of these latent variables can be analytically integrated out of the likelihood function to reduce the computational complexity of the model. Finally, approximating the intractable integral within the likelihood function and first order-derivatives of the log-likelihood function using Gaussian quadrature is detailed.

3.1. Likelihood function

Letting $\boldsymbol{\vartheta}$ denote all of the free and non-redundant model parameters, the likelihood function for individual i follows as:

$$\mathcal{L}_i(\boldsymbol{\vartheta}) = \int f(\mathbf{u}_i|\boldsymbol{\eta}_i) f(\mathbf{y}_i|\boldsymbol{\eta}_i) f(\boldsymbol{\eta}_i) d\boldsymbol{\eta}_i, \tag{6}$$

where

$$\mathbf{y}_i|\boldsymbol{\eta}_i \sim \mathcal{N}(\boldsymbol{\nu}_y + \boldsymbol{\Lambda}_y \boldsymbol{\eta}_i + \mathbf{K}_y \mathbf{x}_i, \boldsymbol{\Theta}), \tag{7}$$

$$\boldsymbol{\eta}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\eta}}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}), \tag{8}$$

$$\boldsymbol{\mu}_{\boldsymbol{\eta}} = (\mathbf{I} - \mathbf{B})^{-1}(\boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i), \quad \text{and} \quad \boldsymbol{\Sigma}_{\boldsymbol{\eta}} = (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\Psi}(\mathbf{I} - \mathbf{B})^{-1'}. \tag{9}$$

As stated previously, there is much flexibility in the specification of $f(\mathbf{u}_i|\boldsymbol{\eta}_i) = \prod_j f_j(u_{ij}|\boldsymbol{\eta}_i)$.

If there are no $\mathbf{u}_i$ variables within the model (i.e., $k = 0$), then the model reduces to a typical SEM and the m-dimensional integral in Equation 6 can be computed in closed-form. But, due to the nonlinearity of $\boldsymbol{\eta}_i$ within $f(\mathbf{u}_i|\boldsymbol{\eta}_i)$, the likelihood function cannot generally be computed in closed-form when $k > 0$. Instead, for likelihood inference the integral must be numerically approximated. Adaptive Gauss-Hermite quadrature is often viewed as the gold standard for numerically approximating such integrals (Pinheiro & Bates, 1995; Rabe-Hesketh et al., 2002), but the computational burden of quadrature based numerical integration increases exponentially as a function of the dimension of numerical integration. In Equation 6, the dimension of integration is equal to the number of latent variables, m. Thus, numerically approximating Equation 6 directly via quadrature based methods in not practical for models involving many latent variables.

However, when some of the latent variables do not influence the categorical variables $\mathbf{u}_i$ directly, the dimension of numerical integration can be reduced. That is, only latent variables that have direct effects on elements of $\mathbf{u}_i$ need to be numerically integrated. These latent variables can be determined using a simple reparameterization of the model.

3.2. Reformulating the model

Direct effects of the latent variables $\boldsymbol{\eta}_i$ on the observed variables $\mathbf{u}_i$ are modeled via $\boldsymbol{\Lambda}_u$ in Equation 3. Suppose $\boldsymbol{\eta}_i$ can be partitioned into two subsets, $\boldsymbol{\eta}_{1i}$ $(m_1 \times 1)$ and $\boldsymbol{\eta}_{2i}$ $(m_2 \times 1)$, such that only elements of $\boldsymbol{\eta}_{1i}$ influence one or more elements of $\mathbf{u}_j$ directly. It is assumed that the elements of $\boldsymbol{\eta}_i$ are ordered such that the combination of Equations 1 and 3 can be reformulated as:

$$\begin{pmatrix} \boldsymbol{\xi}_i \\ \mathbf{y}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{\nu}_u \\ \boldsymbol{\nu}_y \end{pmatrix} + \begin{pmatrix} \boldsymbol{\Lambda}_{u1} & \mathbf{0} \\ \boldsymbol{\Lambda}_{y1} & \boldsymbol{\Lambda}_{y2} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}_{1i} \\ \boldsymbol{\eta}_{2i} \end{pmatrix} + \begin{pmatrix} \mathbf{K}_u \\ \mathbf{K}_y \end{pmatrix} \mathbf{x}_i + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\varepsilon}_i \end{pmatrix}. \tag{10}$$

It is now explicit that $\boldsymbol{\xi}_i$ (and, therefore, $\mathbf{u}_i$) is not modeled as a direct function of $\boldsymbol{\eta}_{2i}$. In contrast, $\boldsymbol{\eta}_{1i}$ may influence both sets of observed variables $\mathbf{u}_i$ and $\mathbf{y}_i$, via the matrices $\boldsymbol{\Lambda}_{u1}$ $(r \times m_1)$ and $\boldsymbol{\Lambda}_{y1}$ $(p \times m_1)$, respectively. When the model can be written in this form, such that $m_2 > 0$, the dimension of numerical integration required for likelihood estimation can be reduced from m to m_1.

Specifically, the likelihood function defined by Equation 6 can be equivalently written as

$$\mathcal{L}_i(\boldsymbol{\vartheta}) = \int\int f(\mathbf{u}_i|\boldsymbol{\eta}_{1i}, \boldsymbol{\eta}_{2i}) f(\mathbf{y}_i|\boldsymbol{\eta}_{1i}, \boldsymbol{\eta}_{2i}) f(\boldsymbol{\eta}_{2i}|\boldsymbol{\eta}_{1i}) f(\boldsymbol{\eta}_{1i}) d\boldsymbol{\eta}_{2i} d\boldsymbol{\eta}_{1i}, \tag{11}$$

where, as before, the dimension of integration is equal to $m = m_1 + m_2$. However, by formulating the model as that defined by Equation 10, it is clear that the observed variables $\mathbf{u}_i$ are independent from $\boldsymbol{\eta}_{2i}$, conditional on $\boldsymbol{\eta}_{1i}$. Therefore, the dependence on $\boldsymbol{\eta}_{2i}$ within $f(\mathbf{u}_i|\boldsymbol{\eta}_{1i}, \boldsymbol{\eta}_{2i})$ can be removed, allowing the function to be pulled out of the interior integral:

$$\mathcal{L}_i(\boldsymbol{\vartheta}) = \int f(\mathbf{u}_i|\boldsymbol{\eta}_{1i}) \underbrace{\left[\int f(\mathbf{y}_i|\boldsymbol{\eta}_{1i}, \boldsymbol{\eta}_{2i}) f(\boldsymbol{\eta}_{2i}|\boldsymbol{\eta}_{1i}) d\boldsymbol{\eta}_{2i}\right]}_{f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})} f(\boldsymbol{\eta}_{1i}) d\boldsymbol{\eta}_{1i}. \tag{12}$$

The utility of this rearrangement is that the interior integral, corresponding to the distribution of $\mathbf{y}_i$ conditional on $\boldsymbol{\eta}_{1i}$, but marginalized over $\boldsymbol{\eta}_{2i}$, can be computed in closed-form. Therefore, the latent variables $\boldsymbol{\eta}_{2i}$ can be analytically integrated, reducing the dimension of numerical integration required. This is equivalent to the method proposed by du Toit and Cudeck (2009) in the context of nonlinear mixed effects models.

3.3. Analytically integrating $\boldsymbol{\eta}_{2i}$ out of the likelihood

Let $\mathbf{\Pi} = (\mathbf{I} - \mathbf{B})^{-1}$, and

$$\mathbf{S}_1 = \begin{pmatrix} \mathbf{I}_{m_1} & \mathbf{0}_{m_1 \times m_2} \end{pmatrix} \quad \text{and} \quad \mathbf{S}_2 = \begin{pmatrix} \mathbf{0}_{m_2 \times m_1} & \mathbf{I}_{m_2} \end{pmatrix} \tag{13}$$

denote zero-one selection matrices such that

$$\mathbf{\Pi}_1 = \mathbf{S}_1 \mathbf{\Pi} \quad \text{and} \quad \mathbf{\Pi}_2 = \mathbf{S}_2 \mathbf{\Pi} \tag{14}$$

correspond to the first m_1 and last m_2 rows of $\mathbf{\Pi}$, respectively. The vector $\boldsymbol{\eta}_i$ is multivariate normal:

$$\begin{pmatrix} \boldsymbol{\eta}_{1i} \\ \boldsymbol{\eta}_{2i} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_1 & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_2 \end{pmatrix} \right], \tag{15}$$

where

$$\boldsymbol{\mu}_1 = \mathbf{\Pi}_1(\boldsymbol{\alpha} + \mathbf{\Gamma}\mathbf{x}_i), \quad \boldsymbol{\mu}_2 = \mathbf{\Pi}_2(\boldsymbol{\alpha} + \mathbf{\Gamma}\mathbf{x}_i), \tag{16}$$

$$\mathbf{\Sigma}_1 = \mathbf{\Pi}_1 \mathbf{\Psi} \mathbf{\Pi}_1', \quad \mathbf{\Sigma}_2 = \mathbf{\Pi}_2 \mathbf{\Psi} \mathbf{\Pi}_2', \quad \text{and} \quad \mathbf{\Sigma}_{21} = \mathbf{\Sigma}_{12}' = \mathbf{\Pi}_2 \mathbf{\Psi} \mathbf{\Pi}_1'. \tag{17}$$

Following, the distribution of $\boldsymbol{\eta}_{2i}$ conditional on $\boldsymbol{\eta}_{1i} = \mathbf{a}$ is also multivariate normal:

$$\boldsymbol{\eta}_{2i} | \boldsymbol{\eta}_{1i} = \mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}_{2\bullet 1}, \mathbf{\Sigma}_{2\bullet 1}), \tag{18}$$

where

$$\boldsymbol{\mu}_{2\bullet 1} = \boldsymbol{\mu}_2 + \mathbf{\Sigma}_{21}\mathbf{\Sigma}_1^{-1}(\mathbf{a} - \boldsymbol{\mu}_1), \tag{19}$$

and

$$\mathbf{\Sigma}_{2\bullet 1} = \mathbf{\Sigma}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{12}. \tag{20}$$

Finally, the distribution of $\mathbf{y}_i$ conditional on $\boldsymbol{\eta}_{1i} = \mathbf{a}$ follows as

$$\mathbf{y}_i | \boldsymbol{\eta}_{1i} = \mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}\bullet 1}, \mathbf{\Sigma}_{\mathbf{y}\bullet 1}), \tag{21}$$

where

$$\boldsymbol{\mu}_{\mathbf{y}\bullet 1} = \boldsymbol{\nu}_y + \mathbf{\Lambda}_{y1}\mathbf{a} + \mathbf{\Lambda}_{y2}\boldsymbol{\mu}_{2\bullet 1} + \mathbf{K}_y \mathbf{x}_i, \tag{22}$$

and

$$\mathbf{\Sigma}_{\mathbf{y}\bullet 1} = \mathbf{\Lambda}_{y2}\mathbf{\Sigma}_{2\bullet 1}\mathbf{\Lambda}_{y2}' + \mathbf{\Theta}. \tag{23}$$

Consequently, the likelihood function defined by Equation 12 can be reformulated as

$$\mathcal{L}_i(\boldsymbol{\vartheta}) = \int f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})f(\boldsymbol{\eta}_{1i})d\boldsymbol{\eta}_{1i}, \tag{24}$$

where

$$f(\mathbf{y}_i|\boldsymbol{\eta}_{1i}=\mathbf{a}) = (2\pi)^{-p/2}|\boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}|^{-1/2}\text{exp}\big\{-\frac{1}{2}(\mathbf{y}_i-\boldsymbol{\mu}_{\mathbf{y}\bullet 1})'\boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}(\mathbf{y}_i-\boldsymbol{\mu}_{\mathbf{y}\bullet 1})\big\}, \tag{25}$$

and

$$f(\mathbf{u}_i|\boldsymbol{\eta}_{1i}=\mathbf{a}) = \prod_j f_j(u_{ij}|\boldsymbol{\nu}_u + \boldsymbol{\Lambda}_{u1}\mathbf{a} + \mathbf{K}_u\mathbf{x}_i). \tag{26}$$

It is clear that the function has now been reduced to an m_1-dimensional integral.

