Introduction to Structural Modelling in R

Lubor Homolka

May 28, 2014

1 CFA - Confirmatory Factor Analysis

CFA is a covariance based method for latent analysis modelling. All latent variables (referred as *constructs*) are related to each other by non-directional link. There are two comfortable ways to assess these relations in R. Library `lavaan` contains function `cfa` with easy to follow guidance how to describe model using path-equation format. In the `sem` library function `cfa` allows several ways how to define model. Some might be useful for more in-depth analysis. Results obtained from both libraries can visualised using two-step procedure utilising `semPlot` package, namely functions `semPlotModel_lavaanModel` for results obtained from `lavaan` library and `semPlotModel` for `sem` results and `semPaths` function. I'll demonstrate CFA on `HBAT` data from Hair's book.

1.1 CFA - library `sem`

In this example five latents constructs are being analysed with no prior knowledge about particular construct relationships. It is only assumed all constructs are related to each other. In this and following sections data are cleaned and imported in the same way.

```
data.sem <- read.table("hair_sem.csv", header=TRUE)
> str(data.sem) # trimmed
'data.frame':   400 obs. of  22 variables:
```

In the following we set the measure model.

```
cfa.hair.model <- cfa()
job: OC1 , OC2 , OC3 , OC4
cow: AC1 , AC2 , AC3 , AC4
env: EP1 , EP2 , EP3 , EP4
int: SI1 , SI2 , SI3 , SI4
com: JS1 , JS2 , JS3 , JS4
```

There are two missing values in the original dataset which are omitted in model matrix.

```
library(sem)
cfa.hair.fit <- sem(cfa.hair.model, data=data.sem)
```

It's also possible to supply `sem` function with covariance matrix and number of observations instead of data sample. Path graph is created employing `semPlot` package.

```
plot.hair <- semPlotModel(cfa.hair.fit)
semPaths(plot.hair, whatLabels="est", layout="spring", edge.label.cex=1)
```

The output of four latent constructs:

Returned results from `sem` are surprisingly concise. Results lack GOF statistics except of p-value of χ^2 test.

```
summary(cfa.hair.fit)
 Model Chisquare =  211.8349   Df =  160 Pr(>Chisq) = 0.003797915
 AIC =  311.8349
```

