

Synthesizing Indirect Effects in Mediation Models with Meta-Analytic Methods: Supplementary Materials 1

Mike Cheung

July 24, 2020

Contents

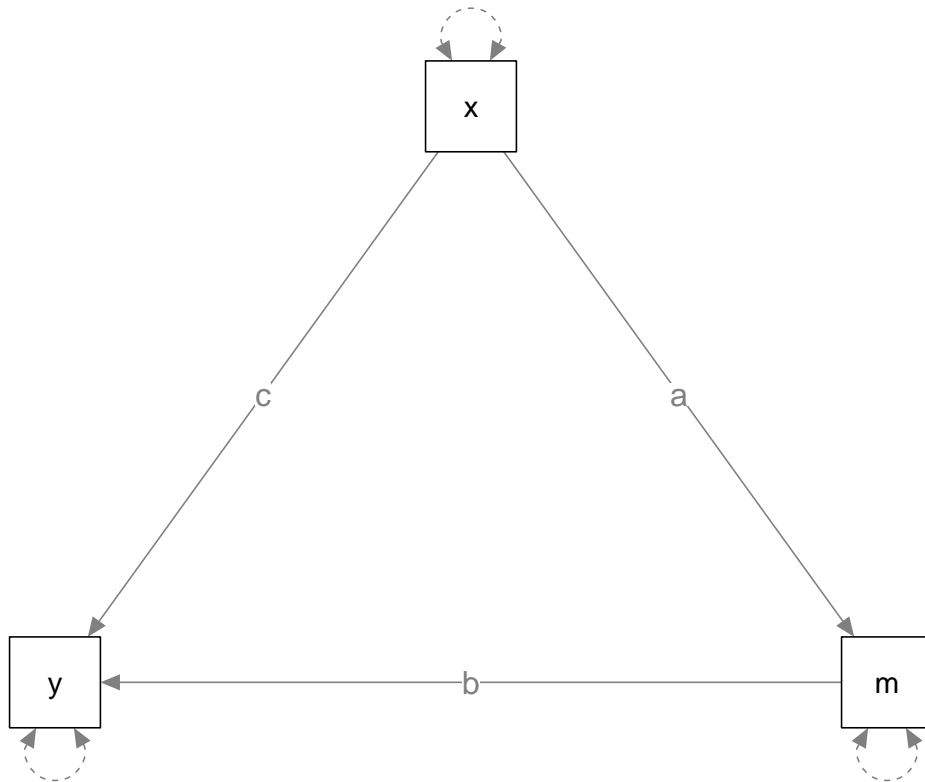
Numeric calculations with the SEM approach	1
One mediator	1
Two parallel mediators	3
Two serial mediators	4
Symbolic calculations	5
One mediator	5
Two parallel mediators	5
Two serial mediators	6
• This file demonstrates how to compute effect sizes and their sampling covariance matrix with two approaches using the delta method. The first approach uses a numeric approach with the structural equation modeling (SEM) framework. The second approach computes the sampling covariance matrix with the symbolic calculations.	

Numeric calculations with the SEM approach

One mediator

```
library(metaSEM)

## Model with one mediator: x -> m -> y
model1 <- "y ~ c*x + b*m
          m ~ a*x
          # Define indirect and direct effects
          Indirect := a*b
          Direct := c"
plot(model1)
```



```

## Sample correlation matrix
my.cor <- matrix(c(1, .5, .3,
                  .5, 1, .4,
                  .3, .4, 1),
                nrow = 3,
                ncol = 3,
                dimnames = list(c("y", "m", "x"),
                              c("y", "m", "x"))))

```

```
my.cor
```

```

##      y    m    x
## y 1.0 0.5 0.3
## m 0.5 1.0 0.4
## x 0.3 0.4 1.0

```

```

## Calculate the indirect and direct effects and their sampling covariance matrix
calEffSizes(model=model11, n=300, Cov=my.cor)