3.4. Numerically approximating the likelihood

Despite the dimension of integration being reduced, the reformulated likelihood function is still intractable and so it must be numerically approximated. Numerous methods have been proposed for approximating the integral. In this section, adaptive and nonadaptive Gaussian quadrature methods are discussed.

3.4.1. Nonadaptive quadrature

Using Q nodes per dimension, the likelihood contribution for individual i can be approximated using Gaussian quadrature as

$$\mathcal{L}_i(\boldsymbol{\vartheta}) = \sum_{q_1=1}^{Q}\cdots\sum_{q_{m_1}=1}^{Q} f(\mathbf{u}_i|\boldsymbol{\eta}_{1i}=\mathbf{t}_q)f(\mathbf{y}_i|\boldsymbol{\eta}_{1i}=\mathbf{t}_q)w_{q_1}\dots w_{q_{m_1}}. \tag{27}$$

The quadrature nodes $\mathbf{t}_q = (t_{q_1},\dots,t_{q_{m_1}})'$ and weights $\mathbf{w}_q = (w_{q_1},\dots,w_{q_{m_1}})'$ are selected such that

$$\mathbf{t}_q = \boldsymbol{\mu}_1 + \sqrt{2}\boldsymbol{\Sigma}_1^{1/2}\mathbf{t}_q^*, \quad \text{and} \quad \mathbf{w}_q = \frac{1}{\sqrt{\pi}}\mathbf{w}_q^*,$$

where $\mathbf{t}_q^*$ and $\mathbf{w}_q^*$ are standard Gaussian quadrature nodes and weights, respectively, and $\boldsymbol{\Sigma}_1^{1/2}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}_1$.

3.4.2. Adaptive quadrature

The approximation method requires the conditional distribution to be evaluated a total of Q^{m_1} times per optimization iteration, where m_1 is fixed dependent on the model and Q is chosen by the user. Using a smaller Q results in quicker computation (at least within the given iteration), but may be too poor of an approximation to accurately estimate the model parameters or ensure that the estimation algorithm converges. In addition, for a fixed number of nodes, the approximation tends to perform more poorly when the the latent variables $\boldsymbol{\eta}_{1i}$ have substantial variability. This results from the quadrature nodes missing important features of the distribution for many individuals (see, e.g., Rabe-Hesketh et al., 2005, for a visual). To circumvent this problem, adaptive quadrature can be used, where the nodes are centered and scaled uniquely for each individual.

One approach for centering and scaling the nodes is to use the posterior means and variances. This approach has been implemented by Rabe-Hesketh et al. (2002) with gllamm and is the default method used by Stata's gsem function. However, obtaining the posterior means and variances also requires numerical integration and so an iterative process is needed. Further, Rabe-Hesketh et al. (2002) note that typically at least five quadrature nodes are required for the method to work properly.

Alternatively, the nodes may be centered around the joint posterior mode of the latent variables, where the scaling of the nodes is determined by the curvature at the mode (Q. Liu & Pierce, 1994; Pinheiro & Bates, 1995). The conditional posterior mode can be found by maximizing the (unnormalized) posterior density of $\boldsymbol{\eta}_{1i}$,

$$h^*(\boldsymbol{\eta}_{1i}|\mathbf{u}_i, \mathbf{y}_i) = f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})f(\boldsymbol{\eta}_{1i}), \tag{28}$$

with respect to $\boldsymbol{\eta}_{1i}$. Although the maximization also requires an iterative procedure, the computational complexity of the maximization does not depend on the number of quadrature points used as numerical integration is not required within this step. This method can also be used with a single node, resulting in the Laplace approximation (Q. Liu & Pierce, 1994; Pinheiro & Bates, 1995). The maximization can be performed using a Newton-Raphson routine, which requires the first- and second-order derivatives. These derivatives are provided in Appendix A.

Letting

$$\hat{\boldsymbol{\eta}}_{1i} = \underset{\boldsymbol{\eta}_{1i}}{\arg\max} \log\{h^*(\boldsymbol{\eta}_{1i}|\mathbf{u}_i, \mathbf{y}_i)\}, \tag{29}$$

denote the conditional posterior mode of $\boldsymbol{\eta}_{1i}$ and

$$\hat{\boldsymbol{\Sigma}}_i = \left[- \frac{\partial^2 \log\{h^*(\boldsymbol{\eta}_{1i}|\mathbf{u}_i, \mathbf{y}_i)\}}{\partial \boldsymbol{\eta}_{1i} \boldsymbol{\eta}'_{1i}} \Big|_{\boldsymbol{\eta}_{1i} = \hat{\boldsymbol{\eta}}_{1i}} \right]^{-1} \tag{30}$$

denote the curvature of the posterior distribution about the mode, a new set of quadrature nodes adapted to individual i can be constructed as

$$\tilde{\mathbf{t}}_q = \hat{\boldsymbol{\eta}}_{1i} + \sqrt{2} \hat{\boldsymbol{\Sigma}}_i^{1/2} \mathbf{t}_q^*, \tag{31}$$

where $\hat{\boldsymbol{\Sigma}}_i^{1/2}$ is the Cholesky decomposition of $\hat{\boldsymbol{\Sigma}}_i$. The likelihood contribution for individual i can then be expressed as:

$$\mathcal{L}_i(\boldsymbol{\vartheta}|\mathbf{u}_i, \mathbf{y}_i) = \int \frac{f(\mathbf{u}_i, \mathbf{y}_i|\boldsymbol{\eta}_{1i}) f(\boldsymbol{\eta}_{1i}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{f(\boldsymbol{\eta}_{1i}|\hat{\boldsymbol{\eta}}_{1i}, \hat{\boldsymbol{\Sigma}}_i)} f(\boldsymbol{\eta}_{1i}|\hat{\boldsymbol{\eta}}_{1i}, \hat{\boldsymbol{\Sigma}}_i) d\boldsymbol{\eta}_{1i},$$

which is approximated using the individual-centered and scaled nodes as:

$$\begin{aligned}
\mathcal{L}_i(\boldsymbol{\vartheta}|\mathbf{u}_i, \mathbf{y}_i) =& \sum_{q_1=1}^{Q} \cdots \sum_{q_{m_1}=1}^{Q} \frac{f(\mathbf{u}_i, \mathbf{y}_i|\boldsymbol{\eta}_{1i} = \tilde{\mathbf{t}}_q) f(\tilde{\mathbf{t}}_q|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{f(\tilde{\mathbf{t}}_q|\hat{\boldsymbol{\eta}}_{1i}, \hat{\boldsymbol{\Sigma}}_i)} w_{q_1} \dots w_{q_{m_1}} \\
=& (2\pi)^{m_1/2} |\hat{\boldsymbol{\Sigma}}_i|^{1/2} \sum_{q_1=1}^{Q} \cdots \sum_{q_{m_1}=1}^{Q} f(\mathbf{u}_i, \mathbf{y}_i|\boldsymbol{\eta}_{1i} = \tilde{\mathbf{t}}_q) f(\tilde{\mathbf{t}}_q|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\
& \times \exp(\mathbf{t}_q^{*'} \mathbf{t}_q^*) w_{q_1} \dots w_{q_{m_1}}.
\end{aligned}$$

For fixed Q, adaptive quadrature is more computationally complex than nonadaptive quadrature because the posterior mode and curvature must be determined for each individual. However, by scaling the nodes adaptively, using adaptive quadrature can typically result in an equally precise approximation as nonadaptive quadrature using much fewer nodes. Thus, estimation requiring multidimensional numerical integration can typically be performed faster using adaptive quadrature relative to nonadaptive quadrature.

3.5. Score Function

The log of the likelihood function can be maximized using any general purpose optimizer. Most optimizers require the first-order partial derivatives of the function being maximized with respect to the parameters (i.e., the gradient vector). When the function being maximized is the log-likelihood function, the gradient vector is commonly referred to as the score function. Although the score function can be numerically approximated, this is often too time consuming given the large number of estimated parameters. Thus, efficient computation of the analytic derivatives is highly advantageous. It turns out that these derivatives can be determined by employing a relation commonly relied upon within the EM algorithm.

Let $\mathbf{c}_i = (\mathbf{y}_i, \mathbf{u}_i, \boldsymbol{\eta}_{1i})$ denote the complete data and

$$\mathcal{L}_{Ci}(\boldsymbol{\vartheta}) = f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})f(\boldsymbol{\eta}_{1i}), \tag{32}$$

denote the complete data likelihood for individual i. Note that in this formulation $\boldsymbol{\eta}_{2i}$ has already been marginalized out of the likelihood function and so $\boldsymbol{\eta}_{2i}$ is not included within the complete data. Thus, $\mathcal{L}_{Ci}(\boldsymbol{\vartheta})$ could be more aptly termed the partially complete data likelihood, but the term complete data likelihood will continue to be used here.

The partial derivatives of the complete data log-likelihood, $l_{Ci}(\boldsymbol{\vartheta}) = \log\{\mathcal{L}_{Ci}(\boldsymbol{\vartheta})\}$, with respect to $\boldsymbol{\vartheta}$ can be denoted as

$$S_{Ci}(\boldsymbol{\vartheta}) = \frac{\partial}{\partial\boldsymbol{\vartheta}} l_{Ci}(\boldsymbol{\vartheta}) = \frac{\partial}{\partial\boldsymbol{\vartheta}}[\log\{f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})\} + \log\{f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})\} + \log\{f(\boldsymbol{\eta}_{1i})\}]. \tag{33}$$

The complete data score function $S_{Ci}(\boldsymbol{\vartheta})$ has a number of interesting properties that result in (relatively) straightforward computation. First, because $f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})$ and $f(\boldsymbol{\eta}_{1i})$ do not depend on the distribution of $\mathbf{u}_i$, neither do the derivatives of the log of each of these components. Therefore, only the derivatives of $\log\{f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})\}$ must be adapted depending on the specific model for $\mathbf{u}_i$. Second, $f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})$ only depends on the parameters $\boldsymbol{\nu}_u$, $\boldsymbol{\Lambda}_{u1}$, and $\mathbf{K}_u$. Thus, the derivatives of the complete data log-likelihood with respect to all other parameters remain unchanged with different specification of the distribution of $\mathbf{u}_i$. Finally, since the u_{ij} are independent of one another conditional on $\boldsymbol{\eta}_{1i}$, the derivatives of the log of $f(\mathbf{u}_{ij}|\boldsymbol{\eta}_{1i})$ are simply the sum of the derivatives of the log of $f_j(u_{ij}|\boldsymbol{\eta}_{1i})$ across j. These derivatives are already well documented for a variety of common models, such as generalized linear models and their extensions (e.g., ZIP models, etc.). The derivatives of the complete data log-likelihood with respect to all of the model parameters are provided in Appendix B.

