Products of Variables in Structural Equation Models

Steven Boker

The University of Virginia, Charlottesville

Timo von Oertzen

University of the Bundeswehr, Munich

Joshual Pritikin

MakerDAO Immunefi Security

Michael D. Hunter, Timothy Brick

The Pennsylvania State University

Andreas Brandmaier

Max Planck Institute for Human Development, Berlin

MSB Medical School Berlin, Berlin

Michael Neale

Virginia Commonwealth University

Draft September 2, 2022

Author Note

Funding for this work was provided in part by NIH Grant R01DA018673, the Max Planck Institute for Human Development, and a fellowship from the University of Zurich Research Priority Program on Dynamics of Healthy Aging. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health. The authors would also like to acknowledge Robert M. Kirkpatrick for his helpful advice on the Monte Carlo simulations reported in this manuscript and the OpenMx development team for their support in user interface design. Correspondence may be addressed to Steven M. Boker, Department of Psychology, The University of Virginia, PO Box 400400, Charlottesville, VA 22903, USA; email sent to boker@virginia.edu; or browsers pointed to https://psychology.as.virginia.edu/people/profile/smb3u

Abstract

A general method is introduced in which variables that are products of other variables in the context of a structural equation model (SEM) can be decomposed into the sources of variance due to the multiplicands. The result is a new category of SEM which we call a Products of Variables Model (PoV). Some useful and practical features of PoV models include estimation of interactions between latent variables, latent variable moderators, manifest moderators with missing values, and manifest or latent squared terms. Expected means and covariances are analytically derived for a simple product of two variables and it is shown that the method reproduces previously published results for this special case. It is shown algebraically that using centered multiplicands results in an unidentified model, but if the multiplicands have non-zero means, the result is identified. The method has been implemented in OpenMx and Ωnyx and is applied in five extensive simulations.

Products of Variables in Structural Equation Models

Introduction

The expected covariance matrix of a Structural Equation Model (SEM) which is composed of a linear combination of variables has long been known (Wright, 1921) as well as its equivalence to a method of path coefficients (Wright, 1934). This equivalence was formalized into what is called the Reticular Action Model (RAM) by McArdle and McDonald (1984). Although linear combinations of variables are powerful and useful, nonlinear models have become increasingly important as latent interaction models, latent moderation models, and nonlinear dynamical systems models have become necessary in the current data-rich research environment.

When a variable in an SEM is the outcome of the product of two variables and at least one of the product terms is a manifest variable, there are a variety of methods that can be used to estimate coefficients in the SEM (e.g., Mehta & Neale, 2005; Neale, 1998). But when both of the product terms are latent, the problem becomes more difficult. Methods have been proposed that require nonlinear constraints on measurement models (the product indicant technique, Kenny & Judd, 1984), approximating non-normal distributions using mixture distributions (the latent moderated structure technique, Klein & Moosbrugger, 2000; Moosbrugger, Schermelleh-Engel, & Klein, 1997), multiple group approaches and estimating latent scores so that they can be dealt with as manifest variables (Schumacker, 2002), and variations on unconstrained approaches (unconstrained and quasi-maximum likelihood techniques, Marsh, Wen, & Hau, 2004; Marsh et al., 2007), the approach proposed and elaborated by Wall and Amemiya (Wall & Amemiya, 2001, 2003), and semiparametric approaches introduced by Bauer, (Bauer, 2005; Bauer & Hussong, 2009). This is only a small sample of the large literature on nonlinear SEM, but most of these methods apply only in limited cases or are difficult to implement in current SEM software.

In the current article we take a different approach and present a novel method for the

decomposition and estimation of the variance and covariance of products of variables in SEM models specified in RAM notation. The approach presented here makes the assumption that the non-normality of a variable constructed as the product of other variables can be completely accounted for if a model can account for all sources of variance that are included in the product and that those sources of variance are multivariate normally distributed. The method is currently implemented in OpenMx (Neale et al., 2016) and Ωnyx (Oertzen, Brandmaier, & Tsang, 2015) and requires only a one line change in the R model script.

We begin our argument by providing an extension to RAM path diagrams (see Boker & McArdle, 2005, for an introduction to RAM matrices and path diagrams) in order to show how the specification of these models can be a minimal change from current user interfaces. By adding one new node to the pallette of RAM path diagrams, we can add products of variables into existing SEM models. We hope that authors of other SEM packages will find that this method is straightforward to implement and has many advantages for end users.

To start, consider the construction of a linear combination: estimated coefficients are constants that are multiplied by variables and then all of these products are summed together and assigned to an outcome variable. For instance, a simple bivariate regression using mean centered multivariate normal variables can be written as,

$$z_i = b_1 x_i + b_2 y_i + e_i \tag{1}$$

where x_i and y_i are the predictor variables and e_i is the residual at each row index i, and b_1, b_2, and 1 are constants. If the variance of x, y, and e are V_x, V_y, and V_e, respectively and the covariance between x and y is C_{xy}, a RAM path diagram isomorphic to this bivariate regression can be drawn as in Figure 1-a. Compare Equation 1 with the path diagram and note that two things are happening when the three one-headed arrows attach to the box representing the manifest variable z. In the equation, $b_1 x_i$, $b_2 y_i$, and e_i are first summed. Second, the result of that sum is assigned to the variable z_i. In the equation,

summation is represented by one symbol, "+", and assignment is represented by another symbol, "=". The path diagram fuses together these two symbols from the equation. This is not a problem as long as *only* linear combinations are used. In that case, summation and assignment are always used together. However, if one wishes to use an n-ary[1] operator other than summation, the assignment operator and the n-ary operator must be disambiguated in the path diagram.

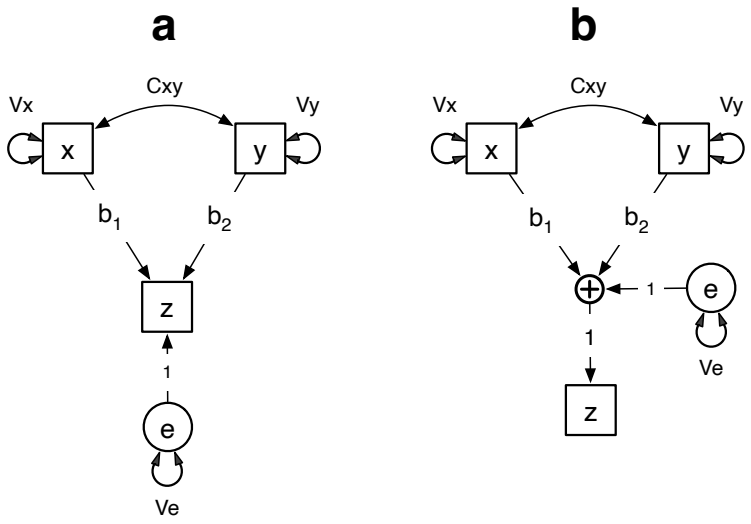

Figure 1. Two equivalent path diagrams of the bivariate regression in Equation 1. (a) Standard RAM path diagram of a bivariate regression. (b) Equivalent path diagram where the n-ary operator addition, represented by $\oplus$, is disambiguated from the assignment operator.

