PQ16 Invariance testing Joachim_Kowalski

Jan Szczypiński

08 10 2021

The MI analysis of PQ16

First loading the packages, data and a function. We use data only from study 2

```
## 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 caluclating 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 = ' factor =~ pq1 + pq2 + pq3 + pq4 + pq5 + pq6 + pq7 + pq8 + pq9 + pq10 +
                     pq11 + pq12 + pq13 + pq14 + pq15 + pq16
pq7 ~~ pq14
pq1 ~~ pq7'

MI_syntax_conf = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
                     ID.cat ='Wu.2016', ID.fac ='unit.variance', parameterization = 'delta',
                     group = "diag", 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 = "diag", 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
##     1886.618      204.000        0.961        0.954        0.049        0.045
```

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)
```

##Treshold 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 = "diag", 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 = "diag", 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 chi^2 difference test

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

```
##      Test statistic   p.value
## [1,]       23.36173 0.1044016
```

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 = "diag", 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 = "diag", 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 chi^2 difference test

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

```
##      Test statistic      p.value
## [1,]       98.30356 2.733098e-14
```

The test was significant so we can't assume metric invariance The next step is to explore which constrained parameter needs to be freed to improve the fit. We use lavTestScore function to print the parameters which have the most influence on the model

```
param = lavTestScore(cfa.metric)$uni
```

```
## Warning in lavTestScore(cfa.metric): lavaan WARNING: se is not 'standard'; not
## implemented yet; falling back to ordinary score test
```

```
head(param[order(param$X2,decreasing =T),],10)
```

```
##
## univariate score tests:
##
##      lhs op    rhs     X2 df p.value
## 1   .p1. == .p117. 91.726  1       0
## 19 .p19. == .p135. 44.944  1       0
## 17 .p17. == .p133. 38.705  1       0
## 53 .p53. == .p169. 19.148  1       0
## 11 .p11. == .p127. 18.511  1       0
## 16 .p16. == .p132. 16.657  1       0
## 13 .p13. == .p129. 15.553  1       0
## 3   .p3. == .p119. 15.112  1       0
## 7   .p7. == .p123. 14.432  1       0
## 54 .p54. == .p170. 14.269  1       0
```

We need to use the parTable function to see what parameter numbers mean

```
head(parTable(cfa.metric),10)
```

```
##    id    lhs op  rhs user block group free ustart exo       label plabel start
## 1   1 factor =~  pq1    1     1     1    1     NA   0  lambda.1_1   .p1. 0.720
## 2   2 factor =~  pq2    1     1     1    2     NA   0  lambda.2_1   .p2. 0.567
## 3   3 factor =~  pq3    1     1     1    3     NA   0  lambda.3_1   .p3. 0.495
## 4   4 factor =~  pq4    1     1     1    4     NA   0  lambda.4_1   .p4. 0.612
## 5   5 factor =~  pq5    1     1     1    5     NA   0  lambda.5_1   .p5. 0.661
## 6   6 factor =~  pq6    1     1     1    6     NA   0  lambda.6_1   .p6. 0.597
## 7   7 factor =~  pq7    1     1     1    7     NA   0  lambda.7_1   .p7. 0.615
## 8   8 factor =~  pq8    1     1     1    8     NA   0  lambda.8_1   .p8. 0.586
## 9   9 factor =~  pq9    1     1     1    9     NA   0  lambda.9_1   .p9. 0.689
## 10 10 factor =~ pq10    1     1     1   10     NA   0 lambda.10_1  .p10. 0.588
##      est    se
## 1  0.562 0.013
## 2  0.533 0.012
## 3  0.547 0.012
```

```
## 4  0.641 0.010
## 5  0.672 0.010
## 6  0.682 0.015
## 7  0.457 0.013
## 8  0.717 0.013
## 9  0.678 0.010
## 10 0.603 0.011
```

We will free the factor =~ pq1 parameter to see if we can get partial metric invariance

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

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

```
cfa.metric.part <- cfa(as.character(MI_syntax_metric.part), data = d, estimator = "WLSMV",
                  group = "diag", missing = 'listwise', ordered = TRUE)

all.results[4,] =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 chi^2 difference test

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

```
##      Test statistic      p.value
## [1,]       63.53148 2.792976e-08
```