The observed data score function for individual i, $S_i(\boldsymbol{\vartheta})$, can then be computed by recognizing that the the observed data score function is equal to the expectation of the complete data score function over the posterior distribution of the latent variables given the observed variables (Cai, 2010; Bianconcini et al., 2014):

$$S_i(\boldsymbol{\vartheta}) = \int S_{Ci}(\boldsymbol{\vartheta})h(\boldsymbol{\eta}_{1i}|\mathbf{y}_i, \mathbf{u}_i)d\boldsymbol{\eta}_{1i}, \tag{34}$$

where $h(\boldsymbol{\eta}_{1i}|\mathbf{y}_i, \mathbf{u}_i)$ is the posterior distribution of $\boldsymbol{\eta}_{1i}$, $h(\boldsymbol{\eta}_{1i}|\mathbf{y}_i, \mathbf{u}_i) = h^*(\boldsymbol{\eta}_{1i}|\mathbf{y}_i, \mathbf{u}_i)/f(\mathbf{y}_i, \mathbf{u}_i)$. This equality is sometimes referred to as Fisher's Identity (Fisher, 1925; Cai, 2010). As with the likelihood function, the observed data score function does not have a closed form solution, but it can be approximated using adaptive quadrature as:

$$S_i(\boldsymbol{\vartheta}) \approx \frac{\sum_{q_1=1}^{Q} \cdots \sum_{q_{m_1}=1}^{Q} S_{Ci}(\boldsymbol{\vartheta}) f(\mathbf{u}_i, \mathbf{y}_i|\boldsymbol{\eta}_{1i} = \tilde{\mathbf{t}}_q) f(\tilde{\mathbf{t}}_q|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\text{exp}(\mathbf{t}_q^{*\prime}\mathbf{t}_q^*) w_{q_1} \ldots w_{q_{m_1}}}{\sum_{q_1=1}^{Q} \cdots \sum_{q_{m_1}=1}^{Q} f(\mathbf{u}_i, \mathbf{y}_i|\boldsymbol{\eta}_{1i} = \tilde{\mathbf{t}}_q) f(\widetilde{\mathbf{t}_q}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\text{exp}(\mathbf{t}_q^{*\prime}\mathbf{t}_q^*) w_{q_1} \ldots w_{q_{m_1}}}, \tag{35}$$

where the complete data score function $S_{Ci}(\boldsymbol{\vartheta})$ is evaluated at $\boldsymbol{\eta}_{1i} = \tilde{\mathbf{t}}_q$. The score function for the entire sample is then $S(\boldsymbol{\vartheta}) = \sum_i S_i(\boldsymbol{\vartheta})$.

4. Example

In this section, the applicability of the recently detailed method for ML estimation is demonstrated by fitting two example models inspired by the study conducted by Collins, Witkiewitz, and Larimer (2011), who were interested in how subjective drinking norms, attitudes towards alcohol, drinking behavior self efficacy, and drinking behavior control relate to drinking intentions which, in turn, relate to changes in drinking behavior over time. The authors measured the drinking norms, attitudes, self efficacy, control, and intentions of 837 college students using four, four, two, three, and two items, respectively. The number of heavy drinking days within the past month was also measured at this time, and each of the following three months for a total of four measurement occasions. The number of heavy drinking days was modeled using the Poisson distribution and the remaining observed variables were assumed to be normally distributed, or at least approximately so. Simulated data, constructed using the descriptive statistics and parameter estimates provided by Collins et al. (2011), are used here so that the data can be shared without restriction.

Two example models are fit to this data. The first model that is fit to the data here is comparable to that used by Collins et al. (2011). This model demonstrates types of models that are common in behavioral science research and how the methods presented within this paper can aid in efficiently estimating the parameters of such models. Next, the data will be used to fit another model corresponding to a Poisson regression with latent predictors and overdispersion. This example demonstrates how the model of interest can sometimes be reparameterized to take advantage of the numerical integration dimension reduction technique described here.

For the current implementation, the likelihood function and first-order derivatives were coded within C++ and maximized using the nlminb algorithm within the optimx R package (Nash, 2014) via Rcpp (Eddelbuettel, 2013) and RcppArmadillo (Eddelbuettel & Sanderson, 2014). All of the code necessary to replicate the analyses is provided at https://osf.io/w8u2q/. The parameter estimates are also compared to those obtained using Mplus and, when possible, the gsem function in Stata, which relies on many of the same underlying computational methods as gllamm with the addition of analytic derivatives.

4.1. Full structural model with latent growth factors as outcomes

4.1.1. Model

This first example uses a model that is comparable to that used by Collins et al. (2011)[1], in which subjective drinking norms, attitudes, control, self efficacy, and intentions are all modeled using latent factors with their respective item indicators. The intentions latent factor is then regressed on the other four (norms, attitudes, control, self efficacy), with estimated covariances among the four exogenous latent variables. The number of

[1] There are two minor differences between the model implemented here and the model used by Collins et al. (2011). First, Collins et al. (2011) modeled subjective drinking norms using a hierarchical factor model with two sub factors, corresponding to descriptive and injunctive norms, each measured by two items. Second, Collins et al. (2011) used a quadratic growth model for the number of drinking episodes, whereas a linear growth model is used here.

heavy drinking episodes at the 4 time points, each assumed to be (conditionally) distributed as Poisson random variables with a log link function, are modeled using a latent growth curve, where the latent growth factors are regressed on the latent drinking intentions factor. A path diagram corresponding to the model is displayed in Figure 1.

[Figure 1 about here.]

The full model specification is:

$$\begin{pmatrix} \xi_{1i} \\ \xi_{2i} \\ \xi_{3i} \\ \xi_{4i} \\ \mathrm{Norm}_{1i} \\ \mathrm{Norm}_{2i} \\ \mathrm{Norm}_{3i} \\ \mathrm{Norm}_{4i} \\ \mathrm{Att}_{1i} \\ \mathrm{Att}_{2i} \\ \mathrm{Att}_{3i} \\ \mathrm{Att}_{4i} \\ \mathrm{Cont}_{1i} \\ \mathrm{Cont}_{2i} \\ \mathrm{Cont}_{3i} \\ \mathrm{SE}_{1i} \\ \mathrm{SE}_{2i} \\ \mathrm{Int}_{1i} \\ \mathrm{Int}_{2i} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \nu_5 \\ \nu_6 \\ \nu_7 \\ \nu_8 \\ \nu_9 \\ \nu_{10} \\ \nu_{11} \\ \nu_{12} \\ \nu_{13} \\ \nu_{14} \\ \nu_{15} \\ \nu_{16} \\ \nu_{17} \\ \nu_{18} \\ \nu_{19} \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{63} & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{73} & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{83} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{10,4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{11,4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{12,4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{14,5} & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{15,5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{17,6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{19,7} \end{pmatrix} \begin{pmatrix} \mathrm{Intercept}_i \\ \mathrm{Slope}_i \\ \mathrm{Norm}_i \\ \mathrm{Att}_i \\ \mathrm{Cont}_i \\ \mathrm{SE}_i \\ \mathrm{Intent}_i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \varepsilon_{5i} \\ \varepsilon_{6i} \\ \varepsilon_{7i} \\ \varepsilon_{8i} \\ \varepsilon_{9i} \\ \varepsilon_{10,i} \\ \varepsilon_{11,i} \\ \varepsilon_{12,i} \\ \varepsilon_{13,i} \\ \varepsilon_{14,i} \\ \varepsilon_{15,i} \\ \varepsilon_{16,i} \\ \varepsilon_{17,i} \\ \varepsilon_{18,i} \\ \varepsilon_{19,i} \end{pmatrix}, \tag{36}$$

$$\mathrm{HED}_{ji} \sim \mathrm{Poisson}(\exp\{\xi_{ji}\}), \quad j = 1, 2, 3, 4, \tag{37}$$

and

$$\begin{pmatrix} \mathrm{Intercept}_i \\ \mathrm{Slope}_i \\ \mathrm{Norm}_i \\ \mathrm{Att}_i \\ \mathrm{Cont}_i \\ \mathrm{SE}_i \\ \mathrm{Intent}_i \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \beta_{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta_{27} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta_{73} & \beta_{74} & \beta_{75} & \beta_{76} & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Intercept}_i \\ \mathrm{Slope}_i \\ \mathrm{Norm}_i \\ \mathrm{Att}_i \\ \mathrm{Cont}_i \\ \mathrm{SE}_i \\ \mathrm{Intent}_i \end{pmatrix} + \begin{pmatrix} \zeta_{1i} \\ \zeta_{2i} \\ \zeta_{3i} \\ \zeta_{4i} \\ \zeta_{5i} \\ \zeta_{6i} \\ \zeta_{7i} \end{pmatrix}, \tag{38}$$

where $\mathbf{\Theta} = \text{diag}(\theta_{55}, \theta_{66}, ..., \theta_{19,19})$ and

$$\mathbf{\Psi} = \begin{pmatrix} \psi_{11} & \psi_{12} & 0 & 0 & 0 & 0 & 0 \\ \psi_{21} & \psi_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \psi_{33} & \psi_{34} & \psi_{35} & \psi_{36} & 0 \\ 0 & 0 & \psi_{43} & \psi_{44} & \psi_{45} & \psi_{46} & 0 \\ 0 & 0 & \psi_{53} & \psi_{54} & \psi_{65} & \psi_{56} & 0 \\ 0 & 0 & \psi_{63} & \psi_{64} & \psi_{75} & \psi_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \psi_{77} \end{pmatrix}. \tag{39}$$

In total, the model has 62 freely estimated parameters and 7 latent variables. Thus, without using the numerical integration dimension reduction technique presented here, ML parameter estimation would require 7 dimensions of numerical integration. However, as shown in Equation 36, the only latent variables that directly influence nonnormal observed variables (i.e., the heavy episodic drinking variables) are the latent intercept and slope growth factors. Thus, only two dimensions of numerical integration are needed when the dimension reduction technique is employed, as once the latent growth factors are conditioned on the other latent factors can be analytically integrated out of the likelihood function.

4.1.2. Parameter Estimates

The model was fit to the data using the estimation routine described in this paper and Mplus. In both implementations, adaptive quadrature was used with 10 nodes per dimensions, resulting in a total of 100 integration nodes. The gsem function in Stata was also used to try to estimate the model parameters. Due to the high dimension of numerical integration required, only three nodes per dimension could be used, resulting in a total of $3^7 = 2187$ total nodes. Unfortunately, the model failed to converge after several days of computation and it would be too computationally intensive to further increase the number of nodes used per dimension of numerical integration (e.g., using 4 nodes per dimension would require $4^7 = 16384$ total nodes).