In Figure 1-b, a new n-ary operator path diagram node is proposed: the summation operator is represented by a plus sign within a circle. If this summation operator is treated just like a latent variable with zero unique variance, then the algorithmic application of the RAM path tracing rules (Boker, McArdle, & Neale, 2002) will correctly produce the components of covariance of the expected covariance matrix of any SEM model that is

[1] Operators often take one or more elements from a set back onto that set. A unary operator maps one member of a set onto the set; whereas a binary operator maps two elements of a set onto that set, and a n-ary operator maps n elements onto that set.

composed solely of linear combinations of variables. Note that in Figure 1-b, the summation operator could have been mixed together with standard addition-and-assignment while the standard RAM tracing rules would continue to produce the correct components of covariance. In many published SEM path diagrams, the plus sign in a circle is drawn without the plus sign and is called a "dummy variable" whose only purpose is to sum its input arrows and send the sum to any output arrows. We suggest that "dummy variables" are actually summation operator nodes since they do not have any residual variance.

Now that the assignment operator has been distinguished from the summation operator, it is possible to propose a new n-ary operation node: multiplication. Consider the simplest case, a product of two predictor variables,

$$z_i = b_1 x_i y_i + e_i \ , \tag{2}$$

where the variables x and y are multivariate normal with mean zero and variances V_x, V_y and covariance C_{xy}, and the residual e is independent normally distributed with mean zero and variance V_e.

A path model for Equation 2 is displayed in Figure 2-a, where the n-ary operator multiplication is represented by an asterisk within a circle. Although the summation operator could be treated as if it were a latent variable with zero residual variance, the multiplication operator cannot, as will be demonstrated below. Thus, we will need to specify that an n-ary operator is a fourth type of node in a path diagram: i) squares represent manifest variables; ii) circles represent latent variables; iii) triangles represent constants for means and intercepts; and iv) n-ary operators are represented by a small circle surrounding the selected operator symbol. This distinction is necessary for two reasons that will be described in more detail later in the article: different n-ary operators have different identity operations and also trigger different path tracing rules.

Diagramming the problem in this way has numerous advantages. Since scalar multiplication is commutative, neither the vector $\mathbf{x}$ nor $\mathbf{y}$ has a privileged place in the

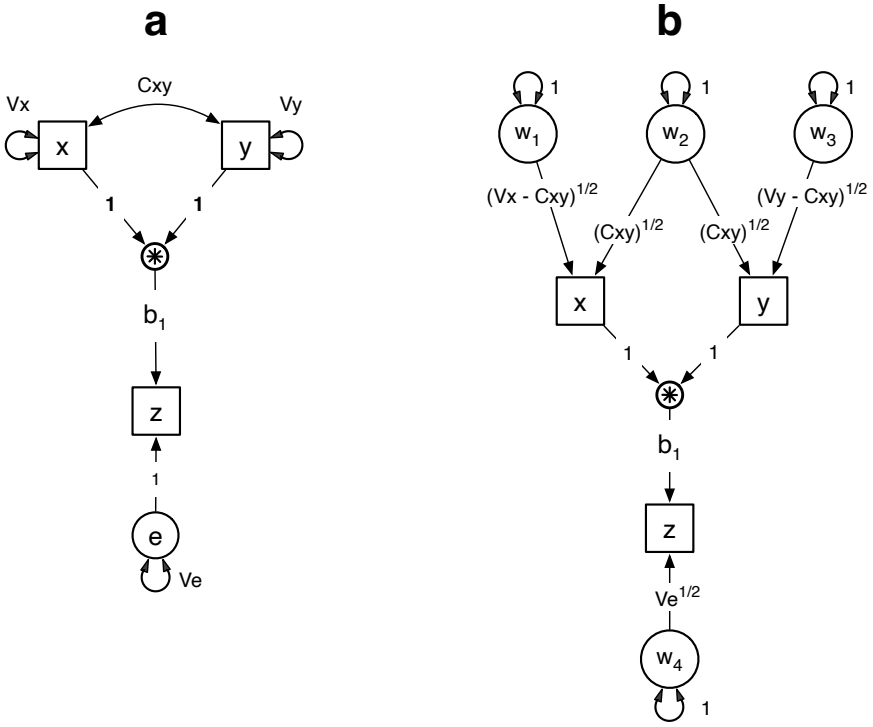

Figure 2. Path diagrams of a product of two variables. The asterisk surrounded by a circle represents the product of the variables connected to it by incoming arrows. (a) Mean-centered variables with a single product term. (b) Equivalent path diagram with variances sources isolated to be independent normally distributed variables with mean zero and variance one.

multiplication, just as one would expect algebraically. This is not explicit in some diagrammatic representations for moderation where a moderating variable is singled out. This might be perceived as just a philosophical difference. However, placing both multiplicands of a commutative binary operation on equal visual footing clarifies that moderation is just interaction without a direct effect of the moderating variable and emphasizes that the SEM network of variables can be perturbed at either multiplicand with an equivalent downstream effect. In addition, by diagramming the model in this way, it sets up how the user interface for specifying SEM models that include products of variables can be implemented as a minimal change from current SEM conventions.

Variance of the Product of Two Variables

Given two vectors $\mathbf{x}$ and $\mathbf{y}$, each with N elements, Goodman (1960) first derived the variance of the product of two variables and showed that this was an unbiased estimator. If $\Delta\mathbf{x} = \mathbf{x} - \mu(\mathbf{x})$ and $\Delta\mathbf{y} = \mathbf{x} - \mu(\mathbf{x})$, the exact variance is given by

$$\begin{aligned} \text{var}(\mathbf{xy}) &= \text{cov}(\Delta\mathbf{x}^2, \Delta\mathbf{y}^2) + \text{var}(\mathbf{x})\text{var}(\mathbf{y}) + \mu(\mathbf{x})^2\text{var}(\mathbf{y}) + \mu(\mathbf{y})^2\text{var}(\mathbf{x}) + \\ &\quad (\mu(\mathbf{x})\mu(\mathbf{y}))^2 + 2\mu(\mathbf{x})\mu(\mathbf{y})\text{cov}(\mathbf{x}, \mathbf{y}) + 2\mu(\mathbf{x})\text{cov}(\Delta\mathbf{x}, \Delta\mathbf{y}^2) + \\ &\quad 2\mu(\mathbf{y})\text{cov}(\Delta\mathbf{x}^2, \Delta\mathbf{y}) - \text{cov}(\mathbf{x}, \mathbf{y})^2 \ . \end{aligned} \tag{3}$$

Bohrnstedt and Goldberger (1969) applied this result to the bivariate normal case and showed that the variance of the product, var($\mathbf{xy}$), can be estimated as (Bohrnstedt & Marwell, 1978, Equation 15)

$$\begin{aligned} \text{var}(\mathbf{xy}) &= \mu(\mathbf{x})^2\text{var}(\mathbf{y}) + \mu(\mathbf{y})^2\text{var}(\mathbf{x}) + 2\mu(\mathbf{x})\mu(\mathbf{y})\text{cov}(\mathbf{x}, \mathbf{y}) + \\ &\quad \text{var}(\mathbf{x})\text{var}(\mathbf{y}) + \text{cov}(\mathbf{x}, \mathbf{y})^2 \ . \end{aligned} \tag{4}$$

Bohrnstedt and Marwell (1978) also derived reliability for this estimator when the observed product variable included a normally distributed error term.