The test is still significant so we will look for another parameter

```
param = lavTestScore(cfa.metric.part)$uni
```

```
## Warning in lavTestScore(cfa.metric.part): lavaan WARNING: se is not 'standard';
## not implemented yet; falling back to ordinary score test
```

```
head(param[order(param$X2,decreasing =T),],10)
```

```
##
## univariate score tests:
##
##      lhs op    rhs     X2 df p.value
## 10 .p11. == .p127. 25.325  1   0.000
## 6   .p7. == .p123. 20.064  1   0.000
## 46 .p47. == .p163. 16.933  1   0.000
## 52 .p53. == .p169. 15.645  1   0.000
## 48 .p49. == .p165. 14.427  1   0.000
## 34 .p35. == .p151. 13.494  1   0.000
## 15 .p16. == .p132. 12.520  1   0.000
## 53 .p54. == .p170. 12.517  1   0.000
## 12 .p13. == .p129. 12.231  1   0.000
## 2   .p3. == .p119. 10.660  1   0.001
```

We need to use the parTable function to see what parameter numbers mean

```
head(parTable(cfa.metric.part),10)
```

```
##    id    lhs op  rhs user block group free ustart exo         label plabel
## 1   1 factor =~  pq1    1     1     1    1     NA   0 lambda.1_1.g1   .p1.
## 2   2 factor =~  pq2    1     1     1    2     NA   0    lambda.2_1   .p2.
## 3   3 factor =~  pq3    1     1     1    3     NA   0    lambda.3_1   .p3.
## 4   4 factor =~  pq4    1     1     1    4     NA   0    lambda.4_1   .p4.
## 5   5 factor =~  pq5    1     1     1    5     NA   0    lambda.5_1   .p5.
## 6   6 factor =~  pq6    1     1     1    6     NA   0    lambda.6_1   .p6.
## 7   7 factor =~  pq7    1     1     1    7     NA   0    lambda.7_1   .p7.
## 8   8 factor =~  pq8    1     1     1    8     NA   0    lambda.8_1   .p8.
## 9   9 factor =~  pq9    1     1     1    9     NA   0    lambda.9_1   .p9.
## 10 10 factor =~ pq10    1     1     1   10     NA   0   lambda.10_1  .p10.
##    start   est    se
## 1  0.720 0.578 0.014
## 2  0.567 0.532 0.012
## 3  0.495 0.546 0.012
## 4  0.612 0.640 0.011
## 5  0.661 0.671 0.010
## 6  0.597 0.681 0.015
## 7  0.615 0.456 0.013
## 8  0.586 0.717 0.013
## 9  0.689 0.677 0.010
## 10 0.588 0.602 0.011
```

We will free the factor =~ pq11 parameter to see if we can get partial metric invariance

```
MI_syntax_metric.part = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
                                ID.cat ='Wu.2016', ID.fac ='unit.variance', parameterization = 'delta',
                                 group = "diag", group.equal = c('thresholds','loadings'),
                                group.partial = c('factor   =~  pq1','factor    =~  pq11'))
```

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

```
cfa.metric.part <- cfa(as.character(MI_syntax_metric.part), data = d, estimator = "WLSMV",
                  group = "diag", missing = 'listwise', ordered = TRUE)

all.results =rbind(all.results,round((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 chi^2 difference test

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

```
##      Test statistic      p.value
## [1,]       49.20141 4.084673e-06
```

We will free the factor =~ pq7 parameter to see if we can get partial metric invariance

```
MI_syntax_metric.part = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
                                ID.cat ='Wu.2016', ID.fac ='unit.variance', parameterization = 'delta',
                                 group = "diag", group.equal = c('thresholds','loadings'),
                                group.partial = c('factor   =~  pq1','factor    =~  pq11','factor   =~
```

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

```
cfa.metric.part <- cfa(as.character(MI_syntax_metric.part), data = d, estimator = "WLSMV",
                  group = "diag", missing = 'listwise', ordered = TRUE)

all.results =rbind(all.results,round((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 chi^2 difference test

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

```
##      Test statistic      p.value
## [1,]        37.7483 0.0001688054
```

We will free the factor =~ pq2 parameter to see if we can get partial metric invariance

```
MI_syntax_metric.part = measEq.syntax(configural.model = model3, data = d, ordered = TRUE,
                                ID.cat ='Wu.2016', ID.fac ='unit.variance', parameterization = 'delta',
                                 group = "diag", group.equal = c('thresholds','loadings'),
                                group.partial = c('factor   =~  pq1','factor    =~  pq11','factor   =~
```

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

```
cfa.metric.part <- cfa(as.character(MI_syntax_metric.part), data = d, estimator = "WLSMV",
                  group = "diag", missing = 'listwise', ordered = TRUE)

all.results =rbind(all.results,round((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 chi^2 difference test

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

```
##      Test statistic     p.value
## [1,]       30.34482 0.001398507
```

Even with 4 freely estimated factor loadings the partial metric invariance still couldn't be met. Thus, we conclude that metric invariance is not present for the pq16