

BAPS Invariance testing Joachim_Kowalski

Jan Szczypiński

08 10 2021

The MI analysis of BAPS

First loading the packages, data and a function

```
## This is lavaan 0.6-8
## lavaan is FREE software! Please report any bugs.

##
```

```
## #####
```

```
## This is semTools 0.5-4
```

```
## All users of R (or SEM) are invited to submit functions or ideas for functions.
```

```
## #####
```

We create a matrix to compare fit indices across models with different constraints

```
all.results<-matrix(NA, nrow = 4, ncol = 6)
```

```
##Configural invariance
```

We follow the Wu and Estabrook (2016) procedure for calculating invariance for models with variables on categorical scale We use delta parametrization. Groups are split based on psychiatric diagnosis (yes/no);

First we create a syntax for a configural (baseline) model using semTools package

```
model3 = 'survival_strategy =~ BAPS2 + BAPS3 + BAPS4 + BAPS5 + BAPS6
          negative_beliefs =~ BAPS7 + BAPS8 + BAPS9 + BAPS10 + BAPS11 + BAPS12
          normalizing_beliefs =~ BAPS13 + BAPS14 + BAPS15 + BAPS16 + BAPS17 + BAPS18
          BAPS13 ~~ BAPS14'

MI_syntax_conf = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
                               ID.cat = 'Wu.2016', ID.fac = 'unit.variance', parameterization = 'delta',
                               group = "diagnosa", group.equal = 'configural')
```

Next, we fit the configural (baseline model). If the fit indices are ok (Hu and Bentler, 1999) we assume configural invariance

```
cfa.config = cfa(as.character(MI_syntax_conf), data = d, estimator = "WLSMV",
  group = "diagnoza", missing = 'listwise', ordered = TRUE)
fitmeasures(cfa.config, c('chisq.scaled', 'df.scaled', 'cfi.scaled', 'tli.scaled',
  'rmsea.scaled', 'srmr'))
```

```
## chisq.scaled    df.scaled    cfi.scaled    tli.scaled rmsea.scaled    srmr
##      853.276      230.000      0.979      0.976      0.072      0.072
```

We write the fit indices to the matrix created earlier.

```
all.results[1,] = round(data.matrix((fitmeasures(cfa.config, c('chisq.scaled',
  'df.scaled', 'cfi.scaled', 'tli.scaled', 'rmsea.scaled', 'srmr')))), digits = 4)
```

Threshold invariance First, we create a syntax for a model with thresholds constrain.

```
MI_syntax_thresh = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
  ID.cat = 'Wu.2016', ID.fac = 'unit.variance', parameterization = 'delta',
  group = "diagnoza", group.equal = 'thresholds')
```

Next, we fit the model and save fit indices to the matrix

```
cfa.thresholds <- cfa(as.character(MI_syntax_thresh), data = d, estimator = "WLSMV",
  group = "diagnoza", missing = 'listwise', ordered = TRUE)
all.results[2,] = round(data.matrix((fitmeasures(cfa.thresholds, c('chisq.scaled',
  'df.scaled', 'cfi.scaled', 'tli.scaled', 'rmsea.scaled', 'srmr')))), digits = 4)
```

Finally we compare baseline model and threshold model using Satorra-Bentler Scaled χ^2 difference test

```
robust_diff(cfa.thresholds, cfa.config)
```

```
##      Test statistic    p.value
## [1,]      13.61865 0.6938881
```

We see that the models do not differ significantly, thus we can assume threshold invariance.

Metric invariance

First, we create a syntax for a model with thresholds and loadings constrains.

```
MI_syntax_metric = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
  ID.cat = 'Wu.2016', ID.fac = 'unit.variance', parameterization = 'delta',
  group = "diagnoza", group.equal = c('thresholds', 'loadings'))
```

Next, we fit the model and save fit indices to the matrix

```
cfa.metric <- cfa(as.character(MI_syntax_metric), data = d, estimator = "WLSMV",
  group = "diagnoza", missing = 'listwise', ordered = TRUE)
all.results[3,] = round(data.matrix((fitmeasures(cfa.metric, c('chisq.scaled',
  'df.scaled', 'cfi.scaled', 'tli.scaled', 'rmsea.scaled', 'srmr')))), digits = 4)
```

Finally we compare threshold model and metric model using Satorra-Bentler Scaled χ^2 difference test

```
robust_diff(cfa.metric,cfa.thresholds)
```

```
##      Test statistic   p.value  
## [1,]      14.48592 0.4141724
```

Again the test was not significant so we assume metric invariance and proceed to the last step

##Scalar invariance

First, we create a syntax for a model with thresholds, loadings, and intercepts constrains.

```
MI_syntax_scalar = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,  
                                ID.cat = 'Wu.2016', ID.fac = 'unit.variance', parameterization = 'delta',  
                                group = "diagnoza", group.equal = c('thresholds','loadings','intercept'))
```

Next, we fit the model and save fit indices to the matrix

```
cfa.scalar <- cfa(as.character(MI_syntax_scalar), data = d, estimator = "WLSMV",  
                 group = "diagnoza", missing = 'listwise', ordered = TRUE)  
  
all.results[4,] = round(data.matrix((fitmeasures(cfa.scalar,c('chisq.scaled',  
                'df.scaled','cfi.scaled','tli.scaled','rmsea.scaled','srmr')))),digits =4)
```

Finally we compare scalar model and metric model using Satorra-Bentler Scaled χ^2 difference test

```
robust_diff(cfa.scalar,cfa.metric)
```

```
##      Test statistic   p.value  
## [1,]      14.4999 0.4131706
```

```
colnames(all.results) = c('chisq.scaled',  
                          'df.scaled','cfi.scaled','tli.scaled','rmsea.scaled','srmr')  
print(all.results)
```

```
##      chisq.scaled df.scaled cfi.scaled tli.scaled rmsea.scaled  srmr  
## [1,]      853.2764      230      0.9794      0.9757      0.0720 0.0723  
## [2,]      882.1445      247      0.9790      0.9769      0.0702 0.0723  
## [3,]      899.6238      261      0.9789      0.9780      0.0684 0.0723  
## [4,]      908.7241      275      0.9791      0.9793      0.0664 0.0725
```

The test was not significant, thus we can assume scalar invariance.