```

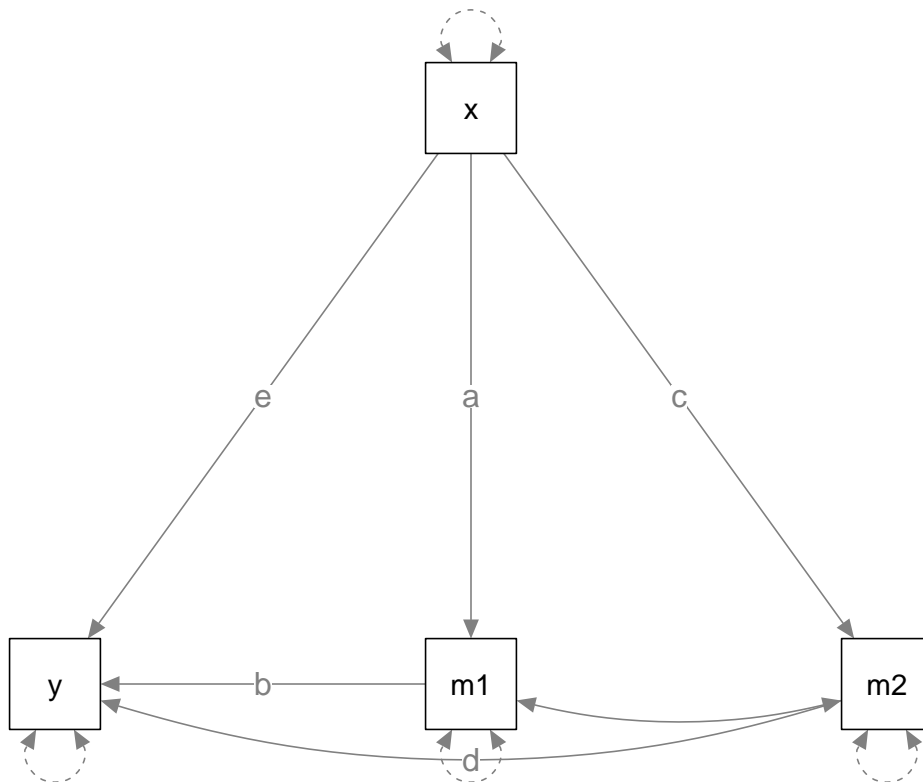
```

## $ES
## Indirect    Direct
## 0.1809524 0.1190476
##
## $VCOV
##           Indirect      Direct
## Indirect 0.0010416478 -0.0004686319
## Direct   -0.0004686319 0.0029289494

```

Two parallel mediators

```
## Model with two specific mediators: x -> m1 -> y and x -> m2 -> y
model2 <- "y ~ e*x + b*m1 + d*m2
          m1 ~ a*x
          m2 ~ c*x
          # m1 and m2 are correlated
          m1 ~~ m2
          # Define indirect and direct effects
          Ind_m1 := a*b
          Ind_m2 := c*d
          Direct := e"
plot(model2)
```



```
## Sample correlation matrix
my.cor <- matrix(c(1, .5, .6, .3,
                  .5, 1, .4, .2,
                  .6, .4, 1, .3,
                  .3, .2, .3, 1),
                nrow = 4,
                ncol = 4,
                dimnames = list(c("y", "m1", "m2", "x"),
                              c("y", "m1", "m2", "x"))))
my.cor
```

