ST 437/537: Applied Multivariate and Longitudinal Data Analysis

Confirmatory Factor Analysis

Arnab Maity
NCSU Department of Statistics
SAS Hall 5240   919-515-1937   amaity[at]ncsu.edu

Introduction

We typically apply the exploratory factor analysis in a pilot study to determine any grouping among the manifest variables and form possible hypotheses. Then we often test these hypotheses (or any other pre-specified hypotheses we might already have) using a separate dataset. We can do so using confirmatory factor analysis (CFA).

To be specific, CFA attempts to formally test a particular factor model, where particular manifest variables are allowed to relate to particular factors.

For example, let us consider the ability data (actually a correlation matrix) in the `MVA` package (Figure 7.1 in Everitt and Hothorn). Six variables were recorded [Calsyn and Kenny (1977)] for 556 eighth-grade students:

SCA: self-concept of ability;

PPE: perceived parental evaluation;

PTE: perceived teacher evaluation;

PFE: perceived friend's evaluation;

EA: educational aspiration;

CP: college plans.

The ability dataset shows the correlation among these variables.

```r
library(MVA)
## code taken from demo("Ch-SEM")
ab <- c(0.73,
         0.70, 0.68,
         0.58, 0.61, 0.57,
         0.46, 0.43, 0.40, 0.37,
         0.56, 0.52, 0.48, 0.41, 0.72)
ability <- diag(6) / 2
ability[upper.tri(ability)] <- ab
ability <- ability + t(ability)
rownames(ability) <- c("SCA","PPE","PTE","PFE","EA","CP")
colnames(ability) <- c("SCA","PPE","PTE","PFE","EA","CP")

ability
```

```
##      SCA  PPE  PTE  PFE   EA   CP
## SCA 1.00 0.73 0.70 0.58 0.46 0.56
## PPE 0.73 1.00 0.68 0.61 0.43 0.52
## PTE 0.70 0.68 1.00 0.57 0.40 0.48
## PFE 0.58 0.61 0.57 1.00 0.37 0.41
## EA  0.46 0.43 0.40 0.37 1.00 0.72
## CP  0.56 0.52 0.48 0.41 0.72 1.00
```

```r
library(corrplot)
```

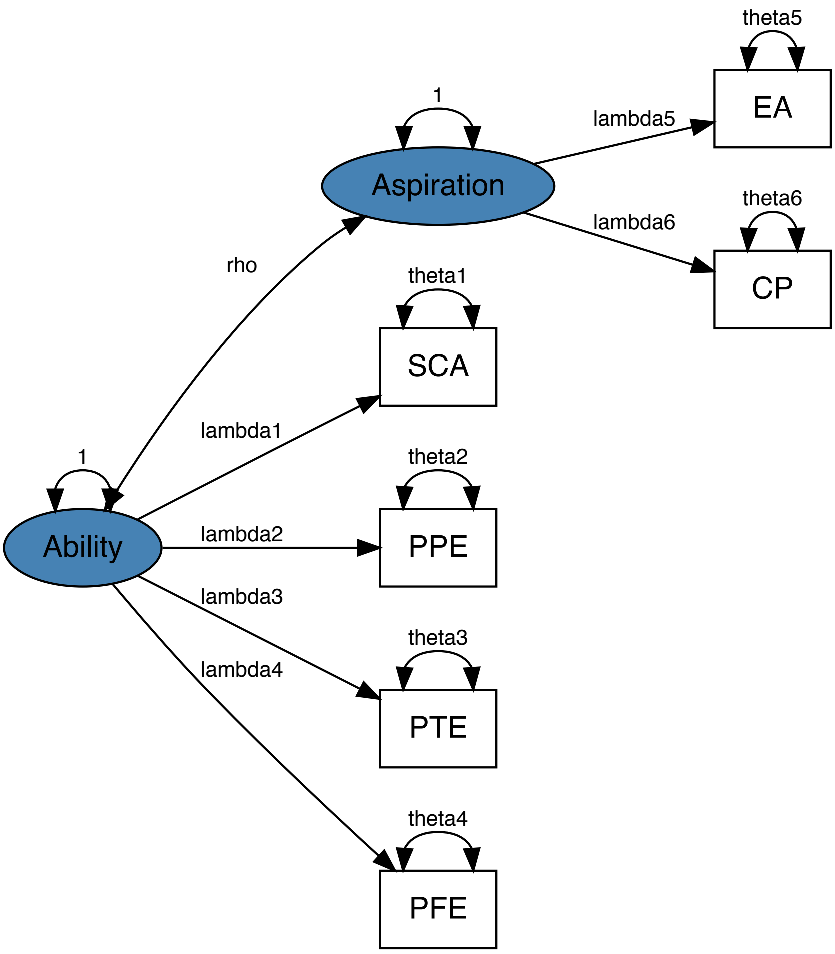
```
## Warning: package 'corrplot' was built under R version 3.5.2
```

```
## corrplot 0.84 loaded
```

```r
corrplot(ability, method = "number", cl.pos = "n")
```