Now consider the regression equation

$$z_i = b_1 x_i y_i + e_i \tag{5}$$

where $\mathbf{x}$ and $\mathbf{y}$ are mean centered, multivariate normal, and the vector of residuals, $\mathbf{e}$, is mean centered, normally distributed, and independent of the predictor variables $\mathbf{x}$ and $\mathbf{y}$. This may be drawn as a path diagram as shown in Figure 2-a. A new set of tracing rules will be required when this product symbol is used. However, note that the total variance at the product symbol can be estimated from Equation 6. The downstream contribution of this variance to $\mathbf{z}$ will be scaled by b_1^2. We can thus linearly decompose the variance of $\mathbf{z}$ into terms due to the contribution of the error term $\mathbf{y}$ and the product of $\mathbf{x}$ and $\mathbf{y}$ such that

$$\begin{aligned} \text{var}(\mathbf{z}) &= b_1^2[\mu(\mathbf{x})^2\text{var}(\mathbf{y}) + \mu(\mathbf{y})^2\text{var}(\mathbf{x}) + 2\mu(\mathbf{x})\mu(\mathbf{y})\text{cov}(\mathbf{x}, \mathbf{y}) + \\ &\quad \text{var}(\mathbf{x})\text{var}(\mathbf{y}) + \text{cov}(\mathbf{x}, \mathbf{y})^2] + \text{var}(\mathbf{e}) \ . \end{aligned} \tag{6}$$

When $\mu(\mathbf{x}) = \mu(\mathbf{y}) = 0$, Equation 6 simplifies to

$$\text{var}(\mathbf{z}) \quad = \quad b_1^2[\text{var}(\mathbf{x})\text{var}(\mathbf{y}) + \text{cov}(\mathbf{x}, \mathbf{y})^2] + \text{var}(\mathbf{e}) \; . \tag{7}$$

We next derive a general estimator for the variance decomposition of products of variables and apply it to the example problem in Equation 7 in order to demonstrate how the known estimator for the variance of the product of two variables is a special case result of the general estimator. This method relies on the model in the equivalent path diagram shown in Figure 2-b.

Expectations by Method of Moments for Products of Variables

Suppose we are given an SEM as a path diagram in which all nodes with incoming edges are additionally labeled as sum or product of their incoming edges. For the time being, we will assume that the graph is acyclic (recursive). It may be that the results can be extended to cases with cycles as well, provided that for all node values, an infinite series calculated along each cycle converges.

We are interested in all moments of the vector of all observed variables. The following describes how those can be computed analytically assuming a fixed set of parameters. The main idea is to represent the variance sources of the SEM as independent standard-normally distributed sources and then represent all variables as polynomial over these sources. Then, all moments become expectations of polynomials, which in turn are sums of monomials. In this way, the computation of the moments is reduced to computing the expectation of a monomial of independently standard-normally distributed variables–a problem that has a known computational solution.

At the highest level, the algorithm proceeds in three steps,

1. All variables in the SEM are represented by a linear combination of some independently normally distributed variables $w_1, ..., w_n$ with known variances such that the covariance matrix of all variables is the symmetrical matrix **S** from the RAM matrix formulation.

2. Progressing top-down in the asymmetrical graph of the path diagram, polynomial representations of all variables in the $w_1, ..., w_n$ are computed.

3. Polynomial representations of all requested moments are computed and evaluated into numbers.

Suppose that we have an SEM represented in standard RAM notation such that the, the model-expected covariance matrix of the observed variables, $\mathbf{C}_{xx}$ is calculated as

$$\mathbf{C}_{xx} = \mathbf{F}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{S}((\mathbf{I} - \mathbf{A})^{-1})^T\mathbf{F}^T \tag{8}$$

where for all variables, both latent and manifest, $\mathbf{A}$ is the matrix of regression coefficients, $\mathbf{S}$ is the matrix of variances and covariances, and $\mathbf{I}$ is the identity matrix. The matrix $\mathbf{F}$ filters out the latent variables so that $\mathbf{C}_{xx}$ contains only the model-expected covariance matrix of the observed variables.

In order to transform the model into a model with only independent variables, we will operate on $\mathbf{S}$, the matrix of model variances and covariances both latent and manifest. We first compute the Eigenvalue decomposition of $\mathbf{S}$,

$$\mathbf{S} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T \tag{9}$$

Let $W = w_1, ..., w_n$ be independently normally distributed variables with zero mean and variances given by the diagonal entries of $\mathbf{D}$, the Eigenvalues of $\mathbf{S}$, where n is the number of total variables in the SEM. Then $\mathbf{Q}W$ is an n-dimensional random variable with zero mean and covariance

$$\mathbb{V}(\mathbf{Q}W) = \mathbb{E}(\mathbf{Q}\sqrt{\mathbf{D}}\sqrt{\mathbf{D}}\mathbf{Q}^T) = \mathbf{S} \tag{10}$$

So each variable can be expressed by a linear combination of the W variables with the corresponding row of Q as weights, plus a constant term that gives the mean of that variable.

Using only product and sum operators, every variable in the model (both observed and latent) x_i can now be represented as a polynomial of the original observed variables.

However, the mathematics of expectations are greatly simplified by using multivariate polynomials of *independent* (i.e., uncorrelated Gaussian) variables. Thus, we express each variable as a multivariate polynomial, f_i, in the independently normally distributed variables $w_1, ..., w_n$. If there are m variables in total, the $k = (k_1, ..., k_m)$-th moment of the joint distribution of the vector $X = (x_1, ..., x_m)$ is

$$M_k(X) = \mathbb{E}(\prod_{i=1}^{m} f_i^{k_i}) \tag{11}$$

where the expectation is taken with respect to the roots w_i. In particular, $M_k(X)$ is the expected value of a polynomial in w_i, where the coefficients are a combination of the regression weights in the SEM. Let this polynomial be $g = \prod_{i=1}^{m} f_i^{k_i}$. This polynomial is a sum of monomials in w_i. Since the expectation is linear, we can separate the computation of the expectation to the single monomials, and thus reduce our problem to computing the expectation of a monomial of independently normally distributed variables, i.e., an expectation of the form

$$\mathbb{E}\left(w_1^{e_1} \cdots w_n^{e_n}\right) \tag{12}$$

which is given by the product of the expectations of the single variables with their exponents,

$$\mathbb{E}\left(w_1^{e_1} \cdots w_n^{e_n}\right) = \prod_{i=1}^{n} \mathbb{E}(w_i^{e_i}) \tag{13}$$

1 For completeness sake, we give a quick proof:

Theorem 1.

$$\mathbb{E}\left(W_1^{e_1} \cdots W_n^{e_n}\right) = \prod_{i=1}^{n} \mathbb{E}(W_i^{e_i}) \tag{14}$$

Proof.

