Measurement Invariance via Equivalence Testing and Projection Method

Ge Jiang, Yujiao Mai, and Ke-Hai Yuan

Introduction

This supplementary document illustrates the use of the accompanying R package equaltestMI. The sample statistics from Table 1 of Lee and Al Otaiba (2015) are used as an example. The description of the data can be found in the original article and the results obtained from equaltestMI are discussed at length in the published article "Advances in Measurement Invariance and Mean Comparison of Latent Variables: Equivalence Testing and A Projection-Based Approach" <doi: 10.3389/fpsyg.2017.01823>.

The R package equaltestMI is available on CRAN and can be downloaded for use on any R platform with version higher than 3.1.0. Users can pass different arguments to the main function eqMI.main() to examine measurement invariance using the conventional multiple-group approach (NHT) or equivalence testing (ET) approach. Under the framework of equivalence testing, users obtain the minimum tolerable size (T-size) and adjusted cutoff values to evaluate the goodness-of-fit of each invariance test. The projection method is also available for testing the equality of latent means.

Important Notes:

This package is developed for complete dataset with two groups/time points. Use of saturated models might lead to problems in calculation of adjusted RMSEA thresholds. Please consider alternative methods if you have datasets that do not satisfy the requirements.

Part 0: Installation

Users can install the package from CRAN:

```r
## load package
# install.packages("equaltestMI")
library(equaltestMI)
```

or install the most recent version from the maintainer's GitHub repository:

```r
# install.packages("devtools")
# library(devtools)
# devtools::install_github("gabriellajg/equaltestMI", force=TRUE)
library(equaltestMI)
```

Part 1: Sample Means and Covariance Matrices of the Example Data

Load data first:

```
data(LeeAlOtaiba)
# contains sample covariance matrices and sample means of four groups
```

We will use two of the groups as illustration:

```
## group 1 = boys ineligible for free-reduced lunches
Group1 <- LeeAlOtaiba$BoysIneligible
Group1 <- as.matrix(Group1)

## group 2 = boys eligible for free-reduced lunches
Group2 <- LeeAlOtaiba$BoysEligible
Group2 <- as.matrix(Group2)

# sample means:
M1 <- Group1[1,]
M2 <- Group2[1,]

# sample covariance matrices:
Cov1 <- Group1[2:7,]
Cov2 <- Group2[2:7,]
```

Sample statistics for boys ineligible for free-reduced lunches (Group 1): Sample means:

```
#>  Letter_Name Letter_Sound     Blending      Elision   Real_Words Pseudo_Words
#>        45.26        40.45        10.91         6.51        23.88        14.12
```

Sample Covariance Matrix:

```
#>              Letter_Name Letter_Sound Blending Elision Real_Words Pseudo_Words
#> Letter_Name      207.360      159.097   32.589  25.805     61.776       45.075
#> Letter_Sound     159.097      280.228   42.888  36.748     76.123       60.204
#> Blending          32.589       42.888   18.233  10.713     19.051       14.219
#> Elision           25.805       36.748   10.713  20.070     20.372       16.709
#> Real_Words        61.776       76.123   19.051  20.372     73.616       47.429
#> Pseudo_Words      45.075       60.204   14.219  16.709     47.429       44.356
```

Sample statistics for boys eligible for free-reduced lunches (Group 2): Sample means:

```
#>  Letter_Name Letter_Sound     Blending      Elision   Real_Words Pseudo_Words
#>        41.32        34.88         9.08         4.45        19.24        11.07
```

Sample Covariance Matrix:

```
#>              Letter_Name Letter_Sound Blending Elision Real_Words Pseudo_Words
#> Letter_Name      295.840      232.200   38.996  20.174     67.593       57.771
#> Letter_Sound     232.200      324.000   43.164  22.824     77.954       60.458
#> Blending          38.996       43.164   19.010   9.260     23.428       16.272
#> Elision           20.174       22.824    9.260  10.049     15.254       11.042
#> Real_Words        67.593       77.954   23.428  15.254     64.320       38.411
#> Pseudo_Words      57.771       60.458   16.272  11.042     38.411       38.688
```

