

Problem with TSSEM

Guoquan (modified by Mike Cheung)

October 06, 2019

<https://openmx.ssri.psu.edu/node/4539>

```
library(metaSEM)

## Loading required package: OpenMx

## To take full advantage of multiple cores, use:
##   mxOption(key='Number of Threads', value=parallel::detectCores()) #now
##   Sys.setenv(OMP_NUM_THREADS=parallel::detectCores()) #before library(OpenMx)

## "SLSQP" is set as the default optimizer in OpenMx.

## mxOption(NULL, "Gradient algorithm") is set at "central".

## mxOption(NULL, "Optimality tolerance") is set at "6.3e-14".

## mxOption(NULL, "Gradient iterations") is set at "2".

# data input
mydatar <- readFullMat(file = "mydata.txt")
mydatar

## $`1`
##      x1    x2    x3 x4
## x1 1.00 0.45 0.65 NA
## x2 0.45 1.00 0.48 NA
## x3 0.65 0.48 1.00 NA
## x4  NA   NA   NA  1
##
## $`2`
##      x1    x2    x3    x4
## x1 1.00 0.56 0.73 0.51
## x2 0.56 1.00 0.76 0.64
## x3 0.73 0.76 1.00 0.70
## x4 0.51 0.64 0.70 1.00
##
## $`3`
##      x1      x2      x3      x4
## x1 1.0000 0.6713 0.7797 0.7600
## x2 0.6713 1.0000 0.7352 0.6512
## x3 0.7797 0.7352 1.0000 0.7234
## x4 0.7600 0.6512 0.7234 1.0000
##
## $`4`
##      x1    x2    x3    x4
## x1 1.000 0.567 0.504 0.477
## x2 0.567 1.000 0.712 0.738
## x3 0.504 0.712 1.000 0.689
## x4 0.477 0.738 0.689 1.000
```

```
##
## $`5`
##      x1    x2    x3 x4
## x1 1.00 0.17 0.47 NA
## x2 0.17 1.00 0.40 NA
## x3 0.47 0.40 1.00 NA
## x4   NA   NA   NA  1
##
## $`6`
##      x1    x2    x3 x4
## x1 1.00 0.79 0.83 NA
## x2 0.79 1.00 0.83 NA
## x3 0.83 0.83 1.00 NA
## x4   NA   NA   NA  1
##
## $`7`
##      x1    x2    x3 x4
## x1 1.00 0.80 0.83 NA
## x2 0.80 1.00 0.88 NA
## x3 0.83 0.88 1.00 NA
## x4   NA   NA   NA  1
##
## $`8`
##      x1    x2    x3    x4
## x1 1.00 0.64 0.71 0.60
## x2 0.64 1.00 0.60 0.63
## x3 0.71 0.60 1.00 0.71
## x4 0.60 0.63 0.71 1.00

mydatan <- c(477, 318, 630, 256, 189, 379, 377, 1475)
mydatan

## [1] 477 318 630 256 189 379 377 1475

# Add names in mydatan matrices
mydatan <- lapply(mydatan, function(x) {
  dimnames(x) <- list(c("IN", "EN", "ADV", "ATT"), c("IN", "EN", "ADV", "ATT"))
  x})
head(mydatan)

## $`1`
##      IN    EN    ADV ATT
## IN  1.00 0.45 0.65  NA
## EN  0.45 1.00 0.48  NA
## ADV 0.65 0.48 1.00  NA
## ATT  NA   NA   NA   1
##
## $`2`
##      IN    EN    ADV ATT
## IN  1.00 0.56 0.73 0.51
## EN  0.56 1.00 0.76 0.64
## ADV 0.73 0.76 1.00 0.70
## ATT 0.51 0.64 0.70 1.00
##
## $`3`
```

```
##          IN      EN      ADV      ATT
## IN  1.0000 0.6713 0.7797 0.7600
## EN  0.6713 1.0000 0.7352 0.6512
## ADV 0.7797 0.7352 1.0000 0.7234
## ATT 0.7600 0.6512 0.7234 1.0000
```

```
##
```

```
## $`4`
```

```
##          IN      EN      ADV      ATT
## IN  1.000 0.567 0.504 0.477
## EN  0.567 1.000 0.712 0.738
## ADV 0.504 0.712 1.000 0.689
## ATT 0.477 0.738 0.689 1.000
```

```
##
```

```
## $`5`
```

```
##          IN      EN      ADV      ATT
## IN  1.00 0.17 0.47  NA
## EN  0.17 1.00 0.40  NA
## ADV 0.47 0.40 1.00  NA
## ATT  NA  NA  NA  1
```

```
##
```

```
## $`6`
```

```
##          IN      EN      ADV      ATT
## IN  1.00 0.79 0.83  NA
## EN  0.79 1.00 0.83  NA
## ADV 0.83 0.83 1.00  NA
## ATT  NA  NA  NA  1
```

```
## First stage analysis
```

```
random1 <- tssem1(mydatar, mydatan, method="REM", RE.type="Diag")
```

```
## Use "autofixtau2 = TRUE" to fix variances that are problematic.
```

```
## You may need the development version of metaSEM by installing it from Github:
```

```
## devtools::install_github("mikewllcheung/metaseM")
```

```
random1 <- rerun(random1, autofixtau2 = TRUE)
```

```
## Running TSSEM1 Correlation with 12 parameters
```

```
##
```

```
## Beginning initial fit attempt
```

```
## Running TSSEM1 Correlation with 12 parameters
```

```
##
```

```
## Lowest minimum so far: -61.6981326059496
```

```
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
```

```
## Not all eigenvalues of the Hessian are positive: 1254523.07808441, 35073.4729856586, 33508.67453987
```

```
##
```

```
## Beginning fit attempt 1 of at maximum 10 extra tries
```

```
## Running TSSEM1 Correlation with 12 parameters
```

```
##
```

```
## Lowest minimum so far: -61.6983344219983
```

```
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
```

```
## Not all eigenvalues of the Hessian are positive: 1249769.68667915, 35057.0580393364, 33535.04550450
```

```

##
## Beginning fit attempt 2 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Lowest minimum so far: -61.6983406168292
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
## Not all eigenvalues of the Hessian are positive: 1245610.08221266, 34989.4309970534, 33428.17949840
##
## Beginning fit attempt 3 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Lowest minimum so far: -61.7205093243637
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
## Not all eigenvalues of the Hessian are positive: 1102762.79574878, 34821.7323090387, 33534.80206574
##
## Beginning fit attempt 4 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Fit attempt worse than current best: -61.7189200101632 vs -61.7205093243637
##
## Beginning fit attempt 5 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Lowest minimum so far: -61.7229207008224
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
## Not all eigenvalues of the Hessian are positive: 1251134.08660995, 35038.0694197052, 33471.67318973
##
## Beginning fit attempt 6 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Lowest minimum so far: -61.7229264183719
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
## Not all eigenvalues of the Hessian are positive: 1252525.07657483, 35062.0957548975, 33582.61088074
##
## Beginning fit attempt 7 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
## OpenMx status code 6 not in list of acceptable status codes, (0,1)
## Not all eigenvalues of the Hessian are positive: 1252525.07657483, 35062.0957548975, 33582.61088074
##
## Beginning fit attempt 8 of at maximum 10 extra tries