The parameter estimates and standard errors for most of the estimates obtained using Mplus and the method introduced here are displayed in Table 1. To ensure the covariance matrices during the R/C++ estimation were positive definite, the matrices were parameterized using variances and correlations with appropriate bound constraints. The

Mplus estimates of correlations were obtained via the option for standardized estimates. Estimates of item intercepts and residual variances are omitted from Table 1 for space-saving purposes.

As is clear, the parameter estimates and final log-likelihood from the R/C++ implementation are nearly identical to those obtained using Mplus. These estimates, which again are based on data simulated to be comparable to that used by Collins et al. (2011), indicate that drinking attitudes and subjective drinking norms are positively associated with drinking intentions ($\hat{\beta}_{74} = 0.44$ and $\hat{\beta}_{73} = 0.11$, respectively), whereas drinking self-efficacy and control are negatively associated with drinking intentions (though the effect of control is not statistically significant; $\hat{\beta}_{76} = -0.17$ and $\hat{\beta}_{75} = -0.07$, respectively). Further, those with higher drinking intentions tend to engage in more heavy episodic drinking events at baseline ($\hat{\beta}_{17} = 0.15$). Individuals with average intentions are estimated to decrease in their heavy episodic drinking behaviors over time ($\hat{\alpha}_2 = -0.05$), but this decrease is less pronounced for those with higher drinking intentions ($\hat{\beta}_{27} = 0.04$). However, even after accounting for drinking intentions, there are significant individual differences in both heavy episodic drinking behaviors at baseline ($\hat{\psi}_{11} = 0.04$) and changes in such behavior over time ($\hat{\psi}_{22} = 0.02$). Readers are encouraged to consult Collins et al. (2011) for the actual substantive conclusions of the analysis.

[Table 1 about here.]

In addition to the estimates being comparable across software implementations, refitting the models using more nodes per dimension resulted in negligible changes to these estimates, suggesting that the 10 nodes per dimension used are sufficient for accurately approximating the likelihood function for this model. Although estimation using Mplus was the fastest program for obtaining estimates (7 seconds), the R/C++ implementation was still fast (9.8 minutes) relative to Stata's gsem function, which took several days before a nonconvergence error was returned.

Differences in estimation time between the Mplus and R/C++ implementation are likely due to a variety of minor differences in computational implementation, including code optimization within Mplus, the optimization routine used (e.g., EM algorithm, nlminb, etc.), and the programming language. For example, Mplus is written in Fortran, which is a very fast compiled language. Although the likelihood function and derivatives for the custom implementation were written in C++, which is also a fast compiled language, the optimization was carried out via R, which is typically orders of magnitude slower than both Fortran and C++. Thus, even if the underlying computations were exactly equivalent between the Mplus and R/C++ implementations, it is expected that Mplus' Fortran implementation would be faster. Further, Mplus uses analytic second-order derivatives, which are useful for both optimization and computation of standard errors. The R/C++ computation of standard errors relies on numerically differentiating the analytic gradient, and the computation of the standard errors was responsible for a large portion of the estimation time for the R/C++ implementation.

Although between-program differences in estimation time are interesting, the primary goal here is to determine the impact of reducing the dimension of numerical integration. Therefore, the models were re-estimated within both Mplus and R/C++ with parameterizations that resulted in all seven latent variables being numerically integrated. This allows for a comparison of estimation times within a given program, which can be used to assess the effect of the dimension of numerical integration after controlling for all other factors that result in between-program differences.

In Mplus, the model was refit using six adaptive quadrature nodes per each of the seven dimension, as using fewer nodes resulted in nonconvergence issues. Although the parameter estimates and final log likelihood (displayed in the final column in Table 1) are generally comparable to those obtained using the dimension reduction technique, the estimation procedure took over 38 hours to complete, which is in stark contrast to the 7 seconds required when only two dimensions were numerically integrated. Within the R/C++ implementation, convergence could be obtained using four adaptive quadrature nodes per dimension, which took over 5 days to complete. Although most of the parameter estimates are comparable to when the model was fit using two dimensions of numerical integration, there are some discrepancies for the estimated variances and correlation for the latent growth factors. There were also issues with the standard errors for these parameters, which likely stemmed from there being too few quadrature nodes to accurately approximate the derivatives. Increasing the number of quadrature nodes could mitigate this issue, though this would substantially increase the computational complexity. Together, these results demonstrate the dramatic effect that the dimension reduction technique can have on estimation time and, in some scenarios, the parameter estimates.

4.2. Poisson regression with latent predictors and overdispersion

In this example, it is demonstrated how a model in which several latent variables directly related to a nonnormal and/or categorical outcome variable can be reparameterized to take advantage of the dimension reduction technique presented here. This example uses only a subset of the latent variables from the full model to allow estimates obtained using Stata's gsem function, which does not have the capability of reducing the dimension of numerical integration, to be compared to the R/C++ and Mplus implementations.

4.2.1. Model

Suppose there is interest on the effect of the exogenous latent factors from the previous analysis on the number of heavy drinking episodes at the final measurement (i.e., HED_{4i}). The equation for the linear function ξ_i of the count outcome in scalar form is

$$\xi_i = \nu_1 + \lambda_{11}\mathrm{Norm}_i + \lambda_{12}\mathrm{Att}_i + \lambda_{13}\mathrm{Cont}_i + \lambda_{14}\mathrm{SE}_i. \tag{40}$$

Count data often has a larger variance than mean, a scenario known as overdispersion. Although it is common to use the negative binomial distribution rather than the Poisson distribution when overdispersion is present, an alternative within latent variable modeling frameworks is to include a random effect/latent factor in the model to account for the overdispersion. A path diagram corresponding to the model with an additional latent factor for overdispersion is presented in Figure 2. The model for ξ_i is

$$\xi_i = \nu_1 + \mathrm{Disp}_i + \lambda_{21}\mathrm{Norm}_i + \lambda_{13}\mathrm{Att}_i + \lambda_{14}\mathrm{Cont}_i + \lambda_{15}\mathrm{SE}_i, \tag{41}$$

where the loading/coefficient for Disp_i, the dispersion factor, is constrained to 1 with a freely estimated variance.

[Figure 2 about here.]

As parameterized, these two models would required four and five dimensions of numerical integration, respectively, as each of the latent factors have direct effects on the count outcome. However, an equivalent model can be specified using a parameterization in which the latent norms, attitude, control, and self efficacy factors directly influence the overdispersion factor, rather than the observed count outcome:

$$\xi_i = \nu_1 + \mathrm{Disp}_i, \quad \text{where} \quad \mathrm{Disp}_i = \beta_{12}\mathrm{Norm}_i + \beta_{13}\mathrm{Att}_i + \beta_{14}\mathrm{Cont}_i + \beta_{15}\mathrm{SE}_i + \zeta_{1i} \tag{42}$$

$$\xi_i = \nu_1 + (\beta_{12}\mathrm{Norm}_i + \beta_{13}\mathrm{Att}_i + \beta_{14}\mathrm{Cont}_i + \beta_{15}\mathrm{SE}_i + \zeta_{1i}). \tag{43}$$

The residual variance of the overdispersion factor in the new parameterization corresponds to the variance of the overdispsersion factor in the original model. With the new parameterization it is clear that only one dimension of numerical integration is required to estimate the parameters of the model. Further, adding additional latent predictors (with conditionally normal indicator variables) of the count outcome via the overdispsersion factor within the new parameterization has no affect on the dimension of numerical integration required. This is because the distribution of the count outcome does not depend on the full multivariate distribution of the latent factors, but only the specific linear function of the latent factors that makes up the Disp_i variable in Equation 42.

In general, a “phantom” factor, such as Disp_i in Equation 42, may be used for each individual nonnormal response variable in a given model, resulting in the dimension of numerical integration required always being less than or equal to the total number of nonnormal response variables. If a phantom factor is used for a nonnormal response and the residual variance of the factor is freely estimated, the variance corresponds to an overdispersion parameter. For nonnormal response variables where overdispersion is not of interest (including nonnormal response variables where an overdispersion parameter would not be identified), the residual variance of the phantom factor could be constrained to 0 (or a small number such as .001) to still take advantage of the reduced dimension of numerical integration resulting from use of the phantom variable.

Depending on the specific model, it may or may not be advantageous to use the phantom variable approach. In the previous latent growth model, it would not be advised since this would require four separate phantom variables that must be numerically integrated, whereas in the original model parameterization there were only two latent factors that influenced nonnormal variables. For other models, it may be beneficial to use a phantom factor for only a subset of the nonnormal responses. Appendix C describes a more thorough account of model reparmeterization and presents a set of steps for determining the reparameterization that minimizes the dimensions of numerical integration needed for any given model. These steps may be conducted algorithmically within the estimation routine, which would take the burden of reparameterization off of the user.

4.2.2. Parameter Estimates

The model with overdispersion was fit using the R/C++ implementation, Mplus, and the gsem function in Stata. For the R/C++ and Mplus implementations, the model was fit using the reparameterized model that only requires one dimension of numerical integration. Ten nodes for this single dimension were deemed to be sufficient for accurate parameter estimation. Within gsem, the parameterization does not change the dimension of numerical integration required for estimation, as both methods require five dimensions. For each dimension, five nodes were used resulting in a total of $5^5 = 3125$ quadrature nodes.

The parameter estimates obtained using the three implementations are displayed in Table 2. All three methods produced near equivalent results that only differed in the hundredths decimal place for a few estimates. However, there was a vast difference in the time required for estimation. Both the R/C++ and Mplus implementations took under a minute. The gsem implementation, on the other hand, took over 3 hours and 45 minutes to complete. Thus, the reduction in the dimension of integration that results from the methods introduced within this paper is instrumental for computationally efficient estimation. Further, because the R/C++ and Mplus implementations only required one dimension of numerical integration, minimal time was required to re-estimate the model using more nodes to ensure that the quadrature-based approximation to the likelihood function is accurate. It would not be practical to try to reestimate the model using gsem with an additional node per dimension, as using 6 nodes per dimension would required $6^5 = 7765$ total nodes, which is more than double what was used initially.

[Table 2 about here.]

To further explore the time required for parameter estimation as a function of the dimension of numerical integration, the model was refit within the R/C++ and Mplus implementations using model parameterizations that required various dimensions of numerical integration ranging from one to five dimensions. Five nodes per dimension were used so that the estimation times, which are displayed in Table 3, can be compared to the gsem estimation method. As with the previous example, Mplus is quite a bit faster than the R/C++ implementations. When five dimensions of numerical integration was required, the Mplus implementation did not converge, whereas the R/C++ implementation was about an hour faster than the gsem implementation. Of primary importance, it is also clear that within both R/C++ and Mplus, estimation time increases dramatically as a function of the dimension of numerical integration. Thus, holding all other implementation variables constant (e.g., optimization algorithm, programming language, etc.), these results suggest that it can be highly beneficial to adopt the dimension reduction technique described throughout this paper.