	SCA	PPE	PTE	PFE	EA	CP
SCA	1	0.73	0.7	0.58	0.46	0.56
PPE	0.73	1	0.68	0.61	0.43	0.52
PTE	0.7	0.68	1	0.57	0.4	0.48
PFE	0.58	0.61	0.57	1	0.37	0.41
EA	0.46	0.43	0.4	0.37	1	0.72
CP	0.56	0.52	0.48	0.41	0.72	1

Calsyn and Kenny (1977) postulated that there are two factors, and they relate to the manifest variables as follows:



The variables in the ellipses are factors, and the variables in the squares are manifest variables. Confirmatory factor analysis can be used here to formally test whether this specific factor model fit the data well.

Recall that in exploratory factor analysis the loadings matrix is not unique (we can rotate them to get the same communality and uniqueness, but get different interpretation). In general, the overall EFA model

$$\mathbf{\Sigma} = \Lambda\Lambda^T + \Psi,$$

cannot be appropriately identified, as we need to estimate all of the parameters in Λ. In contrast, in CFA, by imposing a specific structure (specified by a hypothesis) on Λ, we fix some parameters to be zero (e.g., in the example above, there is no arrow from `Ability` to `EA`, and thus the corresponding loading is set to zero), and decrease the number of the parameters we need to estimate.

The remaining parameters are estimate by the **maximum likelihood (MLE)** approach, and a **chi-squared test** is used to assess the goodness of fit.

Model fitting in R

Let us fit the model in the example above in R. We will use the package `sem`.

The proposed factor model is shown below. The factors f_1 and f_2 are `Ability` and `Aspiration`, respectively.

$$\begin{array}{lclclcl}
CA & = & \lambda_{11} f_1 & & & + & u_1; \\
PPE & = & \lambda_{21} f_1 & & & + & u_2; \\
PTE & = & \lambda_{31} f_1 & & & + & u_3; \\
PFE & = & \lambda_{41} f_1 & & & + & u_4; \\
AE & = & & & \lambda_{52} f_2 & + & u_5; \\
CP & = & & & \lambda_{62} f_2 & + & u_6;
\end{array}$$

In addition, $E(f_1) = E(f_2) = 0$ and $var(f_1) = var(f_2) = 1$. We also assume that the specific variances of the six variables are $\psi_1, \ldots, \psi_6$, respectively. The postulated model also assumes correlated factors, that is, $cor(f_1, f_2) = \rho$, where ρ needs to be estimated.

First we create the file **[`ability_model_first.txt`] (ability_model_first.txt)** file containing the model specified above. The contents of the file is shown below.

```
## Specification of Ability factor
Ability     -> SCA, lambda11, NA
Ability     -> PPE, lambda21, NA
Ability     -> PTE, lambda31, NA
Ability     -> PFE, lambda41, NA
## Specification of Aspiration factor
Aspiration  -> EA, lambda52, NA
Aspiration  -> CP, lambda62, NA
## Uniquenesses for each variable
SCA        <-> SCA, psi1, NA
PPE        <-> PPE, psi2, NA
PTE        <-> PTE, psi3, NA
PFE        <-> PFE, psi4, NA
EA         <-> EA, psi5, NA
CP         <-> CP, psi6, NA
## Fixed variances for the two factors
Ability    <-> Ability, NA, 1
Aspiration <-> Aspiration, NA, 1
## Correlation between two factors
Ability    <-> Aspiration, rho, NA
```

Each line in the file represents one arrow/one equation in the factor model.

