Why we may not need SEM after all

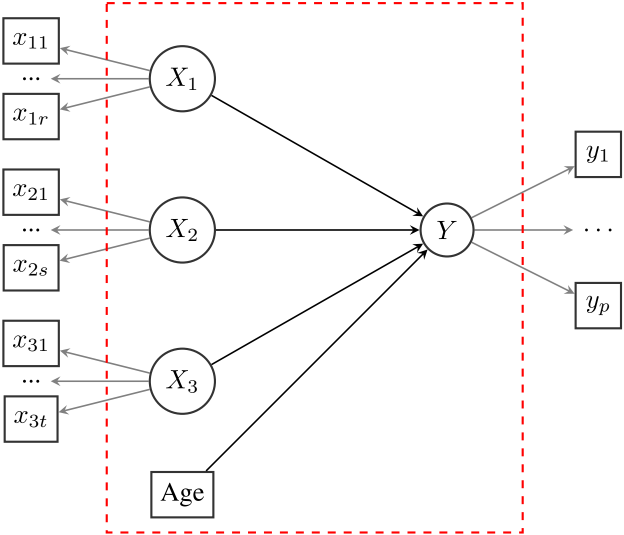
Yves Rosseel & Ines Devlieger
Department of Data Analysis
Ghent University – Belgium

February 15, 2018
Psychoco 2018 – Tübingen

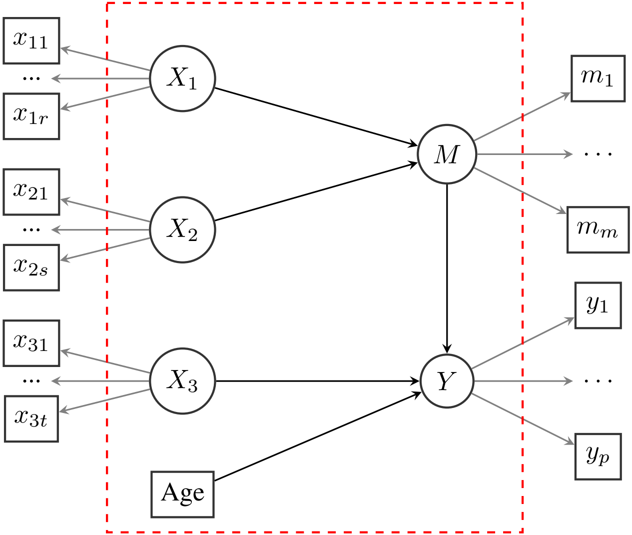
background

- a typical dataset in the social sciences:
  - many constructs (motivation, ability, personality traits, . . .)
  - each construct is measured by a set of (observed) indicators
  - many 'background' variables (age, gender, . . .)
  - (multilevel data, missing data, categorical data)
- the measurement instruments for the latent variables are well established, and usually fit (reasonably) well
- the main focus of the study is the structural part of the model:
  - regression model: variables are either dependent or independent
  - path analysis model: includes mediating effects, perhaps non-recursive
- the sample size is not always very large

structural model: regression model



structural model: path analysis model



the dilemma of the applied researcher

- how to analyze a model with many latent variables, some of them measured by a large number of indicators?
- he/she knows that SEM is the 'golden standard', but ...
  - the full model contains a huge number of free model parameters
  - the sample size is only medium
  - the focus is on the structural part of the model only
  - ~~he/she hesitates to buy a dedicated commercial SEM package~~
  - he/she would very much like to use SPSS to fit the regression part of the model
- a colleague/roommate/supervisor/... suggests to compute sum scores (or factor scores) for each latent variable, to simplify the model
- in the end, he/she decides to talk to the local statistical consultant

the conversation

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

the conversation

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the golden standard.

the conversation

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the golden standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

the conversation

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the golden standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

(consultant:) If you ignore the measurement error, your results will be biased. This is really bad. You should use SEM.

the conversation

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the golden standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

(consultant:) If you ignore the measurement error, your results will be biased. This is really bad. You should use SEM.

(researcher:) So using factor scores followed by regression is really a silly idea? I have seen it in many journals.

the conversation

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the golden standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

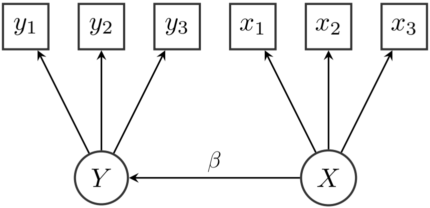
(consultant:) If you ignore the measurement error, your results will be biased. This is really bad. You should use SEM.

(researcher:) So using factor scores followed by regression is really a silly idea? I have seen it in many journals.