Part 2: R Script for Obtaining Results from Equivalence Testing and Projection Method

```
## lavaan model syntax
model <- '
AlphabetKnowledge =~ Letter_Name+ Letter_Sound
PhonologicalAwareness =~ Blending + Elision
Spelling =~ Real_Words + Pseudo_Words
'
```

```
## the results using equivalence testing and projection method
## full R output will be presented in Part 3
test <- eqMI.main(model = model,
    sample.nobs = c(78, 174),
    sample.mean = list(M1, M2),
    sample.cov = list(Cov1, Cov2),
    meanstructure = TRUE,
    output = 'both',
    quiet = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE, bootstrap = FALSE)
```

Part 3: Full R Output of the “test” object in Part 2

```
#>
#> ---------- Equality of Population Covariance Matrices under NHT ----------
#>                Chisq Df       pvalue
#> fit.pop.cov 48.85006 21 0.0005261173
#>
#> -------- Chi-Square and Chi-Square-Difference Test under NHT  --------
#>                         Chisq Df   pvalue Chisq.diff Df.diff   pvalue
#> fit.pop.cov            48.850 21    0.001
#> fit.configural.g1       4.408  6    0.622
#> fit.configural.g2      10.641  6    0.100
#> fit.combine.groups     15.049 12
#> fit.metric             20.033 15    0.171      4.984       3    0.173
#> fit.residuals          42.512 21    0.004     22.479       6    0.001
#> fit.varfactor          54.175 27    0.001     11.663       6    0.070
#> fit.scalar             23.732 18    0.164      3.699       3    0.296
#> fit.strong.means       41.066 21    0.006     17.334       3    0.001
#> fit.strict.residuals   45.968 24    0.004     22.237       6    0.001
#> fit.strict.means       63.630 27    0.000     17.662       3    0.001
#>
#> -------- T-size epsilon, RMSEA, and Adjusted Cutoff Values under ET --------
#>                      epsilon_t  RMESA_t   cut.01   cut.05   cut.08   cut.10
#> fit.pop.cov              0.209    0.141    0.076    0.097    0.121    0.139
#> fit.configural.g1        0.028    0.097    0.116    0.133    0.157    0.175
#> fit.configural.g2        0.071    0.154    0.116    0.133    0.157    0.175
#> fit.metric               0.049    0.181    0.151    0.164    0.187    0.205
#> fit.residuals            0.140    0.216    0.116    0.133    0.157    0.175
#> fit.varfactor            0.078    0.161    0.116    0.133    0.157    0.175
#> fit.scalar               0.040    0.163    0.151    0.164    0.187    0.205
#> fit.strong.means         0.125    0.289    0.151    0.164    0.187    0.205
#> fit.strict.residuals     0.138    0.215    0.116    0.133    0.157    0.175
#> fit.strict.means         0.127    0.291    0.151    0.164    0.187    0.205
#>                      goodness-of-fit
#> fit.pop.cov                     poor
#> fit.configural.g1          excellent
#> fit.configural.g2               fair
#> fit.metric                      fair
#> fit.residuals                   poor
#> fit.varfactor               mediocre
#> fit.scalar                     close
#> fit.strong.means                poor
#> fit.strict.residuals            poor
#> fit.strict.means                poor
#>
#>
#> ---------- Means of Latent and Specific Factors by the Projection Method and under NHT ----------
#>                  Chisq Df       pvalue
#> fit.mvmean   19.906793  6 0.0028771810
#> fit.common   18.672371  3 0.0003195302
#> fit.specific  4.163034  3 0.2443890413
#> Validity Index is 0.98856
#>
#> ---------- Means of Latent and Specific Factors by the Projection Method and under ET ----------
#>              epsilon_t  RMESA_t   cut.01   cut.05   cut.08   cut.10
#> fit.mvmean       0.126    0.205    0.116    0.133    0.157    0.175
#> fit.common       0.133    0.298    0.151    0.164    0.187    0.205
#> fit.specific     0.043    0.170    0.151    0.164    0.187    0.205
#>              goodness-of-fit
#> fit.mvmean              poor
#> fit.common              poor
#> fit.specific            fair
#>
```

