Reliable estimation of Latent Variable Models

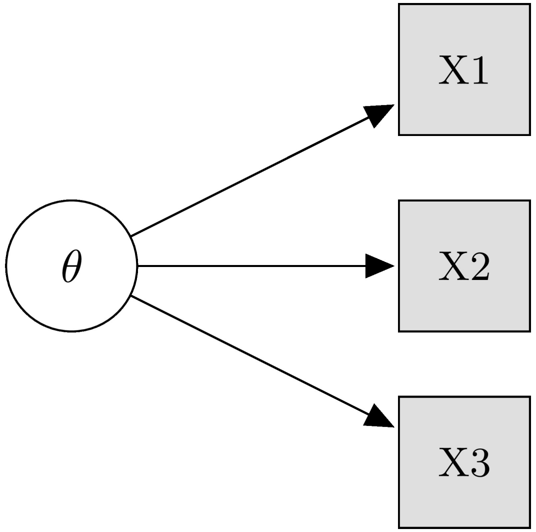
Marcos Jimenez, Mauricio Garnier-Villarreal, & Vithor Rosa Franco

2026-02-22

Latent Variable Modeling

A latent variable model is a way of connecting things we can measure directly (called observed or manifest variables) to hidden qualities we cannot measure directly (called latent variables). These models are used in many areas like biology, computer science, and social sciences. Latent variable models can involve either categorical or continuous observed and hidden variables, as below:

Latent variables	**Continuous (Manifest)**	**Categorical (Manifest)**
Continuous	Factor Analysis	Item Response Theory
Categorical	Latent Profile Analysis	Latent Class Analysis



The latent R package

- lavaan syntax for Structural Equation Models
- Core functions written in C++ with the armadillo library
- Parallelization of multiple random starts to address local maxima
- Customizable models

Console | Terminal | Background Jobs

R 4.5.1 · ~/latent-master/

```
> library(latent)
Welcome to latent!

    __      __             __
   / /___ _/ /____  ____  / /_
  / / __ `/ __/ _ \/ __ \/ __/
 / / /_/ / /_/  __/ / / / /_
/_/\__,_/\__/\___/_/ /_/\__/   version 0.1.0

Type 'citation("latent")' for citing this package in publications.

Report bugs at m.j.jimenezhenriquez@vu.nl
For tutorials, visit marcosjnez.github.io/latent/
```

Confirmatory Factor Analysis Example

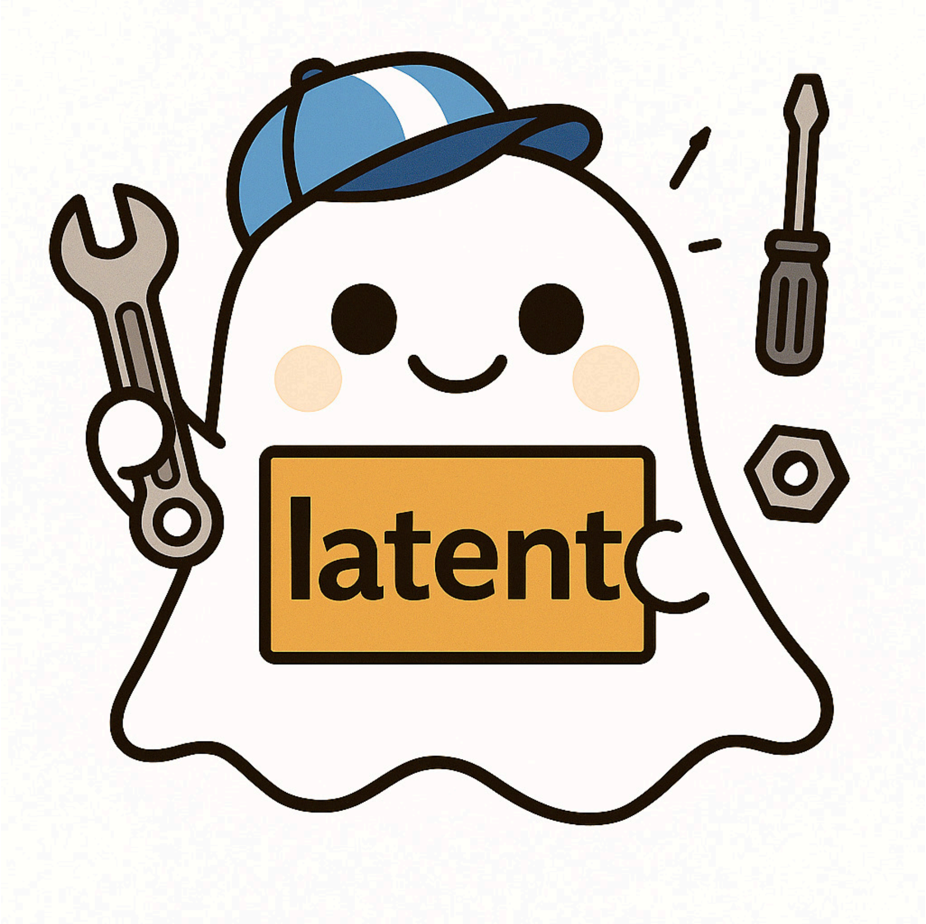
Model fitting:

```r
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '
fit <- lcfa(model = HS.model, data = HolzingerSwineford1939)
```