[Table 3 about here.]

5. Conclusion

A flexible generalized SEM that allows for a combination of response variables from various distributions was introduced and an efficient ML estimation routine was detailed. A major advantage of the routine is the potential for reducing the dimension of numerical integration required for certain model specifications. The first-order derivatives of the log-likelihood function were also provided to improve upon the speed of parameter estimation. The method was demonstrated using two example analyses of alcohol use data similar in structure to that used by Collins et al. (2011).

As demonstrated by the example analyses, the dimension reduction technique can have profound effects on the time required for parameter estimation. These findings are consistent with the orignal assessment of the dimension reduction technique conducted by du Toit and Cudeck (2009) for nonlinear mixed models. Although the performance of the parameter estimates were not directly assessed here, du Toit and Cudeck (2009) also determined that reducing the dimension of numerical integration can result in less biased parameter estimates, as the likelihood function may be better approximated. For multilevel SEMs with random slopes, Rockwood (2020) also found that reducing the dimension of numerical integration required tended to result in less biased parameter estimates, smaller mean square errors for the estimates, and fewer nonconvergence issues. It is reasonable to suggest that these advantages would also be applicable to the generalized SEM discussed here.

Although the proposed method for reducing the dimension of numerical integration required for parameter estimation vastly expands the potential range of models that can be feasibly estimated using quadrature based methods, models with many latent variables that directly affect nonnormal responses may still required a dimension of numerical integration that is not practical. In some circumstances, such as in the second example provided here, the model can be reparameterized following the rules discussed in Appendix C to allow for a final dimension of numerical integration that is practical to approximate using quadrature based methods. For other models with many latent variables, using simulation based methods, such as Monte Carlo integration within the ML framework or Markov Chain Monte Carlo (MCMC) within the Bayesian framework, may be required to obtain parameter estimates.

It would also be worthwhile to extend the methods detailed here to some interesting model extensions that have been implemented in gllamm (Rabe-Hesketh et al., 2004), gsem (StataCorp, 2019b), and Mplus (L. K. Muthén & Muthén, 2017). Mixture models, which rely upon categorical latent variables, have become increasingly popular in substance use, delinquency, and personality research. In addition to more traditional uses of mixture models within the SEM framework, such models can also be used for modeling nonlinear relationships among latent variables (Bauer, 2005; Pek, Sterba, Kok, & Bauer, 2009).

The close relationship between structural equation models and multilevel modeling, which has been detailed extensively (e.g., Bauer, 2003; Curran, 2003), has been exploited to combine such methods within a unifying framework known as multilevel SEM. The software programs focused on here (Mplus, gsem, gllamm) can each fit various forms of multilevel SEMs with generalized responses. However, such models typically have several latent variables and so estimation can be very computationally intensive. As mentioned previously, Rockwood (2020) demonstrated that the numerical integration dimension reduction technique used here can also be used for estimating multilevel SEMs with random slopes for latent covariates and normal response variables. Future research could work to combine the methods introduced here with those presented by Rockwood (2020) for efficient ML estimation of multilevel generalized SEMs.

References

Asparouhov, T., Masyn, K., & Muthen, B. (2006). Continuous time survival in latent variable models. In *Proceedings of the Joint Statistical Meeting in Seattle* (pp. 180–187).

Bartholomew, D. J., & Knott, M. (1999). Kendall's library of statistics 7: Latent variable models and factor analysis. *London: Edward Arnold*.

Bauer, D. J. (2003). Estimating multilevel linear models as structural equation models. *Journal of Educational and Behavioral Statistics*, *28*(2), 135–167.

Bauer, D. J. (2005). A semiparametric approach to modeling nonlinear relations among latent variables. *Structural Equation Modeling*, *12*(4), 513–535.

Bentler, P. M. (2004). *Eqs 6: Structural equations program manual.* Multivariate software.

Bianconcini, S., et al. (2014). Asymptotic properties of adaptive maximum likelihood estimators in latent variable models. *Bernoulli*, *20*(3), 1507–1531.

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

Collins, S. E., Witkiewitz, K., & Larimer, M. E. (2011). The theory of planned behavior as a predictor of growth in risky college drinking. *Journal of Studies on Alcohol and Drugs*, *72*(2), 322–332.

Curran, P. J. (2003). Have multilevel models been structural equation models all along? *Multivariate Behavioral Research*, *38*(4), 529–569.

du Toit, S. H., & Cudeck, R. (2009). Estimation of the nonlinear random coefficient model when some random effects are separable. *Psychometrika*, *74*(1), 65–82.

Eddelbuettel, D. (2013). *Seamless R and C++ integration with Rcpp.* New York: Springer. (ISBN 978-1-4614-6867-7) doi: 10.1007/978-1-4614-6868-4

Eddelbuettel, D., & Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics and Data Analysis*, *71*, 1054–1063. Retrieved from `http://dx.doi.org/10.1016/j.csda.2013.02.005`

Fisher, R. A. (1925). Theory of statistical estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society* (Vol. 22, pp. 700–725).

Fox, J., Nie, Z., & Byrnes, J. (2017). sem: Structural Equation Models [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=sem` (R package version 3.1-9)

Jöreskog, K. G., & Sörbom, D. (1996). *Lisrel 8: User's reference guide.* Scientific Software International.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, *34*(1), 1–14.

Liu, H. (2007). Growth curve models for zero-inflated count data: An application to smoking behavior. *Structural Equation Modeling: A Multidisciplinary Journal*, *14*(2), 247–279.

Liu, Q., & Pierce, D. A. (1994). A note on Gauss—Hermite quadrature. *Biometrika*, *81*(3), 624–629.

Moustaki, I., & Knott, M. (2000). Generalized latent trait models. *Psychometrika*, *65*(3), 391–411.

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika*, *49*(1), 115–132.

Muthén, B., & Asparouhov, T. (2008). Growth mixture modeling: Analysis with non-gaussian random effects. *Longitudinal data analysis*, *143165*.

Muthén, B., & Asparouhov, T. (2011). Beyond multilevel regression modeling: Multilevel analysis in a general latent variable framework. *Handbook of Advanced Multilevel Analysis*, 15–40.

Muthén, B. O. (2002). Beyond SEM: General latent variable modeling. *Behaviormetrika*, *29*(1), 81–117.

Muthén, B. O., Muthén, L. K., & Asparouhov, T. (2017). *Regression and mediation analysis using mplus.* Muthén & Muthén Los Angeles, CA.

Muthén, L. K., & Muthén, B. (2017). Mplus version 8 [Computer software manual]. Los Angeles, CA: Muthén & Muthén.

Nash, J. C. (2014). On best practice optimization methods in R. *Journal of Statistical Software*, *60*(2), 1–14. Retrieved from `http://www.jstatsoft.org/v60/i02/`

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., . . . Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, *81*(2), 535-549.

Pek, J., Sterba, S. K., Kok, B. E., & Bauer, D. J. (2009). Estimating and visualizing nonlinear relations among latent variables: A semiparametric approach. *Multivariate Behavioral Research*, *44*(4), 407–436.

Pinheiro, J. C., & Bates, D. M. (1995). Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics*, *4*(1), 12–35.

R Core Team. (2019). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Rabe-Hesketh, S., & Skrondal, A. (2007). Multilevel and latent variable modeling with composite links and exploded likelihoods. *Psychometrika*, *72*(2), 123.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2002). Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal*, *2*(1), 1–21.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2004). Generalized multilevel structural equation modeling. *Psychometrika*, *69*(2), 167–190. doi: 10.1007/bf02295939

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

Rockwood, N. J. (2020). Maximum likelihood estimation of multilevel structural equation models with random slopes for latent covariates. *Psychometrika*, *85*(2), 275–300.

Rosseel, Y. (2012). Lavaan: An R package for structural equation modeling and more. version 0.5–12 (BETA). *Journal of Statistical Software*, *48*(2), 1–36.

Skrondal, A., & Rabe-Hesketh, S. (2003). Multilevel logistic regression for polytomous data and rankings. *Psychometrika*, *68*(2), 267–287.

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

Skrondal, A., & Rabe-Hesketh, S. (2009). Prediction in multilevel generalized linear models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *172*(3), 659–687.

StataCorp. (2019a). *Stata statistical software: Release 16.* College Station, TX: StataCorp LLC.

StataCorp. (2019b). *Stata structural equation modeling reference manual release 16.* College Station, TX: StataCorp LLC.

Wang, L. (2010). IRT–ZIP modeling for multivariate zero-inflated count data. *Journal of Educational and Behavioral Statistics*, *35*(6), 671–692.

Zheng, X., & Rabe-Hesketh, S. (2007). Estimating parameters of dichotomous and ordinal item response models with gllamm. *The Stata Journal*, *7*(3), 313–333.

Appendix A

In this Appendix the first- and second-order derivatives of the log of the unnormalized conditional posterior distribution of $\boldsymbol{\eta}_{1i}$,

$$\log\{h^*(\boldsymbol{\eta}_{1i}|\mathbf{u}_i,\mathbf{y}_i)\} = \log\{f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})\} + \log\{f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})\} + \log\{f(\boldsymbol{\eta}_{1i})\}, \tag{A1}$$

with respect to $\boldsymbol{\eta}_{1i}$ are provided. These derivatives, which are simply the sum of the derivatives of each of the individual components above, can be used in conjunction with the Newton-Raphson algorithm to obtain the posterior mode and curvature of $\log\{h^*(\boldsymbol{\eta}_{1i}|\mathbf{u}_i,\mathbf{y}_i)\}$ at the mode, which can then be used to center and scale the adaptive quadrature nodes.

First, define the intermediary quantities

$$\mathbf{A} = (\boldsymbol{\Lambda}'_{y1} + \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Lambda}'_{y2})\boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}, \quad \text{and} \quad \mathbf{C} = \mathbf{A}\boldsymbol{\Lambda}'_{y1}, \tag{A2}$$

which do not depend on $\boldsymbol{\eta}_{1i}$ or the data and so only need to be computed once for all individuals (i.e., these quantities remain constant across all i). Using these quantities, the first- and second-order derivatives of the components of $\log\{h^*(\boldsymbol{\eta}_{1i}|\mathbf{u}_i,\mathbf{y}_i)\}$ with respect to $\boldsymbol{\eta}_{1i}$ are

$$\frac{\partial \log\{f(\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}} = -\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\eta}_{1i} - \boldsymbol{\mu}_1), \quad \frac{\partial^2 \log\{f(\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}\partial \boldsymbol{\eta}'_{1i}} = -\boldsymbol{\Sigma}_1^{-1}, \tag{A3}$$

$$\frac{\partial \log\{f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})\}}{\boldsymbol{\eta}_{1i}} = \mathbf{A}(\mathbf{y}_i - \boldsymbol{\mu}_{\mathbf{y}\bullet 1}), \quad \frac{\partial^2 \log\{f(\mathbf{y}_i|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}\boldsymbol{\eta}'_{1i}} = -\mathbf{C}, \tag{A4}$$

$$\frac{\partial \log\{f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}} = \sum_j \frac{\partial \log\{f(u_{ij}|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}}, \quad \text{and} \quad \frac{\partial^2 \log\{f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}\boldsymbol{\eta}'_{1i}} = \sum_j \frac{\partial^2 \log\{f(u_{ij}|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}\boldsymbol{\eta}'_{1i}}. \tag{A5}$$

The final two quantities, corresponding to the first- and second-order derivatives of $\log\{f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})\}$, are the only ones that depend on the distributions of $\mathbf{u}_i$. When u_{ij} is conditionally distributed as a Poisson random variable with mean $\exp(\xi_{ij})$, then

$$\frac{\partial \log\{f(u_{ij}|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}} = u_{ij}\boldsymbol{\Lambda}'_{u1[j]} - \exp(\xi_{ij})\boldsymbol{\Lambda}'_{u1[j]}, \tag{A6}$$

and

$$\frac{\partial^2 \log\{f(u_{ij}|\boldsymbol{\eta}_{1i})\}}{\partial \boldsymbol{\eta}_{1i}\boldsymbol{\eta}'_{1i}} = -\exp(\xi_{ij})\boldsymbol{\Lambda}'_{u1[j]}\boldsymbol{\Lambda}_{u1[j]}, \tag{A7}$$

where $\boldsymbol{\Lambda}_{u1[j]}$ is the jth row of $\boldsymbol{\Lambda}_{u1}$.