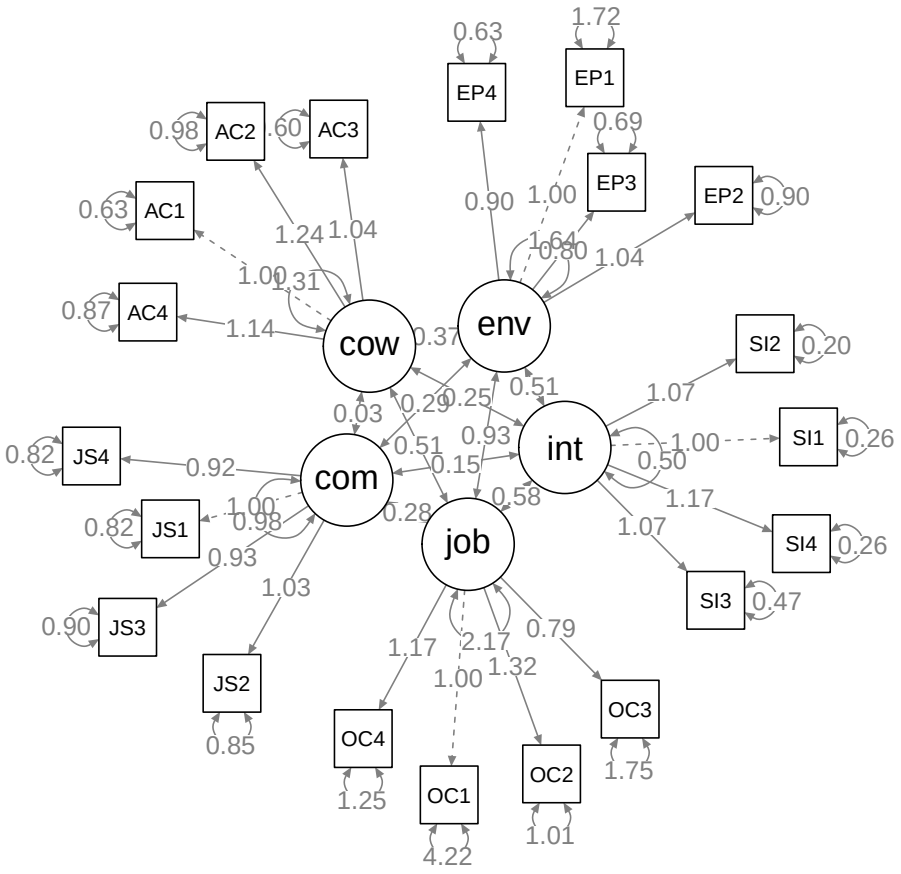

Figure 1: CFA analysis from `sem` library.

```
 BIC =  -745.9975

R-square for Endogenous Variables - AVE for question:
   OC1    OC2    OC3    OC4    AC1    AC2    AC3    AC4
0.3389 0.7886 0.4335 0.7048 0.6746 0.6726 0.6999 0.6631

  Parameter Estimates # Trimmed output
            Estimate  Std Error  z value   Pr(>|z|)
lam[OC3:job] 0.7865337 0.07637945 10.297714 7.215560e-25 OC3 <--- job # indicator - construct
V[job]       2.1650058 0.35912492  6.028559 1.654276e-09 job <--> job # variance of construct
V[OC1]       4.2226746 0.32105136 13.152644 1.643251e-39 OC1 <--> OC1 # variance of indicator
C[cow,com]   0.0304077 0.06667814  0.456037 6.483632e-01 com <--> cow # construct - construct
```

If we compare Hair's results with ours (although we use only four constructs) we do find only minor differences. Hair's estimated loading for OC3 are 0.78 with standard error of 0.08. Our results shows 0.7865 and 0.07637.

1.2 CFA - library `lavaan`

Before proceeding to example using `lavaan` you have to restart your R. Both libraries use same name functions and it might possibly cause unpredictable behaviour. Syntax used by `lavaan` is very simple. It uses $=\sim$ as a symbol for describing dependency relations between constructs and indicators.

```
library(lavaan)

cfa.model <- "
job =~ OC1 + OC2 + OC3 + OC4
cow =~ AC1 + AC2 + AC3 + AC4
env =~ EP1 + EP2 + EP3 + EP4
int =~ SI1 + SI2 + SI3 + SI4
com =~ JS1 + JS2 + JS3 + JS4"

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

In the summarising step fit indices are printed:

```
summary(cfa.fit, fit.measures=TRUE)
```

χ^2 is significant (p-val =0.004), Comparative Fit Index (CFI = 0.987) and Tucker-Lewis Index (TLI = 0.984) are very close to Hair's estimates (both) of 0.99. Confidence interval ($alpha = 0.1$) estimated by `lavaan` ranges from 0.017 to 0.038, while Hair's is 0.015-0.036. All of these results are more restrictive in `lavaan` than in AMOS software.

```
lavaan (0.5-16) converged normally after  52 iterations

                                                  Used       Total
  Number of observations                           398         400

  Estimator                                         ML
  Minimum Function Test Statistic              212.368
  Degrees of freedom                               160
  P-value (Chi-square)                           0.004

Model test baseline model:

  Minimum Function Test Statistic             4200.671
  Degrees of freedom                               190
  P-value                                        0.000
```

```
User model versus baseline model:

  Comparative Fit Index (CFI)                    0.987
  Tucker-Lewis Index (TLI)                       0.984

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -12183.402
  Loglikelihood unrestricted model (H1)     -12077.217

  Number of free parameters                         50
  Akaike (AIC)                               24466.803
  Bayesian (BIC)                             24666.126
  Sample-size adjusted Bayesian (BIC)        24507.474

Root Mean Square Error of Approximation:

  RMSEA                                          0.029
  90 Percent Confidence Interval          0.017  0.038
  P-value RMSEA <= 0.05                          1.000

Standardized Root Mean Square Residual:

  SRMR                                           0.034

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  job =~
    OC1               1.000
    OC2               1.319    0.108   12.193    0.000
    OC3               0.787    0.076   10.311    0.000
    OC4               1.174    0.098   11.961    0.000
  cow =~
    AC1               1.000
    AC2               1.238    0.068   18.307    0.000
    AC3               1.036    0.055   18.769    0.000
    AC4               1.145    0.063   18.142    0.000
  env =~
    EP1               1.000
    EP2               1.035    0.072   14.395    0.000
    EP3               0.805    0.058   13.897    0.000
    EP4               0.897    0.062   14.528    0.000
  int =~
    SI1               1.000
    SI2               1.073    0.055   19.515    0.000
    SI3               1.065    0.066   16.020    0.000
    SI4               1.167    0.061   19.199    0.000
  com =~
    JS1               1.000
    JS2               1.027    0.081   12.635    0.000
```