$$\begin{aligned}
\mathbb{E}\left(W_1^{e_1}\cdots W_n^{e_n}\right) &= \int_{-\infty}^{\infty} w_1^{e_1}\cdots w_n^{e_n} pdf(w_1)\cdots pdf(w_n)dw_1\ldots dw_n \\
&= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} w_1^{e_1} pdf(w_1)dw\right) w_2^{e_2}\cdots w_n^{e_n} pdf(w_2)\cdots pdf(w_n)dw_2\ldots dw_n \\
&= \mathbb{E}(W_1^{e_1})\int_{-\infty}^{\infty} w_2^{e_2}\cdots w_n^{e_n} pdf(w_2)\cdots pdf(w_n)dw_2\ldots dw_n \\
&\vdots \\
&= \prod_{i=1}^{n}\mathbb{E}(W_i^{e_i})
\end{aligned}$$

□

Thus, we are left with computing the higher-order moments of independent standard-normally distributed variables with zero mean and unit variance,

$$\mathbb{E}(W^e)\,, \tag{15}$$

where e is a positive integer. These moments are known (e.g., Papoulis & Pillai, 2002) to be

$$\mathbb{E}(W^e) = \left\{ \begin{array}{ll} 0 & \text{if } e \text{ is odd} \\ \mathbb{V}(W)^{e/2}\prod_{i=1}^{\frac{e-2}{2}}(2i+1) & \text{if } e \text{ is even} \end{array} \right\} \tag{16}$$

Example 1: Bivariate Product of Variables Regression

As an example, we will transform the SEM model in Figure 2-a into the equivalent SEM model shown in Figure 2-b such that all sources of variance and covariances are independent, normally distributed variables with mean zero and unit variance. The first step is to remove all covariances between variables by replacing them with unit variance sources. Thus, we add the latent variable w_2 and replace the covariance path with the value C_{xy} in Figure 2-a with regression paths from w_2 to x and y with values $C_{xy}^{\frac{1}{2}}$. Thus, the covariance between x and y can be calculated as $C_{xy}^{\frac{1}{2}}\cdot 1\cdot C_{xy}^{\frac{1}{2}} = C_{xy}$. However, now the variances for x and y become residual variances that must be reduced by the total effect of w_2 which is $C_{xy}^{\frac{1}{2}}\cdot 1\cdot C_{xy}^{\frac{1}{2}} = C_{xy}$. So, the residual variance of x is $V_x - C_{xy}$ and the residual variance of y is $V_y - C_{xy}$.

We can now replace the variance terms in Figure 2-a with the independent normally distributed unit variance variables w_1, w_3, and w_4 and regression paths to x, y, and e respectively. The regression weights for these variables become the square root of the residual variances for x, y, and e as shown in Figure 2-b. The ith variable in the path diagram in Figure 2-b can now be represented as a polynomial f_i of what we will refer to as *root nodes*, i.e., the independent, normally distributed variables $w_1, \ldots, w_n$ with zero mean and unit variance. This transformation of an SEM model into polynomials of root nodes can be applied to any SEM that can be represented as a RAM model, including models that have n-ary operators such as introduced here. We will call models that conform to RAM conventions along with addition and product n-ary operators Products of Variables (PoV) models.

Returning to the example in Figure 2-b, we can compute the moments, including all joint moments of any variable or pair of variables. Thus, we can decompose the variance of the product variable $\mathbf{z} = b_1\mathbf{xy} + \mathbf{e}$ into components of variance and covariance of the other variables in the SEM as follows:

$$\begin{aligned} \mathbb{V}(z) &= \mathbb{E}(zz) && (17)\\ &= \mathbb{E}((b_1xy + e)(b_1xy + e)) && (18)\\ &= \mathbb{E}(b_1^2x^2y^2 + 2b_1xye + e^2)\ . && (19) \end{aligned}$$

We next transform the variables x, y, and e into their monomial equivalents,

$$\begin{aligned} x &= w_1(V_x - C_{xy})^{\frac{1}{2}} + w_2(C_{xy})^{\frac{1}{2}} && (20)\\ y &= w_3(V_y - C_{xy})^{\frac{1}{2}} + w_2(C_{xy})^{\frac{1}{2}} && (21)\\ e &= w_4(V_e)^{\frac{1}{2}}\ . && (22) \end{aligned}$$

Next we substitute Equations 20, 21 and 22 into Equation 19 and then expand, collect, and cancel terms to find that

$$\mathbb{V}(z) = b_1^2(V_xV_y + C_{xy}^2) + V_e\ , \quad (23)$$

which is the result in Equation 7 derived from the results of Goodman (Goodman, 1960) and Bohrnstedt and colleagues (Bohrnstedt & Marwell, 1978). The complete derivation of this result is lengthy but is included in the supplemental material for this article.

Following the same logic, we can derive the expected covariance matrix of the model $z = b_1 xy + e$ as

$$\mathbb{E}(\Sigma) = \begin{bmatrix} V_x & C_{xy} & 0 \\ C_{xy} & V_y & 0 \\ 0 & 0 & b_1^2(V_x V_y + C_{xy}^2) + V_e \end{bmatrix} \tag{24}$$

What becomes apparent here is that this model is unidentified, that is to say, b_1 and V_e only appear in one cell of the matrix and in that cell they form two parts of a sum. Thus, a smaller b_1 can be compensated by a larger V_e and vice versa. In addition, since b_1 only appears as a squared term, it is unidentified with respect to its sign. Thus the maximum likelihood solution to this problem is not a single point, but lies on a line where all values on that line are equally likely.

When pre-multiplying two variables and then adding them into a regression equation as a way of estimating an interaction, one is taught to always subtract the mean of each variable prior to multiplying them together. This preprocessing is done in order to remove spurious covariances between the multiplicands and the outcome. But in the case of the current PoV method of estimating the effects of products, we can account for these covariances and they provide a way to resolve the underidentification of product coefficient and error term. If we do not remove the means prior to entering the multiplicands into the model and then estimate the means as part of a full information maximum likelihood solution, the model becomes fully identified.

When the expected means and covariances are derived for the same model $z = b_1 xy + e$ we find

$$\mathbb{E}(\Sigma) = \begin{bmatrix} V_x & C_{xy} & b_1(V_x\mu_y + C_{xy}\mu_x) \\ C_{xy} & V_y & b_1(V_y\mu_x + C_{xy}\mu_y) \\ b_1(V_x\mu_y + C_{xy}\mu_x) & b_1(V_y\mu_x + C_{xy})\mu_y & b_1^2(V_x\mu_y^2 + V_y\mu_x^2 + V_xV_y + C_{xy}^2 + C_{xy}(\mu_y + \mu_x)^2 - \mu_x^2\mu_y^2) + \mathbb{V}(e) \end{bmatrix} \quad (25)$$

$$\mathbb{E}(M) = \begin{bmatrix} \mu_x & \mu_y & b_1(C_{xy} + \mu_x\mu_y) \end{bmatrix} \quad (26)$$

Again, the full derivation of this result can be found in the supplemental materials.

Simulations

We ran a series of simulations in order to test the performance of the implementation of the PoV method in OpenMx on five common use cases. The only new syntax required in scripting a PoV model is that one must add a line declaring **productVars=** with the names of any product nodes shown as a circle around an asterisk in the following path diagrams. Then one may use the standard **mxPath()** statements to specify the paths between named variables. All R scripts for each of the example models simulated and estimated below are included in the supplemental materials.