Appendix B

Most optimization algorithms use first-order partial derivates. As discusses in Section 3.5, the derivatives of the (observed data) log-likelihood can be computed using Equastion 35, which is a function of the complete data score function with $\boldsymbol{\eta}_{1i}$ set to the quadrature nodes $\tilde{\mathbf{t}}_q$. In this Appendix, the derivatives of the complete data log-likelihood with respect to each of the parameter vectors/matrices is provided.

As with the derivatives provided in Appendix A, it helps to first define intermediary quantities that remain constant across all individuals and nodes. Specifically, define:

$$\mathbf{V} = \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Pi}_1, \quad \mathbf{Z} = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\boldsymbol{\Lambda}_{y2}\boldsymbol{\Sigma}_{2\bullet 1}, \quad \mathbf{W} = \boldsymbol{\Sigma}_{21}\mathbf{V} - \boldsymbol{\Pi}_2, \quad \mathbf{T} = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\boldsymbol{\Lambda}_{y2}\mathbf{W}, \quad \mathbf{D} = \mathbf{T}\boldsymbol{\Psi}\boldsymbol{\Pi}', \tag{A8}$$

$$\mathbf{E} = \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\mathbf{S}_1 - \mathbf{S}_2, \quad \mathbf{F} = \boldsymbol{\Lambda}_{y2}\mathbf{E}\boldsymbol{\Pi}, \quad \mathbf{G} = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\mathbf{F}, \quad \mathbf{H} = \mathbf{V}\boldsymbol{\Psi}, \quad \mathbf{M} = \boldsymbol{\Sigma}_1^{-1}\mathbf{S}_1\boldsymbol{\Pi}, \tag{A9}$$

$$\tilde{\boldsymbol{\Psi}} = -\mathbf{T}'\boldsymbol{\Lambda}_{y2}\mathbf{W} - \boldsymbol{\Pi}_1'\mathbf{V}, \quad \text{and} \quad \tilde{\mathbf{B}} = -\mathbf{F}'\mathbf{D} - \boldsymbol{\Pi}'\mathbf{S}_1'\mathbf{H}\boldsymbol{\Pi}'. \tag{A10}$$

Next, define the following quantities that do depend on the data and/or quadrature nodes (i.e., $\boldsymbol{\eta}_{1i} = \tilde{\mathbf{t}}_q$):

$$\mathbf{h} = \tilde{\mathbf{t}}_q - \boldsymbol{\mu}_1, \quad \mathbf{r} = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \mathbf{H}\mathbf{h}, \quad \mathbf{e}_{\mathbf{y}} = \mathbf{y}_i - \boldsymbol{\mu}_{\mathbf{y}\bullet 1}, \quad \mathbf{s} = \mathbf{T}'\mathbf{e}_{\mathbf{y}}, \tag{A11}$$

$$\mathbf{U} = \mathbf{s}\mathbf{h}'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Pi}_1, \quad \text{and} \quad \mathbf{w} = \mathbf{V}\mathbf{h}'. \tag{A12}$$

Using these expressions, the first-order derivatives of the complete data log-likelihood with respect to the various parameters that do not depend on the conditional distribution of $\mathbf{u}_i$ follow as:

$$\frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\nu}_y} = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\mathbf{e}_{\mathbf{y}}, \quad \frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \mathbf{K}_y} = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\mathbf{e}_{\mathbf{y}}\mathbf{x}_i', \quad \frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\Lambda}_{y1}} = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\mathbf{e}_{\mathbf{y}}\tilde{\mathbf{t}}_q', \tag{A13}$$

$$\frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\Lambda}_{y2}} = -\mathbf{Z} + \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\mathbf{e}_{\mathbf{y}}(\boldsymbol{\mu}_{2\bullet 1}' - \mathbf{e}_{\mathbf{y}}'\mathbf{Z}), \tag{A14}$$

$$\frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\alpha}} = -(\mathbf{s} - \mathbf{w}), \quad \frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\Gamma}} = -(\mathbf{s} - \mathbf{w})\mathbf{x}_i', \tag{A15}$$

$$\frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \mathbf{B}} = \tilde{\mathbf{B}} + (\mathbf{s} - \mathbf{M}'\mathbf{h})\mathbf{e}_{\mathbf{y}}'\mathbf{D} + (\mathbf{M}'\mathbf{h} - \mathbf{G}'\mathbf{e}_{\mathbf{y}})\mathbf{r}'\boldsymbol{\Pi}' \tag{A16}$$

$$\frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\Theta}} = \boldsymbol{\Theta}^* - .5(\mathbf{I} \circ \boldsymbol{\Theta}^*), \quad \text{and} \quad \frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\Psi}} = \boldsymbol{\Psi}^* - .5(\mathbf{I} \circ \boldsymbol{\Psi}^*), \tag{A17}$$

where

$$\boldsymbol{\Theta}^* = \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1} - \boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}\mathbf{e}_{\mathbf{y}}\mathbf{e}_{\mathbf{y}}'\boldsymbol{\Sigma}_{\mathbf{y}\bullet 1}^{-1}, \quad \text{and} \quad \boldsymbol{\Psi}^* = \tilde{\boldsymbol{\Psi}} + \mathbf{s}\mathbf{s}' - \mathbf{U} - \mathbf{U}' + \mathbf{w}\mathbf{w}'. \tag{A18}$$

The derivatives for $\boldsymbol{\nu}_u$, $\boldsymbol{\Lambda}_{u1}$, and $\mathbf{K}_u$ depend on $f(\mathbf{u}_i|\boldsymbol{\eta}_{1i})$ and so will change with different model specifications. In general,

$$\frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\nu}_u} = \sum_j \frac{\partial f(u_{ij}|\boldsymbol{\eta}_{1i})}{\partial \boldsymbol{\nu}_u}, \quad \frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\Lambda}_{u1}} = \sum_j \frac{\partial f(u_{ij}|\boldsymbol{\eta}_{1i})}{\partial \boldsymbol{\Lambda}_{u1}}, \quad \text{and} \quad \frac{\partial l_{Ci}(\boldsymbol{\vartheta})}{\partial \mathbf{K}_u} = \sum_j \frac{\partial f(u_{ij}|\boldsymbol{\eta}_{1i})}{\partial \mathbf{K}_u}. \tag{A19}$$

When u_{ij} is conditionally distributed as a Poisson random variable with mean $\exp(\xi_{ij})$, then

$$\frac{\partial f(u_{ij}|\boldsymbol{\eta}_{1i})}{\partial \boldsymbol{\nu}_{u[j]}} = u_{ij} - \exp(\xi_{ij}), \quad \frac{\partial f(u_{ij}|\boldsymbol{\eta}_{1i})}{\partial \boldsymbol{\Lambda}_{u1[j]}} = [u_{ij} - \exp(\xi_{ij})]\tilde{\mathbf{t}}_q', \quad \text{and} \quad \frac{\partial f(u_{ij}|\boldsymbol{\eta}_{1i})}{\partial \mathbf{K}_{u[j]}} = [u_{ij} - \exp(\xi_{ij})]\mathbf{x}_i', \tag{A20}$$

where $\boldsymbol{\nu}_{u[j]}$, $\boldsymbol{\Lambda}_{u1[j]}$, and $\mathbf{K}_{u[j]}$ are the jth rows of $\boldsymbol{\nu}_u$, $\boldsymbol{\Lambda}_{u1}$, and $\mathbf{K}_u$, respectively.

Appendix C

A general method for reparameterizing any specified model to minimize the dimension of numerical integration required is discussed in this section. Recall that, without the dimension reduction technique described in this paper, the dimension of numerical integration required for a generalized SEM is equal to the number of latent factors, m. In this paper, it was demonstrated that the dimension can be reduced from m to m_1, where m_1 is the number of latent factors that directly affect the nonnormal response variables $\mathbf{u}_i$. These latent factors can be determined via the columns corresponding to $\boldsymbol{\Lambda}_{u1}$ in Equation 10. Specifically, all columns within $\boldsymbol{\Lambda}$ for which there is at least one nonzero entry in a row corresponding to $\boldsymbol{\xi}_i$ would indicate the latent factors that must be numerically integrated (i.e., $\boldsymbol{\eta}_{1i}$).

In Section 4.2.1 it was demonstrated that the number of factors with nonzero $\boldsymbol{\Lambda}$ entries in the rows corresponding to $\boldsymbol{\xi}_i$ could be reduced by using a phantom variable for each nonnormal response (or each component of a nonnormal response, such as the zero and count components of a ZIP model). Using a phantom variable for each component of a nonnormal response (i.e., each element of $\boldsymbol{\xi}_i$) would result in a total dimension of integration equal to r. Thus, the maximum number of latent factors that must be numerically integration is $\min(m_1, r)$. That is, the minimum of either the original number of latent factors with direct effects on nonnormal responses or the number of phantom variables needed for each of the (components for the) nonnormal responses. In the latent growth factors as outcomes model (Section 4.1.1) the minimum was $m_1 = 2$ (since $r = 4$), whereas in the Poisson regression with latent predictors and overdispersion model (Section 4.2.1) the minimum was $r = 1$ (since $m_1 = 5$).

For other models, using a phantom variable for some, but not all, components of nonnormal responses may result in a dimension of integration fewer than both m_1 and r. For example, consider if the indicators for the Norm_i factor in the latent regression model with overdispersion displayed in Figure 2 were nonnormal. The original parameterization (top of Figure 2) would require five dimensions of numerical integration, corresponding to each of the latent factors ($m_1 = 5$). If a phantom factor was used for all nonnormal response variables (i.e., the four Norm_i indicators and HED_{4i}), the model would still require five dimensions ($r = 5$). If, on the other hand, a phantom factor was only used for HED_{4i} (bottom of Figure 2), then the model would only require two dimensions of numerical integration: one for the Norm_i factor, as it has direct effects on its nonnormal indicator variables, and one for the Disp_i factor, which acts as a phantom factor for HED_{4i}. Using software such as Mplus, which uses the dimension reduction technique based on m_1 but does not automatically reparameterize the model using phantom variables, the user specifying the model should determine which parameterization would result in the fewest latent variables directly influencing nonnormal responses, so that the dimension of numerical integration required is minimized.