```
    JS3               0.929    0.077   12.074    0.000
    JS4               0.920    0.075   12.235    0.000

Covariances:
  job ~~
    cow               0.513    0.107    4.813    0.000
    env               0.931    0.144    6.464    0.000
    int               0.574    0.080    7.172    0.000
    com               0.276    0.090    3.057    0.002
  cow ~~
    env               0.370    0.088    4.183    0.000
    int               0.249    0.048    5.139    0.000
    com               0.030    0.066    0.457    0.648
  env ~~
    int               0.508    0.066    7.746    0.000
    com               0.290    0.079    3.649    0.000
  int ~~
    com               0.149    0.043    3.509    0.000

Variances:
    OC1               4.212    0.320
    OC2               1.006    0.147
    OC3               1.746    0.138
    OC4               1.246    0.137
    AC1               0.631    0.060
    AC2               0.976    0.092
    AC3               0.602    0.060
    AC4               0.871    0.081
    EP1               1.718    0.141
    EP2               0.899    0.090
    EP3               0.687    0.063
    EP4               0.628    0.065
    SI1               0.260    0.023
    SI2               0.195    0.021
    SI3               0.466    0.038
    SI4               0.256    0.026
    JS1               0.818    0.081
    JS2               0.851    0.085
    JS3               0.893    0.081
    JS4               0.820    0.076
    job               2.160    0.358
    cow               1.308    0.136
    env               1.637    0.216
    int               0.500    0.053
    com               0.974    0.126
```

1.3 Other statistics

This sub-chapter should be predecessor of previous two because it deals data-quality problem. Here, we will compute reliability (and not only Cronbach's α) and Average Extracted Variance (AVE) statistics.

```
library(semTools)
reliability(cfa.fit)
```

```
            job       cow       env       int       com
alpha  0.8239700 0.8902763 0.8504382 0.8864902 0.8117312
omega  0.8280547 0.8924000 0.8532772 0.8873356 0.8123543
omega2 0.8280547 0.8924000 0.8532772 0.8873356 0.8123543
omega3 0.8193148 0.8923847 0.8535925 0.8872116 0.8123778
```

Alpha coefficient is famous Cronbach's α which suffers from several flaws (well documented in [1]). Omega coefficients are less restrictive in underlying assumptions and are report better results in terms of statistical efficiency. In our data-sample All constructs show high degree of reliability (they exceed 0.7 which is generally agreed rule of thumbs).

AVE can be assess from the `summary` output from `sem` function package in `sem` library. An useful mean of extracting results from `summary` is to use `inspect` function.

```
> inspect(cfa.fit, "rsquare" ) #AVE
      OC1       OC2       OC3       OC4       AC1       AC2       AC3       AC4
0.3389344 0.7886414 0.4335438 0.7047895 0.6746469 0.6725739 0.6999397 0.6630944

      EP1       EP2       EP3       EP4       SI1       SI2       SI3       SI4
0.4879068 0.6611908 0.6068804 0.6773577 0.6581530 0.7467179 0.5492653 0.7265875

      JS1       JS2       JS3       JS4
0.5436664 0.5473573 0.4850054 0.5012040
```

AVE for constructs is the mean value of single indicators AVE's:

```
ave.dat <- data.frame(matrix(ave, ncol=4, byrow=TRUE))

ave.const <- data.frame(AVE=apply(ave.dat, 1, mean)) # compute mean in each row
rownames(ave.const) <- c("Commitment","Coworkers","Environment","Intentions","Satisfaction")

ave.const # print results
```

	AVE
Commitment	0.57
Coworkers	0.68
Environment	0.61
Intensions	0.67
Satisfaction	0.52

References

[1] Dunn, Thomas J., From alpha to omega: A practical solution to the pervasive problem of internal consistency estimation *British Journal of Psychology*, (2013), Retrieved from: `http://onlinelibrary.wiley.com/doi/10.1111/bjop.12046/abstract`.