Extract model information:

```r
# Get fit indices:
lavaan::fitMeasures(fit)

# Inspect model objects:
lavaan::inspect(fit, what = "est", digits = 3) # Estimates
lavaan::inspect(fit, what = "se", digits = 3)  # Standard errors
```



Latent Class Analysis Example

Model fitting:

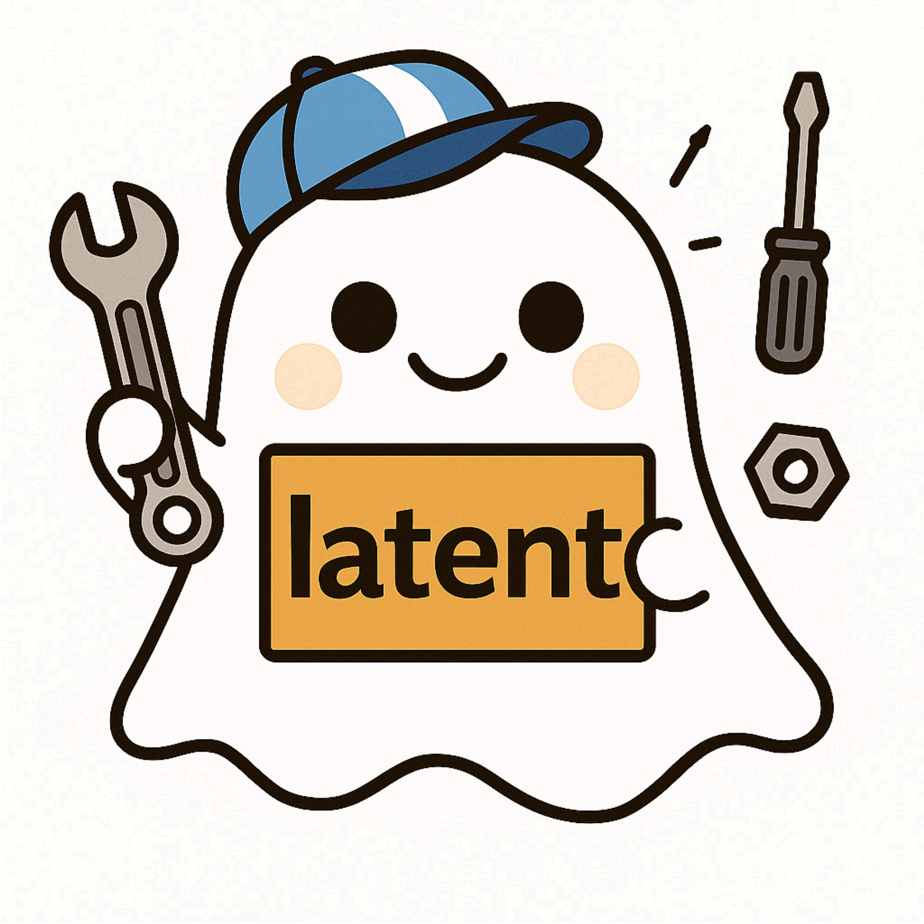
```r
fit <- lca(data = cancer[, 1:6], nclasses = 3L,
           item = c("gaussian", "gaussian",
                    "multinomial", "multinomial",
                    "gaussian", "gaussian"))
fit@loglik                # -5784.701
fit@penalized_loglik      # -5795.573
fit@timing                # 0.1927969
```

Extract model information:

```r
# Get fit indices:
getfit(fit)

# Inspect model objects:
latInspect(fit, what = "coefs", digits = 3)
latInspect(fit, what = "classes", digits = 3)
latInspect(fit, what = "profile", digits = 3)
latInspect(fit, what = "posterior", digits = 3)

# Get confidence intervals:
CI <- ci(fit, type = "standard", confidence = 0.95, digits = 2)
```



Factor Analysis

Factor Analysis (FA) is a method that estimates the influence of K continuous latent variables on a set of J items.