Although determining the proper reparameterization required to minimize the dimension of numerical integration required may seem complex, it is actually possible to determine the correct reparameterization using a basic set of steps:

1. If the original parameterization of a nonnormal response variable already has a single-indicator latent factor (e.g., a dispersion factor), all direct effects of other latent factors on the nonnormal factor should be reparameterized as direct effects on the single-indicator latent factor, which serves as a phantom variable.
2. After any reparameterizations resulting from (1) above, define $\boldsymbol{\Lambda}^*_{u1}$ to be a zero-one

matrix where an element is zero if the corresponding element in $\mathbf{\Lambda}_{u1}$ is also zero and one otherwise (i.e., the element in $\mathbf{\Lambda}_{u1}$ is either a free parameter or a fixed non-zero parameter).

3. Determine each unique row pattern within $\mathbf{\Lambda}^*_{u1}$.
4. For each unique row pattern within $\mathbf{\Lambda}^*_{u1}$:
   (a) Compute the number of nonnormal components (i.e., elements of $\boldsymbol{\xi}_i$) that have the given row pattern (and denote as r^*).
   (b) Compute the number of latent factors with non-zero effects within the pattern (and denote as m^*).
5. The latent factors within all patterns in which the number of latent factors is less than or equal to the number of nonnormal components (i.e., $m^* \leq r^*$) will be numerically integrated.
6. For the remaining patterns, reduce the number of factors to those not included in the list of factors numerically integrated from (5) above (and denote as m^{**}).
7. For the remaining patterns from (6), if the number of nonnormal components is less than the (reduced) number of latent factors (i.e., $r^* < m^{**}$), these nonnormal components should be parameterized using a phantom variable.

To demonstrate the procedure, consider the latent growth factors as outcomes model (Figure 1). Suppose that N_{1i} to N_{4i} were actually nonnormal responses and N_4 cross-loaded onto The Norm_i, Att_i and Cont_i latent factors. Since there are no single-indicator factors, Step One can be skipped. Moving to Step Two,

$$\mathbf{\Lambda}_{u1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \lambda_{63} & 0 & 0 \\ 0 & 0 & \lambda_{73} & 0 & 0 \\ 0 & 0 & \lambda_{83} & \lambda_{84} & \lambda_{85} \end{pmatrix}, \quad \text{and so} \quad \mathbf{\Lambda}^*_{u1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \qquad \text{(A21)}$$

Following Step 3, the four unique row patterns within $\mathbf{\Lambda}^*_{u1}$ are determined and displayed in the first five columns of Table 4. The column labeled r^* contains the number of elements of $\boldsymbol{\xi}_i$ (i.e., the number of rows within $\mathbf{\Lambda}^*_{u1}$) that is represented by each pattern, and the column labeled m^* contains the number of latent factors with non-zero elements within the specific row pattern (i.e., Step 4). The second and third rows corresponds to patterns with fewer latent factors than nonnormal components (i.e., $m^* < r^*$) and so the latent factors with non-zero effects within these patterns will be numerically integrated (Step Five). These include the Intercept_i and Slope_i factors for the second pattern, and the Norm_i factor for the third pattern. The non-zero elements corresponding to these factors are presented in bold within the table.

[Table 4 about here.]

Next, following Step Six, the remaining number of factors (i.e., non-bold elements) are calculated for each of the remaining patterns (patterns one and four). Since Intercept_i and Norm_i will already be numerically integrated based on Step Five, the remaining number of factors for patterns one and four are zero and two, respectively. Finally, following Step Seven, it is determined that the single nonnormal component within pattern one should not be represented using a phantom factor (since $r^* > m^{**}$), whereas the single nonnormal component within pattern four should be reparameterized with a phantom factor (since $r^* < m^{**}$). This nonnormal component corresponds to N_{4i}, which is the indicator variable that loads onto the Norm_i, Att_i, and Cont_i latent factors. The total dimension of numerical integration required for parameter estimation is equal to four, which is the sum of the bold numbers in the final three columns of Table 4. The factors that must be numerically integrated are Intercept_i, Slope_i, Norm_i, and the phantom factor for N_{4i}.

A major advantage of this approach is that it can be implemented within the estimation routine, rather than requiring the user to conduct the reparameterization by hand before model specification. That is, the original parameterization could be used as input from the user, and the estimation routine could first algorithmically reparameterize the model following the steps outlined above and then estimate the parameters from the reparameterized model, resulting in the dimension of numerical integration required being minimized.

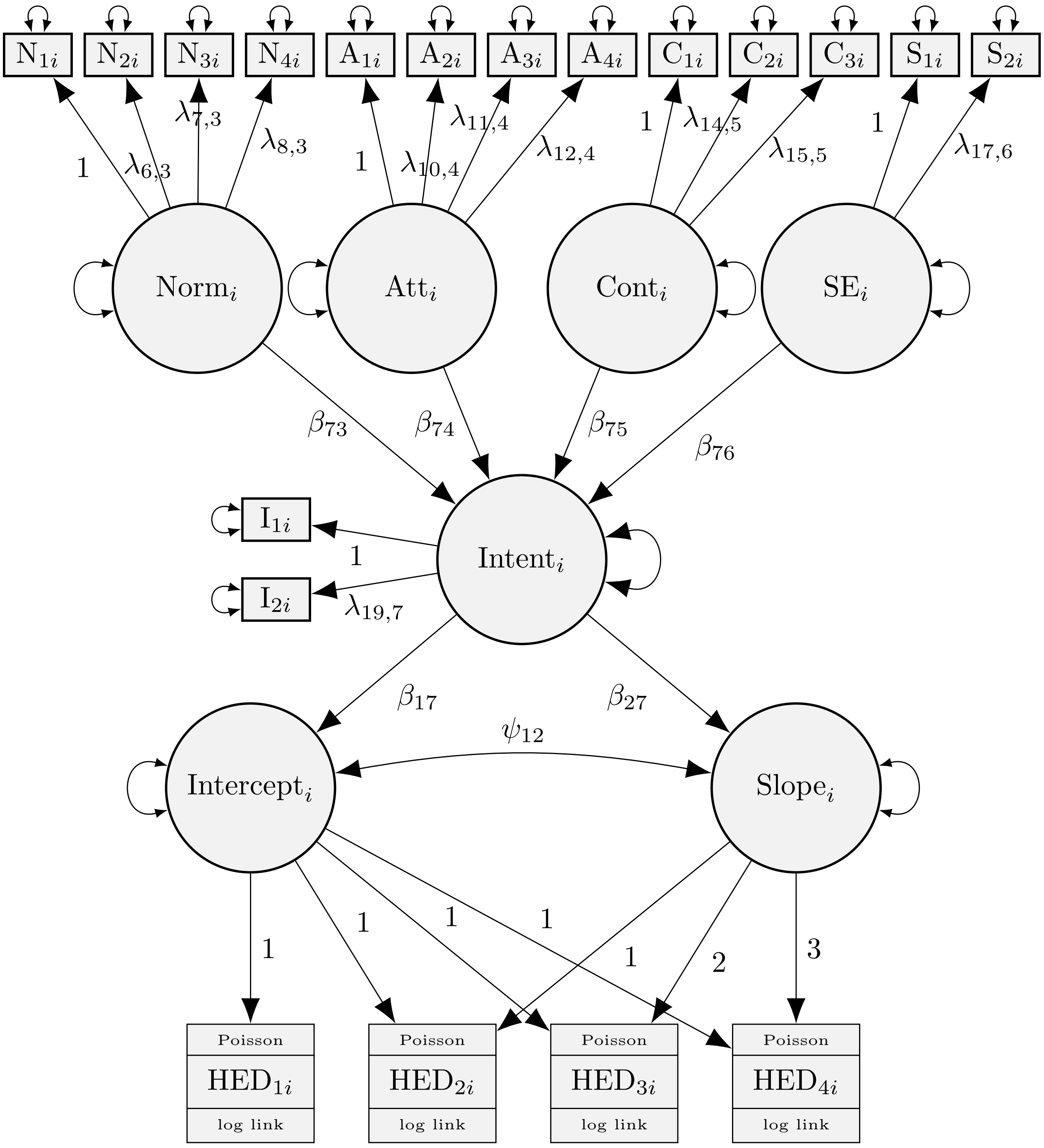

Figure 1.
A path diagram corresponding to the full structural model with latent growth factors as outcomes. To reduce clutter, item and factor intercepts and means, as well as the covariances among the four exogenous factors, have been omitted.

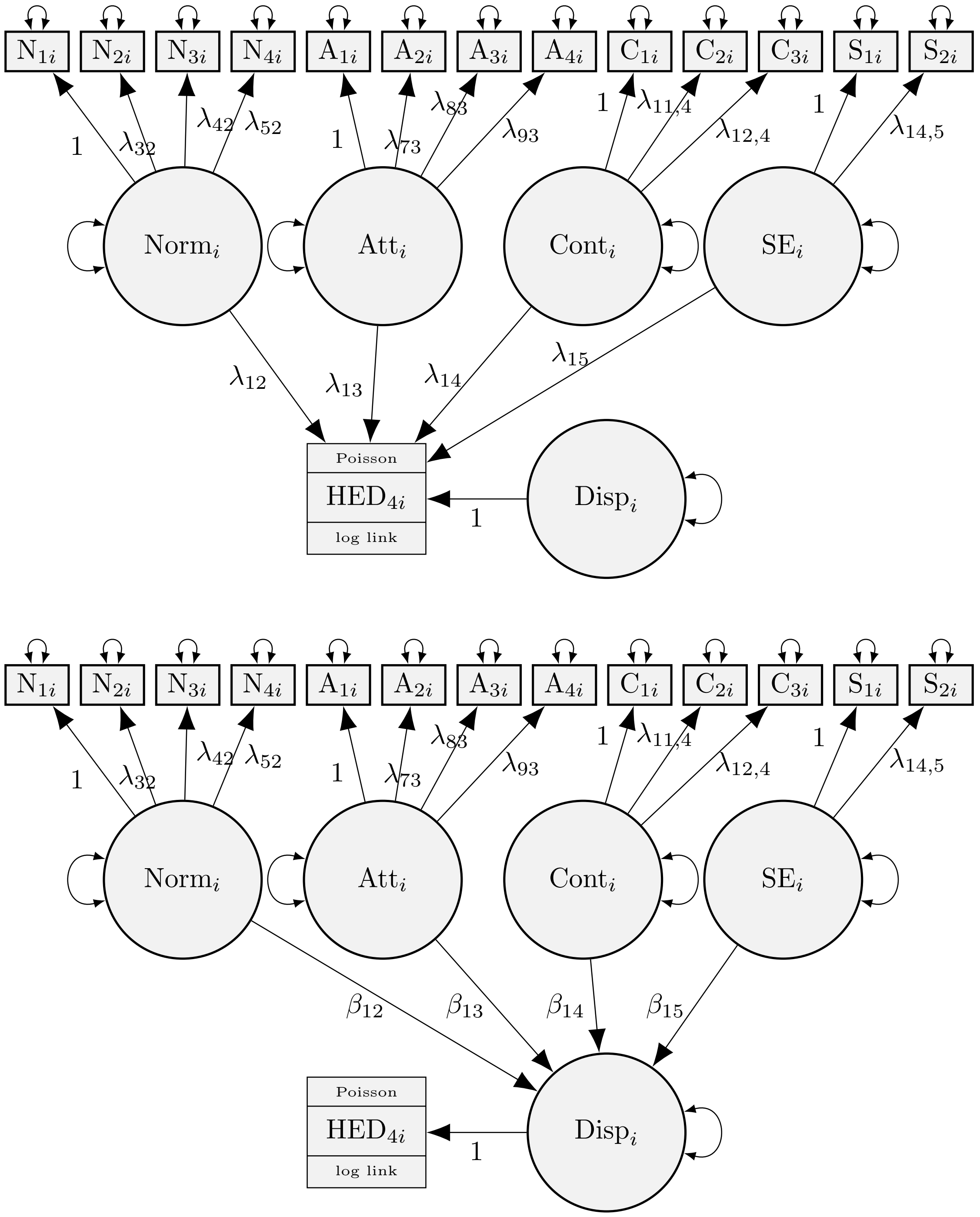

Figure 2.