Part 4: Some Commonly Performed Analysis

1. results using conventional multiple-group SEM approach:

```
test1 <- eqMI.main(model = model,
    sample.nobs = c(78, 174), sample.cov = list(Cov1, Cov2),
    sample.mean = list(M1, M2), meanstructure = TRUE,
    equivalence.test = FALSE, adjRMSEA = FALSE)
```

2. results using conventional multiple-group SEM approach with mean structure tested by projection method:

```
test2 <- eqMI.main(model = model,
    sample.nobs = c(78, 174), sample.cov = list(Cov1, Cov2),
    sample.mean = list(M1, M2), meanstructure = TRUE,
    equivalence.test = FALSE, adjRMSEA = FALSE,
    projection = TRUE)
```

3. results using equivalence testing:

```
test3 <- eqMI.main(model = model,
    sample.nobs = c(78, 174), sample.cov = list(Cov1, Cov2),
```

```r
    sample.mean = list(M1, M2), meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = FALSE)
```

4. results using equivalence testing and adjusted cutoff values:

```r
test4 <- eqMI.main(model = model,
    sample.nobs = c(78, 174), sample.cov = list(Cov1, Cov2),
    sample.mean = list(M1, M2), meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE)
```

5. analysis of 4 with mean structure tested by projection method:

```r
test5 <- eqMI.main(model = model,
    sample.nobs = c(78, 174), sample.cov = list(Cov1, Cov2),
    sample.mean = list(M1, M2), meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE)
```

6. if any of the tests above evaluate only mean structure or only covariance structure, add the argument “structure = ‘mean’ ” or “structure = ‘covariance’ ”, say test5 with only mean structure:

```r
test6 <- eqMI.main(model = model, structure = 'mean',
    sample.nobs = c(78, 174), sample.cov = list(Cov1, Cov2),
    sample.mean = list(M1, M2), meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE)
```

7. if raw data are available, say the name of the raw data is ‘literacy.dat’ and the variable of group membership is ‘FRL’, then test5 is conducted as:

```r
test7 <- eqMI.main(model = model, data = literacy.dat,
    group = "FRL", meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE)
```

8. when raw data are available and project method is used, bootstrap resampling can be used to obtain empirical p-values of the tests conducted by project method by enabling bootstrap = TRUE.

```r
test8 <- eqMI.main(model = model, data = literacy.dat,
    group = "FRL", meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE, bootstrap = TRUE)
```

9. in any of the tests above, the users can suppress the printing of results (not recommended) though the results are still contained in test9$eqMI.stat:

```r
test9 <- eqMI.main(model = model, data = literacy.dat,
    group = "FRL", meanstructure = TRUE,
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE, bootstrap = FALSE,
    quite = TRUE)
```

10. in any of the tests above, the users can allow partial invariance by adding ‘group.partial’:

```r
test10 <- eqMI.main(model = model, data = literacy.dat,
    group = "FRL", meanstructure = TRUE,
    group.partial = c("Spelling=~Real_Words", "Blending~1"),
    equivalence.test = TRUE, adjRMSEA = TRUE,
    projection = TRUE)
```

so that the loadings of ‘Spelling’ on ‘Real_Words’ and the intercept of ‘Blending’ are allowed to vary across groups.

Part 5: Help page of the eqMI.main() function

For a complete view of the help page of function eqMI.main(), please install R package printr and type ?eqMI.main in R console:

```
#> Registered S3 method overwritten by 'printr':
#>   method                from
#>   knit_print.data.frame rmarkdown
#> Rendering development documentation for 'eqMI.main'
```