- The line starting with ## is a comment line. This is optional; used only by the programmaer to comment the code (still highly recommended)
- The line afterward represents one arrow/one equation. For example, the line `Ability -> SCA, lambda11, NA` represents the equation $SCA = \lambda_{11} f_1 + u_1$; the last `NA` tells the program that λ_{11} is not fixed and needs to be estimated.
- The last two line represent the assumption on the factors. For example, the line `Aspiration <-> Aspiration, NA, 1` corresponds to the assumption $var(f_2) = 1$. The second `NA` tells the program that variannce of the factor is not a parameter that needs to be estimated; the last enty 1 says that the variance is fixed at 1.

In general, each line has the following structure:

Path	**Parameter**	**Value**
`Ability -> SCA`	lambda11	NA
`SCA <-> SCA`	psi1	NA
`Aspiration <-> Aspiration`	NA	1

Next we fit the model using the `sem` package. The `specify.model()` function reads the model specification file.

```r
library(sem)
## Input the model in R
ability_model <- specifyModel(file = "ability_model_first.txt")
```

```
## NOTE: it is generally simpler to use specifyEquations() or cfa()
##       see ?specifyEquations
```

```r
ability_model
```

```
##    Path                         Parameter StartValue
## 1  Ability     -> SCA           lambda11
## 2  Ability     -> PPE           lambda21
## 3  Ability     -> PTE           lambda31
## 4  Ability     -> PFE           lambda41
## 5  Aspiration  -> EA            lambda52
## 6  Aspiration  -> CP            lambda62
## 7  SCA        <-> SCA           psi1
## 8  PPE        <-> PPE           psi2
## 9  PTE        <-> PTE           psi3
## 10 PFE        <-> PFE           psi4
## 11 EA         <-> EA            psi5
## 12 CP         <-> CP            psi6
## 13 Ability    <-> Ability    <fixed>   1
## 14 Aspiration <-> Aspiration <fixed>   1
## 15 Ability    <-> Aspiration rho
```

Then the model is fitted using the `sem()` function. At the least, it needs the model, the correlation matrix, and sample size.