Simulation One: Manifest Variable Moderation

A moderation model with only observed variables as shown in Figure 3 was simulated where the parameters took on one of the values $b_1 = \{-0.5, 0.5\}$, $b_2 = \{-0.5, 0.5\}$, $\mu_x = \{-0.5, 0, 0.5\}$, $\mu_y = \{-0.5, 0, 0.5\}$, $C_{xy} = \{-1, 0, 1\}$, $V_e = \{0.2, 0.7, 1.2\}$,

$V_x = V_y = 3.0$, resulting in $2 \times 2 \times 3 \times 3 \times 3 \times 3 = 324$ conditions. Each condition was replicated 30 times, resulting in a total of 9,720 data sets. Prior to simulating the data for $N = 1000$ simulated participants, a normally distributed random number ($\mu = 0$, $\sigma = 0.1$) was added to each parameter to better cover the parameter space.

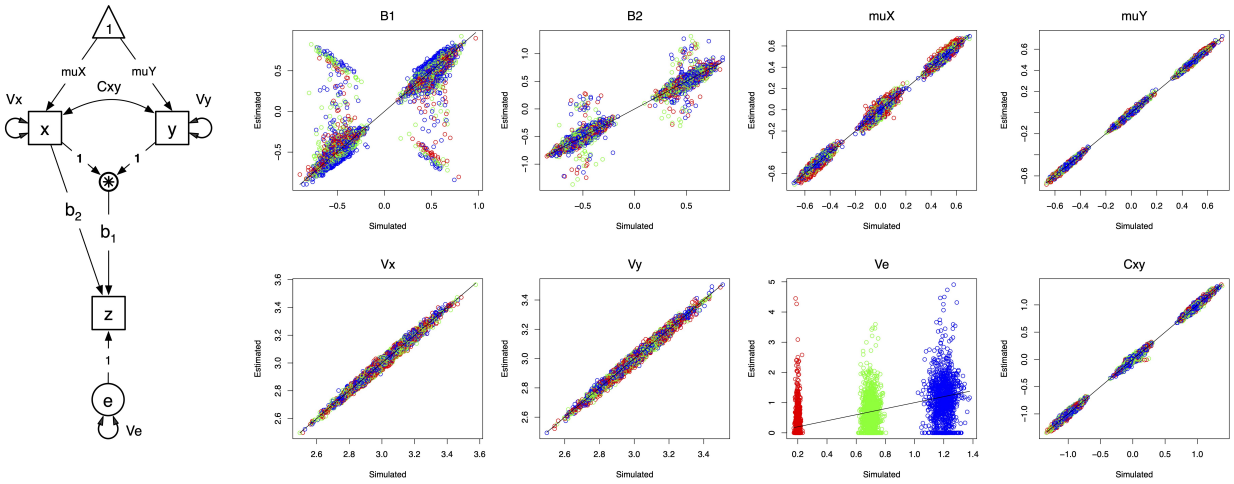

Figure 3. Path diagram and simulation outcomes for a bivariate moderation model. For the scatter plots, the color of the plotted point corresponds to the error variance condition. In the plot for b_1, note that there are some estimated values that have the opposite sign than the generating values. This occurs when both the mean of x and y are very close to zero and the b_1 parameter is underidentified.

The simulation resulted in 99.8% convergence. The estimated parameters from each of the 9,720 data sets was compared to the generating coefficients for that data set. Mean relative bias was calculated as the signed difference between the generating and estimated coefficient divided by the generating coefficient. Mean relative bias for b_1 was 0.046 and for b_2 was < 0.001. Assuming an alpha level of 0.95, coverage as calculated by 1.96 time the parameter standard error was 0.974 for b_1 and was 0.958 for b_2. Thus, standard errors were correct for the direct effect, b_2, but were conservative for the product effect, b_1. The model was estimated on the same 9,720 data sets but with b_1 fixed to zero and the likelihood ratio

test was calculated by subtracting the minus two log likelihood of the full model from the model with the parameter set to zero. Again assuming an alpha level of 0.95 and a χ^2 test with one degree of freedom, this resulted in a type I error rate of 0.0494. Thus, although the standard error for the product direct effect is conservative, the likelihood ratio test performs exactly as expected.

Simulation Two: Manifest Variable Moderation with Missing Values

One of the problems with current methods used to estimate moderation is that they are not able to account for missing data. In order to test whether the PoV method could account for missingness in the same way as full information maximum likelihood, we re-ran Simulation One where we substituted a random selection of 20% of the values of y and a separate random selection of the values of z with NA. Thus, this fits with the definition of values of y and z being missing completely at random. Figure 4 presents the results of this simulation.

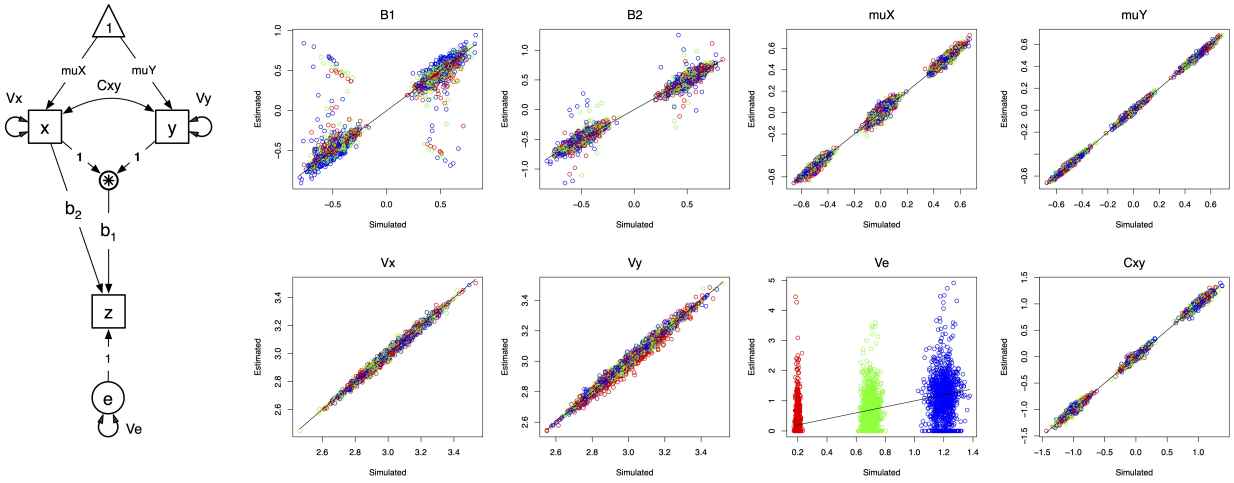

Figure 4. Path diagram and simulation outcomes for a bivariate moderation model where 20% of y and z are missing completely at random.

This simulation resulted in 97.3% convergence. Mean relative bias for b_1 was 0.015 and for b_2 was < 0.001. Assuming an alpha level of 0.95, coverage as calculated by 1.96

time the parameter standard error was 0.969 for b_1 and was 0.950 for b_2. Clearly, full information maximum likelihood is working as expected when data are missing completely at random.

Missingness that can be accounted for by the model, the case of missing at random, was also tested in simulation for this model. Here, missingness in y was set to be conditional on the value of x. When x was more than one standard deviation above its mean, y had a 50% chance of being missing. In addition, z had a 20% chance of being missing, but not conditional on any part of the model. Thus y was missing at random and z was missing completely at random. This simulation resulted in 97.4% convergence. Mean relative bias for b_1 was 0.049 and for b_2 was 0.003. Assuming an alpha level of 0.95, coverage as calculated by 1.96 time the parameter standard error was 0.811 for b_1 and was 0.855 for b_2. Thus, the missing at random case increased bias by a very small amount, but decreased coverage considerably in both the direct effect and the product effect.

