Lecture 5: Latent Variable Models, Interpretations, & Effect Size

Kyle M. Lang

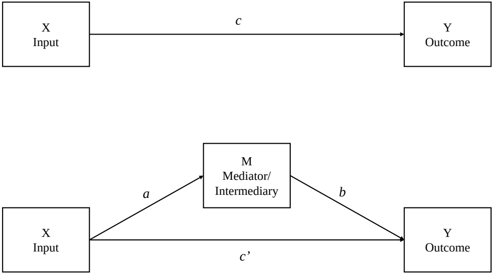
Institute for Measurement, Methodology, Analysis & Policy
Texas Tech University
Lubbock, TX

February 29, 2016

Outline

- Show how to test for indirect effects in latent variable models
- Discuss the interpretation of indirect effects
- Discuss effect size measures for indirect effects
- Talk about term-project ideas

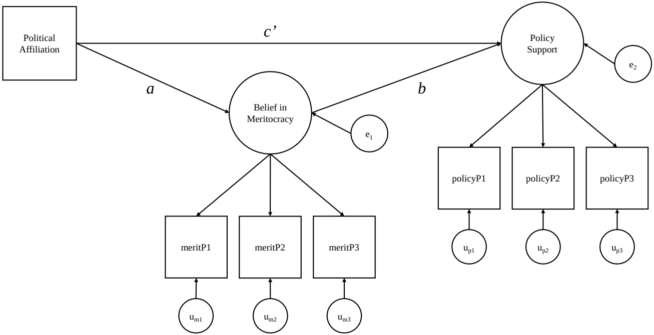
So far, all of our models have been similar to:



But there is no reason that we need to restrict ourselves to mucking about with observed variables.

Better Model

We can (and should) test for indirect effects using *latent variable models* such as:



Measurement error can be a big problem for mediation analysis, so latent variable modeling is highly recommended.

Example

```r
library(lavaan)
dataDir <- "../data/"
dat1 <- readRDS(paste0(dataDir, "adamsKlpsData.rds"))
## Specify the CFA model:
mod1.1 <- "
merit =~ meritP1 + meritP2 + meritP3
policy =~ policyP1 + policyP2 + policyP3
"
## Fit the CFA and check model:
out1.1 <- cfa(mod1.1, data = dat1, std.lv = TRUE)
## Check model fit:
round(fitMeasures(out1.1)[c("chisq", "df", "pvalue", "cfi",
                            "tli", "rmsea", "srmr")], 4)
```

```
  chisq      df  pvalue     cfi     tli   rmsea    srmr
16.8695  8.0000  0.0315  0.9215  0.8529  0.1129  0.0653
```

```r
summary(out1.1)
```

Example

```
lavaan (0.5-19) converged normally after  22 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic               16.869
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.031

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  merit =~
    meritP1           0.690    0.134    5.155    0.000
    meritP2           0.968    0.142    6.830    0.000
    meritP3           0.748    0.137    5.458    0.000
  policy =~
    policyP1          0.851    0.186    4.570    0.000
```

Example

```
    policyP2          0.996    0.167    5.967    0.000
    policyP3          1.121    0.177    6.339    0.000

Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)
  merit ~~
    policy           -0.336    0.131   -2.563    0.010

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    meritP1           0.865    0.165    5.248    0.000
    meritP2           0.445    0.201    2.211    0.027
    meritP3           0.833    0.172    4.857    0.000
    policyP1          1.836    0.324    5.671    0.000
    policyP2          0.942    0.256    3.683    0.000
    policyP3          0.857    0.297    2.882    0.004
    merit             1.000
    policy            1.000
```

Example

```
## Specify the structural model:
mod1.2 <- "
merit =~ meritP1 + meritP2 + meritP3
policy =~ policyP1 + policyP2 + policyP3

policy ~ b*merit + polAffil
merit ~ a*polAffil

ab := a*b
"
## Fit the structural model and test the indirect effect:
out1.2 <- sem(mod1.2, data = dat1, std.lv = TRUE,
              se = "boot", boot = 2500)
summary(out1.2)
```

Example

```
lavaan (0.5-19) converged normally after  24 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic               20.665
  Degrees of freedom                                12
  P-value (Chi-square)                           0.056

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2477

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  merit =~
    meritP1           0.545    0.127    4.301    0.000
    meritP2           0.858    0.132    6.490    0.000
    meritP3           0.609    0.117    5.204    0.000
```

Example

```
  policy =~
    policyP1          0.799    0.191    4.185    0.000
    policyP2          0.924    2.404    0.384    0.701
    policyP3          1.001    2.134    0.469    0.639

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    merit      (b)   -0.195    0.205   -0.950    0.342
    polAffil          0.169    0.134    1.267    0.205
  merit ~
    polAffil   (a)   -0.411    0.100   -4.093    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    meritP1           0.922    0.186    4.947    0.000
    meritP2           0.341    0.224    1.522    0.128
    meritP3           0.869    0.185    4.696    0.000
    policyP1          1.801    0.338    5.326    0.000
    policyP2          0.918  136.895    0.007    0.995
    policyP3          0.922  126.529    0.007    0.994
    merit             1.000
    policy            1.000
```

