lavaan and the history of structural equation modeling

Yves Rosseel
Department of Data Analysis
Ghent University

Psychoco 2012
February 9–10, 2012 – Universität Innsbruck, Austria

Overview

1. what is lavaan; news and updates
2. the history of SEM, from a computational point of view
3. lavaan and the history of SEM

What is lavaan?

- `http://lavaan.org`
- lavaan is an R package for latent variable analysis
- the long-term goal: to provide a collection of tools that can be used to explore, estimate, and understand a wide family of latent variable models, including factor analysis, structural equation, longitudinal, multilevel, latent class, item response, and missing data models
- today: lavaan (version 0.4) is a package for structural equation modeling with continuous data
- one of main attractions of lavaan is its intuitive and easy-to-use model syntax

The lavaan model syntax

```
model.equal <- '
  # measurement model
    ind60 =~ x1 + x2 + x3
    dem60 =~ y1 + a*y2 + b*y3 + c*y4
    dem65 =~ y5 + a*y6 + b*y7 + c*y8

  # regressions
    dem60 ~ ind60
    dem65 ~ ind60 + dem60

  # residual covariances
    y1 ~~ y5
    y2 ~~ y4 + y6
    y3 ~~ y7
    y4 ~~ y8
    y6 ~~ y8
'
fit.equal <- sem(model.equal, data=PoliticalDemocracy)
summary(fit.equal)
```

The lavaan parameter table

```
    id   lhs op   rhs user group free ustart exo label eq.id unco
1    1 ind60 =~    x1    1     1    0      1   0           0    0
2    2 ind60 =~    x2    1     1    1     NA   0           0    1
3    3 ind60 =~    x3    1     1    2     NA   0           0    2
4    4 dem60 =~    y1    1     1    0      1   0           0    0
5    5 dem60 =~    y2    1     1    3     NA   0     a     5    3
6    6 dem60 =~    y3    1     1    4     NA   0     b     6    4
7    7 dem60 =~    y4    1     1    5     NA   0     c     7    5
8    8 dem65 =~    y5    1     1    0      1   0           0    0
9    9 dem65 =~    y6    1     1    3     NA   0     a     5    6
10 10 dem65 =~    y7    1     1    4     NA   0     b     6    7
11 11 dem65 =~    y8    1     1    5     NA   0     c     7    8
12 12 dem60  ~ ind60    1     1    6     NA   0           0    9
13 13 dem65  ~ ind60    1     1    7     NA   0           0   10
14 14 dem65  ~ dem60    1     1    8     NA   0           0   11
15 15    y1 ~~    y5    1     1    9     NA   0           0   12
16 16    y2 ~~    y4    1     1   10     NA   0           0   13
17 17    y2 ~~    y6    1     1   11     NA   0           0   14
18 18    y3 ~~    y7    1     1   12     NA   0           0   15
...
30 30    y7 ~~    y7    0     1   24     NA   0           0   27
31 31    y8 ~~    y8    0     1   25     NA   0           0   28
32 32 ind60 ~~ ind60    0     1   26     NA   0           0   29
33 33 dem60 ~~ dem60    0     1   27     NA   0           0   30
34 34 dem65 ~~ dem65    0     1   28     NA   0           0   31
```


News and updates

linear and nonlinear equality and inequality constraints

```
Data <- data.frame(   y = rnorm(100),
                     x1 = rnorm(100),
                     x2 = rnorm(100),
                     x3 = rnorm(100)  )

model.constr <- ' # model with labeled parameters
                    y ~ b1*x1 + b2*x2 + b3*x3

                  # constraints
                    b1 == (b2 + b3)^2
                    b1 > exp(b2 + b3)
                '

fit <- sem(model.constr, data=Data)
```


defined parameters and mediation analysis

```
X <- rnorm(100)
M <- 0.5*X + rnorm(100)
Y <- 0.7*M + rnorm(100)
Data <- data.frame(X = X, Y = Y, M = M)

model <- ' # direct effect
             Y ~ c*X
           # mediator
             M ~ a*X
             Y ~ b*M

           # indirect effect (a*b)
             ab := a*b
           # total effect
             total := c + (a*b)
         '

fit <- sem(model, data=Data)
```


bootstrapping

```
# The famous Holzinger and Swineford (1939) example
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '

# bootstrapping standard errors
fit <- cfa(HS.model, data=HolzingerSwineford1939, se="bootstrap")

# bootstrapping the test statistic (Bollen-Stine)
fit <- cfa(HS.model, data=HolzingerSwineford1939, test="bootstrap",
           bootstrap=2000, verbose=TRUE)

# bootstrapping anything
fit <- cfa(HS.model, data=HolzingerSwineford1939)

CFI.boot <- bootstrapLavaan(fit, FUN=fitMeasures, R=1000,
                            type="parametric", verbose=TRUE,
                            parallel="multicore", ncups=16,
                            fit.measures="cfi")
```

The history of SEM, from a computational point of view

- several traditions in the SEM (software) world:
  - LISREL (Karl Jöreskog)
  - EQS (Peter Bentler)
  - Mplus (Bengt Muthén)
  - RAM-based approaches (AMOS, Mx, sem, OpenMx, ...)
- superficially, all SEM software packages produce the same results
- there are some subtle (and less subtle) differences in the output
- looking deeper, there are many computational differences

Some differences

- matrix representation
  - standard number of matrices: LISREL: 8; Mplus: 4, EQS: 3, RAM: 2
- optimization algorithm
  - quasi-Newton, gradient-only + quasi-Newton, Gauss-Newton, ...
- variances constrained (strictly positive) versus unrestricted
- constrained optimization algorithm
  - mostly undocumented
  - a Lagrangian-multiplier variant, simple slacks, ...
- normal likelihood versus Wishart likelihood, ML versus GLS-ML (RLS)
  - N versus $N - 1$
  - GLS-ML based chi-square test statistic influences fit measures (CFI!)

Some differences (2)

- Satorra-Bentler/Yuan-Bentler scaled test statistic
  - each program seems to use a different implementation
  - often asymptotically equivalent; but large differences in small samples
- categorical data using the limited information approach
  - Muthén 1984; Jöreskog 1994; Lee, Poon, Bentler (1992)
  - many ways to compute the asymptotic covariance matrix (needed for WLS)
- naive bootstrapping, Bollen-Stine bootstrapping
  - mostly undocumented; one-iteration bootstrap?
  - Bollen-Stine with missing data

...

lavaan and the history of SEM

- lavaan is in many areas still trying to catch up with commercial software; but instead of trying to implement one tradition (based on one program), lavaan tries to implement several traditions
- all fitting functions in lavaan have a `mimic` argument which can be set to `"EQS"` or `"Mplus"` respectively; `"LISREL"` is under development
- this was originally intended to convince users that lavaan could produce 'identical' results as the (commercial) competition
- it is now one of the main design goals of lavaan

lavaan and the future of SEM?

- we need to (re)evaluate old/new/unexplored computational methods in many areas (optimization, constrained inference, Bayesian techniques, limited information estimation, ...)
- lavaan should 'by default' implement best practices in all areas