Simulation Three: Latent Variable Moderation

A moderation model with a latent variable moderating a second latent variable and a latent outcome as shown in Figure 5 was simulated where the parameters took on one of the values $b_1 = \{-0.5, 0, 0.5\}$, $b_6 = \{-0.5, 0, 0.5\}$, $\mu_x = \{-0.5, 0.5\}$, $\mu_y = \{-0.5, 0.5\}$, $C_{xy} = \{-1, 0, 1\}$, and $V_e = \{0.1, 0.6, 1.1\}$. The variance of the predictor latent variables was set to $V_x = V_y = 3.0$ and the residual variance of the outcome latent variable was set to $V_z = V_e$. The loadings for the latent variables were set to $b_2 = b_{x2} = b_{y2} = \{-.05, 0.5\}$ and $b_3 = b_{x3} = b_{y3} = \{-.05, 0.5\}$ and the unique variances were all set to the current value of V_e. This resulted in $3 \times 3 \times 2 \times 2 \times 3 \times 3 \times 2 \times 2 = 1296$ conditions. Each condition was replicated 8 times, resulting in a total of 10,368 data sets, each with $N = 1,000$ simulated participants. Prior to simulating the data, a normally distributed random number ($\mu = 0$, $\sigma = 0.1$) was added to each parameter to better cover the parameter space.

The simulation resulted in 99.9% convergence. Mean relative bias for b_1 was -0.092

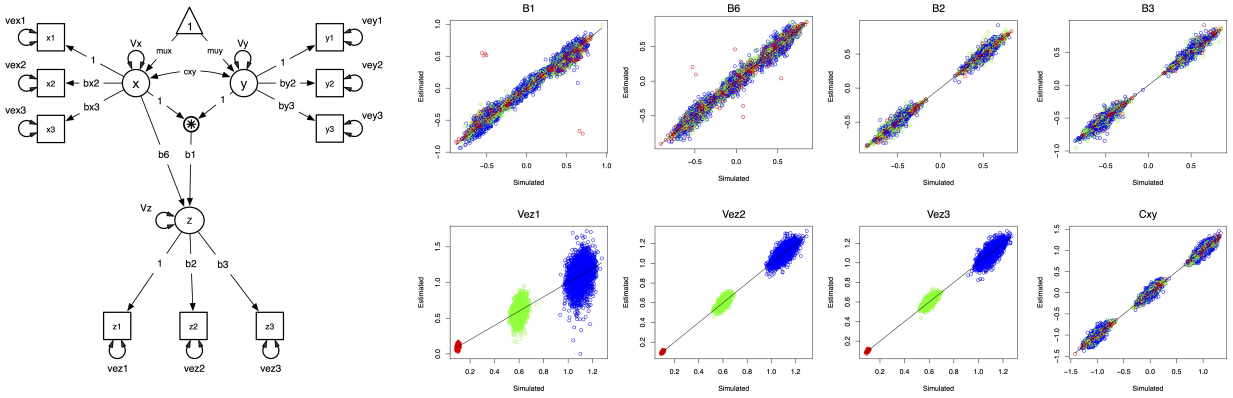

Figure 5. Path diagram and simulation outcomes for a bivariate latent moderation model.

and for b_6 was < 0.001. Assuming an alpha level of 0.95, coverage as calculated by 1.96 time the parameter standard error was 0.975 for b_1 and was 0.958 for b_6. Thus, standard errors were correct for the direct effect, b_6, but were conservative for the product effect, b_1. The model was estimated on the same 9,720 data sets but with b_1 fixed to zero and the likelihood ratio test was calculated by subtracting the minus two log likelihood of the full model from the model with the parameter set to zero. Again assuming an alpha level of 0.95 and a χ^2 test with one degree of freedom, this resulted in a type I error rate of 0.0221. Thus, both the standard error for the product direct effect and the likelihood ratio test were conservative. This was likely due to the fact that in the latent simulation we allowed the true product effect b_1 to take on values very close to zero.

Simulation Four: Latent Variable Interaction

A model with a product of two latent variables and direct effects on a latent outcome as shown in Figure 6 was simulated where the parameters took on one of the values $b_1 = \{-0.5, 0.5\}$, $b_6 = \{-0.5, 0.5\}$, $b_7 = \{-0.5, 0.5\}$, $\mu_x = \{-0.5, 0.5\}$, $\mu_y = \{-0.5, 0.5\}$, $C_{xy} = \{-1, 0, 1\}$, and $V_e = \{0.1, 0.6, 1.1\}$. The variance of the predictor latent variables was set to $V_x = V_y = 3.0$ and the residual variance of the outcome latent variable was set to $V_z = V_e$. The loadings for the latent variables were set to $b_2 = b_{x2} = b_{y2} = \{-.05, 0.5\}$ and

$b_3 = b_{x3} = b_{y3} = \{-.05, 0.5\}$ and the unique variances were all set to the current value of V_e. This resulted in $2 \times 2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 2 \times 2 = 1,152$ conditions. Each condition was replicated 9 times, resulting in a total of 10,368 data sets, each with $N = 1,000$ simulated participants. Prior to simulating the data, a normally distributed random number ($\mu = 0$, $\sigma = 0.1$) was added to each parameter to better cover the parameter space.

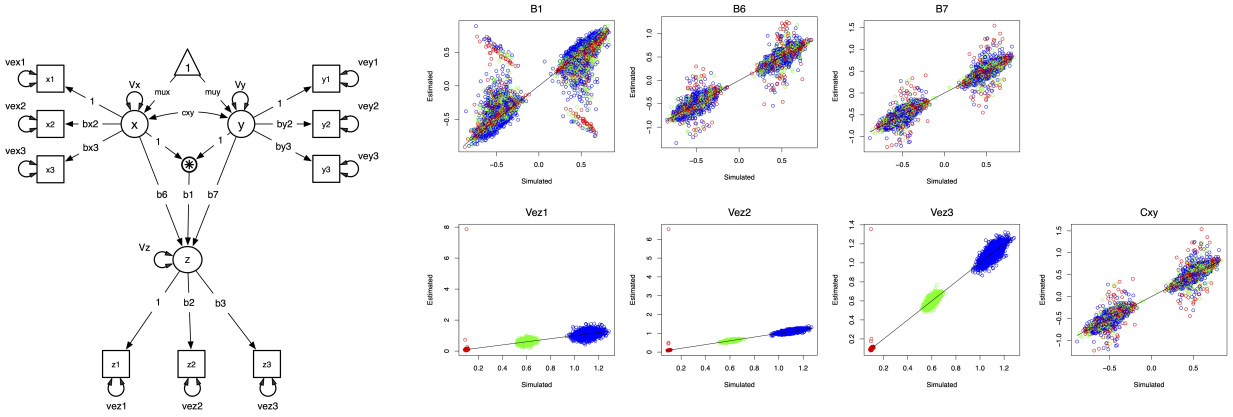

Figure 6. Path diagram and simulation outcomes for a bivariate latent interaction model. Again we see some sign reversals in b_1 due to means of the latent variables being near zero.