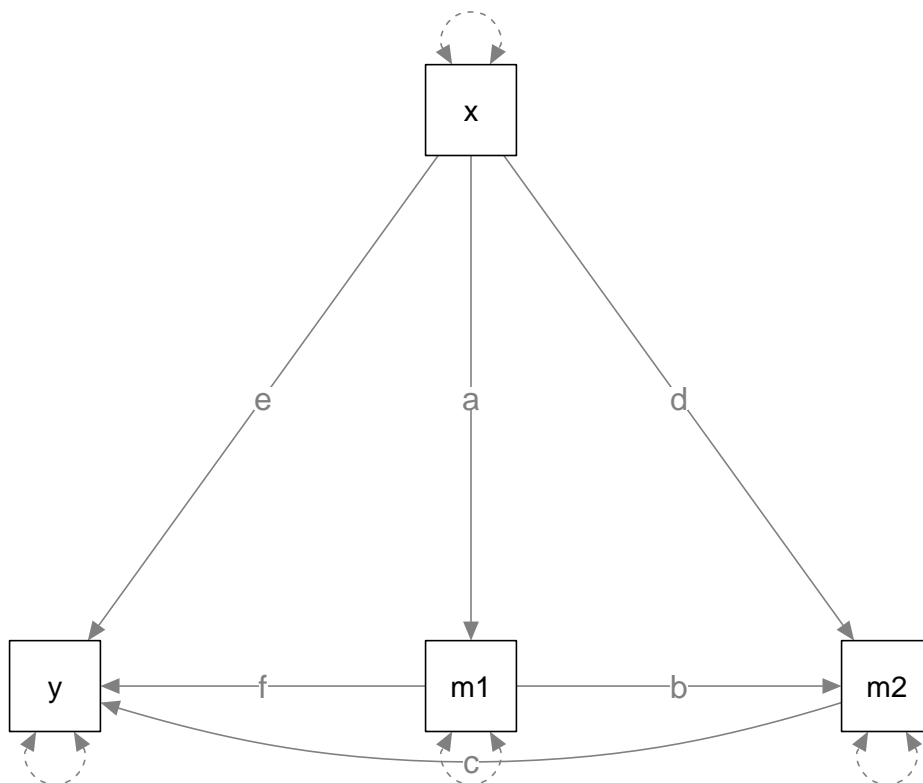
```
##      y  m1  m2  x
## y   1.0 0.5 0.6 0.3
## m1 0.5 1.0 0.4 0.2
## m2 0.6 0.4 1.0 0.3
```

```
## x  0.3 0.2 0.3 1.0
## Calculate the indirect and direct effects and their sampling covariance matrix
calEffSizes(model=model2, n=300, Cov=my.cor)

## $ES
##      Ind_m1      Ind_m2      Direct
## 0.05989446 0.13456464 0.10554090
##
## $VCOV
##           Ind_m1      Ind_m2      Direct
## Ind_m1  0.0003749419  0.0001029453 -0.0000386612
## Ind_m2  0.0001029453  0.0008190651 -0.0001594774
## Direct -0.0000386612 -0.0001594774  0.0020297130
```

Two serial mediators

```
## Model with two intermediate mediators: x -> m1 -> m2 -> y
model3 <- "y ~ e*x + f*m1 + c*m2
m1 ~ a*x
m2 ~ b*m1
m2 ~ d*x
# Define indirect and direct effects
Ind_m1m2 := a*b*c
Direct := e"
plot(model3)
```



```
## Calculate the indirect and direct effects and their sampling covariance matrix
calEffSizes(model=model3, n=300, Cov=my.cor)
```

```
## $ES
##   Ind_m1m2      Direct
## 0.03177221 0.10554090
##
## $VCOV
##           Ind_m1m2      Direct
## Ind_m1m2 0.0001144605 -0.0000376544
## Direct   -0.0000376544 0.0020297130
```

Symbolic calculations

One mediator

```
library(symSEM)

## fn: The effect sizes
## Covfn: Sampling covariance matrix of fn
## Va: Sampling variance of a
## Vb: Sampling variance of b
## Cba: Sampling covariance of a and b
deltamethod(fn="a*b")
```

```
## $fn
##      [,1]
## fn1 "a*b"
##
## $Covfn
##      fn1
## fn1 "b^2*Va+2*b*a*Cba+a^2*Vb"
##
## $vars
## [1] "a" "b"
##
## $Covvars
##      a      b
## a "Va"  "Cba"
## b "Cba" "Vb"
##
## $Jmatrix
##      a      b
## fn1 "b" "a"
```

Two parallel mediators

```
deltamethod(fn=c("a*b", "c*d"))

## $fn
##      [,1]
## fn1 "a*b"
## fn2 "c*d"
##
## $Covfn
##      fn1      fn2
## fn1 "b^2*Va+2*b*a*Cba+a^2*Vb" "b*Cca*d+b*Cda*c+a*Ccb*d+a*Cdb*c"
```

```
## fn2 "d*Cca*b+d*Ccb*a+c*Cda*b+c*Cdb*a" "d^2*Vc+2*d*c*Cdc+c^2*Vd"
##
## $vars
## [1] "a" "b" "c" "d"
##
## $Covvars
##   a      b      c      d
## a "Va"   "Cba" "Cca" "Cda"
## b "Cba"  "Vb"  "Ccb" "Cdb"
## c "Cca"  "Ccb" "Vc"  "Cdc"
## d "Cda"  "Cdb" "Cdc" "Vd"
##
## $Jmatrix
##   a      b      c      d
## fn1 "b" "a" "0" "0"
## fn2 "0" "0" "d" "c"
```