(consultant:) Yeah, idiots are everywhere. You may get away with it in some journals, but not in a good journal. You must use SEM.

a simple example

- consider the regression of a measured latent variable Y on another measured latent variable X:



- we are mainly interested in the question: is there a significant effect from X on Y? We want to test the hypothesis:

$$H_0 : \beta = 0$$

data generation

```
> library(lavaan)
> pop.model <- '
+     # factor loadings
+     Y =~ 1*y1 + 0.8*y2 + 0.6*y3
+     X =~ 1*x1 + 0.8*x2 + 0.6*x3
+
+     # regression part
+     Y ~ 0.25*X
+ '
> set.seed(1234)
> Data <- simulateData(pop.model, sample.nobs = 200L, empirical = TRUE)
```

the golden standard: SEM

```
> model <- '
+     # factor loadings
+     Y =~ y1 + y2 + y3
+     X =~ x1 + x2 + x3
+
+     # regression part
+     Y ~ X
+ '
> fit.sem <- sem(model, data = Data, estimator = "ML")
```

output SEM

```
> parameterEstimates(fit.sem, add.attributes = TRUE, ci = FALSE)[1:7,]

Parameter Estimates:

  Information                                 Expected
  Information saturated (h1) model          Structured
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y =~
    y1                1.000
    y2                0.800    0.161    4.972    0.000
    y3                0.600    0.123    4.881    0.000
  X =~
    x1                1.000
    x2                0.800    0.169    4.735    0.000
    x3                0.600    0.129    4.661    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X                 0.250    0.114    2.189    0.029
```

naive method 1: sum scores

- we replace the latent variables by sum scores:

```
> sumY <- Data$y1 + Data$y2 + Data$y3
> sumX <- Data$x1 + Data$x2 + Data$x3
```

- we fit a simple regression model using these sum scores:

```
> fit.sum <- lm(sumY ~ sumX, data = Data)
> round(summary(fit.sum)$coefficients[2,], 3)

 Estimate Std. Error    t value   Pr(>|t|)
    0.164      0.072      2.297      0.023
```

- bias:
  - downward bias for the point estimate (about 34%)
  - downward bias for the standard error (about 37%)
- the effect is still significant!

naive method 2: factor scores

- we replace the latent variables by factor scores:

```
> fsY <- lavPredict(sem('Y =~ y1 + y2 + y3', data = Data))
> fsX <- lavPredict(sem('X =~ x1 + x2 + x3', data = Data))
```

- we fit a simple regression model using these factor scores:

```
> fit.fs <- lm(fsY ~ fsX)
> round(summary(fit.fs)$coefficients[2,], 3)
```

```
 Estimate Std. Error    t value   Pr(>|t|)
    0.170      0.073      2.329      0.021
```

- bias:
  - downward bias for the point estimate (about 32%)
  - downward bias for the standard error (about 36%)
- the effect is still significant!

alternative method 1: Skrondal & Laake (2001)

- we replace the latent variables by factor scores, but use 'Bartlett' style factor scores for the dependent variable(s), and 'regression' style factor scores for the independent variable(s):

```
> fsYb <- lavPredict(sem('Y =~ y1 + y2 + y3', data = Data),
+                    method = "Bartlett")
> fsX  <- lavPredict(sem('X =~ x1 + x2 + x3', data = Data))
```

- we fit a simple regression model using these factor scores:

```
> fit.sl <- lm(fsYb ~ fsX)
> round(summary(fit.sl)$coefficients[2,], 3)
```

```
Estimate Std. Error    t value   Pr(>|t|)
   0.250      0.107      2.329      0.021
```

- no bias
- limitations:
  - regression setting only (no mediation)
  - does not work for standardized parameters

alternative method 2: Croon's correction

- we replace the latent variables by factor scores, but 'correct' the variance matrix of the factor scores, using a method proposed by Croon (2002)

```
> fit.Y <- sem('Y =~ y1 + y2 + y3', data = Data)
> fsY <- lavPredict(fit.Y, fsm = TRUE)
> fit.X <- sem('X =~ x1 + x2 + x3', data = Data)
> fsX <- lavPredict(fit.X, fsm = TRUE)
```

- from the uncorrected (naive) variance matrix of the factor scores, we can obtain the (biased) regression coefficient $\hat{\beta}$ as $\mathsf{Cov}(F_Y, F_x)/\mathsf{Var}(F_x)$:

```
> nobs <- nobs(fit.Y)
> S.naive <- cov(cbind(fsY, fsX))*(nobs-1)/nobs
> round(S.naive, 3)

      Y     X
Y 0.723 0.113
X 0.113 0.667

> beta.naive <- S.naive["Y","X"]/S.naive["X","X"]
> beta.naive

[1] 0.17
```