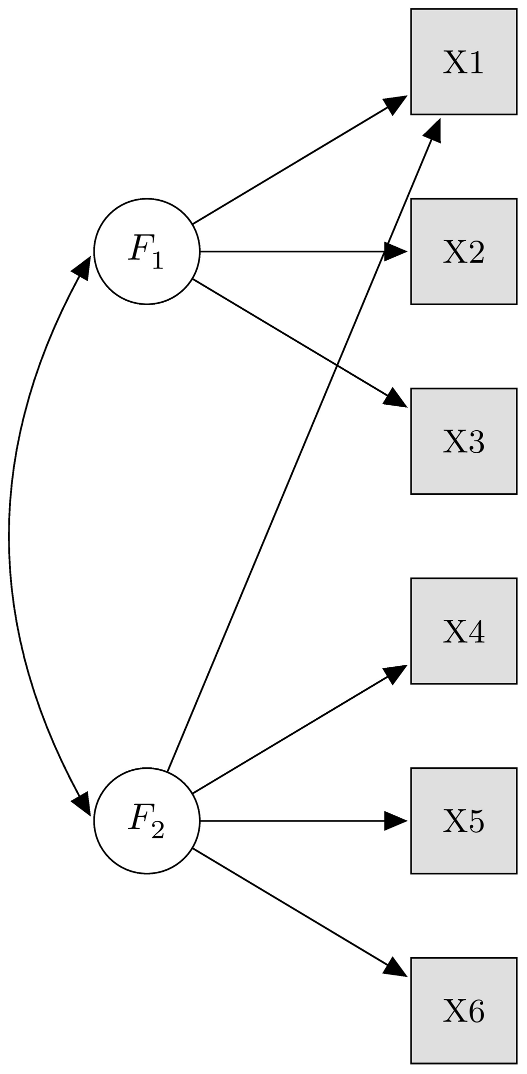
The score in item j is a weighted sum of the K latent factors:

$$X_j = \sum_{k=1}^{K} \lambda_{jk} F_k + \epsilon_j.$$

Under some assumptions, the J regressions can be encoded in a model for the covariance matrix of the items:

$$S = \Lambda \Psi \Lambda^\top + \Theta.$$

- Λ is a $J \times K$ matrix containing the regression coefficients.
- Ψ is the correlation matrix between the K latent factors.
- Θ is the error covariance matrix.



Positive-definite constraints

In the factor model equation,

$$S = \Lambda \Psi \Lambda^\top + \Theta,$$

Latent correlations Ψ and covariances Θ should be at least positive-semidefinite but…

```
Warning message:
lavaan->lav_object_post_check():
   covariance matrix of latent variables is not
   positive definite ; use lavInspect(fit, "cov.lv")
   to investigate.
```

Positive-definite constraints (`lavaan` fails)

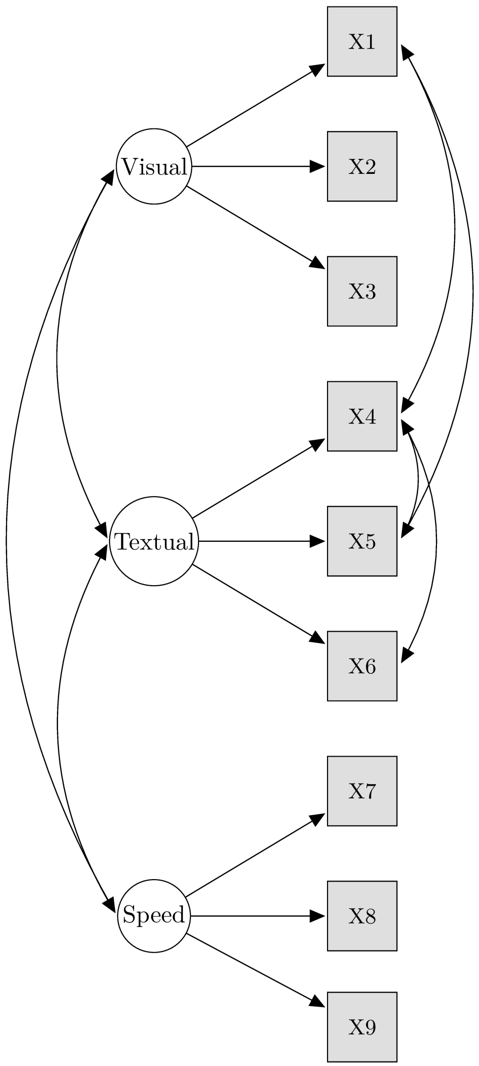
Let's force an instance where lavaan fails to converge to a proper solution.

```r
library(lavaan)
model <- 'visual  =~ x1 + x2 + x3
          textual =~ x4 + x5 + x6
          speed   =~ x7 + x8 + x9
          x1 ~~ x5
          x1 ~~ x4
          x4 ~~ x5
          x4 ~~ x6'
fit <- cfa(data = HolzingerSwineford1939,  model = model,
             estimator = "ml", std.lv = TRUE, std.ov = TRUE)
inspect(fit, what = "est")$theta      # Error covariances
```

```
       x1     x2     x3     x4     x5     x6     x7     x8     x9
x1  0.455
x2  0.000  0.805
x3  0.000  0.000  0.618
x4  0.084  0.000  0.000  0.244
x5  0.060  0.000  0.000  0.132  0.521
x6  0.000  0.000  0.000 -0.202  0.000 -0.087
x7  0.000  0.000  0.000  0.000  0.000  0.000  0.667
x8  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.455
x9  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.575
```

```r
det(inspect(fit, what = "est")$theta) # Check the determinant
```

```
[1] -0.00118551
```