```

```

## Running TSSEM1 Correlation with 12 parameters
##
## Fit attempt worse than current best: -61.6854511944841 vs -61.7229264183719
##
## Beginning fit attempt 9 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Lowest minimum so far: -61.7230204583291
## Not all eigenvalues of the Hessian are positive: 1252691.93200939, 35053.1714831887, 33577.47415138
##
## Beginning fit attempt 10 of at maximum 10 extra tries
## Running TSSEM1 Correlation with 12 parameters
##
## Lowest minimum so far: -61.7230204583292
## Not all eigenvalues of the Hessian are positive: 1252692.709022, 35053.1576860162, 33577.4849424057
##
## Retry limit reached
##
## Retry limit reached; Best fit=-61.72302 (started at -61.697574) (11 attempt(s): 11 valid, 0 errors)
## Running TSSEM1 Correlation with 11 parameters
##
## Beginning initial fit attempt
## Running TSSEM1 Correlation with 11 parameters
##
## Lowest minimum so far: -61.7230209051948
##
## Solution found
##
## Solution found! Final fit=-61.723021 (started at -61.723021) (1 attempt(s): 1 valid, 0 errors)
summary(random1)

##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##   "*Tau2_", 1:no.es, "_", 1:no.es)), RE.lbound = RE.lbound,
##   I2 = I2, model.name = model.name, suppressWarnings = TRUE,
##   silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation (robust=FALSE)
## Coefficients:
##           Estimate Std. Error   lbound   ubound z value
## Intercept1  0.58381151  0.06324956  0.45984465  0.70777837  9.2303
## Intercept2  0.69075643  0.04319090  0.60610382  0.77540903 15.9931
## Intercept3  0.59006723  0.05351673  0.48517637  0.69495809 11.0258
## Intercept4  0.67632267  0.05272848  0.57297675  0.77966859 12.8265