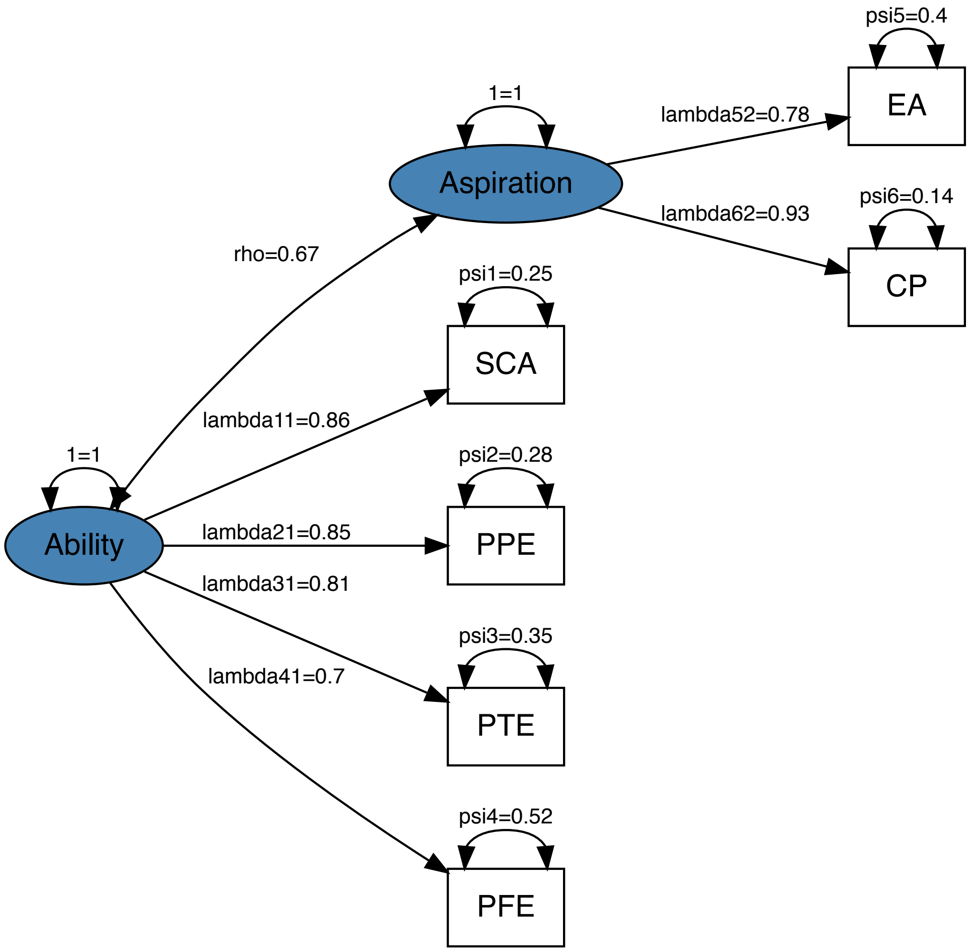
```r
ability_sem <- sem::sem(model = ability_model,  ## factor model
                   S = ability,            ## Correlation matrix
                   N = 556                 ## Sample size
                   )
summary(ability_sem)
```

```
##
##  Model Chisquare =  9.255732   Df =  8 Pr(>Chisq) = 0.3211842
##  AIC =  35.25573
##  BIC =  -41.31041
##
##  Normalized Residuals
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.4409685 -0.1870306 -0.0000018 -0.0130992  0.2107128  0.5333068
##
##  R-square for Endogenous Variables
##    SCA    PPE    PTE    PFE     EA     CP
## 0.7451 0.7213 0.6482 0.4834 0.6008 0.8629
##
##  Parameter Estimates
##          Estimate  Std Error  z value   Pr(>|z|)
## lambda11 0.8632049 0.03514508 24.561188 3.284552e-133
## lambda21 0.8493226 0.03545022 23.958178 7.593661e-127
## lambda31 0.8050861 0.03640470 22.114892 2.272503e-108
## lambda41 0.6952671 0.03863370 17.996387  2.079489e-72
## lambda52 0.7750850 0.04035675 19.205834  3.307658e-82
## lambda62 0.9289304 0.03940959 23.571177 7.615270e-123
## psi1     0.2548772 0.02336722 10.907470  1.061704e-27
## psi2     0.2786512 0.02412754 11.549097  7.460043e-31
## psi3     0.3518366 0.02691875 13.070321  4.865973e-39
## psi4     0.5166036 0.03472534 14.876847  4.659431e-50
## psi5     0.3992432 0.03819583 10.452535  1.426604e-25
## psi6     0.1370884 0.04350459  3.151126  1.626425e-03
## rho      0.6663697 0.03095414 21.527645 8.578257e-103
##
## lambda11 SCA <--- Ability
## lambda21 PPE <--- Ability
## lambda31 PTE <--- Ability
## lambda41 PFE <--- Ability
## lambda52 EA <--- Aspiration
## lambda62 CP <--- Aspiration
## psi1     SCA <--> SCA
## psi2     PPE <--> PPE
## psi3     PTE <--> PTE
## psi4     PFE <--> PFE
## psi5     EA <--> EA
## psi6     CP <--> CP
## rho      Aspiration <--> Ability
##
##  Iterations =  29
```

We can graph the results by using the `DiagrammeR` package. The first function `pathDiagram()` creates a grpah output that can be plotted.

```
pathDiagram(ability_sem,     ## output from `sem()` fit
            ignore.double = FALSE,   ## whether to suppress the variances
            edge.labels = "both",   ## Put both the name and estimated value of edge
 labels
            file = "ability_seb_fitted",  ## Output file name
            output.type = "dot",   ## Output file extension
            node.colors = c("steelblue", "transparent"))  ## Node colors

# Load the DiagrammeR library
library(DiagrammeR)
# Plot the estimated graph
grViz("ability_seb_fitted.dot")
```



The p-value is 0.32, which indicates a good model fit.

Main page: **ST 437/537: Applied Multivariate and Longitudinal Data Analysis (https://maityst537.wordpress.ncsu.edu/)**