Path diagrams corresponding to Poisson regression model with latent predictors and overdispersion. Although both models are equivalent, the top parameterization requires five dimensions of numerical integration, whereas the bottom parameterization requires only one.

		Dimension = 2		Dimension = 7	
Effect	Parameter	New (SE)	Mplus (SE)	New (SE)	Mplus (SE)
Norm$_2$ Loading	λ_{63}	1.010 (0.013)	1.010 (0.013)	1.010 (0.013)	1.010 (0.013)
Norm$_3$ Loading	λ_{73}	0.789 (0.010)	0.789 (0.010)	0.789 (0.010)	0.789 (0.010)
Norm$_4$ Loading	λ_{83}	0.795 (0.011)	0.795 (0.011)	0.795 (0.011)	0.795 (0.011)
Att$_2$ Loading	$\lambda_{10,4}$	0.995 (0.022)	0.995 (0.022)	0.999 (0.022)	0.996 (0.022)
Att$_3$ Loading	$\lambda_{11,4}$	0.660 (0.021)	0.660 (0.021)	0.662 (0.021)	0.660 (0.021)
Att$_4$ Loading	$\lambda_{12,4}$	0.737 (0.021)	0.737 (0.021)	0.739 (0.021)	0.738 (0.021)
Cont$_2$ Loading	$\lambda_{14,5}$	1.042 (0.022)	1.042 (0.022)	1.042 (0.022)	1.043 (0.021)
Cont$_3$ Loading	$\lambda_{15,5}$	0.997 (0.024)	0.997 (0.024)	0.996 (0.024)	0.997 (0.024)
SE$_2$ Loading	$\lambda_{17,6}$	0.572 (0.051)	0.572 (0.049)	0.572 (0.051)	0.558 (0.040)
Intent$_2$ Loading	$\lambda_{19,7}$	1.039 (0.027)	1.039 (0.027)	1.039 (0.027)	1.041 (0.027)
Intercept(Intercept)	α_1	0.203 (0.031)	0.200 (0.031)	0.208 (0.026)	0.201 (0.028)
Intercept(Slope)	α_2	−0.053 (0.017)	−0.052 (0.017)	−0.055 (0.016)	−0.052 (0.016)
Norm → Intent	β_{73}	0.108 (0.033)	0.108 (0.033)	0.109 (0.033)	0.108 (0.033)
Att → Intent	β_{74}	0.440 (0.020)	0.440 (0.020)	0.441 (0.020)	0.441 (0.020)
Cont → Intent	β_{75}	−0.066 (0.036)	−0.066 (0.036)	−0.067 (0.036)	−0.067 (0.035)
SE → Intent	β_{76}	−0.172 (0.052)	−0.172 (0.051)	−0.172 (0.052)	−0.166 (0.049)
Intent → Intercept	β_{17}	0.152 (0.019)	0.152 (0.020)	0.153 (0.019)	0.153 (0.019)
Intent → Slope	β_{27}	0.036 (0.011)	0.036 (0.011)	0.036 (0.011)	0.036 (0.010)
Var(Intercept)	ψ_{11}	0.041 (0.027)	0.045 (0.027)	0.027 (*)	0.043 (0.014)
Var(Slope)	ψ_{22}	0.016 (0.008)	0.016 (0.008)	0.014 (*)	0.018 (0.005)
Var(Norm)	ψ_{33}	1.256 (0.067)	1.256 (0.066)	1.256 (0.067)	1.256 (0.066)
Var(Att)	ψ_{44}	5.676 (0.354)	5.676 (0.339)	5.571 (0.343)	5.665 (0.338)
Var(Cont)	ψ_{55}	1.071 (0.064)	1.071 (0.064)	1.071 (0.064)	1.070 (0.063)
Var(SE)	ψ_{66}	0.793 (0.083)	0.793 (0.077)	0.792 (0.083)	0.812 (0.068)
Var(Intent)	ψ_{77}	0.759 (0.055)	0.759 (0.055)	0.760 (0.055)	0.758 (0.054)
Cor(Intercept, Slope)	ρ_{12}	0.181 (0.539)	0.138 (0.521)	0.568 (*)	0.081 (0.239)
Cor(Norm, Att)	ρ_{34}	0.263 (0.035)	0.263 (0.033)	0.262 (0.035)	0.263 (0.033)
Cor(Norm, Cont)	ρ_{35}	0.010 (0.036)	0.010 (0.036)	0.010 (0.036)	0.010 (0.036)
Cor(Norm, SE)	ρ_{36}	0.071 (0.038)	0.071 (0.037)	0.071 (0.038)	0.071 (0.037)
Cor(Att, Cont)	ρ_{45}	−0.128 (0.036)	−0.128 (0.036)	−0.128 (0.036)	−0.128 (0.036)
Cor(Att, SE)	ρ_{46}	−0.372 (0.037)	−0.372 (0.035)	−0.370 (0.037)	−0.369 (0.034)
Cor(Cont, SE)	ρ_{56}	0.202 (0.038)	0.202 (0.038)	0.202 (0.038)	0.198 (0.037)
Log-likelihood		−19656.32	−19656.34	−19656.51	−19656.32
Parameters		62	62	62	62
Dim. of integration		2	2	7	7
Nodes per dim.		10	10	4	6
Total nodes		100	100	16384	279936

Table 1.
Factor loadings and structural parameter estimates (standard errors) from the model with latent growth factors as outcomes. A * indicates that the standard error could not be computed.

Effect	Parameter	R/C++ (SE)	Mplus (SE)	gsem (SE)
Norm_2 Loading	λ_{32}	1.010 (0.013)	1.010 (0.013)	1.010 (0.013)
Norm_3 Loading	λ_{42}	0.789 (0.010)	0.789 (0.010)	0.789 (0.010)
Norm_4 Loading	λ_{52}	0.795 (0.011)	0.795 (0.011)	0.795 (0.011)
Att_2 Loading	λ_{73}	0.994 (0.022)	0.994 (0.022)	0.994 (0.022)
Att_3 Loading	λ_{83}	0.657 (0.021)	0.657 (0.021)	0.657 (0.021)
Att_4 Loading	λ_{93}	0.736 (0.021)	0.736 (0.021)	0.736 (0.021)
Cont_2 Loading	$\lambda_{11,4}$	1.042 (0.022)	1.043 (0.022)	1.042 (0.022)
Cont_3 Loading	$\lambda_{12,4}$	0.997 (0.024)	0.997 (0.024)	0.997 (0.025)
SE_2 Loading	$\lambda_{14,5}$	0.555 (0.053)	0.554 (0.050)	0.555 (0.050)
Norm → Disp/HED	β_{12}/λ_{12}	−0.031 (0.035)	−0.031 (0.035)	−0.031 (0.035)
Att → Disp/HED	β_{13}/λ_{13}	0.126 (0.019)	0.126 (0.019)	0.126 (0.019)
Cont → Disp/HED	β_{14}/λ_{14}	0.066 (0.037)	0.066 (0.037)	0.066 (0.037)
SE → Disp/HED	β_{15}/λ_{15}	−0.100 (0.051)	−0.099 (0.051)	−0.100 (0.051)
Var(Disp)	ψ_{11}	0.260 (0.053)	0.259 (0.053)	0.260 (0.053)
Var(Norm)	ψ_{22}	1.256 (0.067)	1.256 (0.066)	51.256 (0.066)
Var(Att)	ψ_{33}	5.694 (0.355)	5.695 (0.340)	5.694 (0.340)
Var(Cont)	ψ_{44}	1.071 (0.064)	1.071 (0.064)	1.071 (0.064)
Var(SE)	ψ_{55}	0.817 (0.089)	0.817 (0.083)	0.817 (0.082)
Cor(Norm, Att)	ρ_{23}	0.263 (0.035)	0.263 (0.033)	0.263
Cor(Norm, Cont)	ρ_{24}	0.010 (0.036)	0.010 (0.036)	0.010
Cor(Norm, SE)	ρ_{25}	0.071 (0.037)	0.071 (0.037)	0.071
Cor(Att, Cont)	ρ_{34}	−0.128 (0.036)	−0.128 (0.036)	−0.128
Cor(Att, SE)	ρ_{35}	−0.369 (0.037)	−0.368 (0.035)	−0.369
Cor(Cont, SE)	ρ_{45}	0.197 (0.038)	0.197 (0.038)	0.197
Log-likelihood		−13729.78	−13729.78	−13729.76
Parameters		51	51	51
Dim. of integration		1	1	5
Nodes per dim.		10	10	5
Total nodes		10	10	3125

Table 2.

Factor loadings and structural parameter estimates (standard errors) from the Poisson regression model with latent predictors and overdispersion. For the gsem function, correlations were computed using the estimated variances and covariances. Consequently, standard errors for correlations obtained using gsem are not reported.

Dimension	Total Nodes	R/C++	Mplus	gsem
1	5	22 s	1 s	–
2	25	1.37 min	3 s	–
3	125	5.92 min	14 s	–
4	625	30.11 min	1.00 min	–
5	3125	2.73 hr	DNC	3.77 h

Table 3.
Time required to estimate the parameters of the Poisson latent regression model using five adaptive quadrature nodes per dimension. DNC = Did Not Converge.

Intercept$_i$	Slope$_i$	Norm$_i$	Att$_i$	Cont$_i$	r^*	m^*	m^{**}
1	0	0	0	0	1	1	**0**
1	**1**	0	0	0	3	**2**	–
0	0	**1**	0	0	3	**1**	–
0	0	**1**	1	1	**1**	3	2

Table 4.
The unique row patterns from $\boldsymbol{\Lambda}^*_{u1}$, and the corresponding quantities needed for reparameterizing the model to minimize the dimension of numerical integration required.