

Variance components in OpenMx (corrected)

Mike Cheung

April 02, 2022

One-variable model

- The estimated variances are very close to the population variances:
 - Between-level population variance is 10, and the estimated variance is 10.50.
 - Within-level population variance is 5, and the estimated variance is 5.00.

```
library(OpenMx)
library(MASS)

set.seed(294823)

## Generate level-2 data
k2 <- 1000 # no. of level-2 units

## Between-level variance
SigmaB <- matrix(10)

dfL2 <- mvrnorm(n=k2, mu=0, Sigma=SigmaB)

## Generate level-1 data
k1 <- 100 # no. of level-1 units per level-2 unit, assuming balanced data

## Within-level variance
SigmaW <- matrix(5)

dfL1 <- lapply(split(dfL2, 1:k2),
               function(x) mvrnorm(n=k1, mu=x, Sigma=SigmaW))

## Create a dataset
df <- data.frame(y1=unlist(dfL1, recursive = FALSE))

## Add a cluster label
df$cluster <- as.factor(paste0("c", rep(1:k2, each=k1)))

head(df)

##           y1 cluster
## 1 3.836011      c1
## 2 6.698428      c1
```

```

## 3 5.157043      c1
## 4 6.759612      c1
## 5 4.475435      c1
## 6 5.214640      c1

## Create between data
dfB <- data.frame(cluster=unique(df[, "cluster"]))

## Between model
lat_B <- c("u1")
p <- length(lat_B)
mod_B = mxModel("B", type = "RAM", latentVars = lat_B,
  mxData(dfB, "raw", primaryKey = "cluster"),
  mxMatrix("Diag", p, p, free=TRUE, values=0.1,
    labels=c("taub"), lbound = 0,
    name = "TauB", dimnames = list(lat_B, lat_B)),
  mxMatrix("Zero", p, p, name = "A", dimnames = list(lat_B, lat_B)),
  mxMatrix("Zero", 0, p, name = "F", dimnames = list(NULL, lat_B)),
  mxMatrix("Zero", 1, p, name = "M", dimnames = list(NULL, lat_B)),
  mxExpectationRAM("A", "TauB", "F", "M"))

## Within model
man_W <- c("y1")
mod_W = mxModel("W", type = "RAM", mod_B,
  manifestVars = man_W,
  mxData(df, "raw"),
  mxMatrix("Diag", p, p, free=TRUE, values=0.1,
    labels=c("tauw"), lbound = 0,
    name = "TauW", dimnames = list(man_W, man_W)),
  mxMatrix("Zero", p, p, name = "A", dimnames = list(man_W, man_W)),
  mxMatrix("Iden", p, p, name = "F", dimnames = list(man_W, man_W)),
  mxMatrix("Full", 1, p, T, 0, name = "M", dimnames = list(1, man_W)),
  mxMatrix("Diag", p, p, F, 1, name = "T",
    joinKey = "cluster", joinModel = "B", dimnames = list(man_W, lat_B)),
  mxExpectationRAM("A", "TauW", "F", "M", between = "T"))

fit1 = mxRun(mod_W)
summary(fit1)

## Summary of W
##
## free parameters:
##      name matrix row col   Estimate Std.Error A lbound ubound
## 1   tauw  TauW  y1  y1  5.00373034  0.02248999      0
## 2 W.M[1,1]      M   1  y1 -0.02850115  0.10270172
## 3   taub B.TauB  u1  u1 10.50110397  0.47185752      0
##
## Model Statistics:
##           | Parameters | Degrees of Freedom | Fit (-2lnL units)
##      Model:           3           99997           450157.3
##      Saturated:           NA           NA           NA
##      Independence:           NA           NA           NA
##      Number of observations/statistics: 101000/100000
##
## Information Criteria:

```

```
##          | df Penalty | Parameters Penalty | Sample-Size Adjusted
## AIC:      250163.3          450163.3          450163.3
## BIC:      -702095.7          450191.9          450182.3
## CFI: NA
## TLI: 1    (also known as NNFI)
## RMSEA: 0 [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2022-04-02 18:56:56
## Wall clock time: 1.456449 secs
## optimizer: SLSQP
## OpenMx version number: 2.20.6
## Need help? See help(mxSummary)
```