Positive-definite constraints (`latent` converges)

```r
fit <- lcfa(data = HolzingerSwineford1939, model = model,
            estimator = "ml", std.lv = TRUE, positive = TRUE)
round(latInspect(fit, what = "est")[[1]]$theta, 3) # Error covariances
```

```
      x1    x2    x3     x4    x5     x6    x7    x8    x9
x1 0.453 0.000 0.000  0.082 0.041  0.000 0.000 0.000 0.000
x2 0.000 0.805 0.000  0.000 0.000  0.000 0.000 0.000 0.000
x3 0.000 0.000 0.626  0.000 0.000  0.000 0.000 0.000 0.000
x4 0.082 0.000 0.000  0.336 0.124 -0.080 0.000 0.000 0.000
x5 0.041 0.000 0.000  0.124 0.440  0.000 0.000 0.000 0.000
x6 0.000 0.000 0.000 -0.080 0.000  0.077 0.000 0.000 0.000
x7 0.000 0.000 0.000  0.000 0.000  0.000 0.672 0.000 0.000
x8 0.000 0.000 0.000  0.000 0.000  0.000 0.000 0.462 0.000
x9 0.000 0.000 0.000  0.000 0.000  0.000 0.000 0.000 0.571
```

```r
det(latInspect(fit, what = "est")[[1]]$theta)      # Check the determinant
```

```
[1] 0.0002812675
```

Positive-definite latent covariances

- Parameterize latent covariances as crossproducts:

$$\Psi = Y^\top Y$$
$$\Theta = U^\top U$$

- Constraint **Y** and **U** to be **orthoblique**:

$$X \in \mathbb{R}^{p \times p} : X^\top X = \text{sparse matrix}$$

$$\begin{bmatrix} 0.08 & 1.76 & 0.04 \\ -1.95 & -0.12 & -0.69 \\ -0.67 & -0.08 & 2.02 \end{bmatrix}^\top \begin{bmatrix} 0.08 & 1.76 & 0.04 \\ -1.95 & -0.12 & -0.69 \\ -0.67 & -0.08 & 2.02 \end{bmatrix} = \begin{bmatrix} 4.24 & 0.42 & 0.00 \\ 0.42 & 3.11 & 0.00 \\ 0.00 & 0.00 & 4.56 \end{bmatrix}$$

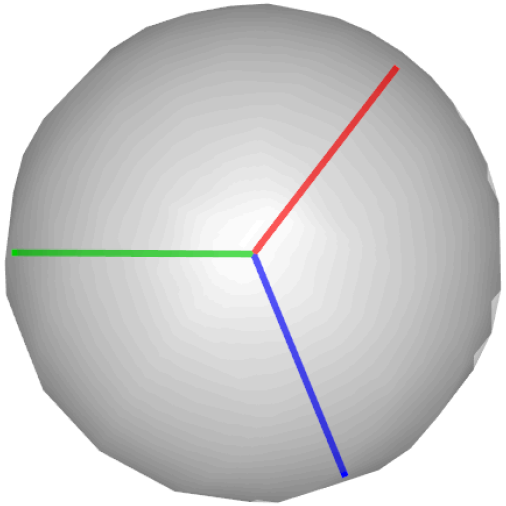
Positive-definite latent correlations

- Parameterize latent covariances as crossproducts:

$$\Psi = Y^\top Y$$
$$\Theta = U^\top U$$

- Constraint **Y** and **U** to be **orthoblique**:

$$X \in \mathbb{R}^{p \times p} : X^\top X = \text{sparse matrix}$$



$$\begin{bmatrix} 0.04 & 1.00 & 0.02 \\ -0.95 & -0.07 & -0.32 \\ -0.32 & -0.05 & 0.95 \end{bmatrix}^\top \begin{bmatrix} 0.04 & 1.00 & 0.02 \\ -0.95 & -0.07 & -0.32 \\ -0.32 & -0.05 & 0.95 \end{bmatrix} = \begin{bmatrix} 1.00 & 0.12 & 0.00 \\ 0.12 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

Factor rotation

PSYCHOMETRIKA—VOL. 66, NO. 2, 289–306
JUNE 2001

A SIMPLE GENERAL PROCEDURE FOR ORTHOGONAL ROTATION

ROBERT I. JENNRICH
UNIVERSITY OF CALIFORNIA AT LOS ANGELES

PSYCHOMETRIKA—VOL. 67, NO. 1, 7–20
MARCH 2002

A SIMPLE GENERAL METHOD FOR OBLIQUE ROTATION

ROBERT I. JENNRICH
UNIVERSITY OF CALIFORNIA AT LOS ANGELES

The orthoblique rotation for exploratory factor analysis

Marcos Jiménez Mauricio Garnier-Villarreal
Vithor R. Franco

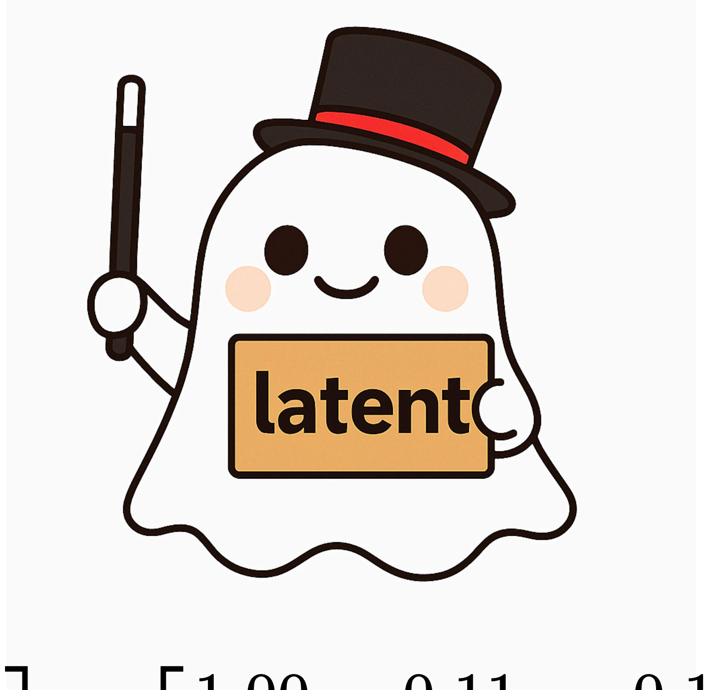
Positive-definite polychoric correlations

- Parameterize the polychoric correlation matrix as a crossproduct:

$$\Sigma = X^\top X$$

- Constraint **X** to be **oblique**:

$$X \in \mathbb{R}^{p \times p} : \mathrm{diag}(X^\top X) = I$$



$$\begin{bmatrix} 0.04 & 1.00 & -0.40 \\ -0.95 & -0.07 & -0.40 \\ -0.32 & -0.04 & 0.83 \end{bmatrix}^\top \begin{bmatrix} 0.04 & 1.00 & -0.40 \\ -0.95 & -0.07 & -0.40 \\ -0.32 & -0.04 & 0.83 \end{bmatrix} = \begin{bmatrix} 1.00 & 0.11 & 0.10 \\ 0.11 & 1.00 & -0.41 \\ 0.10 & -0.41 & 1.00 \end{bmatrix}$$

Cooking new stuff

- (Exploratory) Structural Equation Modeling
- Hidden Markov Models
- (Multidimensional) Item Response Theory

Release date? Soon

Download the **beta version** at https://github.com/Marcosjnez/latent

Contact: m.j.jimenezhenriquez@vu.nl