The simulation resulted in 98.2% convergence. Mean relative bias for b_1 was 0.083, for b_6 was -0.001 and for b_7 was < -0.013. Assuming an alpha level of 0.95, coverage as calculated by 1.96 time the parameter standard error was 0.992 for b_1, 0.983 for b_6 and 0.984 for b_7. Thus, standard errors were conservative for the product effect, b_1, and the two direct effects, b_6 and b_7.

Simulation Five: Squared Predictor Variable

A model with a predictor variable, x and its square, x^2 on a latent outcome as shown in Figure 7 was simulated where the parameters took on one of the values $b_1 = \{-0.5, 0.5\}$, $b_2 = \{-0.5, 0.5\}$, $b_3 = \{-0.5, 0.5\}$, $b_4 = \{-0.5, 0.5\}$, $\mu_x = \{-1.0, -0.5, 0, 0.5, 1.0\}$, and $V_e = \{0.1, 0.35, 0.6, 0.95, 1.1\}$. The variance of the predictor variable was set to $V_x = 3.0$ and the residual variance of the outcome latent variable was set to $V_z = V_e$. The unique

variances were all set to the current value of V_e. This resulted in $2 \times 2 \times 2 \times 2 \times 5 \times 5 = 400$ conditions. Each condition was replicated 25 times, resulting in a total of 10,000 data sets, each with $N = 1,000$ simulated participants. Prior to simulating the data, a normally distributed random number ($\mu = 0$, $\sigma = 0.1$ was added to each parameter to better cover the parameter space.

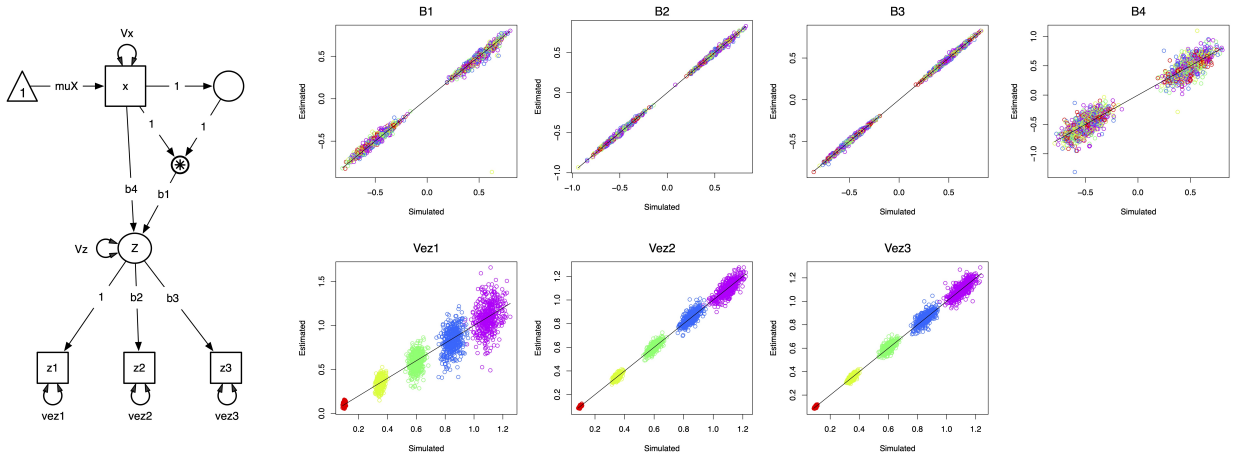

Figure 7. Path diagram and simulation outcomes for a squared predictor.

The simulation resulted in 99.9% convergence. Mean relative bias for b_1 was 0.008 and for b_4 was 0.001. Assuming an alpha level of 0.95, coverage as calculated by 1.96 time the parameter standard error was 0.999 for b_1 and 0.972 for b_4. Thus, standard errors were conservative for the product effect, b_1, and the direct effect, b_4.

Discussion

Estimation of SEM models with Products of Variables is a general method that can provide unbiased estimates of parameters when predictor variables are normally distributed. Coverage of parameters of products is higher than expected when calculated using standard error estimates, but the likelihood ratio test appears to perform as expected.

The primary advantage of the PoV method is in its generality. Most alternative methods for estimating parameters of SEM models containing products focus on special

cases and involve complicated constraints or estimation of parameters with non-normally distributed variability. The PoV method is easy to use as it has been implemented in OpenMx and Ωnyx. The method can handle interaction and moderator variables that are either latent or have missingness. We expect that there are many more models other than the obvious moderation and interaction models for which PoV estimation can be used.

Interaction versus Moderation

The use of the product calculation node in path diagrams has clarified for us the difference between interaction between variables and moderation of one variable by another. Examine the difference between Figure 5 and Figure 6. These two diagrams are exactly the same other than the direct effect between y and z that appears in the interaction diagram, Figure 6, but does not appear in the moderation diagram, Figure 5. Thus, the only thing that distinguishes a moderating variable from the other multiplicand is its lack of a direct effect on the outcome. It becomes clear that when one has a product of two variables, say $z = x \cdot y$ in a model, either x or y or both may be considered to be a moderator depending on presence or absence of a direct effect of x or y on z.

The reader may wonder why we did not use a classic interaction model with direct effects as an example model to illustrate the procedure. Figure 8 illustrates why this model is unidentified using the current method. The reason why it is unidentified as a structural model predicting covariances is apparent by counting seven free parameters while there are only six degrees of freedom in a 3×3 covariance matrix. One must watch for local underidentification in models that include products of variables. The overall model may be globally identified as determined by counting parameters and degrees of freedom in the covariance matrix, but there still be areas of local underidentification resulting in sign reversals such as those seen in simulations 1 and 4.

It may be more difficult to see why identification works when one premultiplies two manifest variables and creates a third variable for an interaction model. In this case there

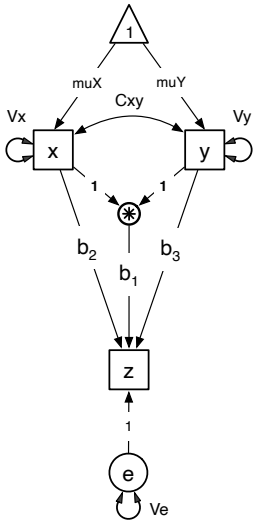

Figure 8. A manifest variable interaction model with both direct effects has more parameters than statistics and thus is unidentified.

is a 4×4 covariance matrix and so there are not negative degrees of freedom. But why is the 4×4 covariance matrix of full rank? This is due to the contributions of higher order moments introduced by the multiplication formed at the individual data rows when the premultiplication is performed. This model is identified because a product of two normal distributions is not itself normal.

Product Calculation Nodes and the Identity Function

One feature of PoV is its reliance on a product node extension of RAM model matrices. Although OpenMx has chosen to call this a ProductVar, it is not really a variable. As such it does not have a mean or a variance, it is simply a node that indicates the multiplication operation. Thus although a product node occupies a row and column in the RAM matrices, OpenMx prevents the user from connecting a variance, covariance, or mean path to this calculation node. However, if one examines the RAM matrices, it becomes apparent that there is an automatically specified mean path to every product node with a fixed value of 1.0. Why is that?