Two-variable model

- The estimated variances are close to the population variances:
 - Between-level population variances are both 10 whereas the estimated variances are 10.05 and 10.77.
 - Within-level population variances are both 5 whereas the estimated variances are 4.99 and 5.00.

```
## Generate level-2 data
k2 <- 1000 # no. of level-2 units

## Between-level variance-covariance matrix
SigmaB <- diag(c(10, 10))

dfL2 <- mvrnorm(n=k2, mu=c(0, 0), Sigma=SigmaB)

## Generate level-1 data
k1 <- 100 # no. of level-1 units per level-2 unit, assuming balanced data

## Within-level variance-covariance matrix
SigmaW <- diag(c(5, 5))

dfL1 <- lapply(split(dfL2, 1:k2),
               function(x) mvrnorm(n=k1, mu=x, Sigma=SigmaW))

## Create a dataset
df <- Reduce(rbind, dfL1)
colnames(df) <- c("y1", "y2")
df <- as.data.frame(df)

## Add a cluster label
df$cluster <- as.factor(paste0("c", rep(1:k2, each=k1)))

head(df)

##          y1          y2 cluster
## 1 9.3354996 2.9676412      c1
## 2 7.4128269 1.0205451      c1
## 3 0.7962157 -0.6011602      c1
```

```
## 4 6.8976459 -4.2873867 c1
## 5 4.1220756 0.5734548 c1
## 6 3.5001996 -0.4353690 c1
```

```
## Create between data
dfB <- data.frame(cluster=unique(df[, "cluster"]))

## Between model
lat_B <- c("u1", "u2")
p <- length(lat_B)
mod_B = mxModel("B", type = "RAM", latentVars = lat_B,
  mxData(dfB, "raw", primaryKey = "cluster"),
  mxMatrix("Diag", p, p, free=TRUE, values=0.1,
    labels=c("taub1", "taub2"), lbound = 0,
    name = "TauB", dimnames = list(lat_B, lat_B)),
  mxMatrix("Zero", p, p, name = "A", dimnames = list(lat_B, lat_B)),
  mxMatrix("Zero", 0, p, name = "F", dimnames = list(NULL, lat_B)),
  mxMatrix("Zero", 1, p, name = "M", dimnames = list(NULL, lat_B)),
  mxExpectationRAM("A", "TauB", "F", "M"))

## Within model
man_W <- c("y1", "y2")
mod_W = mxModel("W", type = "RAM", mod_B,
  manifestVars = man_W,
  mxData(df, "raw"),
  mxMatrix("Diag", p, p, free=TRUE, values=0.1,
    labels=c("tauw1", "tauw2"), lbound = 0,
    name = "TauW", dimnames = list(man_W, man_W)),
  mxMatrix("Zero", p, p, name = "A", dimnames = list(man_W, man_W)),
  mxMatrix("Iden", p, p, name = "F", dimnames = list(man_W, man_W)),
  mxMatrix("Full", 1, p, T, 0, name = "M", dimnames = list(1, man_W)),
  mxMatrix("Diag", p, p, F, 1, name = "T",
    joinKey = "cluster", joinModel = "B", dimnames = list(man_W, lat_B)),
  mxExpectationRAM("A", "TauW", "F", "M", between = "T"))

fit2 = mxRun(mod_W)
summary(fit2)
```

```
## Summary of W
##
## free parameters:
##      name matrix row col  Estimate  Std.Error A lbound ubound
## 1  tauw1  TauW  y1  y1  4.9857221  0.02240912      0
## 2  tauw2  TauW  y2  y2  4.9985456  0.02246666      0
## 3 W.M[1,1]      M   1  y1  0.1009189  0.10043542  !
## 4 W.M[1,2]      M   1  y2 -0.0919679  0.10406222
## 5  taub1 B.TauB  u1  u1  10.0490257  0.45162894      0
## 6  taub2 B.TauB  u2  u2  10.7657251  0.48389966  !      0
##
## Model Statistics:
##      | Parameters | Degrees of Freedom | Fit (-2lnL units)
##      Model:           6                199994                899836
##      Saturated:         NA                NA                NA
##      Independence:      NA                NA                NA
##      Number of observations/statistics: 101000/200000
```

```
##
## Information Criteria:
##      | df Penalty | Parameters Penalty | Sample-Size Adjusted
## AIC:      499848      899848.0      899848.0
## BIC:      -1404670      899905.1      899886.1
## CFI: NA
## TLI: 1 (also known as NNFI)
## RMSEA: 0 [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2022-04-02 18:56:59
## Wall clock time: 1.749797 secs
## optimizer: SLSQP
## OpenMx version number: 2.20.6
## Need help? See help(mxSummary)
```