Two serial mediators

```
deltamethod(fn="a*b*c")
```

```
## $fn
##   [,1]
## fn1 "a*b*c"
##
## $Covfn
##   fn1
## fn1 "b^2*c^2*Va+2*b^2*c*a*Cca+b^2*a^2*Vc+2*b*c^2*a*Cba+2*b*c*a^2*Ccb+c^2*a^2*Vb"
##
## $vars
## [1] "a" "b" "c"
##
## $Covvars
##   a      b      c
## a "Va"   "Cba" "Cca"
## b "Cba"  "Vb"  "Ccb"
## c "Cca"  "Ccb" "Vc"
##
## $Jmatrix
##   a      b      c
## fn1 "b*c" "a*c" "a*b"
```

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04 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=en_SG.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_SG.UTF-8      LC_COLLATE=en_SG.UTF-8
```

```

## [5] LC_MONETARY=en_SG.UTF-8    LC_MESSAGES=en_SG.UTF-8
## [7] LC_PAPER=en_SG.UTF-8       LC_NAME=C
## [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_SG.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] symSEM_0.1      metaSEM_1.2.4.1 OpenMx_2.17.3
##
## loaded via a namespace (and not attached):
## [1] minqa_1.2.4      colorspace_1.4-1  rjson_0.2.20
## [4] ellipsis_0.3.0   rprojroot_1.3-2   htmlTable_1.13.3
## [7] corpcor_1.6.9     base64enc_0.1-3    rstudioapi_0.11
## [10] lavaan_0.6-6      mvtnorm_1.1-0      splines_4.0.2
## [13] mnormt_1.5-7      knitr_1.28         glasso_1.11
## [16] pkgload_1.0.2     Formula_1.2-3      nloptr_1.2.2.1
## [19] cluster_2.1.0     png_0.1-7          regsem_1.5.2
## [22] compiler_4.0.2    backports_1.1.6    assertthat_0.2.1
## [25] Matrix_1.2-18     acepack_1.4.1      htmltools_0.4.0
## [28] tools_4.0.2       igraph_1.2.5       coda_0.19-3
## [31] gtable_0.3.0      glue_1.4.0         reshape2_1.4.4
## [34] dplyr_0.8.5       Rcpp_1.0.4.6       carData_3.0-3
## [37] vctrs_0.2.4       nlme_3.1-147       lisrelToR_0.1.4
## [40] psych_1.9.12.31   xfun_0.13          stringr_1.4.0
## [43] testthat_2.3.2    openxlsx_4.1.5     lme4_1.1-23
## [46] lifecycle_0.2.0   gtools_3.8.2       statmod_1.4.34
## [49] XML_3.99-0.3      MASS_7.3-51.6      scales_1.1.0
## [52] BDgraph_2.62      Ryacas_1.1.3       kutils_1.70
## [55] parallel_4.0.2    huge_1.3.4.1       RColorBrewer_1.1-2
## [58] yaml_2.2.1        pbapply_1.4-2      gridExtra_2.3
## [61] ggplot2_3.3.0     rpart_4.1-15       latticeExtra_0.6-29
## [64] stringi_1.4.6     desc_1.2.0         sem_3.1-9
## [67] checkmate_2.0.0   boot_1.3-25        zip_2.0.4
## [70] truncnorm_1.0-8   rlang_0.4.6        pkgconfig_2.0.3
## [73] d3Network_0.5.2.1 Rsolnp_1.16        arm_1.11-1
## [76] evaluate_0.14     lattice_0.20-41     purrr_0.3.4
## [79] htmlwidgets_1.5.1 tidyselect_1.0.0   plyr_1.8.6
## [82] magrittr_1.5      R6_2.4.1           Hmisc_4.4-0
## [85] pillar_1.4.4      whisker_0.4         foreign_0.8-79
## [88] withr_2.2.0       rockchalk_1.8.144  survival_3.1-12
## [91] semPlot_1.1.2     abind_1.4-5        nnet_7.3-14
## [94] tibble_3.0.1      crayon_1.3.4       fdrtool_1.2.15
## [97] ellipse_0.4.1     rmarkdown_2.1      jpeg_0.1-8.1
## [100] grid_4.0.2        qgraph_1.6.5       data.table_1.12.8
## [103] pbivnorm_0.6.0    matrixcalc_1.0-3   digest_0.6.25
## [106] xtable_1.8-4      mi_1.0             stats4_4.0.2
## [109] munsell_0.5.0

```