Example

```
Defined Parameters:
                   Estimate  Std.Err  Z-value  P(>|z|)
    ab                0.080    0.093    0.859    0.390
```

```
parameterEstimates(out1.2, boot = "bca.simple")[-c(1 : 3)]
```

```
   label    est      se      z pvalue ci.lower ci.upper
1          0.545   0.127  4.301  0.000    0.325    0.824
2          0.858   0.132  6.490  0.000    0.620    1.150
3          0.609   0.117  5.204  0.000    0.375    0.827
4          0.799   0.191  4.185  0.000    0.446    1.144
5          0.924   2.404  0.384  0.701    0.658    1.479
6          1.001   2.134  0.469  0.639    0.574    1.666
7      b -0.195   0.205 -0.950  0.342   -0.622    0.193
8          0.169   0.134  1.267  0.205   -0.087    0.443
9      a -0.411   0.100 -4.093  0.000   -0.625   -0.235
10         0.922   0.186  4.947  0.000    0.615    1.344
11         0.341   0.224  1.522  0.128   -0.127    0.691
12         0.869   0.185  4.696  0.000    0.563    1.284
13         1.801   0.338  5.326  0.000    1.219    2.570
14         0.918 136.895  0.007  0.995    0.341    1.567
```

Example

```
15           0.922  126.529  0.007  0.994     0.019     1.581
16           1.000    0.000     NA     NA     1.000     1.000
17           1.000    0.000     NA     NA     1.000     1.000
18           2.444    0.000     NA     NA     2.444     2.444
19      ab   0.080    0.093  0.859  0.390    -0.081     0.286
```

Interpretation of Indirect Effects

Although indirect effects are composed parameters, they have direct interpretations, independent of the interpretations of their constituent paths:

- The $X \rightarrow M \rightarrow Y$ indirect effect ab is interpreted as:
  - The expected change in Y for a unit change in X that is transmitted indirectly through M, or...
  - For a unit change in X, Y is expected to change by ab units, indirectly through M, or...
  - Participants who differ by one unit on X are expect to differ by ab units on Y as a results of the effect of X on M which, in turn, affects Y.
- The interpretation/scaling of the indirect effect is entirely defined by the input X and outcome Y
  - The scaling of the intermediary variable M does not affect the interpretation of the indirect effect.

Partially Standardized Indirect Effect

$$
\begin{aligned}
ab_{ps} &= \frac{ab}{SD_Y} \\
c'_{ps} &= \frac{c'}{SD_Y} \\
c_{ps} &= \frac{c}{SD_Y} = ab_{ps} + c'_{ps}
\end{aligned}
$$

- Simple
- Removes binding to the scale of Y
- Still scale-bound by X
- Not clear what constitutes a "large" effect

Completely Standardized Indirect Effect

$$
\begin{aligned}
ab_{cs} &= \frac{SD_X\, ab}{SD_Y} \\
c'_{cs} &= \frac{SD_X\, c'}{SD_Y} \\
c_{cs} &= \frac{SD_X\, c}{SD_Y} = ab_{cs} + c'_{cs}
\end{aligned}
$$

- Simple
- Removes all scale binding
- Not clear what constitutes a “large” effect

Ratio of the Indirect Effect to the Total Effect

$$P_M = \frac{ab}{c} = \frac{ab}{c' + ab}$$

- Very simple
- Not bounded by 0 and 1
- Explodes toward $\pm\infty$ as $c \rightarrow 0$
- Very unstable
  - High between-sample variability
  - Requires $N \geq 500$

Ratio of the Indirect Effect to the Direct Effect

$$R_M = \frac{ab}{c'} = \frac{P_M}{1 - P_M}$$

- Very simple
- Not bounded by 0 and 1
- Explodes toward $\pm\infty$ as $c' \rightarrow 0$
- Very unstable
  - High between-sample variability
  - Requires $N \geq 2000$

Proportion of Variance in Y Explained by the Indirect Effect

Developed by Fairchild, MacKinnon, Taborga, and Taylor (2009).

- Given a non-zero total effect, represents the proportion of variance in Y accounted for by the indirect effect.

$$R^2_{med} = r^2_{MY} - \left(R^2_{Y.MX} - r^2_{XY}\right)$$

- Mostly sensible interpretation
- Predicated on the assumption that $\beta_{YX} \neq 0$
- $|ab| > |c| \Rightarrow R^2_{med} < 0$
    - Not a strict proportion

Kappa Squared

Developed by Preacher and Kelley (2011).

- Gives the proportion of the *maximum possible* indirect effect represented by ab.

$$\kappa^2 = \frac{ab}{\max(ab)}$$

- Bounded by 0 and 1
- Values closer to 1.0 indicate a bigger effect
- A bit of a pain to calculate.