- extract some ingredients:

```
> A.y <- attr(fsY, "fsm")[[1]]
> A.x <- attr(fsX, "fsm")[[1]]
> Lambda.y <- lavInspect(fit.Y, "est")$lambda
> Lambda.x <- lavInspect(fit.X, "est")$lambda
> Theta.x <- lavInspect(fit.X, "est")$theta
```

- correction step 1: adjust the covariance

```
> cov.yx <- S.naive["Y","X"]
> scale.yx <- as.numeric(A.x %*% Lambda.x %*% t(Lambda.y) %*% t(A.y))
> cov.yx <- cov.yx / scale.yx
```

- correction step 2: adjust the variance of X:

```
> var.xx <- S.naive["X","X"]
> scale.xx <- as.numeric(crossprod(A.x %*% Lambda.x))
> offset.x <- as.numeric(A.x %*% Theta.x %*% t(A.x))
> var.xx <- (var.xx - offset.x)/scale.xx
```

- compute unbiased regression coefficient

```
> beta.croon <- cov.yx/var.xx
> beta.croon
```

```
[1] 0.25
```

Croon's correction in the lavaan package: function `fsr()`

- experimental version in 0.5-23; better version in 0.6-1
- automates the steps required to perform factor score regression (or path analysis) using Croon's correction:

```
> fit.fsr <- fsr(model, data = Data, se = "standard", output = "lavaan")

> parameterEstimates(fit.fsr, add.attributes = TRUE, ci = FALSE)[1,]
```

```
Parameter Estimates:

  Information                                      Observed
  Observed information based on                     Hessian
  Standard Errors                                  Standard

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X                 0.250    0.071    3.536    0.000
```

- no bias!
- but standard error is too small

getting the standard errors right

- an ad-hoc solution was proposed in Devlieger et. al. (2016), but we need a more general solution

  1. the bootstrap
     - works very good
     - intensive, takes time
  2. robust (sandwich type) standard errors
     - the standard approach needs a huge ACOV matrix
  3. correction for a two-step estimation procedure
     - based on the pseudo ML literature (Gong & Samaniego, 1981)
     - we have a multiple step, not a two-step
     - not trivial to implement in our framework

- work in progress

getting the standard errors right (2)

- the default (for now) is a robust 'sandwich-type' approach:

```
> fit.fsr <- fsr(model, data = Data, se = "robust.sem", output = "lavaan")

> parameterEstimates(fit.fsr, add.attributes = TRUE, ci = FALSE)[1,]

Parameter Estimates:

  Information                                       Observed
  Observed information based on                      Hessian
  Standard Errors                                 Robust.sem

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X                 0.250    0.108    2.325    0.020
```

- works well as long as the number of observed variables is not too large

advantages of the ‘fsr’ approach

- unbiased point estimates for the structural part of the model
- reduction in model complexity
- the ‘fsr’ approach can handle:
  - missing values for indicators (factor scores are always complete)
  - (in principle) categorical indicators (IRT)
- in contrast to ‘system-wide’ estimators (like maximum likelihood) the ‘fsr’ approach is robust against (local) model misspecifications
- conceptual: strict distinction between measurement model(s) and structural model
- for many models, the ‘fsr’ approach might replace SEM altogether

future plans and challenges

- challenge: (analytical) standard errors that perform well in the presence of missing indicators and/or non-normal (but continuous) indicators
- challenge: categorical indicators, nonlinear/interaction effects
- challenge: 'linked' measurement models
  - eg. longitudinal design with correlated residual errors over time
  - equality constraints over measurement models
- solved: extension to multilevel SEM (see talk by Ines on EAM in Jena)
- future plans: study the relationship with other related approaches:
  - consistent PLS (Dijkstra, T.K. 2010, 2014)
  - model-implied instrumental variables estimation (Bollen, 1996, 2001)
  - two-step approaches (eg. Bakk, Z. and Kuha, J. 2017, in LCA setting)
  - ...

Thank you!

some references

Devlieger, I., Mayer, A., & Rosseel, Y. (2016). Hypothesis testing using factor score regression: A comparison of four methods. *Educational and Psychological Measurement, 76*, 741–770.

Devlieger, I., & Rosseel, Y. (2017). Factor Score Path Analysis. *Methodology, 13*, 31–38.

Croon, M. (2002). *Using predicted latent scores in general latent structure models*. In Marcoulides, G., Moustaki, I. (Eds.), Latent variable and latent structure modeling (pp. 195–223). Mahwah, NJ: Lawrence Erlbaum.