```

```

## Intercept5 0.65544143 0.02425776 0.60789709 0.70298576 27.0199
## Intercept6 0.70971617 0.01515123 0.68002030 0.73941203 46.8421
## Tau2_1_1 0.03096452 0.01626245 -0.00090929 0.06283833 1.9041
## Tau2_2_2 0.01426806 0.00758628 -0.00060078 0.02913691 1.8808
## Tau2_3_3 0.01005948 0.00779437 -0.00521720 0.02533617 1.2906
## Tau2_4_4 0.02139460 0.01122850 -0.00061286 0.04340206 1.9054
## Tau2_5_5 0.00066707 0.00148855 -0.00225043 0.00358457 0.4481
## Pr(>|z|)
## Intercept1 < 2e-16 ***
## Intercept2 < 2e-16 ***
## Intercept3 < 2e-16 ***
## Intercept4 < 2e-16 ***
## Intercept5 < 2e-16 ***
## Intercept6 < 2e-16 ***
## Tau2_1_1 0.05690 .
## Tau2_2_2 0.06000 .
## Tau2_3_3 0.19684
## Tau2_4_4 0.05673 .
## Tau2_5_5 0.65406
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 466.9702
## Degrees of freedom of the Q statistic: 30
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
## Estimate
## Intercept1: I2 (Q statistic) 0.9746
## Intercept2: I2 (Q statistic) 0.9654
## Intercept3: I2 (Q statistic) 0.8709
## Intercept4: I2 (Q statistic) 0.9699
## Intercept5: I2 (Q statistic) 0.3177
## Intercept6: I2 (Q statistic) 0.0000
##
## Number of studies (or clusters): 8
## Number of observed statistics: 36
## Number of estimated parameters: 11
## Degrees of freedom: 25
## -2 log likelihood: -61.72302
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)

## outputs pooled covariance matrix (the parameter estimates) extracted
vec2symMat(coef(random1,select = "fixed"), diag = FALSE)

##      [,1]      [,2]      [,3]      [,4]
## [1,] 1.0000000 0.5838115 0.6907564 0.5900672
## [2,] 0.5838115 1.0000000 0.6763227 0.6554414
## [3,] 0.6907564 0.6763227 1.0000000 0.7097162
## [4,] 0.5900672 0.6554414 0.7097162 1.0000000

## Regression coefficients
A1 <- create.mxMatrix(c(0, 0, 0, 0,
0, 0, 0, 0,

```

```

"0*IN2ADV", "0*EN2ADV", 0, 0,
"0*IN2ATT", "0*EN2ATT", "0*ADV2ATT", 0),
type="Full", byrow=TRUE, ncol=4, nrow=4,
as.mxMatrix=FALSE)

## This step is not necessary but it is useful for inspecting the model.
dimnames(A1)[[1]] <- dimnames(A1)[[2]] <- c("IN","EN","ADV","ATT")
A1

```

```

##      IN      EN      ADV      ATT
## IN  "0"      "0"      "0"      "0"
## EN  "0"      "0"      "0"      "0"
## ADV "0*IN2ADV" "0*EN2ADV" "0"      "0"
## ATT "0*IN2ATT" "0*EN2ATT" "0*ADV2ATT" "0"

```

```

## Covariance matrix among the variables
S1 <- create.mxMatrix(c(1,
"0*CorIN_EN", 1,
0, 0, "0.1*VarADV",
0, 0, 0, "0.1*VarATT"),
byrow=TRUE, type="Symm", as.mxMatrix=FALSE)

```

```

## This step is not necessary but it is useful for inspecting the model.
dimnames(S1)[[1]] <- dimnames(S1)[[2]] <- c("IN","EN", "ADV","ATT")
S1

```

```

##      IN      EN      ADV      ATT
## IN  "1"      "0*CorIN_EN" "0"      "0"
## EN  "0*CorIN_EN" "1"      "0"      "0"
## ADV "0"      "0"      "0.1*VarADV" "0"
## ATT "0"      "0"      "0"      "0.1*VarATT"

```

```

checkRAM(Amatrix=A1, Smatrix=S1, cor.analysis=TRUE)

```

```

## Second stage analysis
random2 <- tssem2(random1, Amatrix=A1, Smatrix=S1, diag.constraints=FALSE,
intervals.type="LB")
summary(random2)