Computing $\max(ab)$

$$a \in \left\{ \frac{\sigma_{YM}\sigma_{YX} \pm \sqrt{\sigma_M^2\sigma_Y^2 - \sigma_{YM}^2}\sqrt{\sigma_X^2\sigma_Y^2 - \sigma_{YX}^2}}{\sigma_X^2\sigma_Y^2} \right\} = [a_{low}, a_{high}],$$

$$b \in \left\{ \pm \frac{\sqrt{\sigma_X^2\sigma_Y^2 - \sigma_{YX}^2}}{\sqrt{\sigma_X^2\sigma_M^2 - \sigma_{MX}^2}} \right\} = [b_{low}, b_{high}],$$

$$\max(a) = \begin{cases} a_{high}, & \text{if} \quad \hat{a} > 0 \\ a_{low}, & \text{if} \quad \hat{a} < 0 \end{cases}, \quad \max(b) = \begin{cases} b_{high}, & \text{if} \quad \hat{b} > 0 \\ b_{low}, & \text{if} \quad \hat{b} < 0 \end{cases},$$

$$\max(ab) = \max(a)\max(b)$$

Example

```r
## Specify the model:
mod2 <- "
policy ~ b*sysRac + cp*polAffil
sysRac ~ a*polAffil

ab := a*b
"
## Estimate the model:
out2 <- sem(mod2, data = dat1)
##
## Extract/compute the necessary quantities:
ab <- prod(coef(out2)[c("a", "b")])
ab
```

```
[1] 0.1015958
```

Example

```
cPrime <- coef(out2)["cp"]
##
sdY <- sd(dat1$policy)
sdX <- sd(dat1$polAffil)
##
r2MY <- with(dat1, cor(policy, sysRac))^2
r2XY <- with(dat1, cor(policy, polAffil))^2
R2Y.MX <- inspect(out2, "r2")["policy"]
```

Example

```
## Partially Standardized:
abPS <- ab / sdY
abPS
```

```
[1] 0.08559454
```

```
cPrimePS <- cPrime / sdY
cPrimePS
```

```
       cp
0.1138675
```

```
cPS <- abPS + cPrimePS
cPS
```

```
      cp
0.199462
```

Example

```r
## Completely Standardized:
abCS <- (sdX * ab) / sdY
abCS
```

```
[1] 0.1345859
```

```r
cPrimeCS <- (sdX * cPrime) / sdY
cPrimeCS
```

```
       cp
0.1790413
```

```r
cCS <- abCS + cPrimeCS
cCS
```

```
       cp
0.3136272
```

Example

```r
## Proportions:
pm <- ab / (cPrime + ab)
pm
```

```
      cp
0.429127
```

```r
rm <- ab / cPrime
rm
```

```
       cp
0.7517031
```

```r
## R2:
R2med <- r2MY - (R2Y.MX - r2XY)
R2med
```

```
    policy
0.06905689
```

Compute κ^2

```
## Subset the data:
tmpData <- dat1[ , c("polAffil", "sysRac", "policy")]
colnames(tmpData) <- c("x", "m", "y")
##
## Extract pertinent variance/covariance elements:
cov1 <- cov(tmpData)
sYM <- cov1["x", "m"]
sYX <- cov1["y", "x"]
sMX <- cov1["m", "x"]
s2X <- cov1["x", "x"]
s2M <- cov1["m", "m"]
s2Y <- cov1["y", "y"]
```

Compute κ^2

```
## Possible range of a:
aMarg <- sqrt(s2M * s2Y - sYM^2) * sqrt(s2X * s2Y - sYX^2)
aInt <- c(
    (sYM * sYX - aMarg) / (s2X * s2Y),
    (sYM * sYX + aMarg) / (s2X * s2Y)
)
aInt
```

```
[1] -0.4378558  0.5793099
```

```
##
## Possible range of b:
bMarg <- sqrt(s2X * s2Y - sYX^2) / sqrt(s2X * s2M - sMX^2)
bInt <- c(-1 * bMarg, bMarg)
bInt
```

```
[1] -1.289996  1.289996
```

Compute κ^2

```
##
## max(a):
aMax <- ifelse(coef(out2)["a"] < 0,
               aInt[1],
               aInt[2])
aMax
```

```
        a
0.5793099
```

```
##
## max(b)
bMax <- ifelse(coef(out2)["b"] < 0,
               bInt[1],
               bInt[2])
bMax
```

```
       b
1.289996
```

Compute κ^2

```
##
## max(ab)
abMax <- aMax * bMax
abMax
```

```
        a
0.7473075
```

```
##
## Kappa Squared:
k2 <- ab / abMax
k2
```

```
        a
0.1359491
```

Practice

Suppose:

1. Σ is given by:

	x	m	y
x	1.5		
m	0.3	1.4	
y	0.6	0.45	1.55

2. The estimated paths are:
   - $a = 0.2$
   - $b = 0.246$
   - $ab = 0.049$

Compute κ^2 for the estimated ab.

Fairchild, A. J., MacKinnon, D. P., Taborga, M. P., & Taylor, A. B. (2009). R-squared effect-size measures for mediation analysis. *Behavior Research Methods*, *41*(2), 486–498.

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods*, *16*(2), 93.