Bi-Factor Models and Exploratory Bifactor Rotation

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

Bi-Factor Models and Exploratory Bifactor Rotation

Introduction

- A bifactor model combines the notion of a general factor with the idea of simple structure.
- For example, suppose six items have a general factor, but two groups of 3 items each also have a specific dimension associated with them.
- Then the factor pattern would be of the classic bi-factor form,

$$\mathbf{\Lambda} = \begin{bmatrix} x & x & 0 \\ x & x & 0 \\ x & x & 0 \\ x & 0 & x \\ x & 0 & x \\ x & 0 & x \end{bmatrix} \tag{1}$$

in which x represents a free parameter to be estimated, while other elements are constrained to be zero.

Introduction

- The simple idea of a bi-factor model dates back to the early days of factor analysis, at least as early as 1937, but it is the subject of renewed interest in at least two areas. First, it is a model that makes , and second, it tends to arise in certain testing situations due to "testlet effects."
  1. It is a model that makes sense on substantive grounds, for instance when one expects a general intelligence factor in addition to one or more ability factors
  2. It tends to arise in certain testing situations due to "testlet effects."

Introduction

- Bi-factor models can be exploratory or confirmatory.
- Jennrich, Bentler and associates at UCLA have developed methods for exploratory bi-factor analysis by modifying classic rotation methods to seek out a bi-factor structure automatically, rather than the traditional "simple structure."
- This can be very useful in exploratory factor analysis, as it allows one to deal with the existence of a general factor, while discovering previously unanticipated dimensional structures.

Introduction

- Errors of inference can result when items that are assumed to be essentially unidimensional are plagued by what Professor Sun-Joo Cho calls “nuisance dimensions.”
- As an example, consider the multidimensionality induced in a set of items within a test due to ”testlet” effects.
- Testlets are groups of items based on the same stimulus. For example, on a reading test, a reading passage may be followed by several questions.
- The dependency among the testlet items induced by the fact that they refer to the same passage will cause the assumption of unidimensionality to be violated for this group of items, resulting in turn in errors of estimation if the items are assumed to be unidimensional.

Introduction

- A bi-factor model can be used to help account for additional testlet dimensions in a set of items.
- Reise and associates, among others, have used the bi-factor model to model unwanted dependencies in an item response framework.
- So, for example, $\mathbf{\Lambda}$ above could represent the factor pattern for two testlets composed of 3 items each.

Introduction

- Until very recently, there was no reliable technology for rotating a solution to bi-factor form in the context of simple exploratory factor analysis, because rotation methods were geared to uncovering "simple structure," which a bi-factor pattern violates.
- In a truly exploratory situation, one could obtain a factor pattern, then rotate it via target ("Procrustes") rotation to a particular pattern, but the problem remains — what pattern?
- Ideally, one should have a rotation criterion that produces a bifactor pattern like the one in Equation 1

The Schmid-Leiman Hierarchical Approach

- Unfortunately, the closest thing to automatic bi-factor rotation was the Schmid-Leiman (1957) method.
- Schmid and Leiman discussed a situation in which the factors have a hierarchical configuration.
- For example, suppose one obtains a simple structure from 3 factors, but the factors are correlated. Then these 3 factors have a correlation matrix $\mathbf{\Psi}$.
- One can then factor analyze $\mathbf{\Psi}$ obtain a "higher order" solution.

The Schmid-Leiman Hierarchical Approach

- For example, suppose the first order model is

$$\mathbf{x} = \boldsymbol{\Lambda}\boldsymbol{\xi} + \boldsymbol{\delta} \tag{2}$$

- Under the standard assumptions, this leads to

$$\boldsymbol{\Sigma}_{\mathbf{xx}} = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}' + \boldsymbol{\Theta}_{\delta} \tag{3}$$

- But we also have

$$\boldsymbol{\xi} = \boldsymbol{\gamma} h + \mathbf{d}$$

- In this case, h is a single higher order factor, and we get, by a second application of the fundamental theorem,

$$\boldsymbol{\Phi} = \mathrm{Var}(\boldsymbol{\xi}) = \boldsymbol{\gamma}\boldsymbol{\gamma}' + \boldsymbol{\Theta}_{\mathbf{d}} \tag{4}$$

The Schmid-Leiman Hierarchical Approach

- Combining Equations 3 and 4, we get

$$\mathbf{\Sigma}_{\mathbf{xx}} = \mathbf{\Lambda}(\gamma\gamma' + \mathbf{\Theta}_{\mathbf{d}})\mathbf{\Lambda}' + \mathbf{\Theta}_{\delta} \tag{5}$$

- Notice that this can be written in the following partitioned form:

$$\mathbf{\Sigma} = \mathbf{F}\mathbf{F}' + \mathbf{\Theta}_{\delta} \tag{6}$$

  where

$$\mathbf{F} = \left[\begin{array}{cc} \mathbf{\Lambda}\gamma & \mathbf{\Lambda}\mathbf{\Theta}_{\mathbf{d}}^{1/2} \end{array}\right] \tag{7}$$

- Notice that there is an added factor, and since $\mathbf{\Lambda}\gamma$ is a linear combination of all the columns of the original factor pattern, it often will have non-negligible loadings in all positions.
- Moreover, since $\mathbf{\Lambda}\mathbf{\Theta}_{\mathbf{d}}^{1/2}$ is a diagonal rescaling of the original $\mathbf{\Lambda}$, it will still have simple structure if the original $\mathbf{\Lambda}$ did.