In RAM, removing a path and setting a path to zero are identical operations. However, this is not the case in PoV. Note that the identity function for addition is to add zero whereas the identity function for multiplication is multiply by one. If there is no desired effect between predictor and outcome in a normal SEM, a regression path from the predictor to the outcome is set to zero. However, if a regression path pointing from one multiplicand to a multiplication operator symbol is set to zero in an PoV diagram, then the other multiplicand is multiplied by zero. This may not be the intention of the modeler. When one wishes to use a likelihood ratio to test the difference between a model with and without a product, we recommend to use the outgoing path from the product node as the path to set to zero for the one degree of freedom minus two log likelihood comparison.

Assumptions and Limitations

The algorithm in this article makes the strong assumption that the variables in an PoV allow the transformation of the model into an equivalent model in which all variance sources are represented as linear combinations of independent normally distributed standardized variables. This is a relaxation of the usual SEM assumption of multivariate normality. The outcome of a product of variables will not be normally distributed. However, the PoV assumption is that the non-normality of a product outcome variable can be completely accounted by the non-normality generated by multiplying multivariate normal variance sources. Thus, all variables that are not outcome variables must be multivariate normal. To be explicit, this means that all residuals must be normally distributed, including the residuals of variables that are outcomes of products of variables.

This article presents a novel addition to the already complex infrastructure of structural equation modeling. We acknowledge that there may be special cases that need to be tested before products of variables models can be recommended in those cases. We recommend using simulation and either OpenMx or Ωnyx to verify models that include PoV structures that go beyond the common use cases presented here.

For instance, since the PoV relies on normality of predictor variables, ordinal predictors might not be able to be used as a multiplicand. If one has ordinal manifest variables that one wishes to use in a product, we recommend a method such as OpenMx definition variables (Neale, 1998; Neale et al., 2016). In principal, the method for estimating a normally distributed latent variable from ordinal indicators as implemented in OpenMx should be able to be used to create a latent multiplicand, but we have not yet fully tested this case.

Conclusions

Products of variables can be included in structural equation models and model expectations can be calculated using the methods introduced here. This means that products of combinations of manifest and latent variables are now possible to implement in standard SEM software in an automated and easy-to-specify way. Three important and commonly used models that will immediately benefit from PoV estimation are: i) models with interactions between latent variables, ii) latent moderator models, and iii) moderator models with missingness in the moderator variable.

Future work will certainly find novel and interesting uses for the PoV method. One case that we have begun to explore is the use of PoV to extract a “factor of paths”, creating a latent variable that accounts for the commonality between coefficients that include person-specific variance. This could be construed to be a special case of multilevel models with random coefficients and thus we may find that PoV estimation may prove useful for multilevel models. A second case that we have begun to explore is nonlinear dynamical systems models. Differential equations with squared and cubic terms are at the heart of dynamical systems that exhibit bifurcation and/or chaotic dynamics. We expect that PoV estimation will prove useful in fitting these sorts of models to complex physiological and behavioral timeseries.

We believe that the algorithm presented here represents a paradigm shift for

representing theories within the context of structural equation models. We look forward to seeing PoV estimation appearing as a feature in software other than OpenMx and Ωnyx and with that in mind, we refer SEM software authors to our open source code available on GitHub.

References

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

Bauer, D. J., & Hussong, A. M. (2009). Psychometric approaches for developing commensurate measures across independent studies: traditional and new models. *Psychological methods*, *14*(2), 101.

Bohrnstedt, G. W., & Goldberger, A. S. (1969). On the exact covariance of products of random variables. *Journal of the American Statistical Association*, *64*(328), 1439–1442.

Bohrnstedt, G. W., & Marwell, G. (1978). The reliability of products of two random variables. *Sociological Methodology*, *9*, 254–273.

Boker, S. M., & McArdle, J. J. (2005). Path analysis and path diagrams. In B. Everitt & D. Howell (Eds.), *Encyclopedia of statistics in behavioral science (vol. 3)* (pp. 1529–1531). New York: John Wiley & Sons.

Boker, S. M., McArdle, J. J., & Neale, M. C. (2002). An algorithm for the hierarchical organization of path diagrams and calculation of components of covariance between variables. *Structural Equation Modeling*, *9*(2), 174–194.

Goodman, L. A. (1960). On the exact variance of products. *Journal of the American Statistical Association*, *55*(292), 708–713.

Kenny, D. A., & Judd, C. M. (1984). Estimating the nonlinear and interactive effects of latent variables. *Psychological Bulletin*, *96*(1), 201–210.

Klein, A., & Moosbrugger, H. (2000). Maximum likelihood estimation of latent interaction effects with the lms method. *Psychometrika*, *65*(4), 457–474.

Marsh, H. W., Wen, Z., & Hau, K.-T. (2004). Structural equation models of latent interactions: Evaluation of alternative estimation strategies and indicator construction. *Psychological Methods*, *9*(3), 275.

Marsh, H. W., Wen, Z., Hau, K.-T., Little, T. D., Bovaird, J. A., & Widaman, K. F.

(2007). Unconstrained structural equation models of latent interactions: Contrasting residual-and mean-centered approaches. *Structural Equation Modeling: A Multidisciplinary Journal*, *14*(4), 570–580.

McArdle, J. J., & McDonald, R. P.(1984). Some algebraic properties of the Reticular Action Model for moment structures. *British Journal of Mathematical and Statistical Psychology*, *37*, 234–251.

Mehta, P. D., & Neale, M. C.(2005). People are variables too: Multilevel structural equations modeling. *Psychological Methods*, *10*(3), 259–284.

Moosbrugger, H., Schermelleh-Engel, K., & Klein, A.(1997). Methodological problems of estimating latent interaction effects. *Methods of Psychological Research Online*, *2*(2), 95–111.

Neale, M. C.(1998). Modeling interaction and nonlinear effects with mx: A general approach. In G. Marcoulides & R. Schumacker (Eds.), *Interaction and non-linear effects in structural equation modeling* (pp. 43–61). Mahwah, NJ: Lawrence Erlbaum Associates.

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R., et al.(2016). Openmx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, *81*(2), 535–549. (PMCID: 25622929)

Oertzen, T. von, Brandmaier, A., & Tsang, S.(2015). Structural equation modeling with Ωnyx. *Structural Equation Modeling: A Multidisciplinary Journal*, *22*(1), 148–161.

Papoulis, A., & Pillai, S. U.(2002). *Probability, random variables, and stochastic processes*. New York: Tata McGraw-Hill Education.

Schumacker, R. E.(2002). Latent variable interaction modeling. *Structural Equation Modeling*, *9*(1), 40–54.

Wall, M. M., & Amemiya, Y.(2001). Generalized appended product indicator procedure for nonlinear structural equation analysis. *Journal of Educational and Behavioral Statistics*, *26*(1), 1–29.

Wall, M. M., & Amemiya, Y. (2003). A method of moments technique for fitting interaction effects in structural equation models. *British Journal of Mathematical and Statistical Psychology*, *56*, 47–63.

Wright, S. (1921). Correlation and causation. *Journal of agricultural research*, *20*(7), 557–585.

Wright, S. (1934). The method of path coefficients. *The Annals of Mathematical Statistics*, *5*, 161–215.