```

```

##
## Call:
## wls(Cov = pooledS, aCov = aCov, n = tssem1.obj$total.n, RAM = RAM,
##      Amatrix = Amatrix, Smatrix = Smatrix, Fmatrix = Fmatrix,
##      diag.constraints = diag.constraints, cor.analysis = cor.analysis,
##      intervals.type = intervals.type, mx.algebras = mx.algebras,
##      model.name = model.name, suppressWarnings = suppressWarnings,
##      silent = silent, run = run)
##
## 95% confidence intervals: Likelihood-based statistic
## Coefficients:
##      Estimate Std.Error  lbound  ubound z value Pr(>|z|)
## IN2ADV    0.44892      NA  0.27581  0.60662    NA     NA
## EN2ADV    0.41424      NA  0.22243  0.58712    NA     NA
## IN2ATT    0.12478      NA -0.13662  0.34046    NA     NA
## EN2ATT    0.29652      NA  0.16888  0.41006    NA     NA
## ADV2ATT   0.42298      NA  0.30239  0.57644    NA     NA

```

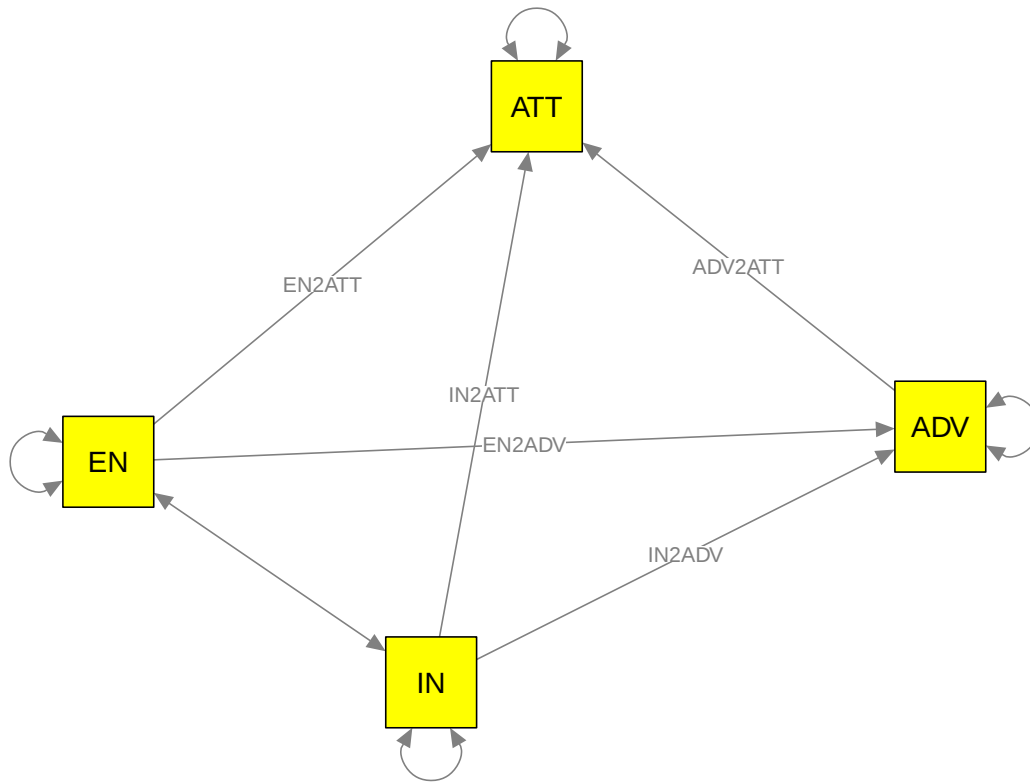
```

## CorIN_EN 0.58381      NA 0.45984 0.70778      NA      NA
##
## Goodness-of-fit indices:
##                                     Value
## Sample size                       4101.0
## Chi-square of target model         0.0
## DF of target model                 0.0
## p value of target model            0.0
## Number of constraints imposed on "Smatrix" 0.0
## DF manually adjusted               0.0
## Chi-square of independence model    2679.7
## DF of independence model           6.0
## RMSEA                             0.0
## RMSEA lower 95% CI                 0.0
## RMSEA upper 95% CI                 0.0
## SRMR                              0.0
## TLI                               -Inf
## CFI                               1.0
## AIC                               0.0
## BIC                               0.0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)

## Plot the model with labels
## The labels are overlapped. We may modify it by using layout="spring"
plot(random2, whatLabels="path", nCharEdges=10, nCharNodes=10,
      layout="spring", color="yellow", edge.label.cex=0.8)

## Registered S3 methods overwritten by 'huge':
##   method      from
##   plot.sim    BDgraph
##   print.sim   BDgraph

```



```
plot(random2, layout="spring", color="green")
```