Three-variable model

- The estimated variances are close to the population variances:
 - Between-level population variances are both 10 whereas the estimated variances are 9.17 to 10.61.
 - Within-level population variances are both 5 whereas the estimated variances are both around 5.00.

```
## Generate level-2 data
k2 <- 1000 # no. of level-2 units

## Between-level variance-covariance matrix
SigmaB <- diag(c(10, 10, 10))

dfL2 <- mvrnorm(n=k2, mu=c(0, 0, 0), Sigma=SigmaB)

## Generate level-1 data
k1 <- 100 # no. of level-1 units per level-2 unit, assuming balanced data

## Within-level variance-covariance matrix
SigmaW <- diag(c(5, 5, 5))

dfL1 <- lapply(split(dfL2, 1:k2),
               function(x) mvrnorm(n=k1, mu=x, Sigma=SigmaW))

## Create a dataset
df <- Reduce(rbind, dfL1)
colnames(df) <- c("y1", "y2", "y3")
df <- as.data.frame(df)

## Add a cluster label
df$cluster <- as.factor(paste0("c", rep(1:k2, each=k1)))

head(df)
```

```
##      y1      y2      y3 cluster
## 1  3.6683994 -0.1532471 -3.31868052      c1
```

```
## 2 -1.2971962  2.0656157 -1.89555706    c1
## 3 -3.0376843  1.4608318 -0.01115027    c1
## 4  0.2784518 -4.0620976  1.45154928    c1
## 5  1.1896494  1.0003728 -2.05436912    c1
## 6 -1.1094104  1.0980635 -0.23295086    c1
```

```
## Create between data
dfB <- data.frame(cluster=unique(df[, "cluster"]))

## Between model
lat_B <- c("u1", "u2", "u3")
p <- length(lat_B)
mod_B = mxModel("B", type = "RAM", latentVars = lat_B,
  mxData(dfB, "raw", primaryKey = "cluster"),
  mxMatrix("Diag", p, p, free=TRUE, values=0.1,
    labels=c("taub1", "taub2", "taub3"), lbound = 0,
    name = "TauB", dimnames = list(lat_B, lat_B)),
  mxMatrix("Zero", p, p, name = "A", dimnames = list(lat_B, lat_B)),
  mxMatrix("Zero", 0, p, name = "F", dimnames = list(NULL, lat_B)),
  mxMatrix("Zero", 1, p, name = "M", dimnames = list(NULL, lat_B)),
  mxExpectationRAM("A", "TauB", "F", "M"))

## Within model
man_W <- c("y1", "y2", "y3")
mod_W = mxModel("W", type = "RAM", mod_B,
  manifestVars = man_W,
  mxData(df, "raw"),
  mxMatrix("Diag", p, p, free=TRUE, values=0.1,
    labels=c("tauw1", "tauw2", "tauw3"), lbound = 0,
    name = "TauW", dimnames = list(man_W, man_W)),
  mxMatrix("Zero", p, p, name = "A", dimnames = list(man_W, man_W)),
  mxMatrix("Iden", p, p, name = "F", dimnames = list(man_W, man_W)),
  mxMatrix("Full", 1, p, T, 0, name = "M", dimnames = list(1, man_W)),
  mxMatrix("Diag", p, p, F, 1, name = "T",
    joinKey = "cluster", joinModel = "B", dimnames = list(man_W, lat_B)),
  mxExpectationRAM("A", "TauW", "F", "M", between = "T"))

fit3 = mxRun(mod_W)
summary(fit3)
```

```
## Summary of W
```

```
##
```

```
## free parameters:
```

##	name	matrix	row	col	Estimate	Std.Error	A	lbound	ubound
## 1	tauw1	TauW	y1	y1	5.016274651	0.02254669		0	
## 2	tauw2	TauW	y2	y2	5.009092562	0.02251437		0	
## 3	tauw3	TauW	y3	y3	4.998192984	0.02246536		0	
## 4	W.M[1,1]	M	1	y1	-0.056651580	0.10075232	!		
## 5	W.M[1,2]	M	1	y2	-0.003615876	0.10321026			
## 6	W.M[1,3]	M	1	y3	-0.019588461	0.09584508	!		
## 7	taub1	B.TauB	u1	u1	10.158141274	0.45628763	!	0	
## 8	taub2	B.TauB	u2	u2	10.607404873	0.47654775	!	0	
## 9	taub3	B.TauB	u3	u3	9.166823345	0.41213644		0	

```
##
```

```
## Model Statistics:
```

```

##           | Parameters | Degrees of Freedom | Fit (-2lnL units)
## Model:           9           299991           1350558
## Saturated:       NA           NA           NA
## Independence:    NA           NA           NA
## Number of observations/statistics: 101000/300000
##
## Information Criteria:
##           | df Penalty | Parameters Penalty | Sample-Size Adjusted
## AIC:       750575.9           1350576           1350576
## BIC:       -2106201.1           1350662           1350633
## CFI: NA
## TLI: 1 (also known as NNFI)
## RMSEA: 0 [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2022-04-02 18:57:02
## Wall clock time: 2.34537 secs
## optimizer: SLSQP
## OpenMx version number: 2.20.6
## Need help? See help(mxSummary)

```

```
sessionInfo()
```

```

## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
## [1] LC_CTYPE=C.UTF-8 LC_NUMERIC=C LC_TIME=C.UTF-8
## [4] LC_COLLATE=C.UTF-8 LC_MONETARY=C.UTF-8 LC_MESSAGES=C.UTF-8
## [7] LC_PAPER=C.UTF-8 LC_NAME=C LC_ADDRESS=C
## [10] LC_TELEPHONE=C LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## other attached packages:
## [1] MASS_7.3-56 OpenMx_2.20.6
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.8.3 lattice_0.20-45 digest_0.6.29 grid_4.1.3
## [5] lifecycle_1.0.1 magrittr_2.0.2 evaluate_0.15 RcppParallel_5.1.5
## [9] rlang_1.0.2 stringi_1.7.6 cli_3.1.1 rstudioapi_0.13
## [13] Matrix_1.4-0 rmarkdown_2.13 tools_4.1.3 stringr_1.4.0
## [17] xfun_0.30 yaml_2.3.5 parallel_4.1.3 fastmap_1.1.0
## [21] compiler_4.1.3 htmltools_0.5.2 knitr_1.37

```