The Jennrich-Bentler Exploratory Bi-Factor Methods

- Unfortunately the Schmid-Leiman method fails on many examples.
- The problem is that, if there is a general factor, the m factor solution will not be close to simple structure to begin with.
- Jennrich and Bentler (2011, 2012) have produced a very clever method that recovers a bi-factor solution by exploratory rotation in many situations where the Schmid-Leiman method fails.
- Here is a simple example of their method.

The Jennrich-Bentler Exploratory Bi-Factor Methods

- Open up R and load the code in the AdvancedFactorFunctions.txt file.

```
> source("http://www.statpower.net/R2101/AdvancedFactorFunctionsV1.05.txt'
```

- The following test matrix **A** exhibits nearly perfect bi-factor structure.

```
> A = matrix(c(1,1,0,2,1,0,1,1,0,2,0,1,1,0,1,2,0,1),6,3,byrow=TRUE)
> A
```

```
     [,1] [,2] [,3]
[1,]    1    1    0
[2,]    2    1    0
[3,]    1    1    0
[4,]    2    0    1
[5,]    1    0    1
[6,]    2    0    1
```

The Jennrich-Bentler Exploratory Bi-Factor Methods

- Next, we rotate $\mathbf{A}$ into a position in which $\mathbf{A}'\mathbf{A}$ is diagonal, and the bifactor structure is clearly unrecognizeable.

```
> T <- eigen(t(A) %*% A)$vectors
> A.r <- A%*%T
> A.r
         [,1]       [,2]        [,3]
[1,] 1.165945 -0.7808688 -0.17554252
[2,] 2.083487 -0.7808688  0.22209836
[3,] 1.165945 -0.7808688 -0.17554252
[4,] 2.145588  0.6246950  0.07880251
[5,] 1.228046  0.6246950 -0.31883837
[6,] 2.145588  0.6246950  0.07880251
```

The Jennrich-Bentler Exploratory Bi-Factor Methods

- Here is a varimax rotation using the GPForth function with a random start. In order to get the optimal rotation, several random starts may be needed. Notice that we could/would reverse the signs in the first column.

```
> output <- GPForth(A,method="varimax",Tmat=Random.Start(3))
> output$Lh
           [,1]      [,2]        [,3]
[1,] -1.2799317 0.5272176  0.28951059
[2,] -1.8107508 1.3115603 -0.03148431
[3,] -1.2799317 0.5272176  0.28951059
[4,] -0.6653284 2.1332146  0.08205809
[5,] -0.1345094 1.3488719  0.40305298
[6,] -0.6653284 2.1332146  0.08205809
```

- Varimax doesn't produce a useful solution. It (of course) does not recover the bifactor solution, and the structure is nothing close to the classic "independent cluster" form characteristic of simple structure.

The Jennrich-Bentler Exploratory Bi-Factor Methods

- Now we try the bi-factor rotation method. My service routine `FindBifactor` runs as many random starts as you want (25 in this case) and returns the best result.

```
> FindBifactor(A.r,25)$Lh
```

```
           [,1]          [,2]          [,3]
[1,] -0.9999962  2.037556e-06  1.000004e+00
[2,] -1.9999962 -2.590201e-06  1.000008e+00
[3,] -0.9999962  2.037556e-06  1.000004e+00
[4,] -2.0000046  9.999907e-01  8.470897e-07
[5,] -1.0000046  9.999954e-01 -2.909104e-06
[6,] -2.0000046  9.999907e-01  8.470897e-07
```

- To clean things up further, I generated another service routine that automatically flips the signs to eliminate negative loadings, and also rounds off the pattern. As you can see from the code, by default, it returns the pattern rounded to 2 digits.

```
> FindBifactorPattern(A.r,25)
```

```
     [,1] [,2] [,3]
[1,]    1    1    0
[2,]    2    1    0
[3,]    1    1    0
[4,]    2    0    1
[5,]    1    0    1
[6,]    2    0    1
```

Applications

- Clearly bi-factor rotation may have something to offer, and there has been a resurgence of interest in the topic (See Reise, 2012, *Multivariate Behavioral Research*, *The Rediscovery of Bifactor Measurement Models*, and references therein).
- I've implemented bi-factor rotation in our Advanced Factor Routines, and *Mplus* now has it fully integrated in its EFA routine.